A NON-NEGATIVE FRAMEWORK FOR JOINT MODELING OF SPECTRAL STRUCTURE AND TEMPORAL DYNAMICS IN SOUND MIXTURES

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF MUSIC
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Gautham J. Mysore

June 2010

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

(Julius O. Smith III) Principal Adviser

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

(Paris Smaragdis)

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

(Malcolm Slaney)

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

(Robert Tibshirani)

Approved for the University Committee on Graduate Studies.

I dedicate this thesis to
my parents Madhura Janakaraj and Mysore N. Janakaraj,
my fiancée Natasha Jaipaul,
and my brother Ashwin J. Mysore.

Abstract

Statistical modeling of audio is an ongoing pursuit. There is a great deal of structure in audio and good models need to make use of this structure. Audio is non-stationary but the statistics of the spectral structure are quite consistent over segments of time. Moreover, there is a structure to the non-stationarity itself in the form of temporal dynamics. Sound mixtures are commonly encountered in practice. Polyphonic music, multiple concurrent speakers, and most environmental sounds are mixtures. Moreover, most real world sound sources are actually a mixture of the source and noise. When dealing with mixtures, the structure of the individual sources becomes particularly important if we wish deal with the sources separately.

In recent years, non-negative spectrogram factorization methods have become quite popular for modeling audio as they provide a rich representation of audio spectra and are amenable to high quality reconstructions. However, they disregard non-stationarity as they use a single dictionary to characterize the statistics of the spectral structure of an entire source. On the other hand, hidden Markov models (HMMs) cater well to non-stationarity and have been used successfully to model temporal dynamics. They can be powerful for audio analysis, as shown by their application to speech recognition. They can also be used for the reconstruction of sources but have certain limitations due to a rigid observation model. This can be an issue for high quality reconstructions.

We propose a new model of single sound sources, the non-negative hidden Markov model (N-HMM), that jointly models the spectral structure and temporal dynamics of a given source. In the proposed model, rather than learning a single dictionary, we learn several small dictionaries that characterize the spectral structure of the

source, catering well to non-stationarity. Moreover, we jointly learn a Markov chain that characterizes the temporal dynamics of the source. This is done with a flexible observation model that allows high quality reconstructions. We demonstrate this model on content–aware audio processing.

We then propose a new model of sound mixtures, the non-negative factorial hidden Markov model (N-FHMM), that combines models of individual sources. This model incorporates the spectral structure and temporal dynamics of each individual source. We demonstrate the model on single channel source separation and show that it yields superior performance to non-negative spectrogram factorization. Although it is demonstrated on source separation, the N-FHMM is a general model of sound mixtures and can be used for various applications.

Acknowledgements

A great deal of people have helped me accomplish this work in various ways. I would like to thank my advisor, Julius O. Smith III, for inspiring me to study audio signal processing even before I arrived at Stanford. He has provided me with a great deal of freedom to explore the topics of my choice and work on the research areas that interest me the most. He has helped me understand complex signal processing concepts through our discussions. He has also welcomed my constant barrage of questions both in and out of his classes.

I owe an immense amount of gratitude to Paris Smaragdis who has been an excellent mentor over the last two years. He has been a great mentor in every sense. He has taught me a great deal about research from general approaches to problem solving to specifics about machine learning and signal processing for audio. We have had constant technical discussions that have been immensely educational. He has also given me a great deal of practical career advice.

I first met Bhiksha Raj in a speech recognition summer school three years ago and I knew from that time that he was born to teach. He has also been an excellent mentor and I thank him for this. His explanations of complex technical topics have been instrumental in my understanding of various concepts.

I would like to thank Malcolm Slaney for many stimulating conversations over the last several years. His wealth of experience has always been enlightening. I would also like to thank him for thorough and detailed comments on this thesis.

I would like to thank Robert Tibshirani for patiently answering my questions on statistics and for welcoming me to his machine learning lab seminars.

No amount of appreciation would be enough for my parents, who have provided

me with constant encouragement throughout the years. I have been truly blessed to have them as my parents. They have instilled the importance of education in me from a very young age. They have provided me with unending love and support over the years and have encouraged me to follow my passion and go after my dreams.

I would like to greatly thank my brother, who has been a constant source of support throughout the years. His encouragement has been immense. He has always helped me with every aspect of anything that I want to do.

I am extremely grateful for having the most amazing fiancée, who has provided me with constant encouragement. She has been truly happy and often more excited than me every time I have achieved one of my educational or career goals. Her unending love has kept me sane through the busiest of times (i.e. writing this thesis).

I would like to thank all of the graduate students, faculty, and staff at CCRMA with whom I have had the pleasure of having spent time with over the last several years. CCRMA has provided me with a truly unique and stimulating interdisciplinary environment. I have had particularly fruitful and enlightening discussions with Juhan Nam, Jonathan Abel, Kyogu Lee, Greg Sell, Nick Bryan, Ryan Cassidy, Aaron Master, Ed Berdahl, Joachim Ganseman, Mofei Zhu, and Hugo Guo.

I would like to thank T.V. Sreenivas for giving me my first opportunity to conduct research. It was at his lab, the speech and audio lab at the Indian Institute of Science (IISc), that I discovered my love for research. I realized that a career in research is the only way that I would be able to satiate my hunger for discovery and invention.

I would like to thank all of the teachers in my high school, the Valley School, for the non-traditional education that I received. Although it was at IISc that I realized that I want to pursue a career in research, it was at the Valley School that I realized that I want to pursue a career related to music. This is a place that truly encouraged its students to find their passion and follow it regardless of traditional career choices. This has been instrumental in leading me to my current career path.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Introduction

We commonly encounter mixtures of sound sources. A significant portion of existing music is a mixture of parts played by different musicians. We often hear multiple people speaking at the same time (cocktail party effect). Most environmental sounds are a mixture of various sources. In fact, most so called single sources are actually a mixture of the source and noise. The human auditory system has an extraordinary ability to differentiate between the constituent sources in a mixture. This is however a challenging problem for computers.

If computers could distinguish between the constituent sources, each source could be dealt with individually and many interesting applications would be possible. This has therefore been an active research area for decades. The problem could be approached in the following two ways:

1. Separate the sources and then process or analyze the individual sources.
2. Process or analyze the individual sources within the mixture without actually performing separation.

In this thesis, we develop a new statistical model of sound mixtures that model the individual sources within the mixture. By modeling the sources, we can attempt the problem in either way. We concentrate on modeling single channel sound mixtures.

This refers to multiple sounds that were recorded with a single microphone as well as sounds that were recorded with multiple microphones but artificially mixed together into a single channel. This is a particularly challenging problem as we do not have spatial information available to us (which is available in multi-channel recordings).

In order to motivate the problem, we discuss some applications that would be possible if we could distinguish between the sources in a mixture.

1. It is often desirable to process a single instrument in a recording. For example, in a single microphone recording of vocals and acoustic guitar, we might want to adjust the volume of the guitar or shift the pitch of the vocals. If we can distinguish between the individual instruments in a mixture, we can process them individually.

2. Speech recognition in the presence of noise, particularly heavy non-stationary noise, is a challenging problem. Speech recognition performance could improve if we could distinguish the speech from the noise and perform recognition on the portion of the mixture that corresponds to speech.

3. In general, when a noisy signal is modeled as a mixture of the signal and noise, denoising can be formulated as a source separation problem.

4. If the vocals can be extracted from a recording, we can generate automatic karaoke tracks. Similarly, if the lead guitar can be extracted from a recording, we can generate automatic "jam tracks". These can both be formulated as source separation problems.

5. Musicians often spend large amounts of time trying to listen to a song and learn the part of a specific instrument by ear. This task becomes more difficult when the given piece of music has numerous parts by numerous instruments (which is often the case). If we can extract the instrument of interest, it could simplify the task of the musician. In practice, this is a common problem for guitar players that try to learn their parts from recordings of bands.

6. Automatic music transcription of polyphonic music is a challenging problem. If

we can model each of the instruments in the mixture, we can transcribe them individually.

7. A number of music information retrieval (MIR) tasks involve extracting information from individual sources. For example, guitar and piano parts could be good indicators of the key of a song. However, the percussion part will rarely have any useful information for this task. Although, the sound mixture can be directly used for many of these tasks, extracting the information from the right source could improve the performance.

Since a number of these applications involve an audio output, we need a model that is amenable to high quality reconstructions.

In the remainder of this chapter, we first discuss structure in audio and the representation of audio that we use. We then discuss the contributions of this thesis. This is followed by an outline of the thesis. Finally, we discuss notation that will be used throughout the thesis.

1.2 Structure in Audio

Although there is some randomness in audio, it has a great deal of structural regularity. Since we deal with single channel sound mixtures, we need some kind of information about the structure of the individual sources that can be used to characterize them. If this information is sufficiently different in the individual sources, we can use it to distinguish between them and therefore use it in models of the individual sources. If we start with the right representation of audio, a machine learning algorithm could potentially automatically discover the structural regularity. It is therefore important to use the right representation.

The spectrogram is a commonly used representation of audio, particularly for sound mixtures [64, 67, 48, 75]. It is the magnitude of the short time Fourier transform (STFT) [70] of an audio signal. The spectrogram is used for several reasons. We start with an example of a few notes of piano music (Fig. 1.1). By visual inspection of the spectrogram, we can see a great deal of structure. For example, we can see that

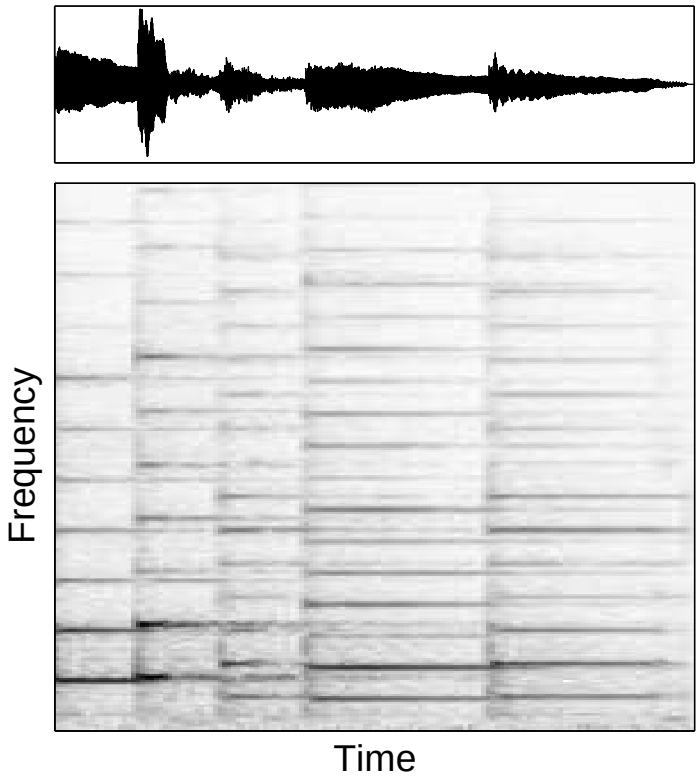


Figure 1.1: Illustration of the time domain signal (top) and the corresponding spectrogram (bottom) of a clip of piano music.

there are five notes in this clip of music. We can also see the fundamental frequency and harmonics of each note. Since we can see this structure, it is plausible that machine learning algorithms can be designed to automatically discover the structure. We approach this problem by developing models that take into account some of the key aspects of audio structure. The learning is performed as parameter estimation in these models. The aspects of audio structure that we address are the following:

1. Spectral structure — Each column of a spectrogram is the magnitude of the Fourier transform over a fixed window of an audio signal. It tells us the spectral content for a given time frame. As seen in Fig. 1.1, there are very clear spectral patterns within the spectrogram that are repeated over several time frames. Although it is not this clear and simple in more complex audio signals, almost all audio has some amount of regularity in spectral structure. Therefore our models make use of spectral structure.

2. Non-stationarity — It is well known that the statistics of audio change with time. Our models account for this by collecting multiple sets of statistics over a given spectrogram rather than amalgamating the statistics of the entire spectrogram into one set.

3. Temporal dynamics — Although the statistics of audio change with time, there is a structure to the non-stationarity. Our model accounts for this by modeling temporal dynamics.

The aspect of spectral structure has been used in various existing models. All three of the above aspects have been addressed in the application of hidden Markov models (HMMs) [47], factorial hidden Markov models (FHMMs) [23], and more generally dynamic Bayesian networks (DBNs) [41] to speech [47, 74, 26, 54, 44] and music [37, 72]. It is important that we model spectral structure in a way that is amenable to high quality reconstruction of the individual sources in dense polyphonic mixtures, while still addressing the other two aspects. This is the concentration of the proposed models.

Another reason that the spectrogram is a useful representation is more specific to sound mixtures. We start with a mathematical definition of a sound mixture in

the time and frequency domains. In the time domain, if we have an audio signal $y(t)$ which is the mixture of the signals $x_1(t)$ and $x_2(t)$, they are related as follows [1]:

$$y(t) = x_1(t) + x_2(t).$$

Due to the linearity of the Fourier transform the spectrum of the sound mixture $Y(f)$ is related to the spectra of the individual sources as follows:

$$Y(f) = X_1(f) + X_2(f).$$

Therefore, the spectrum of the sound mixture is simply a summation of the spectra of the individual sources. Each time–frequency bin of the STFT is therefore the summation of the corresponding time–frequency bins of the STFTs of the individual sources. As noted by Roweis [54, 55] and Yilmaz and Rickard [79], speech spectrograms tend to be sparse. This means that only a small portion of the time–frequency bins in a spectrogram will have a significant amount of energy. Although the sound mixture is the sum of the individual sources, if the sources are independent, most of the energy in a given time–frequency bin can be attributed to a single source. This is generally the case with speech in the context of the cocktail party effect.

Music spectrograms also tend to be sparse as seen in the context of a single source in Fig. 1.1. However, music mixtures are much more synchronized than speech. Individual sources tend to play together, which leads to an overlap in time. The sources are often harmonically related, which leads to an overlap in frequency. However, most real music is not perfectly synchronized in time or frequency. Therefore given enough time and frequency resolution, we can generally circumvent the time–frequency overlap in music. Due to the nature of the spectrogram, there is an inherent tradeoff between time and frequency resolution [70] and we can only have a high resolution in one axis. We generally use a high frequency resolution for music. This

[1] This is a simplification of an acoustic mixing scenario such as the recording of multiple sources with a single microphone, but it conveys the basic idea. In an acoustic mixing scenario, there are also convolutive effects due to room reverberation. However, this is fairly accurate in an artificial mixing scenario such as that of a recording studio.

corresponds to using a long window in the computation of the STFT. Each time–frequency bin, even in music, can then be primarily attributed to a single source.

If the spectrogram is used as the representation in which we apply machine learning algorithms, we would be using only magnitude information and ignoring phase. However, if the algorithm correctly assigns each time–frequency bin to the correct source, we will have a grouping of time–frequency bins. If a given time–frequency bin of the mixture is assigned to a particular source, we use the fact the the phase of the corresponding time–frequency bin can also be primarily attributed to that source. We can therefore model the spectrogram alone (without accounting for phase) and still holistically distinguish between the sources.

1.3 Contributions

The primary contribution of this thesis is the development of a non-negative[2] framework for jointly modeling spectral structure and temporal dynamics of individual sources in sound mixtures. Although the focus is on sound mixtures, a model of single sources was developed in the process and has its own applications. The following two new models are proposed:

1. Non-negative hidden Markov model (N-HMM) for modeling single sound sources.
2. Non-negative factorial hidden Markov model (N-FHMM) for modeling sound mixtures.

Parameter estimation equations are derived for both models. The N-HMM is demonstrated on content–aware audio processing. The N-FHMM is demonstrated on single channel supervised source separation.

1.4 Outline

In this section, we discuss the overall organization of the thesis and the concentration of the individual chapters.

[2]Non-negativity refers to a property that will be discussed in Chapter 2.

In Chapter 2, we discuss models of single sources. We start by describing non-negative spectrogram factorization methods and traditional HMMs, both of which are related to the proposed model of single sources. We then describe the proposed model, the N-HMM, in detail. We discuss the probabilistic model and parameter estimation for the model. This is followed by a description of a useful application of the N-HMM, content–aware audio processing.

In Chapter 3, we discuss models of sound mixtures. We start by describing the use of non-negative spectrogram factorization methods and factorial HMMs to model sound mixtures. We then describe the proposed model, the N-FHMM, in detail. We discuss the probabilistic model and parameter estimation for the model.

In Chapter 4, we discuss source separation. We first give an overview of existing approaches to source separation with a particular concentration on methods that use non-negative spectrogram factorization and factorial HMMs. We then discuss the application of N-HMMs and N-FHMMs to single channel supervised source separation.

We conclude this thesis with Chapter 5. This chapter has some closing remarks as well as future avenues for research. We highlight several applications for which the proposed model can be useful.

1.5 Notation

In this section, we describe the notation that will be used in probability distributions and graphical models.

1.5.1 Probability Distributions

The focus of this thesis is on probabilistic modeling. We therefore encounter various probability distributions throughout the thesis. We encounter both time–varying distributions and distributions that are constant through time. We denote time–varying distributions with a subscript t. For example, $P_t(z|q)$ indicates that we have a separate distribution for each time frame whereas $P(z|q)$ indicates that we have a single distribution for all time frames. As usual, lower case random variables will

denote specific instances of random variables. Therefore, if we come across $P(z_t|q_t)$, it simply means that we evaluate $P(z|q)$ with the assignments z_t and q_t to the random variables. It does not however mean that we have a separate distribution for each time frame (due to the lack of the subscript t after P). On the other hand, if we encounter $P_t(z_t|q_t)$, it means that we have a separate distribution at each time frame. All of the distributions that are encountered are either exactly time–varying or exactly constant over all time frames with two exceptions:

1. In the description of the temporal components $P(t|z)$ in symmetric spectrogram factorization (Sec. 2.2.2), we have a conditional distribution for each value of z. However, these distributions are over time as the random variable is t.

2. The transition probabilities $P(q_{t+1}|q_t)$ are defined over two adjacent time frames. It is however constant over any two adjacent time frames. As there is no subscript t after P, there is only a single distribution. This is the same definition of transition probabilities that are used in traditional HMMs.

In some places, we have multiple draws at each time frame. These are denoted with subscripts. For example, we denote draw v at time frame t of random variable f as $f_{t,v}$. All of the draws of f at time t are represented with the bold notation as $\mathbf{f_t}$. Therefore, we have:

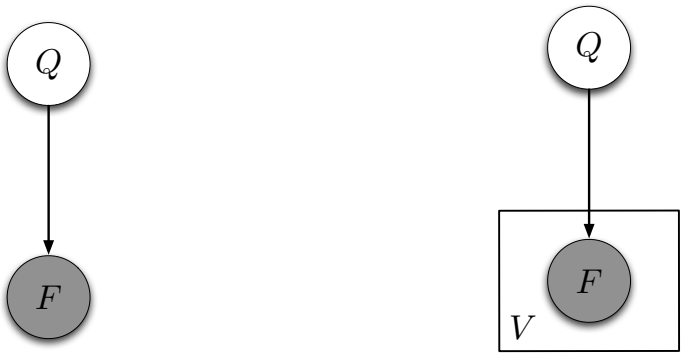
$$P(\mathbf{f_t}) \quad = \quad P(f_{1,v}, f_{2,v}, ..., f_{T,v}).$$

All of the draws over all time frames are represented as $\overline{\mathbf{f}}$. Therefore, we have:

$$P(\overline{\mathbf{f}}) \quad = \quad P(\mathbf{f_1}, \mathbf{f_2}, ..., \mathbf{f_T}).$$

We use the same notation for random variables in which we have only one draw per time frame. For example, in each time frame, if we have only one draw of the random variable q, we will have the sequence, $q_1, q_2, ..., q_T$. The probability of this entire sequence is represented by:

$$P(\overline{\mathbf{q}}) \quad = \quad P(q_1, q_2, ..., q_T).$$



(a) Simple graphical model. (b) Simple graphical model with plate notation.

Figure 1.2: Graphical Models

In chapter 3, we encounter multiple sound sources. These sources have distributions with an identical structure. We therefore use superscripts in parentheses to denote the source number. For example, the transition probabilities of source 1 are denoted by $P(q_{t+1}^{(1)}|q_t^{(1)})$ and that of source 2 are denoted by $P(q_{t+1}^{(2)}|q_t^{(2)})$.

1.5.2 Graphical Models

Graphical models provide a convenient representation of joint probability distributions and clearly indicate the dependencies between random variables in the joint distribution. We will use graphical models to represent various probabilistic models throughout the thesis.

We use the standard convention of representing random variables with nodes and conditional probability distributions with arrows. The direction of the arrows indicate the direction of dependence of random variables. Shaded nodes indicate observed random variables and clear nodes indicate hidden random variables. For example, the graphical model in Fig. 1.2a represents the joint distribution, $P(f, q)$. It tells us that the random variables of the distribution are f and q, the specific conditional distributions of the model are $P(q)$ and $P(f|q)$ (as indicated by the arrows), and that f is observed while q is hidden (as indicated by the shading of the nodes).

In some cases, we will have repetitions of the same random variable. This is a convenient way to represent several draws from the same distribution. This is

represented in the graphical model using the plate notation. The random variables within the plate are repeated a number of times. This specific number is indicated within the plate. For example, the graphical model in Fig. 1.2b has a plate around the random variable f. Since the number within the plate is V, it means that f and therefore the distribution $P(f|q)$ is repeated V times.

Chapter 2

Models of Single Sources

2.1 Introduction

In this chapter, we propose a model of single source sounds, the non-negative hidden Markov model (N-HMM), which jointly captures the spectral structure and temporal dynamics of the source in a way that enables high quality reconstructions. We begin by reviewing non-negative spectrogram factorization methods and traditional hidden Markov models (HMMs), both of which are related to the proposed model.

Non-negative spectrogram factorization methods do a great job of capturing the spectral structure of sound sources but fail to capture the temporal dynamics of the source. They also use a single dictionary to globally capture the spectral structure of the source. This does not cater well to the inherent non-stationarity of audio.

HMMs, on the other hand, have been extensively used to model the temporal dynamics of audio. They can be quite useful for analysis applications such as speech recognition and chord recognition. They can also be used for the reconstruction of sources but have certain limitations due to a rigid observation model. This can be an issue for high quality reconstructions.

The proposed model uses several small dictionaries to capture the spectral structure of a sound source, catering to the non-stationarity of audio. Additionally, a Markov chain is used to model the structure of changes between these dictionaries (temporal dynamics). Given a sound source, the dictionaries and the Markov chain

are jointly learned. A flexible observation model allows high quality reconstructions.

2.2 Non-negative Spectrogram Factorization

There has been a great deal of work on non-negative spectrogram factorization methods in recent years. The idea behind these methods is that that audio spectrograms are effectively low rank non-negative matrices and can therefore be compactly represented in a semantically meaningful way. A typical audio spectrogram can be described by a few spectral patterns. These spectral patterns can be interpreted as a basis or a dictionary of spectral components. Every time frame of the spectrogram can then be explained by a linear combination of these spectral components. A typical model is as follows:

$$v_t \approx \sum_{k=1}^{K} h_{kt} w_k,$$

where v_t is the t-th frame of the spectrogram. The spectrogram is explained by K spectral components. w_k is the k-th component. h_{kt} is the weight of the k-th component at time t. In matrix notation, this can be represented as:

$$\mathbf{V} \approx \mathbf{WH},$$

where the spectrogram $\mathbf{V}$, is an $F \times T$ matrix. The dictionary $\mathbf{W}$ is an $F \times K$ matrix in which each column is a spectral component. The mixture weights are in a $K \times T$ matrix $\mathbf{H}$ in which each row represents the weights for a given component.

Given $\mathbf{V}$, the estimation of $\mathbf{W}$ and $\mathbf{H}$ with the constraint that both matrices must be non-negative, is the non-negative matrix factorization (NMF) problem [35, 36]. This was first shown to be effective for discovering structure in audio spectrograms by Smaragdis and Brown [65]. A simple example of this is shown in Fig. 2.1.

NMF was originally cast as an optimization problem by Lee and Seung [36], in which the cost function was either the Euclidean distance or generalized Kullback

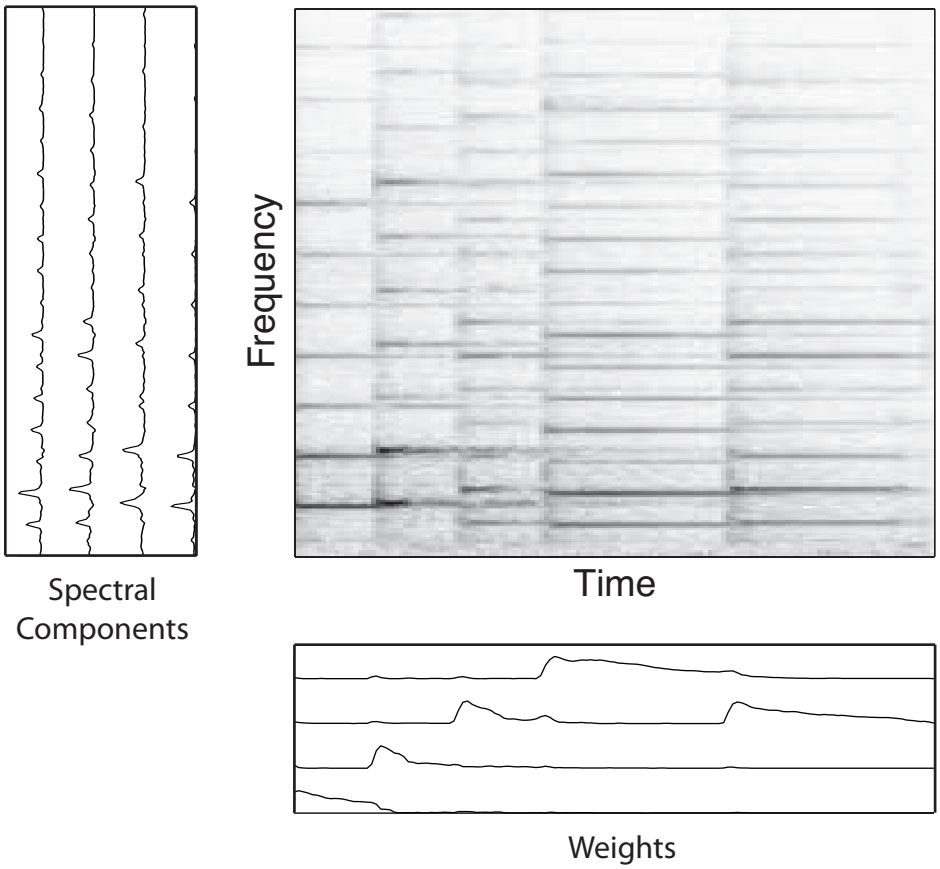


Figure 2.1: Illustration of NMF on the spectrogram of a clip of piano music. The sound clip has five notes as can be seen in the spectrogram. The fifth note is a repetition of the third note so there are four distinct notes. When NMF is applied to the spectrogram (with $K = 4$), four distinct spectral components are learned. Additionally, the weights of these spectral components at each time frame are learned. The attack and decay of each note can be seen in the weights. The repetition of the third note can also be seen in the weights.

Leibler (KL) divergence between $\mathbf{V}$ and $\mathbf{WH}$ (subject to the non-negativity constraints). Dhillon and Sra [15] generalized these cost functions to the Bregman divergence between $\mathbf{V}$ and $\mathbf{WH}$.

Non-negativity plays an important role in this factorization. The idea is to discover spectral components that characterize the spectrogram. Since a spectrogram is by definition non-negative, semantically meaningful spectral components must also be non-negative. Since the spectrogram is modeled as a weighted sum of spectral components, the estimated components could have negative values without an explicit non-negativity constraint. This is seen in algorithms such as principle component analysis (PCA).

Given a spectrogram, the model that is learned is therefore the dictionary of spectral components. This dictionary is useful for representing the general characteristics of the sound source and can generalize to other unseen instances of the source. The weights on the other hand, will only characterize the specific instance of data at hand.

In the following subsections, we briefly discuss probabilistic extensions of NMF and then discuss one family of these extensions, latent variable spectral models.

2.2.1 Probabilistic Extensions

Several researchers have explored probabilistic extensions of NMF in the last few years [67, 61, 59, 60, 76, 12, 17, 57, 44, 66, 22]. This allows the use of sophisticated statistical techniques while still using the general ideas of NMF. These techniques have two general themes. Firstly, they tend to explicitly use the Expectation–Maximization (EM) algorithm [14] and variants of it. Although NMF essentially uses the EM algorithm, the probabilistic framework formalizes it. This allows extensions in a principled manner.

The other theme is the use of prior distributions to encode our knowledge about a given problem. For example Smaragdis and Mysore [66] used Dirichlet priors to encode a user input to select a sound source from a mixture. Ganseman et al. [22] used Dirichlet priors to encode a synthesized score of an instrument that is to be extracted from a mixture. Virtanen et al. [76] used a Gamma–chain prior to encode

spectral smoothness of natural sounds. Apart from encoding prior knowledge about a problem, prior distributions have been used in the general modeling strategies. For example, Shashanka et al. [61] used an entropic prior to achieve sparsity in order to learn overcomplete dictionaries.

In the following subsection, we discuss one specific family of these probabilistic extensions in more detail as they are more closely related to the proposed model.

2.2.2 Latent Variable Spectral Models

The idea behind this family of models is that the given spectrogram is generated by a set of latent components. The latent components are the spectral components as seen in NMF. Given the spectrogram, the goal is to estimate these latent components using the EM algorithm.

The given spectrogram is modeled as a histogram of "sound quanta." The amount of sound quanta in a given time–frequency bin indicates the Fourier magnitude of that bin and is given by V_{ft}[1]. A generative process is used to hypothesize the construction of the histogram. Once normalized, it can be thought of as a joint probability distribution $P(f,t)$ over time and frequency. We therefore need to have a model for $P(f,t)$. We consider two models that correspond to a symmetric and asymmetric factorization of the spectrogram.

Symmetric Factorization

This model, developed by Smaragdis et al. [67], is called probabilistic latent component analysis (PLCA) and is given by:

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

where $P(f|z)$ are the latent components that correspond to spectral components. For a given value of z, $P(f|z)$ is a multinomial distribution. Therefore the probability of each frequency f is a separate parameter. The collection of all $P(f|z)$ form a

[1] In theory, this would involve a scaling of the spectrogram (a single scale factor for all time–frequency bins) such that each time–frequency bin has a whole number of sound quanta.

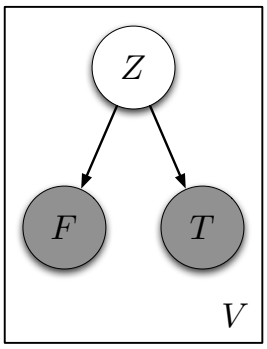


Figure 2.2: Graphical model for symmetric spectrogram factorization [67].

dictionary. $P(t|z)$ are also modeled as latent components and correspond to the occurrences of the spectral components in time. For a given value of z, $P(t|z)$ is a multinomial distribution. Therefore the probability of the occurrence of component z at each time frame t is a separate parameter. $P(z)$ is a distribution of gains and is also a multinomial distribution. In NMF, there are explicit constraints to enforce non-negativity of components. In PLCA, non-negativity is implicitly enforced by mapping the values of the individual spectral magnitudes and time activations to parameters of multinomial distributions, which are by definition non-negative.

For each value of the latent variable z, we therefore have a spectral component $P(f|z)$ and a temporal component $P(t|z)$. Since these components have the same structure, this model is symmetric over time and frequency and therefore corresponds to a symmetric factorization of the spectrogram. In Fig. 2.1, $P(f|z)$ corresponds to the spectral components and $P(t|z)$ correspond to the weights. In that example, the contribution of each value of z gives us an outer product expansion that can be interpreted as the contribution of a given note. The contribution of a given value of z is therefore given by:

$$P(f,t|z) \quad = \quad P(f|z)P(t|z).$$

$P(z)$ can be interpreted as giving an overall gain to each $P(f,t|z)$.

The generative process is as follows (the graphical model is shown in Fig. 2.2) :

1. Choose a latent variable according to $P(z)$.

2. Choose a frequency according to $P(f|z)$ and choose a time frame according to $P(t|z)$.

3. Repeat steps 1 and 2 V times, where $V = \sum_f \sum_t V_{ft}$ (the total number of observed sound quanta).

Given the spectrogram, we estimate the parameters of the model using the EM algorithm as follows:

E Step

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

M Step

$$\begin{aligned} P(f|z) &= \frac{\sum_t V_{ft}P(z|f,t)}{\sum_f \sum_t V_{ft}P(z|f,t)}, \\ P(t|z) &= \frac{\sum_f V_{ft}P(z|f,t)}{\sum_t \sum_f V_{ft}P(z|f,t)}, \\ P(z) &= \frac{\sum_f \sum_t V_{ft}P(z|f,t)}{\sum_z \sum_f \sum_t V_{ft}P(z|f,t)}. \end{aligned}$$

It should be noted that the temporal components $P(t|z)$ independently indicate the temporal dynamics of each spectral component. However, they do not indicate the dynamics between the different components or relate the components in any way. They simply indicate the occurrence of each spectral component in time. Moreover, they simply characterize the specific instance of the data at hand rather than indicating the general temporal characteristics of the the given class of data.

Asymmetric Factorization

In this model, developed by Raj and Smargdis [48], each time frame is modeled as a linear combination of spectral components. We therefore have a different distribution of weights for each time frame. Since the weights for a given time frame form a

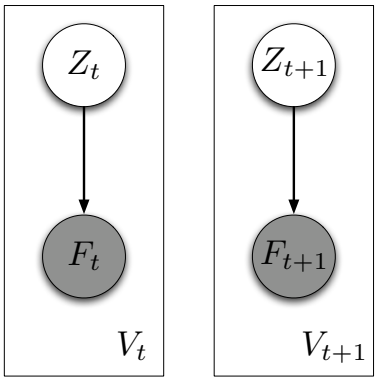


Figure 2.3: Graphical model for asymmetric spectrogram factorization [48]. The independence between time frames can be seen here.

distribution, they can be interpreted as "mixture weights". The model is given by:

$$P_t(f_t) = \sum_{z_t} P(f_t|z_t)P_t(z_t),$$

where $P_t(f_t)$ corresponds to the normalized spectrogram at time frame t. $P(f_t|z_t)$ is a spectral component[2]. $P_t(z_t)$ is a distribution of mixture weights at time frame t. All of the distributions are multinomial.

The main difference between this model and the symmetric factorization model is the normalization of the weights. In this model, the weights are normalized across components (at each time frame) to yield probability distributions. In the symmetric factorization model, the weights for a given component are normalized across time to yield a probability distribution. In this model, we therefore gained the property of relating the components in a given time frame at the cost of independence between time frames.

The generative process for a given time frame is as follows (the graphical model is shown in Fig. 2.3):

1. Choose a latent variable according to $P_t(z_t)$.

[2]It should be noted that $P(f_t|z_t)$ is not time dependent. The subscripts are used here to indicate the time dependence of the spectrogram and the mixture weights. We however estimate a single spectral component for each value of z.

2. Choose a frequency according to $P(f|z_t)$.

3. Repeat steps 1 and 2 V_t times, where $V_t = \sum_t V_{ft}$ (the total number of observed sound quanta at time frame t).

Given the spectrogram, we estimate a set of weights at each time frame but a single set of spectral components. This is done using the EM algorithm as follows:

E Step

$$P_t(z_t|f_t) = \frac{P_t(z_t)P(f_t|z_t)}{\sum_{z_t} P_t(z_t)P(f_t|z_t)}.$$

M Step

$$P(f|z) = \frac{\sum_t V_{ft}P_t(z|f)}{\sum_f \sum_t V_{ft}P_t(z|f)},$$
$$P_t(z_t) = \frac{\sum_{f_t} V_{ft}P_t(z_t|f_t)}{\sum_{z_t} \sum_{f_t} V_{ft}P_t(z_t|f_t)}.$$

As can be seen, there is an inherent tradeoff in which either the components are independent (symmetric factorization) or the time frames are independent (asymmetric factorization). The relation between the time frames and components are both however important. Moreover, the relation between components over time frames is important as that characterizes the temporal dynamics of the source. This is not modeled by this family of models and non-negative spectrogram factorization methods in general and is a serious limitation.

Another limitation is that these algorithms estimate a single dictionary of spectral components to model an entire sound source. Audio is inherently non-stationary so the use of a single dictionary to describe a sound source over all time frames is perhaps not the best strategy. A more effective strategy could be to use multiple small dictionaries to describe the spectral characteristics at different points of time. In Sec. 2.4, we will see how the proposed model overcomes these limitations.

2.3 Hidden Markov Models

Hidden Markov models (HMMs) have been used for decades to model temporal dynamics. They are particularly useful for audio due to the strong temporal structure in almost any form of audio. They has been seen to be particularly effective in modeling the dynamics of speech [47]. They have also been effective in modeling different temporal aspects of music. For example, Lee and Slaney [37] have modeled chord progressions using HMMs.

HMMs are used to model time series data that can be described by a Markov process. In such data, there is a sequence of states (one per time frame) that describe something that is of interest to us. The transition from the state in one time frame to the state in the following time frame can be described by a transition probability. The problem is that the sequence of states are not observed. However, the state at each time frame gives rise to an observation. The relationship between the state and the observation can be described by an observation model (emission probability).

The state sequence generally has an intuitive interpretation. For example, a state can correspond to a chord [37]. The dynamics of the state sequence could then conform to music theory rules. For example, if the state at a given time frame corresponds to a dominant (V) chord, there would be a high probability that state in the following time frame would correspond to a tonic (I) chord. In speech, a state can correspond to a subunit such as a phoneme [47]. The dynamics of the state sequence would correspond to phoneme transition probabilities.

To formalize this, we wish to model a hidden state sequence, $q_1, q_2, ..., q_T$, that we approximate as a Markov process. The Markovian dynamics are described by the transition probability, $P(q_{t+1}|q_t)$ (generally represented as a matrix). At every time step, there is an observation, f_t. This observation is related to the hidden state through the observation model, $P(f_t|q_t)$. There is also an initial state distribution $P(q_1)$.

The model (in terms of the observations) is given by:

$$P(\overline{\mathbf{f}}) \;=\; \sum_{\overline{\mathbf{q}}} P(q_1) \prod_{t=1}^{T-1} P(q_{t+1}|q_t) \prod_{t=1}^{T} P(f_t|q_t),$$

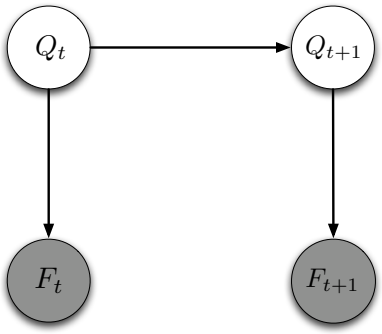


Figure 2.4: Graphical model for an HMM.

where $P(\overline{\mathbf{f}})$ indicates the probability of all of the observations.

For a given value of q_t, $P(q_{t+1}|q_t)$ is a multinomial distribution. These conditional distributions for all values of q_t can therefore be represented by a matrix. If the state at a given time frame can have N values, then the matrix will be size $N \times N$. $P(q_1)$ is a multinomial distribution over the N possible values of the state.

Traditionally, the observation f_t is a feature vector and the observation model $P(f_t|q_t)$ is a multivariate distribution. If we are modeling a spectrogram, the feature vector would correspond to a time frame of the spectrogram. A multivariate Gaussian is commonly used for the observation model in audio applications.

The generative process for a traditional HMM is as follows (the graphical model is shown in Fig. 2.4):

1. Choose an initial state according to $P(q_1)$.
2. Set $t = 1$.
3. Choose an observation according to $P(f_t|q_t)$.
4. Transit to a new state according to $P(q_{t+1}|q_t)$.
5. Set $t = t + 1$ and go to step 3 if $t < T$.

Given a sequence of observations, $f_1, f_2, ..., f_T$, we can estimate the parameters of the model using the EM algorithm. This is often called the Baum-Welch algorithm [3, 47].

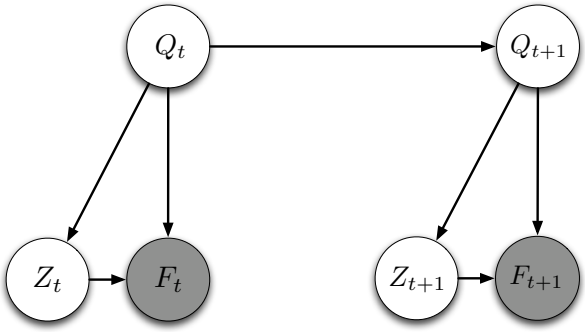


Figure 2.5: Graphical model for an HMM with a mixture model for the observation model.

A common extension of these HMMs is to use a mixture model for the observation model. For example, a Gaussian mixture model (GMM) is commonly used for this purpose. In this case, each state will correspond to an entire GMM. If the GMM has K Gaussians, the observation model would be:

$$P(f_t|q_t) = \sum_{z_t=1}^{K} P(z_t|q_t)P(f_t|z_t, q_t),$$

where $P(f_t|z_t, q_t)$ is a multivariate Gaussian distribution and $P(z_t|q_t)$ are the mixture weights for state q_t. An important point is that for a given state, the structure of the observation model is fixed. This is to say that for a given state, the mixture weights are constant regardless of the time frame. This is of course the case for $P(f_t|z_t, q_t)$ as well. The subscripts merely indicate the instances of the random variable.

The model of the HMM (in terms of the observations) using this observation model is given by:

$$P(\overline{\mathbf{f}}) = \sum_{\overline{\mathbf{q}}} P(q_1) \prod_{t=1}^{T-1} P(q_{t+1}|q_t) \prod_{t=1}^{T} \left(\sum_{z_t=1}^{K} P(z_t|q_t)P(f_t|z_t, q_t) \right).$$

The generative process for this HMM is as follows (the graphical model is shown in Fig. 2.5):

1. Choose an initial state according to $P(q_1)$.

2. Set $t = 1$.

3. Choose a Gaussian according to $P(z_t|q_t)$.

4. Choose an observation according to $P(f_t|z_t, q_t)$.

5. Transit to a new state according to $P(q_{t+1}|q_t)$.

6. Set $t = t + 1$ and go to step 3 if $t < T$.

Using a mixture model for the observation model can greatly increase the expressive power of HMMs. However, the inflexibility of the observation model can still be a hindrance. Also, when modeling spectrograms, there are no explicit non-negativity constraints, which could be an issue for certain applications.

2.4 Non-negative Hidden Markov Models

In this section, we describe the proposed model, the non-negative hidden Markov model (N-HMM). We start with non-negative spectrogram factorizations and build a conceptual explanation of N-HMMs. We then start with traditional HMMs and build up a formal probabilistic model of N-HMMs. This is followed by an explanation of parameter estimation for the model. We show the results of parameter estimation on a few examples. Finally, we discuss the application of this model to content–aware audio processing. In this application, a certain aspect of an input signal, which can be specified by high level information, is processed. It should be noted that the focus of this thesis is the modeling of sound mixtures as will be discussed in the following chapters. However, content–aware audio processing is performed on a single source. It has been included as it is an interesting application of the N-HMM that makes use of some of the key properties of the model.

2.4.1 Conceptual Explanation

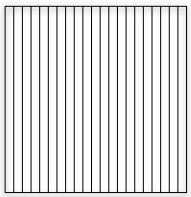
Non-negative spectrogram factorizations model each column of a spectrogram as a linear combination of spectral components. Let us start with the example in Fig. 2.1.

The learned model of the piano is a dictionary of four spectral components. These spectral components correspond to the four notes in the clip of music. This dictionary is represented in Fig. 2.6a. Each note is therefore represented with a single spectral component. The variations of a given note are simply represented by a time–varying weight. In reality, the variations are much more than a time–varying weight. For example, the spectral shape of the attack of a note can be quite different from the spectral shape of the decay.

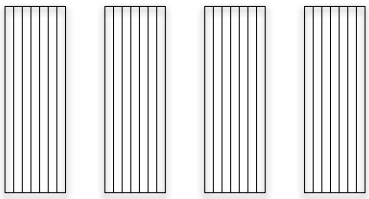
Although the use of one component per note is a reasonable approximation and can be quite useful for analysis applications, it can leave a lot to be desired in terms of reconstruction, which is important for several applications. One way to circumvent this problem is to learn a much larger dictionary as is shown in Fig. 2.6b. This will do a better job of capturing the nuances of the sound. It can greatly reduce the approximation error of the factorization (used to learn the components). Each column of the spectrogram will be explained by a linear combination of the spectral components. The problem is that multiple components will correspond to each note since there will be more components than notes. The algorithm does not give us a grouping of these components so we will not be able to process or analyze each note individually. Moreover, it is quite likely that some components will correspond to many notes. For example, if more than one note has a common harmonic, the harmonic alone could be captured by a single component. A common harmonic is a very likely occurrence in a sequence of notes of music.

Another problem with using a single large dictionary to model a sound source has to do with the modeling philosophy. Non-negative spectrogram factorizations learn a single dictionary to characterize the statistics of the sound source over a long period of time. The same dictionary is used to explain every column of the spectrogram. It is well know that audio is non-stationary. Therefore, amalgamating the statistics over all time frames into a single dictionary is perhaps not the best strategy. A method that conforms better to non-stationarity could be to learn several small dictionaries to explain different aspects of the sound source. Each column of the spectrogram can then be explained by a linear combination of the spectral components from one (out of the many) dictionaries. The N-HMM, does this by jointly learning several

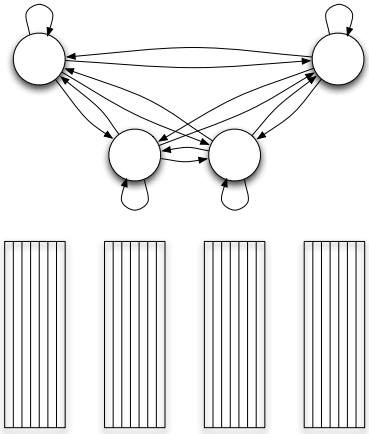
(a) Dictionary of four components learned by non-negative spectrogram factorization.



(b) Dictionary with a large number of components learned by non-negative spectrogram factorization.



(c) Four separate small learned dictionaries rather than one large dictionary.



(d) Four separate small dictionaries and a Markov chain jointly learned using the N-HMM.

Figure 2.6: Illustration of learned dictionaries. We start with a single small dictionary that is learned using non-negative spectrogram factorization and work up to several small dictionaries and a Markov chain jointly learned by the N-HMM. In the N-HMM, each dictionary corresponds to a state of the Markov chain. An ergodic (fully-connected) model has been shown for illustration purposes but any kind of Markov chain can be learned.

small dictionaries, as shown in Fig. 2.6c, to explain different aspects of the sound source. The different dictionaries are automatically learned from data but they often correspond to intuitive notions. For example, in a sequence of notes of music such as the piano example in Fig. 2.1, each dictionary is likely to correspond to a note. In speech, each dictionary is likely to correspond to a phoneme or part of a phoneme.

Learning several small dictionaries rather than a single large dictionary is consistent with the non-stationarity of audio. However, there is still more structure that we can use. Since each dictionary corresponds to a different aspect of the sound source, the transitions between dictionaries corresponds to the temporal dynamics of the source. This can be modeled by a Markov chain and learned from the data as well. We can therefore jointly learn several small dictionaries and a Markov chain from the given spectrogram as seen in Fig. 2.6d, using the N-HMM. The Markov chain also has intuitive interpretations. For example, if the given spectrogram corresponds to a sequence of notes, it could conform to music theory rules. It would then tell us the probability of a given note in a given time frame given the note in the previous time frame.

The following are therefore the key points of the N-HMM:

1. Rich spectral modeling capability of non-negative spectrogram factorizations
2. Models non-stationarity of audio by learning several small dictionaries of spectral components rather than a single large dictionary
3. Models temporal dynamics between dictionaries by learning a Markov chain

2.4.2 Probabilistic Model

We now describe the probabilistic model for N-HMMs starting with simple HMMs and building up from there in a step by step manner. We then describe the generative process of the N-HMM.

We start with the simple HMM as seen in Fig. 2.4. The first thing to do is to use a multinomial distribution $P(f|q)$ as the observation model. This is analogous to a spectral component in non-negative spectrogram factorizations. Each state will

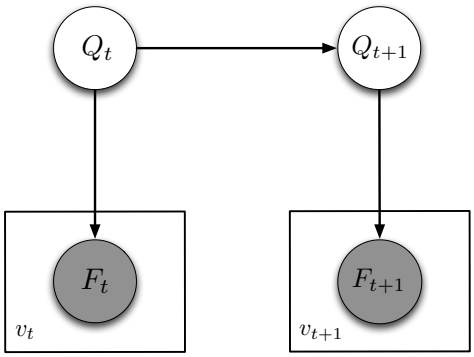


Figure 2.7: Graphical model for an HMM with multiple draws at every time frame.

then correspond to its own spectral component. In every time frame, we would draw a single frequency from the multinomial. Our observation sequence would then be a single frequency at every time frame. In order to model an entire magnitude spectrum, we need to have multiple draws at every time frame. This is shown in the graphical model in Fig. 2.7. This brings up another issue. We need to know how many times we need to draw from the given multinomial distribution in each time frame. In order to do this, we explicitly model the number of draws for a given state with a Gaussian distribution $P(v|q)$. The number of draws that were made to explain a given time frame intuitively corresponds to the energy of the spectrogram at that time frame. We therefore call it the "energy distribution." We now have a way of generating a magnitude spectrum at every time frame.

The use of a single spectral component per state can be quite limiting. We therefore extend this to the use of a whole dictionary of spectral components per state. This is modeled using a multinomial mixture model. The spectral components for state q are given by $P(f|z,q)$. Since we have multiple spectral components per state, we need to have a set of mixture weights $P(z|q)$ for each state. If $P(z|q)$ were not time dependent (as in traditional HMMs), then every occurrence of state q would have the same mixture weights and $P(f|q)$ would be fixed for each q. This would greatly reduce the expressive power of the model. This is especially an issue when using a mixture of multinomial distributions as the mixture would collapse to a single multinomial distribution, defeating the purpose. For these reasons, we use time dependent

distributions $P_t(z_t|q_t)$ for the mixture weights. This gives us the interpretation of explaining each column of the spectrogram by a linear combination of the spectral components from the dictionary that corresponds to state q. Time dependent weights in the context of a mixture model have previously been used in Gaussian scaled mixture models by Benaroya et al. [5]. Regardless of the type of distribution used, if time dependent weights are not used, we would either lose accuracy in modeling variations of a state or we would have to use a very large number of states. Using a large number of states can lead to a combinatorial blowup in complexity when using factorial HMMs (FHMMs) as will be seen in section 4.2.3.

The observation model is therefore given by:

$$P_t(f_t|q_t) = \sum_{z_t} P_t(z_t|q_t)P(f_t|z_t,q_t).$$

We have now arrived at the model for the N-HMM (the graphical model is shown in Fig. 2.8). The model includes the spectral components, weights distributions, energy distributions, transition matrix, and initial state probabilities. The model is in terms of the observations. We therefore sum over all of the hidden variables ($\overline{\mathbf{q}}$ and $\overline{\mathbf{z}}$). It is given by:

$$\begin{aligned} P(\overline{\mathbf{f}},\overline{\mathbf{v}}) = & \sum_{\overline{\mathbf{q}}}\sum_{\overline{\mathbf{z}}} P(q_1)\left(\prod_{t=1}^{T-1} P(q_{t+1}|q_t)\right)\left(\prod_{t=1}^{T} P(v_t|q_t)\right)... \\ & \left(\prod_{t=1}^{T}\prod_{v}^{v_t} P_t(z_{t,v}|q_t)P(f_{t,v}|z_{t,v},q_t)\right). \end{aligned}$$

Since we explicitly model the number of draws at each each time frame v_t and it is simply the total number of sound quanta (over all frequencies) at that time frame, it is an observed random variable.

The generative process for the N-HMM is as follows:

1. Choose an initial state according according to $P(q_1)$.

2. Set $t = 1$.

3. Choose the number of draws for the given time frame according to $P(v_t|q_t)$.

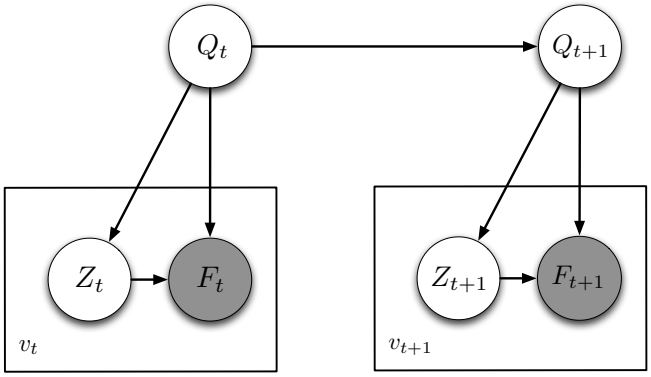


Figure 2.8: Graphical model for the N-HMM.

4. Repeat the following steps v_t times:

   (a) Choose a spectral component according to $P_t(z_{t,v}|q_t)$.

   (b) Choose a frequency according to $P(f_{t,v}|z_{t,v}, q_t)$.

5. Transit to a new state q_{t+1} according to $P(q_{t+1}|q_t)$.

6. Set $t = t + 1$ and go to step 3 if $t < T$.

The following are therefore the key differences between the N-HMM and traditional HMMs that use a mixture model for the observation model:

1. In the N-HMM, we have multiple draws at every time frame rather than a single draw as in traditional HMMs.

2. N-HMMs have a flexible observation model allowing us to model variations in a given state by using time dependent mixture weights. This is opposed to the rigid observation model in traditional HMMs with a fixed distribution of mixture weights for each state.

3. The energy of each state is explicitly modeled in the N-HMM.

2.4.3 Parameter Estimation

We now describe the procedure for the estimation of the parameters of the N-HMM from a given spectrogram. The parameters of the model are as follows:

1. Spectral components (multinomial distributions) — $P(f|z,q)$
2. Mixture weights (multinomial distributions) — $P_t(z_t|q_t)$
3. Transition matrix (multinomial distributions) — $P(q_{t+1}|q_t)$
4. Initial state probabilities (multinomial distribution) —$P(q_1)$
5. Energy distributions (Gaussian distributions) — $P(v|q)$

The parameters are estimated by maximizing the log-likelihood of the data. Since some of the random variables are hidden, we perform the maximization using the Expectation–Maximization (EM) algorithm [14]. It is an iterative algorithm in which the log-likelihood increases with each iteration and tends to converge after a certain number of iterations. Specifically, we iterate between the following steps:

1. Expectation step (E step) — Compute the posterior distribution using the estimated parameters.
2. Maximization step (M step) — Estimate the parameters by maximizing the expected value of the the complete data log likelihood with respect to the posterior distribution.

The posterior distribution of the model is the probability of the hidden variables given the observations and is given by:

$$P(\overline{\mathbf{z}}, \overline{\mathbf{q}}|\overline{\mathbf{f}}, \overline{\mathbf{v}}).$$

It should be noted that apart from $\overline{\mathbf{f}}$, we have also explicitly modeled $\overline{\mathbf{v}}$ as an observation. Since v_t is the total number of draws at time t, it is observed.

The complete data log likelihood is given by:

$$\begin{aligned}\log P(\overline{\mathbf{f}}, \overline{\mathbf{z}}, \overline{\mathbf{q}}, \overline{\mathbf{v}}) &= \log P(q_1) + \sum_{t=1}^{T-1} \log P(q_{t+1}|q_t) + \sum_{t=1}^{T} \log P(v_t|q_t) + ... \\ & \sum_{t=1}^{T}\sum_{v=1}^{v_t} \log P_t(z_{t,v}|q_t) + \sum_{t=1}^{T}\sum_{v=1}^{v_t} \log P(f_{t,v}|z_{t,v}, q_t). \end{aligned} \tag{2.1}$$

The expected value of the complete data log likelihood with respect to the posterior distribution is given by:

$$\begin{aligned}\mathcal{L} &= E_{\overline{\mathbf{z}},\overline{\mathbf{q}}|\overline{\mathbf{f}},\overline{\mathbf{v}}}\left[\log P(\overline{\mathbf{f}}, \overline{\mathbf{z}}, \overline{\mathbf{q}}, \overline{\mathbf{v}})\right] \\ &= \sum_{\overline{\mathbf{q}}}\sum_{\overline{\mathbf{z}}} P(\overline{\mathbf{z}}, \overline{\mathbf{q}}|\overline{\mathbf{f}}, \overline{\mathbf{v}}) \log P(\overline{\mathbf{f}}, \overline{\mathbf{z}}, \overline{\mathbf{q}}, \overline{\mathbf{v}}) \\ &= \sum_{\overline{\mathbf{q}}}\sum_{\overline{\mathbf{z}}} P(\overline{\mathbf{z}}, \overline{\mathbf{q}}|\overline{\mathbf{f}}, \overline{\mathbf{v}}) \log P(q_1) + \sum_{t=1}^{T-1}\sum_{\overline{\mathbf{q}}}\sum_{\overline{\mathbf{z}}} P(\overline{\mathbf{z}}, \overline{\mathbf{q}}|\overline{\mathbf{f}}, \overline{\mathbf{v}}) \log P(q_{t+1}|q_t)... \\ &\quad + \sum_{t=1}^{T}\sum_{\overline{\mathbf{q}}}\sum_{\overline{\mathbf{z}}} P(\overline{\mathbf{z}}, \overline{\mathbf{q}}|\overline{\mathbf{f}}, \overline{\mathbf{v}}) \log P(v_t|q_t) + \sum_{t=1}^{T}\sum_{v=1}^{v_t}\sum_{\overline{\mathbf{q}}}\sum_{\overline{\mathbf{z}}} P(\overline{\mathbf{z}}, \overline{\mathbf{q}}|\overline{\mathbf{f}}, \overline{\mathbf{v}}) \log P_t(z_{t,v}|q_t)... \\ &\quad + \sum_{t=1}^{T}\sum_{v=1}^{v_t}\sum_{\overline{\mathbf{q}}}\sum_{\overline{\mathbf{z}}} P(\overline{\mathbf{z}}, \overline{\mathbf{q}}|\overline{\mathbf{f}}, \overline{\mathbf{v}}) \log P(f_{t,v}|z_{t,v}, q_t). \end{aligned}$$

Certain variables get marginalized in each of the above terms. This gives us the following equation:

$$\begin{aligned}\mathcal{L} &= \sum_{q_1} P(q_1|\overline{\mathbf{f}}, \overline{\mathbf{v}}) \log P(q_1) + \sum_{t=1}^{T-1}\sum_{q_t}\sum_{q_{t+1}} P_t(q_t, q_{t+1}|\overline{\mathbf{f}}, \overline{\mathbf{v}}) \log P(q_{t+1}|q_t)... \\ &\quad + \sum_{t=1}^{T}\sum_{q_t} P_t(q_t|\overline{\mathbf{f}}, \overline{\mathbf{v}}) \log P(v_t|q_t) + \sum_{t=1}^{T}\sum_{v=1}^{v_t}\sum_{q_t}\sum_{z_{t,v}} P_t(z_{t,v}, q_t|\overline{\mathbf{f}}, \overline{\mathbf{v}}) \log P_t(z_{t,v}|q_t)... \\ &\quad + \sum_{t=1}^{T}\sum_{v=1}^{v_t}\sum_{q_t}\sum_{z_{t,v}} P_t(z_{t,v}, q_t|\overline{\mathbf{f}}, \overline{\mathbf{v}}) \log P(f_{t,v}|z_{t,v}, q_t). \end{aligned}$$

In the last two terms of the last step, we change the summations to be over frequencies rather than draws by counting the number of draws of frequency f_t at time t. This is given by V_{ft} (scaled spectrogram). Since we are summing over frequencies, we have changed the distribution $P_t(z_{t,v}, q_t|\overline{\mathbf{f}}, \overline{\mathbf{v}})$ to $P_t(z_t, q_t|f_t, \overline{\mathbf{f}}, \overline{\mathbf{v}})$. This gives us a

distribution over z_t and q_t for all draws at time t that result in the observation f_t. The above equation then becomes:

$$\begin{aligned}\mathcal{L} &= \sum_{q_1} P(q_1|\overline{\mathbf{f}},\overline{\mathbf{v}}) \log P(q_1) + \sum_{t=1}^{T-1}\sum_{q_t}\sum_{q_{t+1}} P_t(q_t,q_{t+1}|\overline{\mathbf{f}},\overline{\mathbf{v}}) \log P(q_{t+1}|q_t)... \\ &+ \sum_{t=1}^{T}\sum_{q_t} P_t(q_t|\overline{\mathbf{f}},\overline{\mathbf{v}}) \log P(v_t|q_t) + \sum_{t=1}^{T}\sum_{q_t}\sum_{z_t}\sum_{f_t} V_{ft} P_t(z_t,q_t|f_t,\overline{\mathbf{f}},\overline{\mathbf{v}}) \log P_t(z_t|q_t)... \\ &+ \sum_{t=1}^{T}\sum_{q_t}\sum_{z_t}\sum_{f_t} V_{ft} P_t(z_t,q_t|f_t,\overline{\mathbf{f}},\overline{\mathbf{v}}) \log P(f_t|z_t,q_t). \end{aligned} \tag{2.2}$$

In order to ensure that the multinomial distributions sum to 1, we incorporate constraints using Lagrange multipliers, κ, μ_{q_t}, τ_{q_t}, and $\rho_{z,q}$. With the constraints, the above equation becomes:

$$\begin{aligned}\mathcal{L} &= \sum_{q_1} P(q_1|\overline{\mathbf{f}},\overline{\mathbf{v}}) \log P(q_1) + \sum_{t=1}^{T-1}\sum_{q_t}\sum_{q_{t+1}} P_t(q_t,q_{t+1}|\overline{\mathbf{f}},\overline{\mathbf{v}}) \log P(q_{t+1}|q_t)... \\ &+ \sum_{t=1}^{T}\sum_{q_t} P_t(q_t|\overline{\mathbf{f}},\overline{\mathbf{v}}) \log P(v_t|q_t) + \sum_{t=1}^{T}\sum_{q_t}\sum_{z_t}\sum_{f_t} V_{ft} P_t(z_t,q_t|f_t,\overline{\mathbf{f}},\overline{\mathbf{v}}) \log P_t(z_t|q_t)... \\ &+ \sum_{t=1}^{T}\sum_{q_t}\sum_{z_t}\sum_{f_t} V_{ft} P_t(z_t,q_t|f_t,\overline{\mathbf{f}},\overline{\mathbf{v}}) \log P(f_t|z_t,q_t)... \\ &+ \kappa\left(1-\sum_{q_1} P(q_1)\right) + \sum_{q_t}\mu_{q_t}\left(1-\sum_{q_{t+1}} P(q_{t+1}|q_t)\right)... \\ &+ \sum_{t=1}^{T}\sum_{q_t}\tau_{q_t}\left(1-\sum_{z_t} P(z_t|q_t)\right)... \\ &+ \sum_{q}\sum_{z}\rho_{z,q}\left(1-\sum_{f} P(f|z,q)\right). \end{aligned} \tag{2.3}$$

In the M step, we estimate the parameters that maximize the above equation. The posterior distribution is $P(\overline{\mathbf{z}},\overline{\mathbf{q}}|\overline{\mathbf{f}},\overline{\mathbf{v}})$. However, by comparing the third and the fourth steps of Eq. 2.2, we see that we only need a few specific marginalizations of the posterior rather than the entire posterior. Therefore, we only compute the following required marginalizations in the E step:

1. Marginalized posteriors for spectral components and mixture weights — $P_t(z_t, q_t|f_t, \overline{\mathbf{f}}, \overline{\mathbf{v}})$
2. Marginalized posteriors for transition matrix — $P_t(q_t, q_{t+1}|\overline{\mathbf{f}}, \overline{\mathbf{v}})$
3. Marginalized posterior for initial state probabilities — $P(q_1|\overline{\mathbf{f}}, \overline{\mathbf{v}})$
4. Marginalized posteriors for energy distributions — $P_t(q_t|\overline{\mathbf{f}}, \overline{\mathbf{v}})$

We now perform a step by step derivation of the EM equations. We first derive each of the marginalized posteriors that are computed in the E step. We then take the derivative of Eq. 2.3 with respect to each of the parameters to obtain the M step equations.

Marginalized posteriors for spectral components and mixture weights (E Step)

We start with the marginalized posterior that corresponds to an individual draw v at time t. We have:

$$\begin{aligned} P_t(z_{t,v}, q_t|\overline{\mathbf{f}}, \overline{\mathbf{v}}) &= \frac{P_t(z_{t,v}, q_t, \overline{\mathbf{f}}, \overline{\mathbf{v}})}{P(\overline{\mathbf{f}}, \overline{\mathbf{v}})} \\ &= \frac{P_t(z_{t,v}|\overline{\mathbf{f}}, \overline{\mathbf{v}}, q_t)P_t(\overline{\mathbf{f}}, \overline{\mathbf{v}}, q_t)}{P(\overline{\mathbf{f}}, \overline{\mathbf{v}})} \\ &= \frac{P_t(z_{t,v}|f_{t,v}, q_t)P_t(\overline{\mathbf{f}}, \overline{\mathbf{v}}, q_t)}{P(\overline{\mathbf{f}}, \overline{\mathbf{v}})} \\ &= \frac{P_t(\overline{\mathbf{f}}, \overline{\mathbf{v}}, q_t)}{P(\overline{\mathbf{f}}, \overline{\mathbf{v}})} P_t(z_{t,v}|f_{t,v}, q_t) \\ &= \gamma_t(q_t)P_t(z_{t,v}|f_{t,v}, q_t), \end{aligned} \tag{2.4}$$

where $\gamma_t(q_t) = P_t(q_t|\overline{\mathbf{f}}, \overline{\mathbf{v}})$.

In the third step of Eq. 2.4, we use the fact that $z_{t,v}$ is independent of all f's except for $f_{t,v}$ when we are given q_t.

This marginalized posterior is a distribution over $z_{t,v}$ and q_t. The distribution will be the same for all draws at time t for which the observation $f_{t,v}$ is the same.

Since we observe the $f_{t,v}$'s, in theory, we can determine which draws have the same distribution. The problem is that we have the scaled spectrogram data (which is viewed as a number of sound quanta in each time–frequency bin) but we do not know which sound quanta came from which draw. We therefore compute the distribution for each possible value of $f_{t,v}$. This gives us the following relation for all draws in which the observation is f_t (this is the same substitution that was done in Eq. 2.2):

$$P_t(z_{t,v}, q_t|\overline{\mathbf{f}}, \overline{\mathbf{v}}) \;=\; P_t(z_t, q_t|f_t, \overline{\mathbf{f}}, \overline{\mathbf{v}}).$$

Since we have a single distribution for all draws with a given value of $f_{t,v}$, we drop the subscript v. The marginalized posterior is therefore given by:

$$P_t(z_t, q_t|f_t, \overline{\mathbf{f}}, \overline{\mathbf{v}}) \;=\; \gamma_t(q_t) P_t(z_t|f_t, q_t). \tag{2.5}$$

In order to compute this posterior, we therefore need to compute $\gamma_t(q_t)$ and $P_t(z_t|f_t, q_t)$. We start with the computation of $\gamma_t(q_t)$. Since $\gamma_t(q_t) = P_t(q_t|\overline{\mathbf{f}}, \overline{\mathbf{v}})$, we need all observations over all time frames to compute this distribution for a single time frame. This can be done efficiently by using the forward–backward algorithm as in traditional HMMs [47]. We start by defining the forward ($\alpha_t(q_t)$) and backward ($\beta_t(q_t)$) variable as follows:

$$\begin{aligned}
\alpha_t(q_t) &= P(\mathbf{f_1}, \mathbf{f_2}, ..., \mathbf{f_t}, v_1, v_2, ..., v_t, q_t), \\
\beta_t(q_t) &= P(\mathbf{f_{t+1}}, \mathbf{f_{t+2}}, ..., \mathbf{f_T}, v_{t+1}, v_{t+2}, ..., v_T|q_t).
\end{aligned}$$

$\gamma_t(q_t)$ can be computed from the forward and backward variables as follows:

$$\begin{aligned}
\gamma_t(q_t) &= P_t(q_t|\overline{\mathbf{f}}, \overline{\mathbf{v}}) \\
&= \frac{P_t(\overline{\mathbf{f}}, \overline{\mathbf{v}}, q_t)}{\sum_{q_t} P_t(\overline{\mathbf{f}}, \overline{\mathbf{v}}, q_t)} \\
&= \frac{\alpha_t(q_t)\beta_t(q_t)}{\sum_{q_t} \alpha_t(q_t)\beta_t(q_t)}.
\end{aligned}$$

The last step above uses the following relation:

$$\begin{aligned} P_t(\overline{\mathbf{f}}, \overline{\mathbf{v}}, q_t) &= P(\mathbf{f_{t+1}}, \mathbf{f_{t+2}}, ..., \mathbf{f_T}, v_{t+1}, v_{t+2}, ..., v_T | \mathbf{f_1}, \mathbf{f_2}, ..., \mathbf{f_t}, v_1, v_2, ..., v_t, q_t)... \\ &\quad P(\mathbf{f_1}, \mathbf{f_2}, ..., \mathbf{f_t}, v_1, v_2, ..., v_t, q_t) \\ &= P(\mathbf{f_{t+1}}, \mathbf{f_{t+2}}, ..., \mathbf{f_T}, v_{t+1}, v_{t+2}, ..., v_T | q_t)... \\ &\quad P(\mathbf{f_1}, \mathbf{f_2}, ..., \mathbf{f_t}, v_1, v_2, ..., v_t, q_t) \\ &= \alpha_t(q_t)\beta_t(q_t). \end{aligned}$$

Therefore, we can compute $\gamma_t(q_t)$ by computing the forward and backward variables at every time frame. Since it is a distribution over the states, it indicates the contribution of each dictionary at time frame t. In practice, within the first few EM iterations, each time frame is almost entirely explained by a single dictionary. This means that $\gamma_t(q_t)$ will have a value of almost 0 for all states except for a single state, which will have a value of almost 1. This happens because the EM algorithm essentially performs a soft clustering. The fact that $\gamma_t(q_t)$ is still a distribution rather than a hard assignment to a single state is beneficial as it allows small contributions from other states. This is useful for modeling things like the decay of a note from a previous time frame.

The forward and backward variables themselves can be computed recursively. We start with the forward variables. We first initialize the recursion as follows:

$$\begin{aligned} \alpha_1(q_1) &= P(\mathbf{f_1}, v_1, q_1) \\ &= P(\mathbf{f_1}, v_1 | q_1) P(q_1). \end{aligned}$$

We then compute each $\alpha_{t+1}(q_{t+1})$ as follows:

$$\begin{aligned} \alpha_{t+1}(q_{t+1}) &= P(\mathbf{f_1}, \mathbf{f_2}, ..., \mathbf{f_{t+1}}, v_1, v_2, ..., v_{t+1}, q_{t+1}) \\ &= P(\mathbf{f_{t+1}}, v_{t+1} | \mathbf{f_1}, \mathbf{f_2}, ..., \mathbf{f_t}, v_1, v_2, ..., v_t, q_{t+1})... \\ &\quad P(\mathbf{f_1}, \mathbf{f_2}, ..., \mathbf{f_t}, v_1, v_2, ..., v_t, q_{t+1}) \\ &= P(\mathbf{f_{t+1}}, v_{t+1} | q_{t+1}) P(\mathbf{f_1}, \mathbf{f_2}, ..., \mathbf{f_t}, v_1, v_2, ..., v_t, q_{t+1}) \end{aligned}$$

$$
\begin{aligned}
&= \left(\sum_{q_t} P(\mathbf{f_1}, \mathbf{f_2}, ..., \mathbf{f_t}, v_1, v_2, ..., v_t, q_{t+1}, q_t) \right) P(\mathbf{f_{t+1}}, v_{t+1} | q_{t+1}) \\
&= \left(\sum_{q_t} P(q_{t+1} | \mathbf{f_1}, \mathbf{f_2}, ..., \mathbf{f_t}, v_1, v_2, ..., v_t, q_t) P(\mathbf{f_1}, \mathbf{f_2}, ..., \mathbf{f_t}, v_1, v_2, ..., v_t, q_t) \right) ... \\
&\quad P(\mathbf{f_{t+1}}, v_{t+1} | q_{t+1}) \\
&= \left(\sum_{q_t} P(q_{t+1} | q_t) P(\mathbf{f_1}, \mathbf{f_2}, ..., \mathbf{f_t}, v_1, v_2, ..., v_t, q_t) \right) P(\mathbf{f_{t+1}}, v_{t+1} | q_{t+1}) \\
&= \left(\sum_{q_t} P(q_{t+1} | q_t) \alpha_t(q_t) \right) P(\mathbf{f_{t+1}}, v_{t+1} | q_{t+1}).
\end{aligned}
$$

A number of the steps in the above derivation use conditional independence relations.

We now move onto the backward variables. We begin with the final time step T and initialize the recursion as follows:

$$\beta_T(q_T) = 1.$$

We then compute each $\beta_t(q_t)$ as follows:

$$
\begin{aligned}
\beta_t(q_t) &= P(\mathbf{f_{t+1}}, \mathbf{f_{t+2}}, ..., \mathbf{f_T}, v_{t+1}, v_{t+2}, ..., v_T | q_t) \\
&= \sum_{q_{t+1}} P(\mathbf{f_{t+1}}, \mathbf{f_{t+2}}, ..., \mathbf{f_T}, v_{t+1}, v_{t+2}, ..., v_T, q_{t+1} | q_t) \\
&= \sum_{q_{t+1}} P(\mathbf{f_{t+2}}, ..., \mathbf{f_T}, v_{t+2}, ..., v_T | \mathbf{f_{t+1}}, v_{t+1}, q_{t+1}, q_t) P(\mathbf{f_{t+1}}, v_{t+1}, q_{t+1} | q_t) \\
&= \sum_{q_{t+1}} P(\mathbf{f_{t+2}}, ..., \mathbf{f_T}, v_{t+2}, ..., v_T | q_{t+1}) P(\mathbf{f_{t+1}}, v_{t+1}, q_{t+1} | q_t) \\
&= \sum_{q_{t+1}} \beta_{t+1}(q_{t+1}) P(\mathbf{f_{t+1}}, v_{t+1}, q_{t+1} | q_t) \\
&= \sum_{q_{t+1}} \beta_{t+1}(q_{t+1}) P(\mathbf{f_{t+1}}, v_{t+1} | q_{t+1}, q_t) P(q_{t+1} | q_t) \\
&= \sum_{q_{t+1}} \beta_{t+1}(q_{t+1}) P(q_{t+1} | q_t) P(\mathbf{f_{t+1}}, v_{t+1} | q_{t+1}).
\end{aligned}
$$

As with the forward variables, a number of the steps in the above derivation use conditional independence relations.

In the computation of the forward and backward variables, we need the likelihoods,

$P(\mathbf{f_t}, v_t|q_t)$. This gives us the likelihood of a particular state for the given data at time t. It is computed as follows:

$$\begin{aligned} P(\mathbf{f_t}, v_t|q_t) &= P(v_t|q_t)P(\mathbf{f_t}|q_t) \\ &= P(v_t|q_t)\prod_{v=1}^{v_t} P_t(f_{t,v}|q_t) \\ &= P(v_t|q_t)\prod_{f_t} (P_t(f_t|q_t))^{V_{ft}} \\ &= P(v_t|q_t)\prod_{f_t} \left(\sum_{z_t} P_t(f_t, z_t|q_t)\right)^{V_{ft}} \\ &= P(v_t|q_t)\prod_{f_t} \left(\sum_{z_t} P(f_t|z_t, q_t)P_t(z_t|q_t)\right)^{V_{ft}}. \end{aligned}$$

In the first step, we use the fact that the number of draws at time t is independent of the value of the actual draws given the state. Moreover, the draws are independent from each other given the state (second step). In the third step, we group all of the draws for which the observation is f_t. At time t, we would have exactly V_{ft} such draws.

The computation of the likelihoods benefits from the explicit modeling of $\overline{\mathbf{v}}$. If it were not explicitly modeled, the likelihood of a given state at time t would be $P(\mathbf{f_t}|q_t)$. The explicit modeling effectively modulates $P(\mathbf{f_t}|q_t)$ by the energy distribution $P(v_t|q_t)$, whose parameters are estimated in the M step. The likelihood of the state therefore depends not only the shape of corresponding spectra but also on the energy associated with the state. This is particularly useful for data such as speech as certain phonemes consistently have a high energy and certain other phonemes consistently have a low energy. This is also useful for automatically assigning one of the states to explain silence (very low energy).

We now have a way of computing $\gamma_t(q_t)$. As seen in Eq. 2.5, the other distribution that is needed to compute the marginalized posterior $P_t(z_t, q_t|f_t, \overline{\mathbf{f}}, \overline{\mathbf{v}})$ is $P_t(z_t|f_t, q_t)$. This is computed using Baye's rule as follows:

$$P_t(z_t|f_t, q_t) = \frac{P_t(z_t|q_t)P(f_t|z_t, q_t)}{\sum_{z_t} P_t(z_t|q_t)P(f_t|z_t, q_t)}.$$

Marginalized posteriors for transition matrix (E Step)

These marginalized posteriors use the forward and backward variables as well as the likelihoods that were derived above. Apart from these distributions, they use the distribution of the transition matrix, $P(q_{t+1}|q_t)$. For a given pair of time frames, they are computed as follows:

$$\begin{aligned} P_t(q_t, q_{t+1}|\overline{\mathbf{f}}, \overline{\mathbf{v}}) &= \frac{P_t(\overline{\mathbf{f}}, \overline{\mathbf{v}}, q_t, q_{t+1})}{P(\overline{\mathbf{f}}, \overline{\mathbf{v}})} \\ &= \frac{P_t(\overline{\mathbf{f}}, \overline{\mathbf{v}}, q_t, q_{t+1})}{\sum_{q_t}\sum_{q_{t+1}} P_t(\overline{\mathbf{f}}, \overline{\mathbf{v}}, q_t, q_{t+1})}. \end{aligned}$$

$P_t(\overline{\mathbf{f}}, \overline{\mathbf{v}}, q_t, q_{t+1})$, which is in the numerator and the denominator of the above equation, is computed as follows:

$$\begin{aligned} P_t(\overline{\mathbf{f}}, \overline{\mathbf{v}}, q_t, q_{t+1}) &= P(q_{t+1}, \mathbf{f_{t+1}}, \mathbf{f_{t+2}}, ..., \mathbf{f_T}, v_{t+1}, v_{t+2}, ..., v_T|\mathbf{f_1}, \mathbf{f_2}, ..., \mathbf{f_t}, v_1, v_2, ..., v_t, q_t)... \\ &\quad P(\mathbf{f_1}, \mathbf{f_2}, ..., \mathbf{f_t}, v_1, v_2, ..., v_t, q_t) \\ &= P(q_{t+1}, \mathbf{f_{t+1}}, \mathbf{f_{t+2}}, ..., \mathbf{f_T}, v_{t+1}, v_{t+2}, ..., v_T|q_t)\alpha_t(q_t) \\ &= P(\mathbf{f_{t+1}}, \mathbf{f_{t+2}}, ..., \mathbf{f_T}, v_{t+1}, v_{t+2}, ..., v_T|q_{t+1}, q_t)P(q_{t+1}|q_t)\alpha_t(q_t) \\ &= P(\mathbf{f_{t+1}}, \mathbf{f_{t+2}}, ..., \mathbf{f_T}, v_{t+1}, v_{t+2}, ..., v_T|q_{t+1})P(q_{t+1}|q_t)\alpha_t(q_t) \\ &= P(\mathbf{f_{t+1}}, v_{t+1}|\mathbf{f_{t+2}}, ..., \mathbf{f_T}, v_{t+2}, ..., v_T, q_{t+1})... \\ &\quad P(\mathbf{f_{t+2}}, ..., \mathbf{f_T}, v_{t+2}, ..., v_T|q_{t+1})P(q_{t+1}|q_t)\alpha_t(q_t) \\ &= P(\mathbf{f_{t+1}}, v_{t+1}|q_{t+1})\beta_{t+1}(q_{t+1})P(q_{t+1}|q_t)\alpha_t(q_t). \end{aligned}$$

Therefore, the marginalized posterior for a given pair of time frames is given by:

$$P_t(q_t, q_{t+1}|\overline{\mathbf{f}}, \overline{\mathbf{v}}) = \frac{\alpha_t(q_t)P(q_{t+1}|q_t)\beta_{t+1}(q_{t+1})P(\mathbf{f_{t+1}}, v_{t+1}|q_{t+1})}{\sum_{q_t}\sum_{q_{t+1}} \alpha_t(q_t)P(q_{t+1}|q_t)\beta_{t+1}(q_{t+1})P(\mathbf{f_{t+1}}, v_{t+1}|q_{t+1})}.$$

Marginalized posterior for initial state probabilities (E Step)

This marginalized posterior is simply $\gamma_1(q_1)$.

Marginalized posteriors for energy distribution (E Step)

These marginalized posteriors are simply $\gamma_t(q_t)$.

Spectral components (M Step)

The spectral components are multinomial distributions. For a given component z of a given state q, the probability of each frequency is a separate parameter. We take the derivative of Eq. 2.3 with respect to each of these parameters and set them to 0. This gives us the following set of equations (one equation for each value of f):

$$\frac{\sum_t V_{ft} P_t(z_t, q_t | f_t, \overline{\mathbf{f}}, \overline{\mathbf{v}})}{P(f_t | z_t, q_t)} - \rho_{z,q} = 0.$$

By eliminating the Lagrange multiplier, we get the following M step equation:

$$P(f|z,q) = \frac{\sum_t V_{ft} P_t(z, q | f, \overline{\mathbf{f}}, \overline{\mathbf{v}})}{\sum_f \sum_t V_{ft} P_t(z, q | f, \overline{\mathbf{f}}, \overline{\mathbf{v}})}. \tag{2.6}$$

This has an intuitive interpretation. First, consider the numerator. V_{ft} tells us the amount of sound quanta at time-frequency bin f, t. Each sound quanta can have a contribution from each spectral component of each dictionary (although the contributions tend to be almost 0 from all but one dictionaries at a given time frame). $P_t(z, q | f, \overline{\mathbf{f}}, \overline{\mathbf{v}})$ tells us the contribution of spectral component z from dictionary (state) q at time–frequency bin f, t. We multiply this contribution by the total number of sound quanta at that time–frequency bin. The denominator merely normalizes the numerator so that the distribution sums to 1.

Mixture weights (M Step)

The mixture weights are a set of multinomial distributions at every time frame. We have a distribution of mixture weights $P_t(z_t|q_t)$ for each state at each time frame. For a given time frame t and state q_t, the probability of each component z_t is a separate parameter. We take the derivative of Eq. 2.3 with respect to each of these parameters and set them to 0. This gives us the following set of equations (one equation for each

value of z_t):

$$\frac{\sum_{f_t} V_{ft} P_t(z_t, q_t | f_t, \overline{\mathbf{f}}, \overline{\mathbf{v}})}{P_t(z_t|q_t)} - \tau_{q_t} = 0.$$

By eliminating the Lagrange multiplier, we get the following M step equation:

$$P_t(z_t|q_t) = \frac{\sum_{f_t} V_{ft} P_t(z_t, q_t | f_t, \overline{\mathbf{f}}, \overline{\mathbf{v}})}{\sum_{z_t} \sum_{f_t} V_{ft} P_t(z_t, q_t | f_t, \overline{\mathbf{f}}, \overline{\mathbf{v}})}.$$

Like the Eq. 2.6, this equation also has an intuitive interpretation. The numerator is identical to that of Eq. 2.6 except for the summation variable. We estimate the distributions $P_t(z_t|q_t)$ independently for each time frame. For a given time frame, we sum the contributions over all frequencies. The denominator is again merely used for normalization.

Transition matrix (M Step)

The transition matrix is a separate multinomial distribution for each value of the state q_t. For a given state q_t, the probability of each state at the next time frame q_{t+1} is a separate parameter. We take the derivative of Eq. 2.3 with respect to each of these parameters and set them to 0. This gives us the following set of equations (one equation for each value of q_{t+1}):

$$\frac{\sum_{t=1}^{T-1} P(q_t, q_{t+1} | \overline{\mathbf{f}}, \overline{\mathbf{v}})}{P(q_{t+1}|q_t)} - \mu_{q_t} = 0.$$

By eliminating the Lagrange multiplier, we get the following M step equation:

$$P(q_{t+1}|q_t) = \frac{\sum_{t=1}^{T-1} P(q_t, q_{t+1} | \overline{\mathbf{f}}, \overline{\mathbf{v}})}{\sum_{q_{t+1}} \sum_{t=1}^{T-1} P(q_t, q_{t+1} | \overline{\mathbf{f}}, \overline{\mathbf{v}})}.$$

Initial state probabilities (M Step)

This initial state probabilities are a single multinomial distribution. The parameters are the probabilities of starting on a given state. We take the derivative of Eq. 2.3

with respect to each of these parameters and set them to 0. This gives us the following set of equations:

$$\frac{P_1(q_1|\overline{\mathbf{f}},\overline{\mathbf{v}})}{P(q_1)} - \kappa \;=\; 0.$$

By eliminating the Lagrange multiplier, we get the following equation:

$$P(q_1) \;=\; \frac{P_1(q_1|\overline{\mathbf{f}},\overline{\mathbf{v}})}{\sum_{q_1} P_1(q_1|\overline{\mathbf{f}},\overline{\mathbf{v}})}.$$

However, the numerator is the same as $\gamma_1(q_1)$ which is computed in the E step and is already normalized so we can simply set it to $\gamma_1(q_1)$. The M step equation is therefore simply the following assignment:

$$P(q_1) \;=\; \gamma_1(q_1).$$

Energy distributions (M Step)

The energy distributions are univariate Gaussian distributions (one for each state q). They are therefore parameterized by a mean μ_q and a variance σ_q^2. The energy at time t is simply the total number of draws at that time frame[3] and is given by $v_t = \sum_f V_{ft}$. In terms of the data, we have:

$$p(v_t|q) \;=\; \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{\left(\sum_f V_{ft} - \mu_q\right)}{2\sigma^2}\right).$$

We take the derivative of Eq. 2.3 with respect to μ_q and set it to 0. This gives us the following equation:

$$\sum_t P_t(q|\overline{\mathbf{f}},\overline{\mathbf{v}}) \left(\sum_f V_{ft} - \mu_q\right) \;=\; 0.$$

[3]The total number of draws does not give us the actual energy but rather has an intuitive interpretation that is analogous to energy.

Solving for μ_q, we get the following M step equation:

$$\begin{aligned} \mu_q &= \frac{\sum_t P_t(q|\overline{\mathbf{f}}, \overline{\mathbf{v}}) \sum_f V_{ft}}{\sum_t P_t(q|\overline{\mathbf{f}}, \overline{\mathbf{v}})} \\ &= \frac{\sum_t \gamma_t(q) \sum_f V_{ft}}{\sum_t \gamma_t(q)}. \end{aligned}$$

Similarly, we obtain the following M step equation for σ_q^2:

$$\sigma_q^2 = \frac{\sum_t \left(\mu_q - \gamma_t(q) \sum_f V_{ft}\right)^2}{\sum_t \gamma_t(q)}.$$

This has an intuitive interpretation. $\sum_f V_{ft}$ gives us the total number of sound quanta in a given time frame. μ_q tells us the mean sound quanta for state q. If the model was such that each time frame had to be generated by a single state (dictionary), we would simply sum $\sum_f V_{ft}$ over all of the time frames generated by q and divide it by the the total number of time frames generated by q. In practice, this is almost what we are doing. Since each time frame can be generated by multiple states, the amount that time frame t was generated by state q is given by $\gamma_t(q)$. To estimate μ_q, we therefore perform the summation over all time frames but weight each time frame by $\gamma_t(q)$. The intuition for the variance is a straightforward extension of this.

Summary of EM equations

In the E step, we first perform some intermediate computations using the parameters of the model. We then compute the marginalized posteriors using these intermediate computations and the parameters. In the M step, we compute the parameters using the marginalized posteriors. All of the equations are summarized as follows:

E Step - Intermediate Computations

$$
\begin{aligned}
P(\mathbf{f_t}, v_t|q_t) &= P(v_t|q_t)\prod_{f_t}\left(\sum_{z_t} P(f_t|z_t,q_t)P_t(z_t|q_t)\right)^{V_{ft}},\\
\alpha_1(q_1) &= P(\mathbf{f_1}, v_1|q_1)P(q_1),\\
\alpha_{t+1}(q_{t+1}) &= \left(\sum_{q_t} P(q_{t+1}|q_t)\alpha_t(q_t)\right)P(\mathbf{f_{t+1}}, v_{t+1}|q_{t+1}),\\
\beta_T(q_T) &= 1,\\
\beta_t(q_t) &= \sum_{q_{t+1}} \beta_{t+1}(q_{t+1})P(q_{t+1}|q_t)P(\mathbf{f_{t+1}}, v_{t+1}|q_{t+1}),\\
P_t(z_t|f_t,q_t) &= \frac{P_t(z_t|q_t)P(f_t|z_t,q_t)}{\sum_{z_t} P_t(z_t|q_t)P(f_t|z_t,q_t)}.
\end{aligned}
$$

E Step - Marginalized Posteriors

$$
\begin{aligned}
\gamma_t(q_t) &= \frac{\alpha_t(q_t)\beta_t(q_t)}{\sum_{q_t}\alpha_t(q_t)\beta_t(q_t)},\\
P_t(z_t,q_t|f_t,\overline{\mathbf{f}},\overline{\mathbf{v}}) &= \gamma_t(q_t)P_t(z_t|f_t,q_t),\\
P_t(q_t,q_{t+1}|\overline{\mathbf{f}},\overline{\mathbf{v}}) &= \frac{\alpha_t(q_t)P(q_{t+1}|q_t)\beta_{t+1}(q_{t+1})P(\mathbf{f_{t+1}}, v_{t+1}|q_{t+1})}{\sum_{q_t}\sum_{q_{t+1}}\alpha_t(q_t)P(q_{t+1}|q_t)\beta_{t+1}(q_{t+1})P(\mathbf{f_{t+1}}, v_{t+1}|q_{t+1})}.
\end{aligned}
$$

M Step

$$
\begin{aligned}
P(f|z,q) &= \frac{\sum_t V_{ft}P_t(z,q|f,\overline{\mathbf{f}},\overline{\mathbf{v}})}{\sum_f\sum_t V_{ft}P_t(z,q|f,\overline{\mathbf{f}},\overline{\mathbf{v}})},\\
P_t(z_t|q_t) &= \frac{\sum_{f_t} V_{ft}P_t(z_t,q_t|f_t,\overline{\mathbf{f}},\overline{\mathbf{v}})}{\sum_{z_t}\sum_{f_t} V_{ft}P_t(z_t,q_t|f_t,\overline{\mathbf{f}},\overline{\mathbf{v}})},\\
P(q_{t+1}|q_t) &= \frac{\sum_{t=1}^{T-1} P(q_t,q_{t+1}|\overline{\mathbf{f}},\overline{\mathbf{v}})}{\sum_{q_{t+1}}\sum_{t=1}^{T-1} P(q_t,q_{t+1}|\overline{\mathbf{f}},\overline{\mathbf{v}})},\\
P(q_1) &= \gamma_1(q_1),\\
\mu_q &= \frac{\sum_t \gamma_t(q)\sum_f V_{ft}}{\sum_t \gamma_t(q)},\\
\sigma_q^2 &= \frac{\sum_t \left(\mu_q - \gamma_t(q)\sum_f V_{ft}\right)^2}{\sum_t \gamma_t(q)}.
\end{aligned}
$$

Reconstructions

After performing the EM iterations, we may wish to reconstruct the contribution from each dictionary. This is useful for certain applications such as content–aware audio processing, which will be described in Sec. 2.4.6. The reconstruction of the contribution from state q_t at time t is as follows:

$$\begin{aligned} P_t(f_t, q_t|\overline{\mathbf{f}}, \overline{\mathbf{v}}) &= P_t(q_t|\overline{\mathbf{f}}, \overline{\mathbf{v}}) P_t(f_t|q_t, \overline{\mathbf{f}}, \overline{\mathbf{v}}) \\ &= \gamma_t(q_t) P_t(f_t|q_t) \\ &= \gamma_t(q_t) \sum_{z_t} P_t(z_t|q_t) P(f_t|z_t, q_t). \end{aligned} \tag{2.7}$$

This gives us the contribution of each state with respect to the other states at each time frame. However, if we were to sum the contribution of all states, all time frames would have the same gain. We therefore need to modulate $P_t(f_t, q_t|\overline{\mathbf{f}}, \overline{\mathbf{v}})$ by the gain of the original spectrogram. The final reconstruction of the contribution from state q_t to the input data at time t is therefore given by:

$$P_t(f_t, q_t|\overline{\mathbf{f}}, \overline{\mathbf{v}}) \sum_f V_{ft}.$$

2.4.4 Model Selection

Model selection for a given model is the optimal choice of model configurations or user-defined parameters. Akaike information criterion (AIC) [1], Bayesian information criterion (BIC) [58] , and minimum description length (MDL) [52] are commonly used for this purpose. The goal of using these metrics it to find a model that fits the data well while still keeping the model complexity low. We show the use of AIC for N-HMM model selection. The two user-defined parameters in the N-HMM are the number of dictionaries and the number of spectral components per dictionary. AIC is defined as follows:

$$\text{AIC} = -2 \log L + 2k, \tag{2.8}$$

where L is the likelihood of the data with respect to a given set of user-defined parameters (this is different from the complete data log-likelihood in Eq. 2.2). k is the number of parameters in the model (this is different from the user-defined parameters). A lower AIC is considered to be better. Eq. 2.8 has an intuitive interpretation. Log-likelihood (in the first term) is a measure of how well the model fits the data. Higher log-likelihoods are therefore preferred and correspond to a lower AIC. On the other hand, there is a penalty for more complex models as seen in the second term. To use AIC for model selection, we simply compute the AIC for different sets of user-defined parameters and choose the one with the lowest AIC.

Log-likelihood of the data for a given set of user-defined parameters is defined as follows:

$$\begin{aligned} \log L &= \log P(\mathbf{f_1}, \mathbf{f_2}, ..., \mathbf{f_T}, v_1, v_2, ..., v_T) \\ &= \log \sum_{q_T} P(\mathbf{f_1}, \mathbf{f_2}, ..., \mathbf{f_T}, v_1, v_2, ..., v_T, q_T) \\ &= \log \sum_{q_T} \alpha_T(q_T). \end{aligned}$$

Therefore, we can compute AIC for a given model configuration by simply summing the $\alpha_T(q_T)$ over all values of q_T and taking the log. This is identical to the procedure that Rabiner uses to compute the log-likelihood in traditional HMMs [47].

We have used AIC to potentially find the optimal set of user-defined parameters for the N-HMM when the input data is speech. Specifically, the data is the concatenation of nine sentences that have been spoken by a given speaker. The data is from the TIMIT database. We have shown the results of parameter estimation with one such example in Fig. 2.12. We have computed the AIC on 16 different sets of such data (8 male speakers and 8 female speakers) and have reported the average results for the following two cases:

1. We fix the number of spectral components per dictionary at 10 and use different numbers of dictionaries.

2. We fix the number of dictionaries at 40 and use different numbers of spectral components per dictionary.

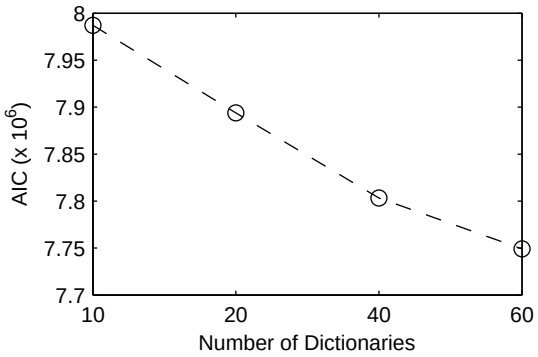


Figure 2.9: AIC for finding the optimal number of dictionaries (the number of spectral components per dictionary has been fixed at 10).

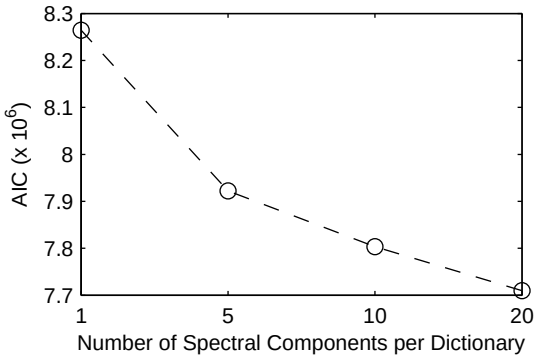


Figure 2.10: AIC for finding the optimal number of spectral components per dictionary (the number of dictionaries has been fixed at 40).

The plots of the AIC when using these metrics are shown in Figs. 2.9, 2.10. We argue that these are not appropriate metrics for model selection in the proposed model for the following two reasons:

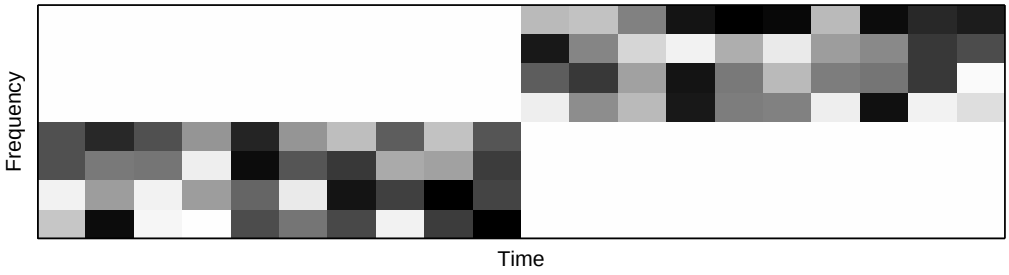
1. The penalty with respect to the number of parameters in the model is insufficient for the proposed model. The first term in Eq. 2.8 therefore dominates the computed AIC. More complex models fit the data better and therefore give a higher log-likelihood. This metric will therefore simply tell us to choose the most complex model. Even if we were to come up with a more appropriate penalty term, we have the following issue.

2. We argue that the optimal choice of user-defined parameters depends on the application. Two different applications can have different optimal parameters. Therefore we suggest that model selection should be performed using metrics that are appropriate for a given application. In Sec. 4.3.3, we perform model selection using source separation metrics that were defined by Vincent et al. [73, 18]. According to Figs. 2.9, 2.10, we should simply choose the most complex model. However the source separation metrics disagree with this hypothesis. Moreover, we hypothesize that the optimal user-defined parameters could be different for other applications. For example, in Sec. 5.2.2, we discuss how one might use the proposed models for automatic music transcription. As discussed there, we hypothesize that the optimal user-defined parameters would be different from those used in source separation. Therefore, even if we were to modify the second term in Eq. 2.8 such that it would not simply tell us to choose the most complex model, the resulting optimal model would not necessarily be optimal for all applications.

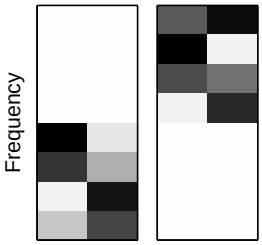
Therefore, we argue that model selection needs to be performed with respect to a given application.

2.4.5 Examples

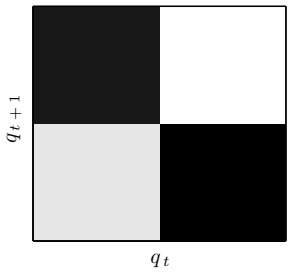
In this section, we illustrate N-HMM parameter estimation on a few examples. In all of the examples, the input to the algorithm is a spectrogram. We show that the estimated parameters in all of these examples have intuitive interpretations. We start with a toy example that was artificially generated and explain the estimated parameters. We then show an example of speech and explain the estimated parameters. Finally, we show an example of notes of music in which we illustrate the reconstructions. This example will be revisited in Sec. 2.4.6. An ergodic (fully-connected) Markov chain has been used in all examples. However, a left–right Markov chain or any other kind of Markov chain could be used for a given data set if desired.



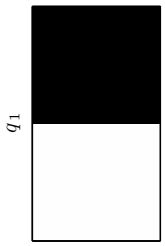
(a) Input Data



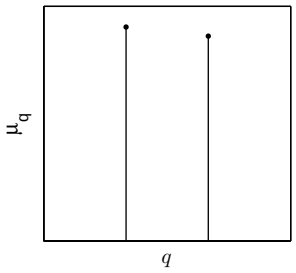
(b) Dictionaries



(c) Transition Matrix



(d) Initial State Probabilities



(e) Energy Means

Figure 2.11: Illustration of N-HMM parameter estimation on a toy example. In all of the figures, black represents a value of 1 and white represents a value of 0.

Toy example

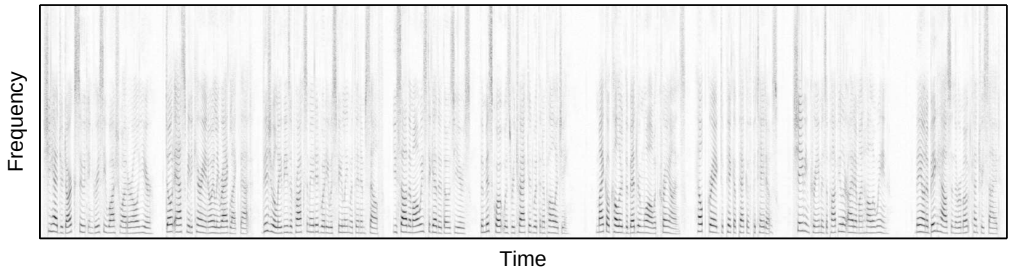
We start with a simple example (Fig. 2.11) in which a matrix that represents a spectrogram (Fig. 2.11a) was artificially generated. This matrix is the input data to the algorithm. It has eight frequencies and twenty time frames. As can be seen, the data in the first ten time frames are quite similar (energy only in the low frequencies) suggesting that it should be explained by a dictionary (state). Similarly, the data in the last ten time frames are quite similar (energy only in the high frequencies) suggesting that it should be explained by another dictionary.

We performed N-HMM parameter estimation on this data using 2 dictionaries and 2 spectral components per dictionary. If we want to use different dictionaries to explain different parts of the data in an intuitive way, it is important to use 2 dictionaries as the input data clearly has two different kinds of segments. The parameters of the model have been jointly learned from the input data as is shown in Figs. 2.11b, 2.11c, 2.11d, and 2.11e. We see that the dictionaries (Fig. 2.11b) have been correctly learned such that each dictionary explains a different segment of the data. The first dictionary can clearly be used to explain the first ten time frames of the data and similarly, the second dictionary can be used to explain the last ten time frames of the data. Particularly, each time frame of the data can be explained as a linear combination of the spectral components in one of the dictionaries. When we look at the spectral components in a given dictionary, we see that both components do not tend to have a high (or low) energy at the same frequency. Either one of the components has a high energy and the other component has a low energy at a given frequency or both components have a moderate energy. This corresponds to the fact that the spectral components in a given dictionary explain different aspects of the data.

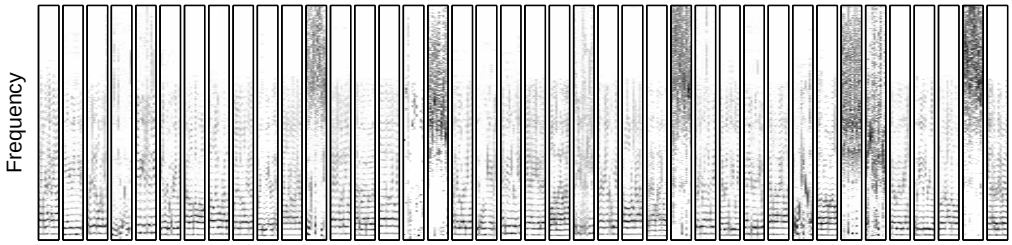
The transition matrix (Fig. 2.11c) is also quite intuitive. As can be seen in the input data, the probability of remaining in a given state (state persistence) is quite high. This can be seen be the strong diagonal in the transition matrix. We can also see from the data that at one of the time frames, there is a transition from state 1 to state 2. This corresponds to the small non-zero probability of $P(q_{t+1} = 2|q_t = 1)$ in the transition matrix. In fact, that probability is 0.1, which corresponds to the fact that there is a transition to state 2 in one out of the ten occurrences of state 1. On the other hand, we have $P(q_{t+1} = 1|q_t = 2) = 0$. This corresponds to the data since there is never a transition from state 2 to state 1.

The initial state probabilities have also been correctly learned as can be seen in Fig. 2.11d). This tells us that the data starts in state 1 with a probability of 1.

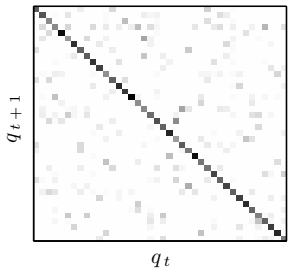
As can be seen in the input data, the segments that correspond to each of the states have a similar energy. The mean of the energy distribution that corresponds to each state, μ_q should therefore be quite similar. This has been correctly learned



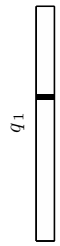
(a) Input Data



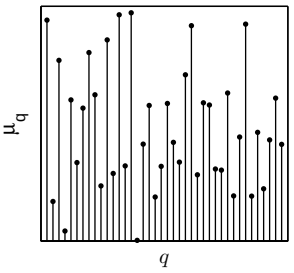
(b) Dictionaries



(c) Transition Matrix



(d) Initial State Probabilities



(e) Energy Means

Figure 2.12: Illustration of N-HMM parameter estimation on a speech example.

as seen in Fig. 2.11e.

Speech example

In this example (Fig. 2.12), we estimate the N-HMM model parameters for a single speaker. The input data is the spectrogram of the concatenation of nine sentences spoken by the given speaker (Fig. 2.12a). This data is from the TIMIT database. The spectrogram was computed using a window size of 64ms and a hop size of 16ms.

We performed N-HMM parameter estimation on this data using 40 dictionaries and 10 spectral components per dictionary. As can be seen in Fig. 2.12b, each

dictionary could plausibly correspond to a phoneme or a part of a phoneme. Some of the dictionaries explain parts of voiced phonemes and some of the dictionaries explain parts of unvoiced phonemes. A given dictionary captures a fair amount of the variations within a given phoneme such as changes in pitch in a voiced phoneme. However, when there are large changes in pitch, different dictionaries tend to be used to explain the variations. If we were to use more dictionaries, more subtle variations within a phoneme would be explained by different dictionaries. On the other hand, if we were to use less dictionaries, more variations would be explained by a single dictionary. Also, a single dictionary might then explain multiple phonemes.

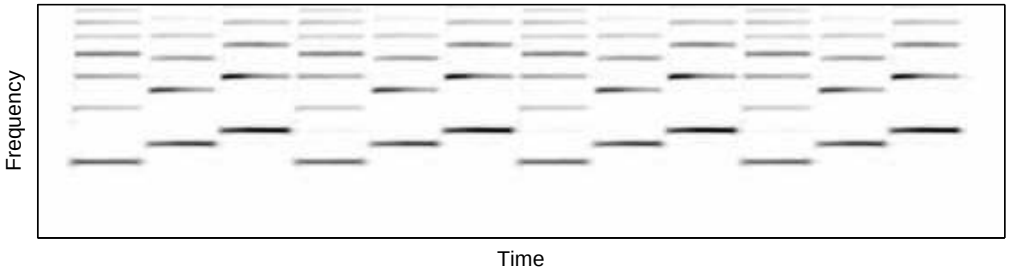
The transition matrix (Fig. 2.12c) correctly learns state persistence as can be seen by the strong diagonal. This is to say that the transition matrix indicates that each given state tends to explain several adjacent time frames, which make intuitive sense.

The initial state probabilities (Fig. 2.12d) indicate that the first time frame should be explained by state 16. μ_q for the energy distribution (Fig. 2.12e) that corresponds to state 16 is almost 0. This indicates that the first frame of the input data has an energy of almost 0. This makes sense as the first frame actually corresponds to silence (low energy noise).

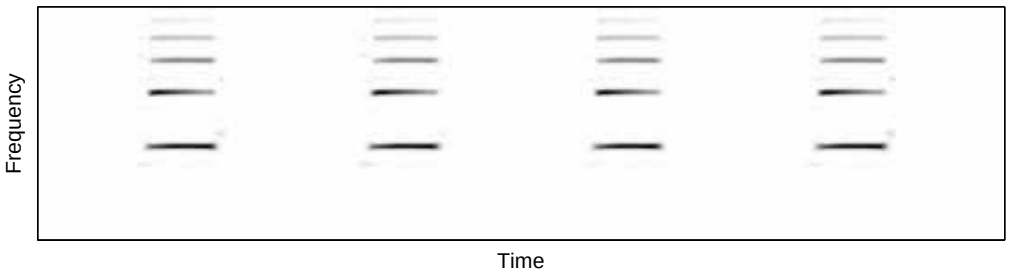
Notes example

In this example (Fig. 2.13), we illustrate the reconstructions of the contributions from the individual dictionaries. The input data is the spectrogram of a synthesized saxophone playing a C major arpeggio four times (Fig. 2.13a). Therefore, we have four repetitions of the sequence C-E-G. The spectrogram was computed using a window size of 100ms and a hop size of 25ms. The constant-Q transform [9] has been used for display purposes (only) so that we can clearly see the fundamental frequencies of the different notes and the relation between the fundamental frequencies.

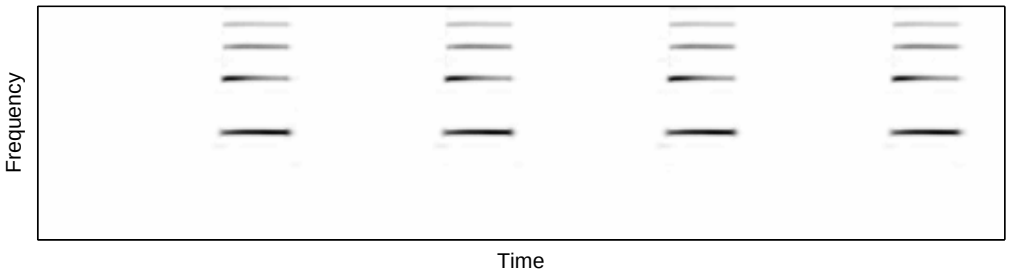
As the data has 3 distinct notes, we performed N-HMM parameter estimation using 3 dictionaries. We used 5 spectral components per dictionary. Using the estimated parameters, we reconstructed the contributions from each of the three dictionaries using Eq. 2.7. The reconstructions are shown in Figs. 2.13b, 2.13c, and 2.13d. We see that each reconstruction correctly captures the contribution from a single note. If



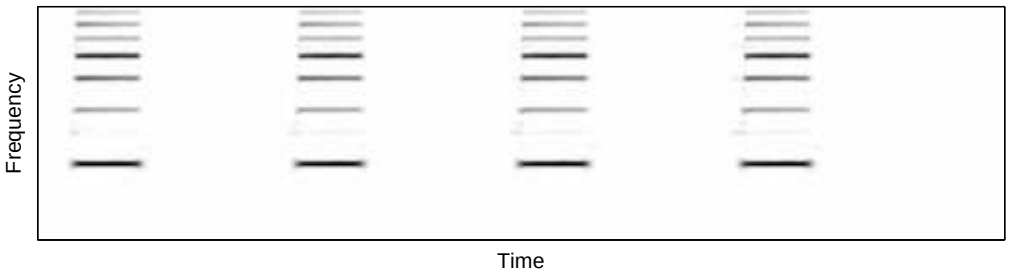
(a) Input Data



(b) Reconstruction from dictionary 1



(c) Reconstruction from dictionary 2



(d) Reconstruction from dictionary 3

Figure 2.13: Illustration of the reconstructions from individual dictionaries. The input data (top figure) is a repetition of a sequence of three notes (major arpeggio). The other three figures show the reconstructions from the individual dictionaries.

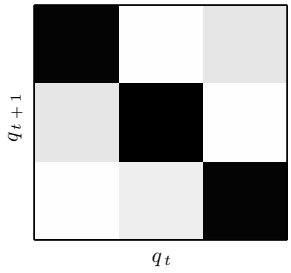


Figure 2.14: Transition matrix for the example of the sequence of notes in Fig. 2.13.

we wish to obtain the audio signals that correspond to each of these reconstructions, we can use the phase of the original STFT and invert each of these reconstructions, going back to the time domain.

The transition matrix (Fig. 2.14) correctly explains the data as well. The strong diagonal corresponds to state persistence. Also, we see that given a particular note, we can transit to one other note with a small probability as indicated by the light gray squares in the transition matrix. We also see that for each note, there is zero probability of transiting to one of the other notes.

2.4.6 Content–Aware Audio Processing

In this section, we discuss an interesting application of the N-HMM, content–aware audio processing. This means that a certain aspect of an input signal, which can be specified by high level information, is processed. This has numerous practical applications from making subtle volume changes to a particular aspect of a recording to completely changing the musical structure of a recording.

The procedure is as follows:

1. Estimate the N-HMM parameters of a given spectrogram.

2. Reconstruct the spectrograms that correspond to the contributions of each dictionary.

3. Using the reconstructed spectrograms and the phase of the original STFT, obtain the time domain signals that correspond to each dictionary, using inverse STFTs.

4. Process a subset of the time domain signals.

5. Sum all of the time domain signals (including the processed ones).

It should be noted that the phase of the original STFT is used to obtain the time domain signals that correspond to each of the individual dictionaries. Each time frame of the spectrogram is explained almost exclusively by a single dictionary, as shown in Fig. 2.13. In the reconstructed spectrograms (that correspond to the individual dictionaries), each time frame either corresponds almost exactly to the original spectrogram or has a magnitude of almost zero. Therefore, the portions of a given reconstructed spectrogram that corresponds to the original spectrogram will correspond to the phase of the original STFT. The other portions will not correspond to the phase of the original STFT but will have a magnitude of almost zero and are inconsequential. Therefore, the phase of the original STFT can be used to obtain the time domain signals from the reconstructed spectrograms.

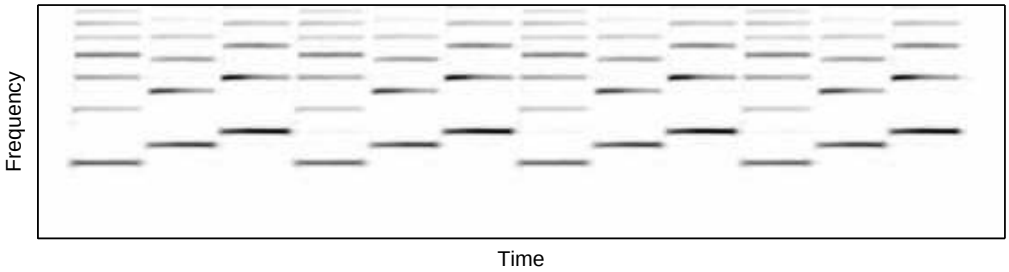
Conversion of a Major Arpeggio to Minor Arpeggios

We start with the example in Fig. 2.13. We performed N-HMM parameter estimation and reconstructed the contribution of the individual dictionaries. We then had a separate reconstruction of each note of the arpeggio. We could therefore pitch shift any of them without affecting the other reconstructions. In the first example (Fig. 2.15b), we pitch shifted the third (contribution from the first dictionary) down by a semitone. We then got the arpeggio, C-Eb-G, in the parallel minor scale.

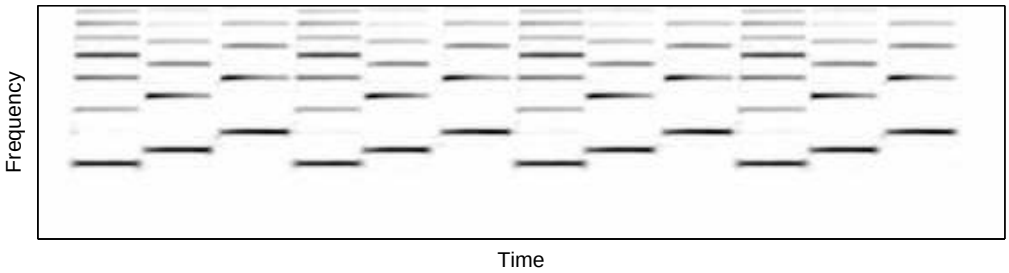
In the next example (Fig. 2.15c), we pitch shifted the fifth (contribution from the second dictionary) down by 10 semitones. This shifted the note G down to A. The sequence A-C-E is an arpeggio in the relative minor scale. In this example, we therefore have this arpeggio but start on the third (C).

Processing the Snare Drum in a Drum Loop

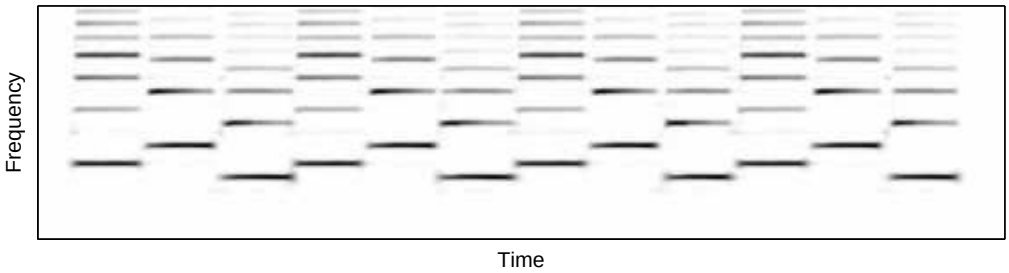
In this example (Fig. 2.16), we performed N-HMM parameter estimation on the spectrogram of a drum loop using 5 dictionaries and 20 components per dictionary. We then performed the reconstructions from the dictionaries. We found that one



(a) Input data is a major arpeggio.

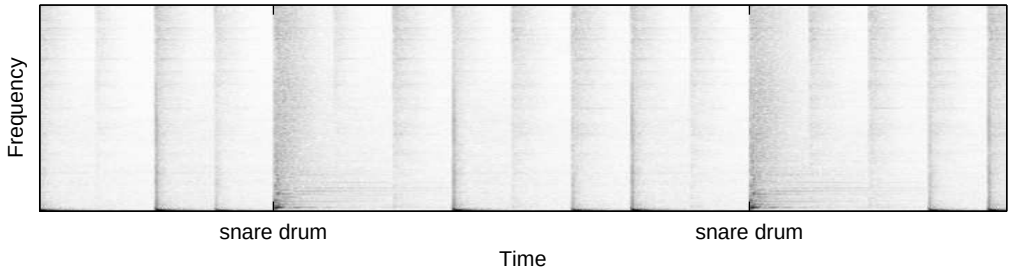


(b) Conversion to parallel minor arpeggio by flattening the third by a semitone.

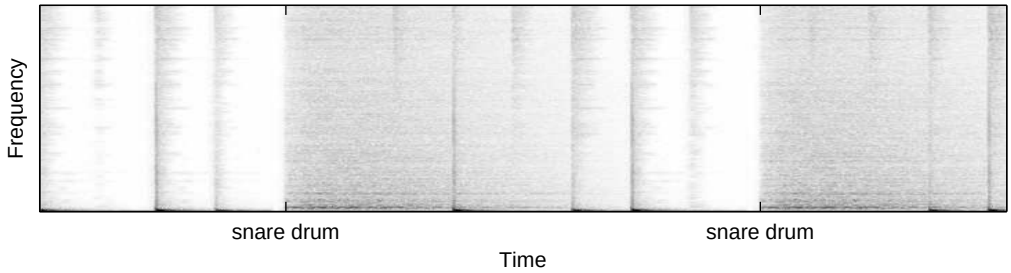


(c) Conversion to relative minor arpeggio (starting on the third) by flattening the fifth by ten semitones.

Figure 2.15: Illustration of the conversion of a major arpeggio to minor arpeggios.



(a) Input data is a drum loop.



(b) Artificial reverberation has been applied to only the snare drum as can be seen by two sections of heavy shading.

Figure 2.16: Illustration of processing the snare drum in a drum loop.

of the reconstructions corresponded to the snare drum. Artificial reverberation was applied to this reconstruction and then all of the reconstructions were put back together (without processing the other drums). The end effect is processing only the snare drum in a drum loop without affecting the other drums. This is shown in Fig. 2.16b. Note the reverberation tail that follows each snare drum hit.

2.5 Conclusions

In this chapter, we presented a new probabilistic model of single sources, the non-negative hidden Markov model (N-HMM). Spectral structure and temporal dynamics of a given source are jointly modeled by the N-HMM.

We first described two classes of existing models of single sources. Non-negative spectrogram factorization methods can be interpreted as dictionary learning methods that give us rich spectral models of sources but fail to model the non-stationarity of audio and do not capture the temporal dynamics. Traditional hidden Markov models,

on the other hand, can model the temporal dynamics of a source but have limitations when it comes to reconstructions, due to a rigid observation model.

The N-HMM yields rich spectral models by using the concept of spectral dictionaries. However, they account for non-stationarity by learning multiple small dictionaries rather than a single large dictionary. They model temporal dynamics by learning a Markov chain. They also have a flexible observation model with time–varying mixture weights that make them amenable to high quality reconstructions.

We conceptually described the N-HMM using the concept of dictionaries. We then described the probabilistic model using an HMM type framework. This was followed by a derivation of parameter estimation in the N-HMM. We illustrated this with a few examples. We then discussed model selection and argued that it should be performed with respect to an application.

Finally, we described an application of the N-HMM, content–aware audio processing. This is just one application of the model. Another important place in which the model will be used is in single channel supervised source separation as will be described in Chapter 4. The N-HMM will be used to estimate model parameters of single sources. Certain parameters of this model will subsequently be used in the model of sound mixtures which will be described in Chapter 3. Although source separation is one application of the model of sound mixtures in which the N-HMM parameter estimation will form in an important part, there are several other applications that can be explored. This will be discussed in Chapter 5.

Chapter 3

Models of Sound Mixtures

3.1 Introduction

In this chapter, we propose a model of sound mixtures, the non-negative factorial hidden Markov model (N-FHMM). We begin by reviewing extensions of non-negative spectrogram factorizations for modeling sound mixtures. We then introduce factorial hidden Markov models (FHMMs). This will be followed by a detailed explanation of the proposed model.

The general idea behind the proposed model and existing models of sound mixtures that we address is to combine models of single sources into a coherent framework to model mixtures. Specifically, N-HMMs of individual sources are combined in order to form the N-FHMM. Throughout the chapter, the models will be explained in the context of two sources. However, extending the models to more sources is straightforward.

3.2 Non-negative Spectrogram Factorization

Non-negative spectrogram factorizations, as discussed in Sec. 2.2, are used to learn a dictionary of spectral components for a given source. This idea can be extended to sound mixtures. Our discussion will focus on extending latent variable spectral models, as discussed in Sec. 2.2.2. Particularly, we concentrate on the asymmetric

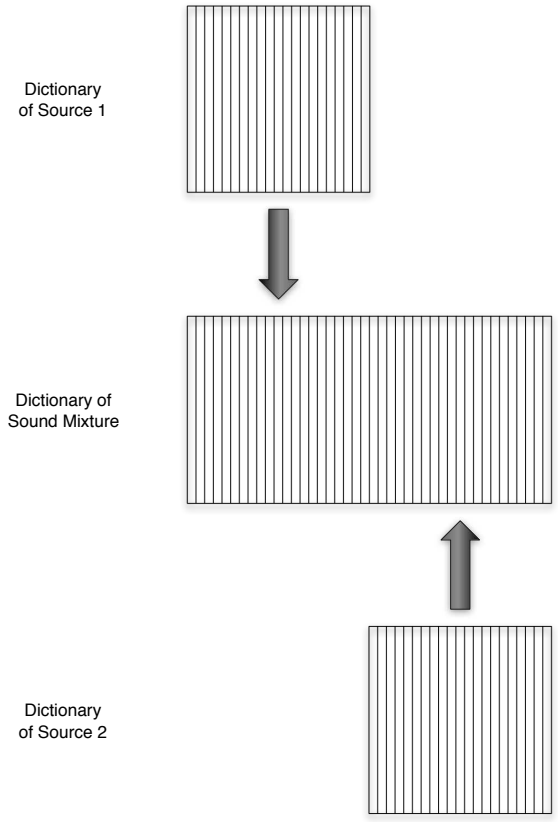


Figure 3.1: Illustration of concatenating dictionaries of individual sources to model a sound mixture.

factorization, developed by Raj and Smaragdis [48]. However, conceptually similar ideas are used to extend most such non-negative spectrogram factorization methods. For example, the same ideas have been used in the case of the symmetric factorization model developed by Smaragdis et al. [68].

In the asymmetric factorization model, the idea is that each sound source in the mixture will correspond to its own dictionary. We model the mixture with a dictionary that is simply the concatenation of the individual dictionaries (Fig. 3.1). A (different) linear combination of the spectral components of the concatenated dictionary explains each column of the mixture spectrogram. The model is given by:

$$P_t(f_t) \;=\; \sum_{s_t}\sum_{z_t} P(f_t|z_t,s_t)P_t(z_t,s_t),$$

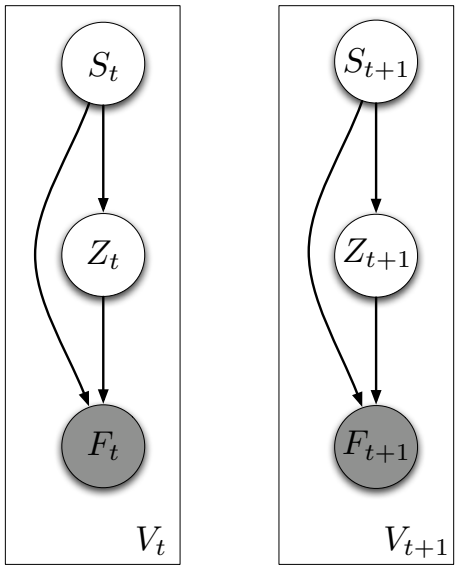


Figure 3.2: Graphical model for asymmetric factorization [48] of spectrograms of sound mixtures.

where $P(f_t|z_t, s_t)$ is spectral component z_t of source s_t[1]. $P_t(z_t, s_t)$ is the distribution of mixture weights at time t. Using the fact that $P_t(z_t, s_t) = P_t(s_t)P_t(z_t|s_t)$, we express the model as:

$$P_t(f_t) \quad = \quad \sum_{s_t} P_t(s_t) \sum_{z_t} P_t(z_t|s_t) P(f_t|z_t, s_t).$$

This gives us a distribution of mixture weights $P_t(s_t)$ at the source level as well as a distribution of mixture weights $P_t(z_t|s_t)$ for the spectral components of a given source.

The generative process for a given time frame is as follows (the graphical model is shown in Fig. 3.2):

1. Choose a source according to $P_t(s_t)$.

2. Choose a latent variable according to $P_t(z_t|s_t)$.

[1]As with single source models, it should be noted that $P(f_t|z_t, s_t)$ is not time dependent. The subscripts are used here to indicate the time dependence of the spectrogram and the mixture weights. We however estimate a single spectral component for each value of z and s.

3. Choose a frequency according to $P(f|z_t, s_t)$.

4. Repeat steps 1–3 V_t times, where $V_t = \sum_t V_{ft}$ (the total number of observed sound quanta at time frame t).

As with single source models, given the spectrogram, we can estimate the parameters of the model using the EM algorithm. This is done as follows:

E Step

$$P_t(z_t, s_t|f_t) = \frac{P_t(s_t)P_t(z_t|s_t)P(f_t|z_t, s_t)}{\sum_{s_t}\sum_{z_t} P_t(s_t)P_t(z_t|s_t)P(f_t|z_t, s_t)}.$$

M Step

$$\begin{aligned} P(f|z, s) &= \frac{\sum_t V_{ft}P_t(z, s|f)}{\sum_f \sum_t V_{ft}P_t(z, s|f)}, \\ P_t(z_t|s_t) &= \frac{\sum_{f_t} V_{ft}P_t(z_t, s_t|f_t)}{\sum_{z_t}\sum_{f_t} V_{ft}P_t(z_t, s_t|f_t)}, \\ P_t(s_t) &= \frac{\sum_{z_t}\sum_{f_t} V_{ft}P_t(z_t, s_t|f_t)}{\sum_{s_t}\sum_{z_t}\sum_{f_t} V_{ft}P_t(z_t, s_t|f_t)}. \end{aligned}$$

The parameters of the model include spectral components for each source as well as all of the mixture weights. This is a highly unconstrained problem and in practice will not yield meaningful solutions if we estimate all of the parameters from the spectrogram of the sound mixture. We therefore fix a subset of the parameters and estimate the remaining parameters from the spectrogram of the sound mixture. The specific parameters that we fix depends on the application. For example, in supervised source separation, we learn the spectral components of each individual source from training data of that source. When we are given the sound mixture, we fix the pre-learned spectral components of the individual sources and estimate only the mixture weights. This will be revisited in Chapter 4.

In semi-supervised separation, we learn the spectral components of one of the sources from training data. When we are given the sound mixture, we fix the pre-learned spectral components of that source and estimate the spectral components of the other source as well as the mixture weights over both sources.

It should be noted that rather than fixing certain parameters, we could also apply priors on the parameters as done by Smaragdis and Mysore [66] and Ganseman et al. [22].

As with single source models, the time frames are modeled as being independent. This is a serious limitation as it the ignores temporal dynamics of the individual sources.

3.3 Factorial Hidden Markov Models

Ghahramani and Jordan introduced factorial hidden Markov models (FHMMs) [23], which can be used to model multiple time series data that have a common observation (the graphical model[2] is shown in Fig. 3.3). They are therefore naturally suited to model sound mixtures. In these models, each source is modeled by a separate HMM. Therefore, there is an independent sequence of hidden states that correspond to each source. With two sources, $q_1^{(1)}, q_2^{(1)}, ..., q_T^{(1)}$ corresponds to the first source and $q_1^{(2)}, q_2^{(2)}, ..., q_T^{(2)}$ corresponds to the second source. At time t, the probability of a pair of hidden states given the pair of hidden states in the previous time frame, is given by $P(q_{t+1}^{(1)}, q_{t+1}^{(2)} | q_t^{(1)}, q_t^{(2)})$. However, due to the independence of the state sequences of the two sources, we have:

$$P(q_{t+1}^{(1)}, q_{t+1}^{(2)} | q_t^{(1)}, q_t^{(2)}) \quad = \quad P(q_{t+1}^{(1)} | q_t^{(1)}) P(q_{t+1}^{(2)} | q_t^{(2)}).$$

The temporal dynamics of each source is therefore decoupled and incorporated in the model. They can be learned from separate training data of each individual source. Although the state sequences are a priori independent, they give rise to a common observation f_t at each time frame that is dependent on the states $q_t^{(1)}$ and $q_t^{(2)}$ of each individual source at that time frame. The interaction model[3] is therefore

[2]The graphical model is diagrammatically a bit different from that of the original paper [23] in order to facilitate easier comparison to the proposed model. However, the models are identical and the difference is only diagrammatic. Such diagrammatic changes will be seen with other graphical models in the thesis as well in order to facilitate easier comparisons.

[3]This is analogous to the observation model in HMMs. It is called the interaction model because it describes the generation of the observation as a function of the interaction of the individual sources.

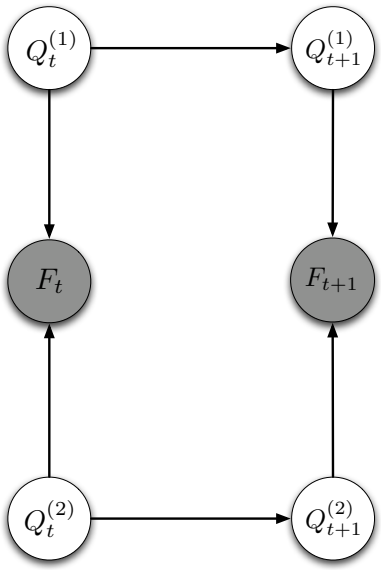


Figure 3.3: Graphical model for the FHMM [23].

$P(f_t|q_t^{(1)}, q_t^{(2)})$. The states at a given time frame therefore become coupled, given the observation at that time frame (v-structure in a graphical model). At each time frame, we therefore have to consider $P(f_t|q_t^{(1)}, q_t^{(2)})$ for each pair of states. If each source has N possible states, we need to consider N^2 possible state combinations.

The model (in terms of the observations) is given by:

$$P(\overline{\mathbf{f}}) = \sum_{\overline{\mathbf{q}}} P(q_1^{(1)})P(q_1^{(2)}) \prod_{t=1}^{T-1} P(q_{t+1}^{(1)}|q_t^{(1)})P(q_{t+1}^{(2)}|q_t^{(2)}) \prod_{t=1}^{T} P(f_t|q_t^{(1)}, q_t^{(2)}).$$

The generative process is as follows:

1. Choose initial states according to $P(q_1^{(1)})$ and $P(q_1^{(2)})$.
2. Set $t = 1$.
3. Choose an observation according to $P(f_t|q_t^{(1)}, q_t^{(2)})$.
4. Transit to new states according to $P(q_{t+1}^{(1)}|q_t^{(1)})$ and $P(q_{t+1}^{(2)}|q_t^{(2)})$.
5. Set $t = t + 1$ and go to step 3 if $t < T$.

FHMMs have been used for various applications that involve sound mixtures. They have been used for source separation by Roweis [54], Hershey et al. [25], Virtanen [74], and Ozerov et al. [44]. They have been used in concurrent speech recognition of multiple speakers by Hershey et al. [26] and Virtanen [74]. They have been used in the concurrent pitch estimation of multiple instruments by Bach and Jordan [2]. We discuss the application of FHMMs to source separation in more detail in Chapter 4.

3.4 Non-negative Factorial Hidden Markov Models

In this section, we describe the proposed model, the non-negative factorial hidden Markov model (N-FHMM). We start with a conceptual explanation of the model in terms of dictionaries. We then describe the probabilistic model. This is followed by an explanation of parameter estimation in the model.

3.4.1 Conceptual Explanation

As with non-negative spectrogram factorizations, N-FHMMs model each column of the spectrogram as a linear combination of the spectral components from a dictionary. In non-negative spectrogram factorizations, each column of the mixture spectrogram is modeled by a dictionary that is a concatenation of the dictionaries of the individual sources.

In the N-FHMM however, each source has multiple dictionaries. As with the N-HMM, each dictionary of a given source corresponds to a state of that source. In a given time frame, each source will be in a particular state. Therefore, each source is modeled by a single dictionary in that time frame. The sound mixture will then be modeled by a dictionary that is the concatenation of the active dictionaries of the individual sources. If each source has N states, the sound mixture can be explained with any one of the N^2 possible combinations of dictionaries in that time frame. This is illustrated in Fig. 3.4 with a simple example in which each source has two dictionaries.

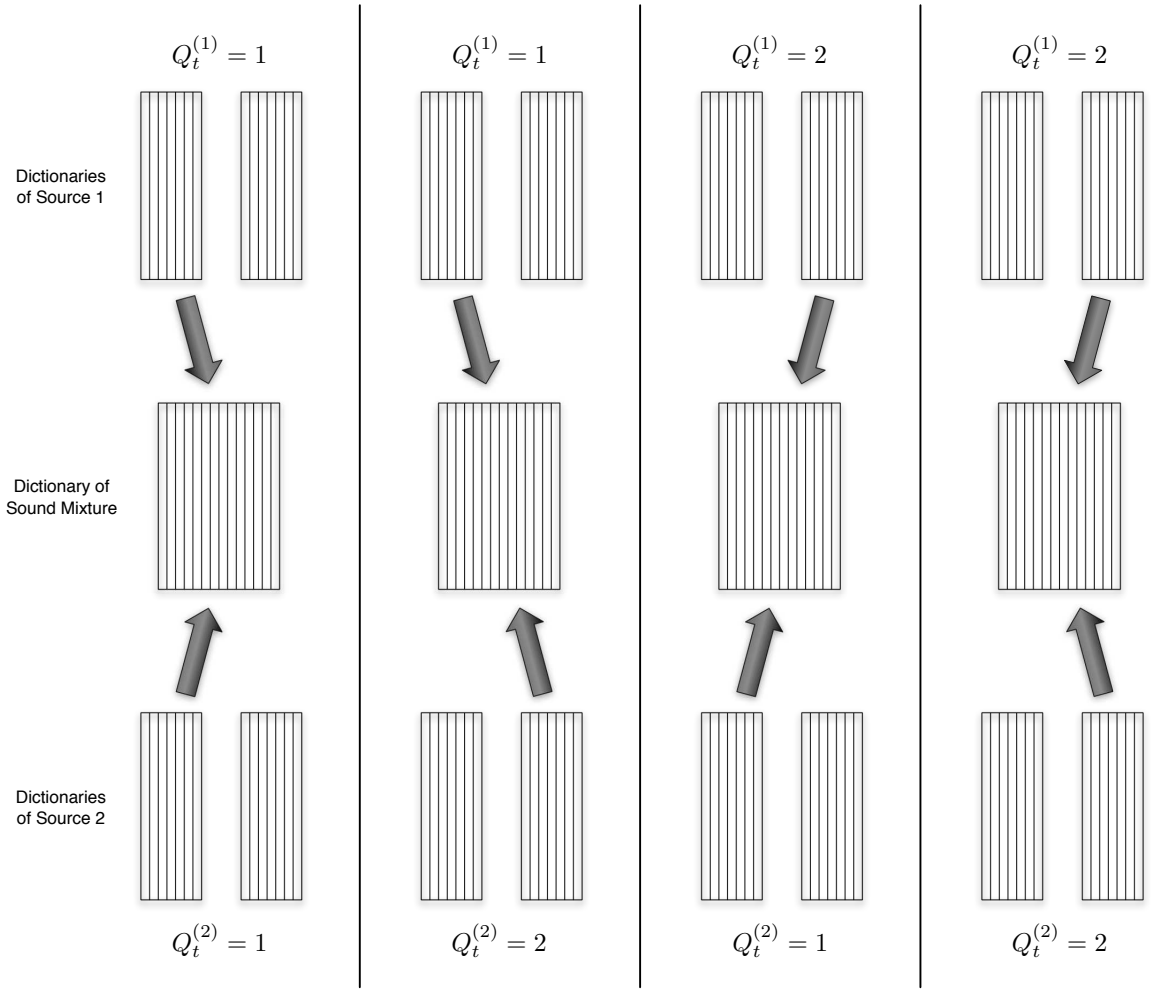


Figure 3.4: Illustration of the different combinations of dictionaries that can be used to model a time frame using the N-FHMM.

3.4.2 Probabilistic Model

The probabilistic model of the N-FHMM combines N-HMMs of individual sources through an interaction model. Since sound sources are additive, we use an additive interaction model. Since we model the spectrogram rather than the complex STFT, the sources are not strictly additive. However, we found that they can be well approximated as additive, as commonly done with non-negative spectrogram factorizations [48, 67].

We introduce a random variable s_t that is used to model the mixing proportions of the sources at time frame t. The interaction model tells us the probability of an observation f_t as a function of the interaction between the sources. If source 1 is in state $q_t^{(1)}$ and source 2 is in state $q_t^{(2)}$, the interaction model is given by:

$$\begin{aligned} P(f_t|q_t^{(1)}, q_t^{(2)}) &= \sum_{s_t} P_t(s_t|q_t^{(1)}, q_t^{(2)}) P(f_t|s_t, q_t^{(1)}, q_t^{(2)}) \\ &= \sum_{s_t} P_t(s_t|q_t^{(1)}, q_t^{(2)}) P(f_t|s_t, q_t^{(s_t)}). \end{aligned}$$

$P(f_t|s_t, q_t^{(s_t)})$ gives us a distribution by which the observation of source s_t is generated. It is essentially the observation model of the corresponding N-HMM. In the second step, we used the fact that an observation that is generated by a given source is independent of the state of the other source. $P_t(s_t|q_t^{(1)}, q_t^{(2)})$ gives us a time-dependent distribution by which each of these independent observations are mixed. This is reflective of the additive interaction.

The observation model of an individual source is given by:

$$P_t(f_t|s_t, q_t^{(s_t)}) = \sum_{z_t} P_t(z_t|s_t, q_t^{(s_t)}) P(f_t|z_t, s_t, q_t^{(s_t)}),$$

where $P(f_t|z_t, s_t, q_t^{(s)})$ is the spectral component z_t of state q_t of source s_t. $P_t(z_t|s_t, q_t^{(s_t)})$ is the distribution of mixture weights for state q_t of source s_t.

The graphical model shown is in Fig. 3.5. Comparing to Fig. 2.8, we see the graphical models of the individual N-HMMs (one on top and one on the bottom). Also, given a pairs of states, we model the observation as a linear combination of the

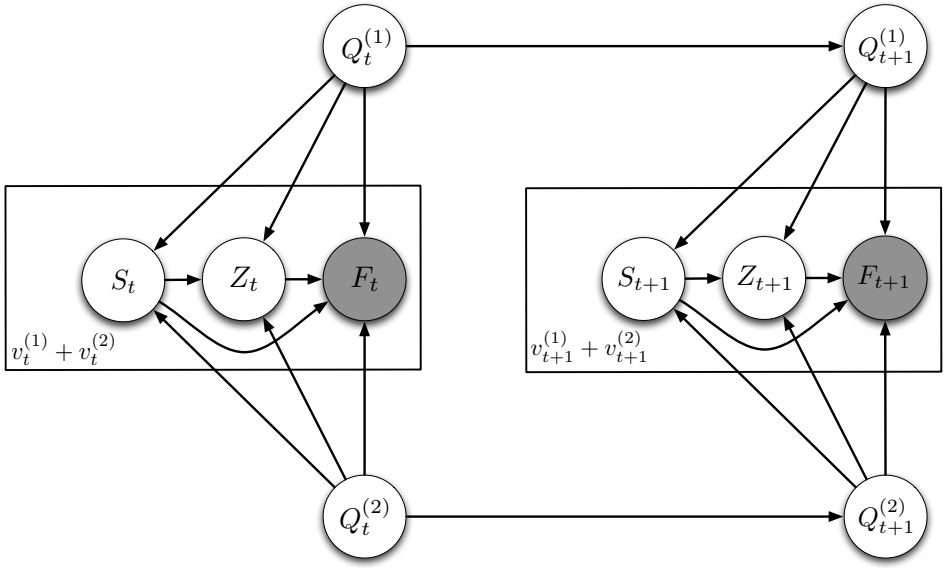


Figure 3.5: Graphical model for the N-FHMM

spectral components of the appropriate concatenated dictionary. Therefore, given a pair of states in a particular time frame, the graphical model is essentially the same as that of non-negative spectrogram factorization. This can be seen by comparing to the graphical model in Fig. 3.2. Particularly, the model for each time frame in Fig. 3.2, is horizontally laid out in Fig. 3.5.

The total number of draws v_t in a given time frame is observed. However, the number of draws from the individual sources, $v_t^{(1)}$ and $v_t^{(2)}$, are hidden. They are related as follows:

$$v_t \;=\; v_t^{(1)} + v_t^{(2)}.$$

Since v_t is the sum of two random variables, its distribution (given the states) is the convolution of the distributions of $v_t^{(1)}$ and $v_t^{(2)}$. Therefore, we have:

$$P(v_t|q_t^{(1)}, q_t^{(2)}) \;=\; \int_1^{v_t^{(1)}} P(v_t^{(1)}|q_t^{(1)})P(v_t - v_t^{(1)}|q_t^{(2)})\mathrm{d}v_t^{(1)}.$$

Since $P(v_t^{(1)}|q_t^{(1)})$ and $P(v_t^{(2)}|q_t^{(2)})$ are independent Gaussian distributions, if the

parameters are known, the parameters of $P(v_t|q_t^{(1)}, q_t^{(2)})$ will also be known. Specifically, if we have:

$$v_t^{(1)}|q_t^{(1)} \sim \mathcal{N}\left(\mu_{q_t^{(1)}}, \sigma^2_{q_t^{(1)}}\right),$$

and

$$v_t^{(2)}|q_t^{(2)} \sim \mathcal{N}\left(\mu_{q_t^{(2)}}, \sigma^2_{q_t^{(2)}}\right),$$

then we will have:

$$v_t|q_t^{(1)}, q_t^{(2)} \sim \mathcal{N}\left(\mu_{q_t^{(1)}} + \mu_{q_t^{(2)}}, \sigma^2_{q_t^{(1)}} + \sigma^2_{q_t^{(2)}}\right).$$

Along with the interaction model and the energy distributions, the model of the N-FHMM also contains the transition matrices, $P(q_{t+1}^{(1)}|q_t^{(1)})$ and $P(q_{t+1}^{(2)}|q_t^{(2)})$, and the initial state probabilities, $P(q_1^{(1)})$ and $P(q_1^{(2)})$. Putting all of these things together, the model of the N-FHMM (in terms of the observations as draws) is given by:

$$\begin{aligned} P(\overline{\mathbf{f}}, \overline{\mathbf{v}}) \;=\; & \sum_{\mathbf{q^{(1)}}} \sum_{\mathbf{q^{(2)}}} \sum_{\overline{\mathbf{s}}} \sum_{\overline{\mathbf{z}}} P(q_1^{(1)}) P(q_1^{(2)}) \ldots \\ & \left(\prod_{t=1}^{T-1} P(q_{t+1}^{(1)}|q_t^{(1)})\right) \left(\prod_{t=1}^{T-1} P(q_{t+1}^{(2)}|q_t^{(2)})\right) \left(\prod_{t=1}^{T} P(v_t|q_t^{(1)}, q_t^{(2)})\right) \ldots \\ & \left(\prod_{t=1}^{T-1} \prod_{v=1}^{v_t} P_t(s_{t,v}|q_t^{(1)}, q_t^{(2)}) P_t(z_{t,v}|s_{t,v}, q_t^{(s_{t,v})}) P(f_{t,v}|z_{t,v}, s_{t,v}, q_t^{(s_{t,v})})\right). \end{aligned}$$

The generative process is as follows:

1. Choose initial states according to $P(q_1^{(1)})$ and $P(q_1^{(2)})$.
2. Set $t = 1$.
3. Choose the number of draws for the given time frame according to $P(v_t^{(1)}|q_t^{(1)})$ and $P(v_t^{(2)}|q_t^{(2)})$.
4. Repeat the following steps $v_t^{(1)} + v_t^{(2)}$ times:
   (a) Choose a source according to $P_t(s_{t,v}|q_t^{(1)}, q_t^{(2)})$.

(b) Choose a spectral component according to $P_t(z_{t,v}|s_{t,v}, q_t^{(s_{t,v})})$.

(c) Choose a frequency according to $P(f_{t,v}|z_{t,v}, s_{t,v}, q_t^{(s_{t,v})})$.

5. Transit to new states according to $P(q_{t+1}^{(1)}|q_t^{(1)})$ and $P(q_{t+1}^{(2)}|q_t^{(2)})$.

6. Set $t = t + 1$ and go to step 3 if $t < T$.

In a given time frame, the sources are additive as described by $P_t(s_t|q_t^{(1)}, q_t^{(2)})$ and the spectral components within a given source are additive according to $P_t(z_t|s_t, q_t^{(s_t)})$. Therefore, the spectral components of the two sources are additive. This corresponds to the fact that a given time frame is explained by a linear combination of the spectral components that correspond to the concatenation of the active dictionaries of each source. We therefore use the distribution $P_t(z_t, s_t|q_t^{(1)}, q_t^{(2)})$ to model a single set of mixture weights over both sources rather than using the individual distributions. They are related as follows:

$$\begin{aligned} P_t(z_t, s_t|q_t^{(1)}, q_t^{(2)}) &= P_t(s_t|q_t^{(1)}, q_t^{(2)})P_t(z_t|s_t, q_t^{(1)}, q_t^{(2)}) \\ &= P_t(s_t|q_t^{(1)}, q_t^{(2)})P_t(z_t|s_t, q_t^{(s_t)}). \end{aligned}$$

If a given application requires the individual distributions, they can be easily obtained by marginalizing $P_t(z_t, s_t|q_t^{(1)}, q_t^{(2)})$.

3.4.3 Parameter Estimation

We now describe the procedure for the estimation of the parameters of the N-FHMM from a given spectrogram. The parameters that we estimate are as follows:

1. Spectral components (multinomial distributions) — $P(f|z, s, q^{(s)})$

2. Mixture weights (multinomial distributions) — $P_t(z_t, s_t|q_t^{(1)}, q_t^{(2)})$

3. Transition matrices (multinomial distributions) — $P(q_{t+1}^{(1)}|q_t^{(1)})$ and $P(q_{t+1}^{(2)}|q_t^{(2)})$

4. Initial state probabilities (multinomial distribution) —$P(q_1^{(1)})$ and $P(q_1^{(2)})$

Given an input scaled spectrogram V_{ft} of a sound mixture, we estimate the parameters using the EM algorithm. As with non-negative spectrogram factorization, discussed in Sec. 3.2, this is a highly unconstrained problem and will not yield meaningful solutions if we estimate all of the parameters from the mixture spectrogram. Therefore, in practice, we fix some of the parameters and estimate the remaining parameters. The specific parameters that we fix depends on the application. As an alternative to fixing these parameters, we could apply priors on them. We therefore derive the parameter estimation equations for all parameters (except the energy distributions).

In the E step of the EM algorithm, we compute the posterior distribution given the parameters. In the M step,we estimate the parameters by maximizing the expected value of the the complete data log likelihood with respect to the posterior distribution.

The posterior distribution of the model is the probability of the hidden variables given the observations and is given by:

$$P(\overline{\mathbf{z}}, \overline{\mathbf{s}}, \overline{\mathbf{q^{(1)}}}, \overline{\mathbf{q^{(2)}}} | \overline{\mathbf{f}}, \overline{\mathbf{v}}).$$

The complete data log-likelihood is given by:

$$\begin{aligned}
\log P(\overline{\mathbf{z}}, \overline{\mathbf{s}}, \overline{\mathbf{q^{(1)}}}, \overline{\mathbf{q^{(2)}}}, \overline{\mathbf{f}}, \overline{\mathbf{v}}) &= \log P(q_1^{(1)}) + \log P(q_1^{(2)}) + \sum_{t=1}^{T-1} \log P(q_{t+1}^{(1)} | q_t^{(1)}) ... \\
&+ \sum_{t=1}^{T-1} \log P(q_{t+1}^{(2)} | q_t^{(2)}) + \sum_{t=1}^{T} \log P(v_t | q_t^{(1)}, q_t^{(2)}) ... \\
&+ \sum_{t=1}^{T} \sum_{v=1}^{v_t} \log P_t(z_{t,v}, s_{t,v} | q_t^{(1)}, q_t^{(2)}) ... \\
&+ \sum_{t=1}^{T} \sum_{v=1}^{v_t} \log P(f_{t,v} | z_{t,v}, s_{t,v}, q_t^{(s_{t,v})}).
\end{aligned}$$

The expected value of the complete data log likelihood with respect to the posterior distribution is given by:

$$\begin{aligned}
\mathcal{L} &= E_{\overline{\mathbf{z}}, \overline{\mathbf{s}}, \overline{\mathbf{q^{(1)}}}, \overline{\mathbf{q^{(2)}}} | \overline{\mathbf{f}}, \overline{\mathbf{v}}} \left[\log P(\overline{\mathbf{z}}, \overline{\mathbf{s}}, \overline{\mathbf{q^{(1)}}}, \overline{\mathbf{q^{(2)}}}, \overline{\mathbf{f}}, \overline{\mathbf{v}}) \right] \\
&= \sum_{\overline{\mathbf{q^{(1)}}}} \sum_{\overline{\mathbf{q^{(2)}}}} \sum_{\overline{\mathbf{s}}} \sum_{\overline{\mathbf{z}}} P(\overline{\mathbf{z}}, \overline{\mathbf{s}}, \overline{\mathbf{q^{(1)}}}, \overline{\mathbf{q^{(2)}}} | \overline{\mathbf{f}}, \overline{\mathbf{v}}) \log P(\overline{\mathbf{z}}, \overline{\mathbf{s}}, \overline{\mathbf{q^{(1)}}}, \overline{\mathbf{q^{(2)}}}, \overline{\mathbf{f}}, \overline{\mathbf{v}})
\end{aligned}$$

$$
\begin{aligned}
= & \sum_{\overline{\mathbf{q}^{(1)}}} \sum_{\overline{\mathbf{q}^{(2)}}} \sum_{\overline{\mathbf{s}}} \sum_{\overline{\mathbf{z}}} P(\overline{\mathbf{z}}, \overline{\mathbf{s}}, \overline{\mathbf{q}^{(1)}}, \overline{\mathbf{q}^{(2)}} | \overline{\mathbf{f}}, \overline{\mathbf{v}}) \log P(q_1^{(1)}) \\
& + \sum_{\overline{\mathbf{q}^{(1)}}} \sum_{\overline{\mathbf{q}^{(2)}}} \sum_{\overline{\mathbf{s}}} \sum_{\overline{\mathbf{z}}} P(\overline{\mathbf{z}}, \overline{\mathbf{s}}, \overline{\mathbf{q}^{(1)}}, \overline{\mathbf{q}^{(2)}} | \overline{\mathbf{f}}, \overline{\mathbf{v}}) \log P(q_1^{(2)}) ... \\
& + \sum_{t=1}^{T-1} \sum_{\overline{\mathbf{q}^{(1)}}} \sum_{\overline{\mathbf{q}^{(2)}}} \sum_{\overline{\mathbf{s}}} \sum_{\overline{\mathbf{z}}} P(\overline{\mathbf{z}}, \overline{\mathbf{s}}, \overline{\mathbf{q}^{(1)}}, \overline{\mathbf{q}^{(2)}} | \overline{\mathbf{f}}, \overline{\mathbf{v}}) \log P(q_{t+1}^{(1)} | q_t^{(1)}) ... \\
& + \sum_{t=1}^{T-1} \sum_{\overline{\mathbf{q}^{(1)}}} \sum_{\overline{\mathbf{q}^{(2)}}} \sum_{\overline{\mathbf{s}}} \sum_{\overline{\mathbf{z}}} P(\overline{\mathbf{z}}, \overline{\mathbf{s}}, \overline{\mathbf{q}^{(1)}}, \overline{\mathbf{q}^{(2)}} | \overline{\mathbf{f}}, \overline{\mathbf{v}}) \log P(q_{t+1}^{(2)} | q_t^{(2)}) ... \\
& + \sum_{t=1}^{T} \sum_{\overline{\mathbf{q}^{(1)}}} \sum_{\overline{\mathbf{q}^{(2)}}} \sum_{\overline{\mathbf{s}}} \sum_{\overline{\mathbf{z}}} P(\overline{\mathbf{z}}, \overline{\mathbf{s}}, \overline{\mathbf{q}^{(1)}}, \overline{\mathbf{q}^{(2)}} | \overline{\mathbf{f}}, \overline{\mathbf{v}}) \log P(v_t | q_t^{(1)}, q_t^{(2)}) ... \\
& + \sum_{t=1}^{T} \sum_{v=1}^{v_t} \sum_{\overline{\mathbf{q}^{(1)}}} \sum_{\overline{\mathbf{q}^{(2)}}} \sum_{\overline{\mathbf{s}}} \sum_{\overline{\mathbf{z}}} P(\overline{\mathbf{z}}, \overline{\mathbf{s}}, \overline{\mathbf{q}^{(1)}}, \overline{\mathbf{q}^{(2)}} | \overline{\mathbf{f}}, \overline{\mathbf{v}}) \log P_t(z_{t,v}, s_{t,v} | q_t^{(1)}, q_t^{(2)}) ... \\
& + \sum_{t=1}^{T} \sum_{v=1}^{v_t} \sum_{\overline{\mathbf{q}^{(1)}}} \sum_{\overline{\mathbf{q}^{(2)}}} \sum_{\overline{\mathbf{s}}} \sum_{\overline{\mathbf{z}}} P(\overline{\mathbf{z}}, \overline{\mathbf{s}}, \overline{\mathbf{q}^{(1)}}, \overline{\mathbf{q}^{(2)}} | \overline{\mathbf{f}}, \overline{\mathbf{v}}) \log P(f_{t,v} | z_{t,v}, s_{t,v}, q_t^{(s_{t,v})}). \quad (3.1)
\end{aligned}
$$

Certain variables get marginalized in each of the above terms. This gives us the following equation:

$$
\begin{aligned}
\mathcal{L} = & \sum_{q_1^{(1)}} P(q_1^{(1)} | \overline{\mathbf{f}}, \overline{\mathbf{v}}) \log P(q_1^{(1)}) + \sum_{q_1^{(2)}} P(q_1^{(2)} | \overline{\mathbf{f}}, \overline{\mathbf{v}}) \log P(q_1^{(2)}) \\
& + \sum_{t=1}^{T-1} \sum_{q_t^{(1)}} \sum_{q_{t+1}^{(1)}} P(q_t^{(1)}, q_{t+1}^{(1)} | \overline{\mathbf{f}}, \overline{\mathbf{v}}) \log P(q_{t+1}^{(1)} | q_t^{(1)}) ... \\
& + \sum_{t=1}^{T-1} \sum_{q_t^{(2)}} \sum_{q_{t+1}^{(2)}} P(q_t^{(2)}, q_{t+1}^{(2)} | \overline{\mathbf{f}}, \overline{\mathbf{v}}) \log P(q_{t+1}^{(2)} | q_t^{(2)}) ... \\
& + \sum_{t=1}^{T} \sum_{q_t^{(1)}} \sum_{q_t^{(2)}} P(q_t^{(1)}, q_t^{(2)} | \overline{\mathbf{f}}, \overline{\mathbf{v}}) \log P(v_t | q_t^{(1)}, q_t^{(2)}) ... \\
& + \sum_{t=1}^{T} \sum_{v=1}^{v_t} \sum_{q_t^{(1)}} \sum_{q_t^{(2)}} \sum_{s_{t,v}} \sum_{z_{t,v}} P(z_{t,v}, s_{t,v}, q_t^{(1)}, q_t^{(2)} | \overline{\mathbf{f}}, \overline{\mathbf{v}}) \log P_t(z_{t,v}, s_{t,v} | q_t^{(1)}, q_t^{(2)}) ... \\
& + \sum_{t=1}^{T} \sum_{v=1}^{v_t} \sum_{q_t^{(1)}} \sum_{q_t^{(2)}} \sum_{s_{t,v}} \sum_{z_{t,v}} P(z_{t,v}, s_{t,v}, q_t^{(1)}, q_t^{(2)} | \overline{\mathbf{f}}, \overline{\mathbf{v}}) \log P(f_{t,v} | z_{t,v}, s_{t,v}, q_t^{(s_{t,v})}) \quad (3.2)
\end{aligned}
$$

In the last two terms, we change the summations to be over frequencies rather than draws by counting the number of draws of frequency f_t at time t. This is given by V_{ft} (scaled spectrogram). Since we sum over frequencies, we change the distribution $P_t(z_{t,v}, s_{t,v}, q_t^{(1)}, q_t^{(2)}|\overline{\mathbf{f}}, \overline{\mathbf{v}})$ to $P_t(z_t, s_t, q_t^{(1)}, q_t^{(2)}|f_t, \overline{\mathbf{f}}, \overline{\mathbf{v}})$. This gives us a distribution over z_t, s_t, $q_t^{(1)}$, and $q_t^{(2)}$ for all draws at time t that result in the observation f_t. The above equation then becomes:

$$\begin{aligned}
\mathcal{L} &= \sum_{q_1^{(1)}} P(q_1^{(1)}|\overline{\mathbf{f}}, \overline{\mathbf{v}}) \log P(q_1^{(1)}) + \sum_{q_1^{(2)}} P(q_1^{(2)}|\overline{\mathbf{f}}, \overline{\mathbf{v}}) \log P(q_1^{(2)}) \\
&+ \sum_{t=1}^{T-1} \sum_{q_t^{(1)}} \sum_{q_{t+1}^{(1)}} P(q_t^{(1)}, q_{t+1}^{(1)}|\overline{\mathbf{f}}, \overline{\mathbf{v}}) \log P(q_{t+1}^{(1)}|q_t^{(1)}) ... \\
&+ \sum_{t=1}^{T-1} \sum_{q_t^{(2)}} \sum_{q_{t+1}^{(2)}} P(q_t^{(2)}, q_{t+1}^{(2)}|\overline{\mathbf{f}}, \overline{\mathbf{v}}) \log P(q_{t+1}^{(2)}|q_t^{(2)}) ... \\
&+ \sum_{t=1}^{T} \sum_{q_t^{(1)}} \sum_{q_t^{(2)}} P(q_t^{(1)}, q_t^{(2)}|\overline{\mathbf{f}}, \overline{\mathbf{v}}) \log P(v_t|q_t^{(1)}, q_t^{(2)}) ... \\
&+ \sum_{t=1}^{T} \sum_{q_t^{(1)}} \sum_{q_t^{(2)}} \sum_{s_t} \sum_{z_t} \sum_{f_t} V_{ft} P_t(z_t, s_t, q_t^{(1)}, q_t^{(2)}|f_t, \overline{\mathbf{f}}, \overline{\mathbf{v}}) \log P_t(z_t, s_t|q_t^{(1)}, q_t^{(2)}) ... \\
&+ \sum_{t=1}^{T} \sum_{q_t^{(1)}} \sum_{q_t^{(2)}} \sum_{s_t} \sum_{z_t} \sum_{f_t} V_{ft} P_t(z_t, s_t, q_t^{(1)}, q_t^{(2)}|f_t, \overline{\mathbf{f}}, \overline{\mathbf{v}}) \log P(f_t|z_t, s_t, q_t^{(s_t)}). \quad (3.3)
\end{aligned}$$

In order to ensure that the multinomial distributions sum to 1, we incorporate constraints using Lagrange multipliers, $\kappa^{(1)}$, $\kappa^{(2)}$, $\mu_{q_t^{(1)}}$, $\mu_{q_t^{(2)}}$, $\tau_{q_t^{(1)}, q_t^{(2)}}$, and $\rho_{z,s,q}$. With the constraints, the above equation becomes:

$$\begin{aligned}
\mathcal{L} &= \sum_{q_1^{(1)}} P(q_1^{(1)}|\overline{\mathbf{f}}, \overline{\mathbf{v}}) \log P(q_1^{(1)}) + \sum_{q_1^{(2)}} P(q_1^{(2)}|\overline{\mathbf{f}}, \overline{\mathbf{v}}) \log P(q_1^{(2)}) \\
&+ \sum_{t=1}^{T-1} \sum_{q_t^{(1)}} \sum_{q_{t+1}^{(1)}} P(q_t^{(1)}, q_{t+1}^{(1)}|\overline{\mathbf{f}}, \overline{\mathbf{v}}) \log P(q_{t+1}^{(1)}|q_t^{(1)}) ...
\end{aligned}$$

$$
\begin{aligned}
&+\sum_{t=1}^{T-1}\sum_{q_t^{(2)}}\sum_{q_{t+1}^{(2)}} P(q_t^{(2)}, q_{t+1}^{(2)}|\overline{\mathbf{f}},\overline{\mathbf{v}}) \log P(q_{t+1}^{(2)}|q_t^{(2)})... \\
&+\sum_{t=1}^{T}\sum_{q_t^{(1)}}\sum_{q_t^{(2)}} P(q_t^{(1)}, q_t^{(2)}|\overline{\mathbf{f}},\overline{\mathbf{v}}) \log P(v_t|q_t^{(1)}, q_t^{(2)})... \\
&+\sum_{t=1}^{T}\sum_{q_t^{(1)}}\sum_{q_t^{(2)}}\sum_{s_t}\sum_{z_t}\sum_{f_t} V_{ft} P_t(z_t, s_t, q_t^{(1)}, q_t^{(2)}|f_t,\overline{\mathbf{f}},\overline{\mathbf{v}}) \log P_t(z_t, s_t|q_t^{(1)}, q_t^{(2)})... \\
&+\sum_{t=1}^{T}\sum_{q_t^{(1)}}\sum_{q_t^{(2)}}\sum_{s_t}\sum_{z_t}\sum_{f_t} V_{ft} P_t(z_t, s_t, q_t^{(1)}, q_t^{(2)}|f_t,\overline{\mathbf{f}},\overline{\mathbf{v}}) \log P(f_t|z_t, s_t, q_t^{(s_t)}) \\
&+\kappa^{(1)}\left(1-\sum_{q_1^{(1)}} P(q_1^{(1)})\right)+\kappa^{(2)}\left(1-\sum_{q_1^{(2)}} P(q_1^{(2)})\right) \\
&+\sum_{q_t^{(1)}}\mu_{q_t^{(1)}}\left(1-\sum_{q_{t+1}^{(1)}} P(q_{t+1}^{(1)}|q_t^{(1)})\right)+\sum_{q_t^{(2)}}\mu_{q_t^{(2)}}\left(1-\sum_{q_{t+1}^{(2)}} P(q_{t+1}^{(2)}|q_t^{(2)})\right)... \\
&+\sum_{t=1}^{T}\sum_{q_t^{(1)}}\sum_{q_t^{(2)}}\tau_{q_t^{(1)},q_t^{(2)}}\left(1-\sum_{s_t}\sum_{z_t} P_t(z_t, s_t|q_t^{(1)}, q_t^{(2)})\right)... \\
&+\sum_{s}\sum_{q}\sum_{z}\rho_{z,s,q}\left(1-\sum_{f} P(f|z, s, q^{(s)})\right). \qquad (3.4)
\end{aligned}
$$

In the M step, we estimate the parameters that maximize the above equation. The posterior distribution is $P(\overline{\mathbf{z}},\overline{\mathbf{s}},\overline{\mathbf{q^{(1)}}},\overline{\mathbf{q^{(2)}}}|\overline{\mathbf{f}},\overline{\mathbf{v}})$. However, by comparing Eq. 3.1 and Eq. 3.2, we see that we only need a few specific marginalizations of the posterior rather than the entire posterior. Therefore, we only compute the following required marginalizations in the E step:

1. Marginalized posteriors for spectral components and weights — $P_t(z_t, s_t, q_t^{(1)}, q_t^{(2)}|f_t,\overline{\mathbf{f}},\overline{\mathbf{v}})$

2. Marginalized posteriors for transition matrices — $P(q_t^{(1)}, q_{t+1}^{(1)}|\overline{\mathbf{f}},\overline{\mathbf{v}})$ and $P(q_t^{(2)}, q_{t+1}^{(2)}|\overline{\mathbf{f}},\overline{\mathbf{v}})$

3. Marginalized posteriors for initial state probabilities — $P(q_1^{(1)}|\overline{\mathbf{f}},\overline{\mathbf{v}})$ and $P(q_1^{(2)}|\overline{\mathbf{f}},\overline{\mathbf{v}})$

We now perform a step by step derivation of the EM equations. We first derive each of the marginalized posteriors that are computed in the E step. We then take the derivative of Eq. 3.4 with respect to each of the parameters to obtain the M step equations.

Marginalized posteriors for spectral components and mixture weights (E Step)

We start with the marginalized posterior that corresponds to an individual draw v at time t. We have:

$$\begin{aligned}
P_t(z_{t,v}, s_{t,v}, q_t^{(1)}, q_t^{(2)}|\overline{\mathbf{f}}, \overline{\mathbf{v}}) &= \frac{P_t(z_{t,v}, s_{t,v}, q_t^{(1)}, q_t^{(2)}, \overline{\mathbf{f}}, \overline{\mathbf{v}})}{P(\overline{\mathbf{f}}, \overline{\mathbf{v}})} \\
&= \frac{P_t(z_{t,v}, s_{t,v}|\overline{\mathbf{f}}, \overline{\mathbf{v}}, q_t^{(1)}, q_t^{(2)}) P_t(\overline{\mathbf{f}}, \overline{\mathbf{v}}, q_t^{(1)}, q_t^{(2)})}{P(\overline{\mathbf{f}}, \overline{\mathbf{v}})} \\
&= \frac{P_t(z_{t,v}, s_{t,v}|f_{t,v}, q_t^{(1)}, q_t^{(2)}) P_t(\overline{\mathbf{f}}, \overline{\mathbf{v}}, q_t^{(1)}, q_t^{(2)})}{P(\overline{\mathbf{f}}, \overline{\mathbf{v}})} \\
&= \frac{P_t(\overline{\mathbf{f}}, \overline{\mathbf{v}}, q_t^{(1)}, q_t^{(2)})}{P(\overline{\mathbf{f}}, \overline{\mathbf{v}})} P_t(z_{t,v}, s_{t,v}|f_{t,v}, q_t^{(1)}, q_t^{(2)}) \\
&= \gamma_t(q_t^{(1)}, q_t^{(2)}) P_t(z_{t,v}, s_{t,v}|f_{t,v}, q_t^{(1)}, q_t^{(2)}), \qquad (3.5)
\end{aligned}$$

where $\gamma_t(q_t^{(1)}, q_t^{(2)}) = P_t(q_t^{(1)}, q_t^{(2)}|\overline{\mathbf{f}}, \overline{\mathbf{v}})$.

In the third step of Eq. 3.5, we use the fact that $z_{t,v}$ and $s_{t,v}$ are independent of all f's except for $f_{t,v}$ when we are given $q_t^{(1)}$ and $q_t^{(2)}$.

This posterior is a distribution over $z_{t,v}$, $s_{t,v}$, $q_t^{(1)}$, and $q_t^{(2)}$. The distribution will be the same for all draws at time t for which the observation $f_{t,v}$ is the same. Since we observe the $f_{t,v}$'s, in theory, we can determine which draws have the same distribution. The problem is that we have the scaled spectrogram data (which is viewed as a number of sound quanta in each time–frequency bin) but we do not know which sound quanta came from which draw. We therefore compute the distribution for each possible value of $f_{t,v}$. This gives us the following relation for all draws in

which the observation is f_t (this is the same substitution that was done in Eq. 3.3):

$$P_t(z_{t,v}, s_{t,v}, q_t^{(1)}, q_t^{(2)}|\overline{\mathbf{f}}, \overline{\mathbf{v}}) \;=\; P_t(z_t, s_t, q_t^{(1)}, q_t^{(2)}|f_t, \overline{\mathbf{f}}, \overline{\mathbf{v}}).$$

Since we have a single distribution for all draws with a given value of $f_{t,v}$, we drop the subscript v. The marginalized posterior is therefore given by:

$$P_t(z_t, s_t, q_t^{(1)}, q_t^{(2)}|f_t, \overline{\mathbf{f}}, \overline{\mathbf{v}}) \;=\; \gamma_t(q_t^{(1)}, q_t^{(2)}) P_t(z_t, s_t|f_t, q_t^{(1)}, q_t^{(2)}). \tag{3.6}$$

In order to compute this posterior, we therefore need to compute $\gamma_t(q_t^{(1)}, q_t^{(2)})$ and $P_t(z_t, s_t|f_t, q_t^{(1)}, q_t^{(2)})$. We start with the computation of $\gamma_t(q_t^{(1)}, q_t^{(2)})$. This is done with a two dimensional forward–backward algorithm. We start by defining the forward ($\alpha_t(q_t^{(1)}, q_t^{(2)})$) and backward ($\beta_t(q_t^{(1)}, q_t^{(2)})$) variable as follows:

$$\begin{aligned}
\alpha_t(q_t^{(1)}, q_t^{(2)}) &= P(\mathbf{f_1}, \mathbf{f_2}, ..., \mathbf{f_t}, v_1, v_2, ..., v_t, q_t^{(1)}, q_t^{(2)}), \\
\beta_t(q_t^{(1)}, q_t^{(2)}) &= P(\mathbf{f_{t+1}}, \mathbf{f_{t+2}}, ..., \mathbf{f_T}, v_{t+1}, v_{t+2}, ..., v_T|q_t^{(1)}, q_t^{(2)}).
\end{aligned}$$

$\gamma_t(q_t^{(1)}, q_t^{(2)})$ can be computed from the forward and backward variables as follows:

$$\begin{aligned}
\gamma_t(q_t^{(1)}, q_t^{(2)}) &= P_t(q_t^{(1)}, q_t^{(2)}|\overline{\mathbf{f}}, \overline{\mathbf{v}}) \\
&= \frac{P_t(\overline{\mathbf{f}}, \overline{\mathbf{v}}, q_t^{(1)}, q_t^{(2)})}{\sum_{q_t^{(1)}} \sum_{q_t^{(2)}} P_t(\overline{\mathbf{f}}, \overline{\mathbf{v}}, q_t^{(1)}, q_t^{(2)})} \\
&= \frac{\alpha_t(q_t^{(1)}, q_t^{(2)}) \beta_t(q_t^{(1)}, q_t^{(2)})}{\sum_{q_t^{(1)}} \sum_{q_t^{(2)}} \alpha_t(q_t^{(1)}, q_t^{(2)}) \beta_t(q_t^{(1)}, q_t^{(2)})}.
\end{aligned}$$

The last step above uses the following relation:

$$\begin{aligned}
P_t(\overline{\mathbf{f}}, \overline{\mathbf{v}}, q_t^{(1)}, q_t^{(2)}) &= P(\mathbf{f_{t+1}}, \mathbf{f_{t+2}}, ..., \mathbf{f_T}, v_{t+1}, v_{t+2}, ..., v_T|\mathbf{f_1}, \mathbf{f_2}, ..., \mathbf{f_t}, v_1, v_2, ..., v_t, q_t^{(1)}, q_t^{(2)}) ... \\
&\quad P(\mathbf{f_1}, \mathbf{f_2}, ..., \mathbf{f_t}, v_1, v_2, ..., v_t, q_t^{(1)}, q_t^{(2)}) \\
&= P(\mathbf{f_{t+1}}, \mathbf{f_{t+2}}, ..., \mathbf{f_T}, v_{t+1}, v_{t+2}, ..., v_T|q_t^{(1)}, q_t^{(2)}) ... \\
&\quad P(\mathbf{f_1}, \mathbf{f_2}, ..., \mathbf{f_t}, v_1, v_2, ..., v_t, q_t^{(1)}, q_t^{(2)}) \\
&= \alpha_t(q_t^{(1)}, q_t^{(2)}) \beta_t(q_t^{(1)}, q_t^{(2)}).
\end{aligned}$$

Therefore, we can compute $\gamma_t(q_t^{(1)}, q_t^{(2)})$ by computing the forward and backward variables at every time frame. Since it is a distribution over the states, it indicates the contribution of each pair of dictionaries at time frame t. In practice, within the first few EM iterations, each time frame is almost entirely explained by a single pair of dictionaries. This means that $\gamma_t(q_t^{(1)}, q_t^{(2)})$ will have a value of almost 0 for all pairs of states except for a single pair, which will have a value of almost 1.

The forward and backward variables themselves can be computed recursively. We start with the forward variables. We first initialize the recursion as follows:

$$\begin{aligned}
\alpha_1(q_1^{(1)}, q_1^{(2)}) &= P(\mathbf{f_1}, v_1, q_1^{(1)}, q_1^{(2)}) \\
&= P(\mathbf{f_1}, v_1 | q_1^{(1)}, q_1^{(2)}) P(q_1^{(1)}, q_1^{(2)}) \\
&= P(\mathbf{f_1}, v_1 | q_1^{(1)}, q_1^{(2)}) P(q_1^{(1)}) P(q_1^{(2)}).
\end{aligned}$$

In the last step we used the fact that the initial state distributions of the individual sources are independent when they are not conditioned on any observations.

We then compute each $\alpha_{t+1}(q_1^{(1)}, q_1^{(2)})$ as follows:

$$\begin{aligned}
\alpha_{t+1}(q_{t+1}^{(1)}, q_{t+1}^{(2)}) &= P(\mathbf{f_1}, \mathbf{f_2}, ..., \mathbf{f_{t+1}}, v_1, v_2, ..., v_{t+1}, q_{t+1}^{(1)}, q_{t+1}^{(2)}) \\
&= P(\mathbf{f_{t+1}}, v_{t+1} | \mathbf{f_1}, \mathbf{f_2}, ..., \mathbf{f_t}, v_1, v_2, ..., v_t, q_{t+1}^{(1)}, q_{t+1}^{(2)}) ... \\
&\quad P(\mathbf{f_1}, \mathbf{f_2}, ..., \mathbf{f_t}, v_1, v_2, ..., v_t, q_{t+1}^{(1)}, q_{t+1}^{(2)}) \\
&= P(\mathbf{f_{t+1}}, v_{t+1} | q_{t+1}^{(1)}, q_{t+1}^{(2)}) P(\mathbf{f_1}, \mathbf{f_2}, ..., \mathbf{f_t}, v_1, v_2, ..., v_t, q_{t+1}^{(1)}, q_{t+1}^{(2)}) \\
&= \left(\sum_{q_t^{(1)}} \sum_{q_t^{(2)}} P(\mathbf{f_1}, \mathbf{f_2}, ..., \mathbf{f_t}, v_1, v_2, ..., v_t, q_{t+1}^{(1)}, q_{t+1}^{(2)}, q_t^{(1)}, q_t^{(2)}) \right) ... \\
&\quad P(\mathbf{f_{t+1}}, v_{t+1} | q_{t+1}^{(1)}, q_{t+1}^{(2)}) \\
&= \left(\sum_{q_t^{(1)}} \sum_{q_t^{(2)}} P(q_{t+1}^{(1)}, q_{t+1}^{(2)} | \mathbf{f_1}, \mathbf{f_2}, ..., \mathbf{f_t}, v_1, v_2, ..., v_t, q_t^{(1)}, q_t^{(2)}) ... \right. \\
&\quad \left. P(\mathbf{f_1}, \mathbf{f_2}, ..., \mathbf{f_t}, v_1, v_2, ..., v_t, q_t^{(1)}, q_t^{(2)}) \right) P(\mathbf{f_{t+1}}, v_{t+1} | q_{t+1}^{(1)}, q_{t+1}^{(2)}) \\
&= \left(\sum_{q_t^{(1)}} \sum_{q_t^{(2)}} P(q_{t+1}^{(1)}, q_{t+1}^{(2)} | q_t^{(1)}, q_t^{(2)}) P(\mathbf{f_1}, \mathbf{f_2}, ..., \mathbf{f_t}, v_1, v_2, ..., v_t, q_t^{(1)}, q_t^{(2)}) \right) ...
\end{aligned}$$

$$
\begin{aligned}
& P(\mathbf{f_{t+1}}, v_{t+1} | q_{t+1}^{(1)}, q_{t+1}^{(2)}) \\
= & \left(\sum_{q_t^{(1)}} \sum_{q_t^{(2)}} P(q_{t+1}^{(1)}, q_{t+1}^{(2)} | q_t^{(1)}, q_t^{(2)}) \alpha_t(q_t^{(1)}, q_t^{(2)}) \right) P(\mathbf{f_{t+1}}, v_{t+1} | q_{t+1}^{(1)}, q_{t+1}^{(2)}) \\
= & \left(\sum_{q_t^{(1)}} P(q_{t+1}^{(1)} | q_t^{(1)}) \sum_{q_t^{(2)}} P(q_{t+1}^{(2)} | q_t^{(2)}) \alpha_t(q_t^{(1)}, q_t^{(2)}) \right) P(\mathbf{f_{t+1}}, v_{t+1} | q_{t+1}^{(1)}, q_{t+1}^{(2)}).
\end{aligned}
$$

A number of the steps in the above derivation use conditional independence relations. As with the initial state probabilities, we used the fact that the transition probabilities of the individual sources are independent when they are not conditioned on any observations (as seen in the last step).

We now move onto the backward variables. We begin with the final time step T and initialize the recursion as follows:

$$\beta_T(q_T^{(1)}, q_T^{(2)}) = 1.$$

We then compute each $\beta_t(q_t^{(1)}, q_t^{(2)})$ as follows:

$$
\begin{aligned}
\beta_t(q_t^{(1)}, q_t^{(2)}) &= P(\mathbf{f_{t+1}}, \mathbf{f_{t+2}}, ..., \mathbf{f_T}, v_{t+1}, v_{t+2}, ..., v_T | q_t^{(1)}, q_t^{(2)}) \\
&= \sum_{q_{t+1}^{(1)}} \sum_{q_{t+1}^{(2)}} P(\mathbf{f_{t+1}}, \mathbf{f_{t+2}}, ..., \mathbf{f_T}, v_{t+1}, v_{t+2}, ..., v_T, q_{t+1}^{(1)}, q_{t+1}^{(2)} | q_t^{(1)}, q_t^{(2)}) \\
&= \sum_{q_{t+1}^{(1)}} \sum_{q_{t+1}^{(2)}} P(\mathbf{f_{t+2}}, ..., \mathbf{f_T}, v_{t+2}, ..., v_T | \mathbf{f_{t+1}}, v_{t+1}, q_{t+1}^{(1)}, q_{t+1}^{(2)}, q_t^{(1)}, q_t^{(2)}) ... \\
&\quad P(\mathbf{f_{t+1}}, v_{t+1}, q_{t+1}^{(1)}, q_{t+1}^{(2)} | q_t^{(1)}, q_t^{(2)}) \\
&= \sum_{q_{t+1}^{(1)}} \sum_{q_{t+1}^{(2)}} P(\mathbf{f_{t+2}}, ..., \mathbf{f_T}, v_{t+2}, ..., v_T | q_{t+1}^{(1)}, q_{t+1}^{(2)}) ... \\
&\quad P(\mathbf{f_{t+1}}, v_{t+1}, q_{t+1}^{(1)}, q_{t+1}^{(2)} | q_t^{(1)}, q_t^{(2)}) \\
&= \sum_{q_{t+1}^{(1)}} \sum_{q_{t+1}^{(2)}} \beta_{t+1}(q_{t+1}^{(1)}, q_{t+1}^{(2)}) P(\mathbf{f_{t+1}}, v_{t+1}, q_{t+1}^{(1)}, q_{t+1}^{(2)} | q_t^{(1)}, q_t^{(2)}) \\
&= \sum_{q_{t+1}^{(1)}} \sum_{q_{t+1}^{(2)}} \beta_{t+1}(q_{t+1}^{(1)}, q_{t+1}^{(2)}) P(\mathbf{f_{t+1}}, v_{t+1} | q_{t+1}^{(1)}, q_{t+1}^{(2)}, q_t^{(1)}, q_t^{(2)}) ... \\
&\quad P(q_{t+1}^{(1)}, q_{t+1}^{(2)} | q_t^{(1)}, q_t^{(2)})
\end{aligned}
$$

$$= \sum_{q_{t+1}^{(1)}} P(q_{t+1}^{(1)}|q_t^{(1)}) \sum_{q_{t+1}^{(2)}} P(q_{t+1}^{(2)}|q_t^{(2)}) \beta_{t+1}(q_{t+1}^{(1)}, q_{t+1}^{(2)}) P(\mathbf{f_{t+1}}, v_{t+1}|q_{t+1}^{(1)}, q_{t+1}^{(2)}).$$

In the computation of the forward and backward variables, we need the likelihoods, $P(\mathbf{f_t}, v_t|q_t^{(1)}, q_t^{(2)})$. This gives us the likelihood of a particular pair of states for the given data at time t. It is computed as follows:

$$\begin{aligned}
P(\mathbf{f_t}, v_t|q_t^{(1)}, q_t^{(2)}) &= P(v_t|q_t^{(1)}, q_t^{(2)}) P(\mathbf{f_t}|q_t^{(1)}, q_t^{(2)}) \\
&= P(v_t|q_t^{(1)}, q_t^{(2)}) \prod_{v=1}^{v_t} P_t(f_{t,v}|q_t^{(1)}, q_t^{(2)}) \\
&= P(v_t|q_t^{(1)}, q_t^{(2)}) \prod_{f_t} \left(P_t(f_t|q_t^{(1)}, q_t^{(2)}) \right)^{V_{ft}} \\
&= P(v_t|q_t^{(1)}, q_t^{(2)}) \prod_{f_t} \left(\sum_{s_t} \sum_{z_t} P_t(f_t, z_t, s_t|q_t^{(1)}, q_t^{(2)}) \right)^{V_{ft}} \\
&= P(v_t|q_t^{(1)}, q_t^{(2)}) \prod_{f_t} \left(\sum_{s_t} \sum_{z_t} P(f_t|z_t, s_t, q_t^{(1)}, q_t^{(2)}) P_t(z_t, s_t|q_t^{(1)}, q_t^{(2)}) \right)^{V_{ft}} \\
&= P(v_t|q_t^{(1)}, q_t^{(2)}) \prod_{f_t} \left(\sum_{s_t} \sum_{z_t} P(f_t|z_t, s_t, q_t^{(s_t)}) P_t(z_t, s_t|q_t^{(1)}, q_t^{(2)}) \right)^{V_{ft}}.
\end{aligned}$$

In the first step, we use the fact that the number of draws at time t is independent of the value of the actual draws given the pair of states. Moreover, the draws are independent from each other given the pair states (second step). In the third step, we group all of the draws for which the observation is f_t. At time t, we would have exactly V_{ft} such draws. $P(v_t|q_t^{(1)}, q_t^{(2)})$ is computed using the parameters of the energy distributions of the individual sources as described earlier in this section.

We now have a way of computing $\gamma_t(q_t^{(1)}, q_t^{(2)})$. As seen in Eq. 3.6, the other distribution that is needed in order to compute the marginalized posterior $P_t(z_t, s_t, q_t^{(1)}, q_t^{(2)}|f_t, \overline{\mathbf{f}}, \overline{\mathbf{v}})$ is $P_t(z_t, s_t|f_t, q_t^{(1)}, q_t^{(2)})$. This is computed using Baye's rule as follows:

$$P_t(z_t, s_t|f_t, q_t^{(1)}, q_t^{(2)}) = \frac{P(z_t, s_t|q_t^{(1)}, q_t^{(2)}) P(f_t|z_t, s_t, q_t^{(s_t)})}{\sum_{z_t} \sum_{s_t} P(z_t, s_t|q_t^{(1)}, q_t^{(2)}) P(f_t|z_t, s_t, q_t^{(s_t)})}.$$

Marginalized posteriors for transition matrices (E Step)

These marginalized posteriors use the forward and backward variables as well as the likelihoods that were derived in the previous subsection. Apart from these distributions, they use the distributions of the transition matrices, $P(q_{t+1}^{(1)}|q_t^{(1)})$ and $P(q_{t+1}^{(2)}|q_t^{(2)})$. For a given pair of time frames, the marginalized posterior for the first source is computed as follows:

$$\begin{aligned} P_t(q_t^{(1)}, q_{t+1}^{(1)}|\overline{\mathbf{f}}, \overline{\mathbf{v}}) &= \frac{\sum_{q_t^{(2)}} \sum_{q_{t+1}^{(2)}} P_t(\overline{\mathbf{f}}, \overline{\mathbf{v}}, q_t^{(1)}, q_t^{(2)}, q_{t+1}^{(1)}, q_{t+1}^{(2)})}{P(\overline{\mathbf{f}}, \overline{\mathbf{v}})} \\ &= \frac{\sum_{q_t^{(2)}} \sum_{q_{t+1}^{(2)}} P_t(\overline{\mathbf{f}}, \overline{\mathbf{v}}, q_t^{(1)}, q_t^{(2)}, q_{t+1}^{(1)}, q_{t+1}^{(2)})}{\sum_{q_t^{(1)}} \sum_{q_{t+1}^{(1)}} \sum_{q_t^{(2)}} \sum_{q_{t+1}^{(2)}} P_t(\overline{\mathbf{f}}, \overline{\mathbf{v}}, q_t^{(1)}, q_t^{(2)}, q_{t+1}^{(1)}, q_{t+1}^{(2)})}. \end{aligned}$$

The marginalized posterior for the second source, for a given pair of time frames, is similarly computed as follows:

$$P_t(q_t^{(2)}, q_{t+1}^{(2)}|\overline{\mathbf{f}}, \overline{\mathbf{v}}) = \frac{\sum_{q_t^{(1)}} \sum_{q_{t+1}^{(1)}} P_t(\overline{\mathbf{f}}, \overline{\mathbf{v}}, q_t^{(1)}, q_t^{(2)}, q_{t+1}^{(1)}, q_{t+1}^{(2)})}{\sum_{q_t^{(2)}} \sum_{q_{t+1}^{(2)}} \sum_{q_t^{(1)}} \sum_{q_{t+1}^{(1)}} P_t(\overline{\mathbf{f}}, \overline{\mathbf{v}}, q_t^{(1)}, q_t^{(2)}, q_{t+1}^{(1)}, q_{t+1}^{(2)})}.$$

$P_t(\overline{\mathbf{f}}, \overline{\mathbf{v}}, q_t^{(1)}, q_t^{(2)}, q_{t+1}^{(1)}, q_{t+1}^{(2)})$ is computed as follows:

$P_t(\overline{\mathbf{f}}, \overline{\mathbf{v}}, q_t^{(1)}, q_t^{(2)}, q_{t+1}^{(1)}, q_{t+1}^{(2)})$

$$\begin{aligned} &= P(q_{t+1}^{(1)}, q_{t+1}^{(2)}, \mathbf{f_{t+1}}, \mathbf{f_{t+2}}, ..., \mathbf{f_T}, v_{t+1}, v_{t+2}, ..., v_T|\mathbf{f_1}, \mathbf{f_2}, ..., \mathbf{f_t}, v_1, v_2, ..., v_t, q_t^{(1)}, q_t^{(2)})... \\ &\quad P(\mathbf{f_1}, \mathbf{f_2}, ..., \mathbf{f_t}, v_1, v_2, ..., v_t, q_t^{(1)}, q_t^{(2)}) \\ &= P(q_{t+1}^{(1)}, q_{t+1}^{(2)}, \mathbf{f_{t+1}}, \mathbf{f_{t+2}}, ..., \mathbf{f_T}, v_{t+1}, v_{t+2}, ..., v_T|q_t^{(1)}, q_t^{(2)})\alpha_t(q_t^{(1)}, q_t^{(2)}) \\ &= P(\mathbf{f_{t+1}}, \mathbf{f_{t+2}}, ..., \mathbf{f_T}, v_{t+1}, v_{t+2}, ..., v_T|q_{t+1}^{(1)}, q_{t+1}^{(2)}, q_t^{(1)}, q_t^{(2)})P(q_{t+1}^{(1)}, q_{t+1}^{(2)}|q_t^{(1)}, q_t^{(2)})\alpha_t(q_t^{(1)}, q_t^{(2)}) \\ &= P(\mathbf{f_{t+1}}, \mathbf{f_{t+2}}, ..., \mathbf{f_T}, v_{t+1}, v_{t+2}, ..., v_T|q_{t+1}^{(1)}, q_{t+1}^{(2)})P(q_{t+1}^{(1)}|q_t^{(1)})P(q_{t+1}^{(2)}|q_t^{(2)})\alpha_t(q_t^{(1)}, q_t^{(2)}) \\ &= P(\mathbf{f_{t+1}}, v_{t+1}|\mathbf{f_{t+2}}, ..., \mathbf{f_T}, v_{t+2}, ..., v_T, q_{t+1}^{(1)}, q_{t+1}^{(2)})... \\ &\quad P(\mathbf{f_{t+2}}, ..., \mathbf{f_T}, v_{t+2}, ..., v_T|q_{t+1}^{(1)}, q_{t+1}^{(2)})P(q_{t+1}^{(1)}|q_t^{(1)})P(q_{t+1}^{(2)}|q_t^{(2)})\alpha_t(q_t^{(1)}, q_t^{(2)}) \end{aligned}$$

$$
\begin{aligned}
&= P(\mathbf{f_{t+1}}, v_{t+1}|q_{t+1}^{(1)}, q_{t+1}^{(2)})\beta_{t+1}(q_{t+1}^{(1)}, q_{t+1}^{(2)})P(q_{t+1}^{(1)}|q_t^{(1)})P(q_{t+1}^{(2)}|q_t^{(2)})\alpha_t(q_t^{(1)}, q_t^{(2)}) \\
&= \alpha_t(q_t^{(1)}, q_t^{(2)})P(q_{t+1}^{(1)}|q_t^{(1)})P(q_{t+1}^{(2)}|q_t^{(2)})\beta_{t+1}(q_{t+1}^{(1)}, q_{t+1}^{(2)})P(\mathbf{f_{t+1}}, v_{t+1}|q_{t+1}^{(1)}, q_{t+1}^{(2)}).
\end{aligned}
$$

Marginalized posteriors for initial state probabilities (E Step)

This marginalized posterior is computed for the first source using $\gamma_1(q_1^{(1)}, q_1^{(2)})$ as follows:

$$
\begin{aligned}
P(q_1^{(1)}|\overline{\mathbf{f}}, \overline{\mathbf{v}}) &= \sum_{q_1^{(2)}} P(q_1^{(1)}, q_1^{(2)}|\overline{\mathbf{f}}, \overline{\mathbf{v}}) \\
&= \sum_{q_1^{(2)}} \gamma_1(q_1^{(1)}, q_1^{(2)}).
\end{aligned}
$$

It is similarly computed for the second source as follows:

$$
P(q_1^{(2)}|\overline{\mathbf{f}}, \overline{\mathbf{v}}) = \sum_{q_1^{(1)}} \gamma_1(q_1^{(1)}, q_1^{(2)}).
$$

Spectral components (M Step)

The spectral components are multinomial distributions. For a given component z of a given state $q^{(s)}$ of a given source s, the probability of each frequency is a separate parameter. We take the derivative of Eq. 3.4 with respect to each of these parameters and set them to 0. This gives us the following set of equations for the first source (one equation for each value of f):

$$
\frac{\sum_t \sum_{q_t^{(2)}} V_{ft} P_t(z, s=1, q^{(1)}, q^{(2)}|f, \overline{\mathbf{f}}, \overline{\mathbf{v}})}{P(f|z, s=1, q^{(1)})} - \rho_{z,s=1,q} = 0.
$$

By eliminating the Lagrange multiplier, we get the following M step equation::

$$
P(f|z, s=1, q^{(1)}) = \frac{\sum_t \sum_{q_t^{(2)}} V_{ft} P_t(z, s=1, q^{(1)}, q^{(2)}|f, \overline{\mathbf{f}}, \overline{\mathbf{v}})}{\sum_f \sum_t \sum_{q_t^{(2)}} V_{ft} P_t(z, s=1, q^{(1)}, q^{(2)}|f, \overline{\mathbf{f}}, \overline{\mathbf{v}})}.
$$

Similarly, the M step equation for the second source is as follows:

$$P(f|z, s=2, q^{(2)}) \quad = \quad \frac{\sum_t \sum_{q_t^{(1)}} V_{ft} P_t(z, s=2, q^{(1)}, q^{(2)}|f, \overline{\mathbf{f}}, \overline{\mathbf{v}})}{\sum_f \sum_t \sum_{q_t^{(1)}} V_{ft} P_t(z, s=2, q^{(1)}, q^{(2)}|f, \overline{\mathbf{f}}, \overline{\mathbf{v}})}.$$

Mixture weights (M Step)

The weights are a set of multinomial distributions at every time frame. We have a distribution of weights $P_t(z_t, s_t|q_t^{(1)}, q_t^{(2)})$ over the components of both sources for a given pair of states. For a given time frame t and pair of states $q_t^{(1)}, q_t^{(2)}$, the probability of each component z_t of source s_t is a separate parameter. We take the derivative of Eq. 3.4 with respect to each of these parameters and set them to 0. This gives us the following set of equations (one equation for each value of z_t):

$$\frac{\sum_{f_t} V_{ft} P_t(z_t, s_t, q_t^{(1)}, q_t^{(2)}|f_t, \overline{\mathbf{f}}, \overline{\mathbf{v}})}{P_t(z_t, s_t|q_t^{(1)}, q_t^{(2)})} - \tau_{q_t^{(1)}, q_t^{(2)}} \quad = \quad 0.$$

By eliminating the Lagrange multiplier, we get the following M step equation:

$$P_t(z_t, s_t|q_t^{(1)}, q_t^{(2)}) \quad = \quad \frac{\sum_{f_t} V_{ft} P_t(z_t, s_t, q_t^{(1)}, q_t^{(2)}|f_t, \overline{\mathbf{f}}, \overline{\mathbf{v}})}{\sum_{s_t} \sum_{z_t} \sum_{f_t} V_{ft} P_t(z_t, s_t, q_t^{(1)}, q_t^{(2)}|f_t, \overline{\mathbf{f}}, \overline{\mathbf{v}})}.$$

Transition matrices (M Step)

The transition matrix for each source is a separate multinomial distribution for each value of the state $q_t^{(s)}$. For a given state $q_t^{(s)}$, the probability of each state at the next time frame $q_{t+1}^{(s)}$ is a separate parameter. We take the derivative of Eq. 3.4 with respect to each of these parameters and set them to 0. This gives us the following set of equations for the first source (one equation for each value of q_{t+1}):

$$\frac{\sum_{t=1}^{T-1} P(q_t^{(1)}, q_{t+1}^{(1)}|\overline{\mathbf{f}}, \overline{\mathbf{v}})}{P(q_{t+1}^{(1)}|q_t^{(1)})} - \mu_{q_t^{(1)}} \quad = \quad 0.$$

By eliminating the Lagrange multiplier, we get the following M step equation:

$$P(q_{t+1}^{(1)}|q_t^{(1)}) = \frac{\sum_{t=1}^{T-1} P(q_t^{(1)}, q_{t+1}^{(1)}|\overline{\mathbf{f}}, \overline{\mathbf{v}})}{\sum_{q_{t+1}^{(1)}} \sum_{t=1}^{T-1} P(q_t^{(1)}, q_{t+1}^{(1)}|\overline{\mathbf{f}}, \overline{\mathbf{v}})}.$$

Similarly, the M step equation for the second source is as follows:

$$P(q_{t+1}^{(2)}|q_t^{(2)}) = \frac{\sum_{t=1}^{T-1} P(q_t^{(2)}, q_{t+1}^{(2)}|\overline{\mathbf{f}}, \overline{\mathbf{v}})}{\sum_{q_{t+1}^{(2)}} \sum_{t=1}^{T-1} P(q_t^{(2)}, q_{t+1}^{(2)}|\overline{\mathbf{f}}, \overline{\mathbf{v}})}.$$

Initial state probabilities (M Step)

The initial state probabilities for a given source is a single multinomial distribution. The parameters are the probabilities of starting on a given state. We take the derivative of Eq. 3.4 with respect to each of these parameters and set them to 0. This gives us the following set of equations for the first source:

$$\frac{P_1(q_1^{(1)}|\overline{\mathbf{f}}, \overline{\mathbf{v}})}{P(q_1^{(1)})} - \kappa^{(1)} = 0.$$

By eliminating the Lagrange multiplier, we get the following equation:

$$P(q_1^{(1)}) = \frac{P_1(q_1^{(1)}|\overline{\mathbf{f}}, \overline{\mathbf{v}})}{\sum_{q_1} P_1(q_1^{(1)}|\overline{\mathbf{f}}, \overline{\mathbf{v}})}.$$

The denominator is simply a normalization step. However, $P_1(q_1^{(1)}|\overline{\mathbf{f}}, \overline{\mathbf{v}})$ is already normalized. Therefore, the M step equation is simply the following assignment:

$$P(q_1^{(1)}) = P_1(q_1^{(1)}|\overline{\mathbf{f}}, \overline{\mathbf{v}}).$$

Similarly, the M step equation for the second source is the following assignment:

$$P(q_1^{(2)}) = P_1(q_1^{(2)}|\overline{\mathbf{f}}, \overline{\mathbf{v}}).$$

Summary of EM equations

In the E step, we first perform some intermediate computations using the parameters of the model. We then compute the marginalized posteriors using these intermediate computations and the parameters. In the M step, we compute the parameters using the marginalized posteriors. All of the equations are summarized on the following pages:

E Step - Intermediate Computations

$$P(\mathbf{f_t}, v_t|q_t^{(1)}, q_t^{(2)}) = P(v_t|q_t^{(1)}, q_t^{(2)}) \prod_{f_t} \left(\sum_{s_t} \sum_{z_t} P(f_t|z_t, s_t, q_t^{(s_t)}) P_t(z_t, s_t|q_t^{(1)}, q_t^{(2)}) \right)^{V_{ft}},$$

$$\begin{aligned} \alpha_1(q_1^{(1)}, q_1^{(2)}) &= P(\mathbf{f_1}, v_1|q_1^{(1)}, q_1^{(2)}) P(q_1^{(1)}) P(q_1^{(2)}), \\ \alpha_{t+1}(q_{t+1}^{(1)}, q_{t+1}^{(2)}) &= \left(\sum_{q_t^{(1)}} P(q_{t+1}^{(1)}|q_t^{(1)}) \sum_{q_t^{(2)}} P(q_{t+1}^{(2)}|q_t^{(2)}) \alpha_t(q_t^{(1)}, q_t^{(2)}) \right) ... \\ &\quad P(\mathbf{f_{t+1}}, v_{t+1}|q_{t+1}^{(1)}, q_{t+1}^{(2)}), \end{aligned}$$

$$\begin{aligned} \beta_T(q_T^{(1)}, q_T^{(2)}) &= 1, \\ \beta_t(q_t^{(1)}, q_t^{(2)}) &= \sum_{q_{t+1}^{(1)}} P(q_{t+1}^{(1)}|q_t^{(1)}) \sum_{q_{t+1}^{(2)}} P(q_{t+1}^{(2)}|q_t^{(2)}) \beta_{t+1}(q_{t+1}^{(1)}, q_{t+1}^{(2)}) P(\mathbf{f_{t+1}}, v_{t+1}|q_{t+1}^{(1)}, q_{t+1}^{(2)}), \end{aligned}$$

$$P_t(z_t, s_t|f_t, q_t^{(1)}, q_t^{(2)}) = \frac{P(z_t, s_t|q_t^{(1)}, q_t^{(2)}) P(f_t|z_t, s_t, q_t^{(s_t)})}{\sum_{z_t} \sum_{s_t} P(z_t, s_t|q_t^{(1)}, q_t^{(2)}) P(f_t|z_t, s_t, q_t^{(s_t)})},$$

$$\begin{aligned} &P_t(\overline{\mathbf{f}}, \overline{\mathbf{v}}, q_t^{(1)}, q_t^{(2)}, q_{t+1}^{(1)}, q_{t+1}^{(2)}) \\ &= \alpha_t(q_t^{(1)}, q_t^{(2)}) P(q_{t+1}^{(1)}|q_t^{(1)}) P(q_{t+1}^{(2)}|q_t^{(2)}) \beta_{t+1}(q_{t+1}^{(1)}, q_{t+1}^{(2)}) P(\mathbf{f_{t+1}}, v_{t+1}|q_{t+1}^{(1)}, q_{t+1}^{(2)}), \end{aligned}$$

$$\gamma_t(q_t^{(1)}, q_t^{(2)}) = \frac{\alpha_t(q_t^{(1)}, q_t^{(2)}) \beta_t(q_t^{(1)}, q_t^{(2)})}{\sum_{q_t^{(1)}} \sum_{q_t^{(2)}} \alpha_t(q_t^{(1)}, q_t^{(2)}) \beta_t(q_t^{(1)}, q_t^{(2)})}.$$

E Step - Marginalized Posteriors

$$P_t(z_t, s_t, q_t^{(1)}, q_t^{(2)} | f_t, \overline{\mathbf{f}}, \overline{\mathbf{v}}) = \gamma_t(q_t^{(1)}, q_t^{(2)}) P_t(z_t, s_t | f_t, q_t^{(1)}, q_t^{(2)}),$$

$$P_t(q_t^{(1)}, q_{t+1}^{(1)} | \overline{\mathbf{f}}, \overline{\mathbf{v}}) = \frac{\sum_{q_t^{(2)}} \sum_{q_{t+1}^{(2)}} P_t(\overline{\mathbf{f}}, \overline{\mathbf{v}}, q_t^{(1)}, q_t^{(2)}, q_{t+1}^{(1)}, q_{t+1}^{(2)})}{\sum_{q_t^{(1)}} \sum_{q_{t+1}^{(1)}} \sum_{q_t^{(2)}} \sum_{q_{t+1}^{(2)}} P_t(\overline{\mathbf{f}}, \overline{\mathbf{v}}, q_t^{(1)}, q_t^{(2)}, q_{t+1}^{(1)}, q_{t+1}^{(2)})},$$

$$P_t(q_t^{(2)}, q_{t+1}^{(2)} | \overline{\mathbf{f}}, \overline{\mathbf{v}}) = \frac{\sum_{q_t^{(1)}} \sum_{q_{t+1}^{(1)}} P_t(\overline{\mathbf{f}}, \overline{\mathbf{v}}, q_t^{(1)}, q_t^{(2)}, q_{t+1}^{(1)}, q_{t+1}^{(2)})}{\sum_{q_t^{(2)}} \sum_{q_{t+1}^{(2)}} \sum_{q_t^{(1)}} \sum_{q_{t+1}^{(1)}} P_t(\overline{\mathbf{f}}, \overline{\mathbf{v}}, q_t^{(1)}, q_t^{(2)}, q_{t+1}^{(1)}, q_{t+1}^{(2)})},$$

$$P(q_1^{(1)} | \overline{\mathbf{f}}, \overline{\mathbf{v}}) = \sum_{q_1^{(2)}} \gamma_1(q_1^{(1)}, q_1^{(2)}),$$

$$P(q_1^{(2)} | \overline{\mathbf{f}}, \overline{\mathbf{v}}) = \sum_{q_1^{(1)}} \gamma_1(q_1^{(1)}, q_1^{(2)}).$$

M Step

$$P(f|z,s=1,q^{(1)}) = \frac{\sum_t \sum_{q_t^{(2)}} V_{ft} P_t(z,s=1,q^{(1)},q^{(2)}|f,\overline{\mathbf{f}},\overline{\mathbf{v}})}{\sum_f \sum_t \sum_{q_t^{(2)}} V_{ft} P_t(z,s=1,q^{(1)},q^{(2)}|f,\overline{\mathbf{f}},\overline{\mathbf{v}})},$$

$$P(f|z,s=2,q^{(2)}) = \frac{\sum_t \sum_{q_t^{(1)}} V_{ft} P_t(z,s=2,q^{(1)},q^{(2)}|f,\overline{\mathbf{f}},\overline{\mathbf{v}})}{\sum_f \sum_t \sum_{q_t^{(1)}} V_{ft} P_t(z,s=2,q^{(1)},q^{(2)}|f,\overline{\mathbf{f}},\overline{\mathbf{v}})},$$

$$P_t(z_t,s_t|q_t^{(1)},q_t^{(2)}) = \frac{\sum_{f_t} V_{ft} P_t(z_t,s_t,q_t^{(1)},q_t^{(2)}|f_t,\overline{\mathbf{f}},\overline{\mathbf{v}})}{\sum_{s_t}\sum_{z_t}\sum_{f_t} V_{ft} P_t(z_t,s_t,q_t^{(1)},q_t^{(2)}|f_t,\overline{\mathbf{f}},\overline{\mathbf{v}})},$$

$$P(q_{t+1}^{(1)}|q_t^{(1)}) = \frac{\sum_{t=1}^{T-1} P(q_t^{(1)},q_{t+1}^{(1)}|\overline{\mathbf{f}},\overline{\mathbf{v}})}{\sum_{q_{t+1}^{(1)}} \sum_{t=1}^{T-1} P(q_t^{(1)},q_{t+1}^{(1)}|\overline{\mathbf{f}},\overline{\mathbf{v}})},$$

$$P(q_{t+1}^{(2)}|q_t^{(2)}) = \frac{\sum_{t=1}^{T-1} P(q_t^{(2)},q_{t+1}^{(2)}|\overline{\mathbf{f}},\overline{\mathbf{v}})}{\sum_{q_{t+1}^{(2)}} \sum_{t=1}^{T-1} P(q_t^{(2)},q_{t+1}^{(2)}|\overline{\mathbf{f}},\overline{\mathbf{v}})},$$

$$P(q_1^{(1)}) = P_1(q_1^{(1)}|\overline{\mathbf{f}},\overline{\mathbf{v}}),$$
$$P(q_1^{(2)}) = P_1(q_1^{(2)}|\overline{\mathbf{f}},\overline{\mathbf{v}}).$$

Reconstructions

After performing the EM iterations, we may wish to reconstruct each of the sources. This can be done from the estimated parameters and the marginalized posteriors as follows:

$$\begin{aligned} P(f_t,s_t|\overline{\mathbf{f}},\overline{\mathbf{v}}) &= \sum_{q_t^{(1)}}\sum_{q_t^{(2)}} P(f_t,s_t,q_t^{(1)},q_t^{(2)}|\overline{\mathbf{f}},\overline{\mathbf{v}}) \\ &= \sum_{q_t^{(1)}}\sum_{q_t^{(2)}} P(q_t^{(1)},q_t^{(2)}|\overline{\mathbf{f}},\overline{\mathbf{v}}) P(f_t,s_t|q_t^{(1)},q_t^{(2)},\overline{\mathbf{f}},\overline{\mathbf{v}}) \\ &= \sum_{q_t^{(1)}}\sum_{q_t^{(2)}} P(q_t^{(1)},q_t^{(2)}|\overline{\mathbf{f}},\overline{\mathbf{v}}) P(f_t,s_t|q_t^{(1)},q_t^{(2)}) \end{aligned}$$

$$
\begin{aligned}
&= \sum_{q_t^{(1)}} \sum_{q_t^{(2)}} P(q_t^{(1)}, q_t^{(2)} | \overline{\mathbf{f}}, \overline{\mathbf{v}}) \sum_{z_t} P(f_t, z_t, s_t | q_t^{(1)}, q_t^{(2)}) \\
&= \sum_{q_t^{(1)}} \sum_{q_t^{(2)}} P(q_t^{(1)}, q_t^{(2)} | \overline{\mathbf{f}}, \overline{\mathbf{v}}) \sum_{z_t} P(f_t | z_t, s_t, q_t^{(1)}, q_t^{(2)}) P(z_t, s_t | q_t^{(1)}, q_t^{(2)}) \\
&= \sum_{q_t^{(1)}} \sum_{q_t^{(2)}} P(q_t^{(1)}, q_t^{(2)} | \overline{\mathbf{f}}, \overline{\mathbf{v}}) \sum_{z_t} P(f_t | z_t, s_t, q_t^{(s_t)}) P(z_t, s_t | q_t^{(1)}, q_t^{(2)}). \quad (3.7)
\end{aligned}
$$

As shown in the above equation, the reconstruction of a given source has contributions from all of its constituent dictionaries. However, in a given time frame, the reconstructions from all but one dictionaries will be nearly zero. The non-zero contributions are useful for modeling things like decays from a note in a previous frame that was explained by a different dictionary.

Although this will yield a reconstruction of the spectrograms of the individual sources, there are a few additional steps that are required for source separation. This will be discussed in Sec. 4.3.

3.5 Conclusions

In this chapter, we presented a new probabilistic model of sound mixtures, the non-negative factorial hidden Markov model (N-FHMM). The spectral structure and temporal dynamics of each source within the mixture has been modeled by a constituent N-HMM.

We first described two classes of existing models of sound mixtures. Non-negative spectrogram factorization methods model sound mixtures by concatenating the dictionaries of the individual sources. Factorial HMMs model sound mixtures by combining HMMs of individual sources through an interaction model.

The N-FHMM models sound mixtures by combining dictionaries from N-HMMs of individual sources through an interaction model. The interaction model is additive to reflect the additive nature of the individual sources. Specifically, at each time frame, the mixture is modeled as a linear combination of the spectral components from a concatenated dictionary, which is formed from the active dictionaries of the individual sources at that time frame.

The parameter estimation equations for the model were derived. As this is a highly unconstrained estimation problem, it will not yield meaningful solutions if we estimate all of the parameters from the mixture spectrogram. Therefore, in practice, some of the parameters are fixed and the remaining parameters are estimated. The specific parameters that are fixed depends on the application. As an alternative to fixing these parameters, we could apply priors on them.

In Chapter 4, we apply the N-FHMM to single channel supervised source separation by fixing some of the parameters of the individual sources, which are estimated from isolated training data. Although source separation is an important application of the N-FHMM, it is a general model of sound mixtures and can be used for various other applications as will be discussed in Chapter 5.

Chapter 4

Source Separation

4.1 Introduction

The N-FHMM is a general model of sound mixtures. It can be used for various applications that involve sound mixtures. Source separation is essential for many of these applications, as noted in Chapter 1. In this chapter, we therefore discuss source separation in more detail. We first give a brief overview of existing approaches to source separation. We then discuss non-negative spectrogram factorization and FHMM approaches to the problem in more detail as they are closely related to the N-FHMM. This is followed by a discussion of the application of the N-HMM and the N-FHMM to single channel supervised source separation, with examples and experimental results.

4.2 Overview of Source Separation Techniques

Source separation techniques can be broadly divided into multi-channel and single channel techniques. The number of channels refers to the number of channels available to us as an input to the source-separation algorithm. For example, single channel source separation refers to multiple sounds that were recorded with a single microphone as well as sounds that were recorded with multiple microphones but artificially mixed together into a single channel. There is some amount of overlap in the two

classes of techniques and we discuss this when applicable.

4.2.1 Multi-Channel Techniques

In certain scenarios, multiple recordings of a given sound mixture will be available to us. Such recordings are commonly made with microphone arrays. The proportion and time of arrival of each source in each microphone will vary with the microphones. These two factors could be considered as spatial information because they depend on the location of each source and each microphone. An important aspect of multi-channel source separation techniques is that they make use of this spatial information.

To formalize the problem, there are I source signals $x_i(t)$ that we wish to obtain. We have J recordings of mixtures $y_j(t)$ of the source signals (one recording per microphone). The mixtures are related to the source signals according to the gains h_{ij} from source i to microphone j as follows:

$$y_j(t) = \sum_i h_{ij} x_i(t). \tag{4.1}$$

This is called an instantaneous mixing model because each $y_j(t)$ depends on only the value $x_i(t)$ at time t. However, a more realistic mixing model accounts for room reverberation and is called a convolutive mixing model. In this model, each $x_i(t)$ is convolved with a filter $h_{ij}(\tau)$. Each $y_j(t)$ is then a sum of filtered source signals as follows:

$$y_j(t) = \sum_i \sum_\tau h_{ij}(\tau) x_i(t-\tau). \tag{4.2}$$

There are two different kinds of multi-channel mixtures:

1. Over-determined and determined mixtures – The number of microphones is greater than or equal to the number of sources ($J \geq I$).

2. Under-determined mixtures – The number of microphones is less than the number of sources ($J < I$).

We consider each of the above cases.

Over-Determined and Determined Mixtures

In this case, it is possible (in theory) to achieve perfect separation without strong assumptions about the sources and the mixing process. These methods have therefore been termed blind source separation (BSS).

A commonly used class of methods for this type of source separation is called independent component analysis (ICA). These methods exploit an assumption of statistical independence of the sources. A great deal of the well known ICA methods, such as the one proposed by Bell and Sejnowski [4], are meant for the instantaneous mixing case and assume that the individual samples of each source are independent draws from a stationary non-Gaussian distribution. Extensive reviews of these methods have been made by Cardoso [10], Hyvarinen et al. [28].

There has also been work addressing the use of ICA for the convolutive mixing model. This has been surveyed by Pedersen et al. [46]. One of the strategies that has been used in this scenario is to perform source separation in the frequency domain, as done by Smaragdis [62] and Ikeda and Murata [29].

Extensions of ICA that assume a convolutive mixing model with time–varying filters have been proposed by Mukai et al. [40].

Another approach to performing multi-channel source separation is the use of beamforming. This has been extensively reviewed by Brandstein and Ward [6] and Van Veen and Buckley [71]. Saruwatari et al. [56] have combined ICA and beamforming to perform source separation.

Under-Determined Mixtures

We now consider cases in which $J < I$ but J is still greater than one. We start with a discussion of time–frequency masks as it is relevant in this section as well as future sections (single channel techniques).

As discussed in Sec. 1.2, Roweis [54, 55] and Yilmaz and Rickard [79] have pointed out that speech spectrograms tend to be sparse. This means that only a small portion of the time–frequency bins in a spectrogram will have a significant amount of energy. Although the sound mixture is the sum of the individual sources, if the sources are

independent, most of the energy in a given time–frequency bin can be attributed to a single source. This is generally the case with speech in the context of the cocktail party effect. Music sources, on the other hand, are more synchronized. However, with enough frequency resolution, the imperfections in real music give music the same property (although it is often to a lesser degree).

We start with the idea that every time–frequency bin in the spectrogram can be assumed to have some contribution (however small) from each source. If we can estimate these contributions, then we can reconstruct the separate sources. The relative contributions of each source at each time-frequency bin can be defined by a soft mask such that the sum of the contributions sum to 1. The contribution of source s at time frequency-bin f, t is defined by the mask $M_{ft}^{(s)}$. Therefore, if we have two sources, we have:

$$M_{ft}^{(1)} + M_{ft}^{(2)} = 1.$$

The above argument of course implies that the value of the soft mask will be almost zero for all but one source in a given time–frequency bin. We could therefore just assign all of the energy in that time–frequency bin to that single source. This will effectively yield a binary mask. This can sometimes work reasonably well. However, it often leads to artifacts such as musical noise. The problem is that source-separation algorithms generally do not perform a perfect assignment of time–frequency bins. These mistakes can lead to the artifacts.

Another strategy is to use the soft mask directly without ever performing a hard assignment. We can modulate the mixture spectrogram by the soft mask to obtain the estimates of the spectrograms of the individual sources. Therefore we have:

$$\begin{aligned} \hat{V}_{ft}^{(1)} &= V_{ft} M_{ft}^{(1)}, \\ \hat{V}_{ft}^{(2)} &= V_{ft} M_{ft}^{(2)}, \end{aligned} \tag{4.3}$$

where $\hat{V}_{ft}^{(s)}$ is the estimated spectrogram of separated source s and V_{ft} is the spectrogram of the mixture. Since the masks sum to 1 at every time–frequency bin, we

could also formulate them in terms of probabilities as follows:

$$P(s_t|f_t) \;=\; M_{ft}^{(s_t)}.$$

In some source-separation algorithms, we can perform a reconstruction of the individual sources in the form $P(f_t, s_t)$. For example, this has been done using the N-FHMM in Eq. 3.7. Since these are estimates, certain nuances of the mixture are sometimes not captured by either source. However, all of the information is still present in the mixture. If we were to instead use a soft mask, we would account for all of the information in the mixture. It is possible to estimate a soft mask using the reconstructions as follows:

$$P(s_t|f_t) \;=\; \frac{P(f_t, s_t)}{\sum_{s_t} P(f_t, s_t)}. \tag{4.4}$$

Looking at the structure of Eq. 4.4, we can also interpret it as a time-varying Wiener filter, as pointed out by Roweis [54].

We now move onto specific source separation techniques for under-determined mixtures. Although, the number of microphones is less than the number of sources, the number of microphones is still greater than one. Therefore some amount of spatial information is still available. These techniques generally estimate time–frequency masks using the available spatial information in conjunction with the idea that spectrograms are sparse.

Jourjine et al. [31] have developed the degenerate unmixing estimation technique (DUET), which was extended by Yilmaz and Rickard [79]. They use the above idea to separate sources that have been mixed in an anechoic environment. Master [38] points out that when dealing with stereo music, some of the information about the panning (spatial information) is known. He used this information in a Bayesian extension of DUET.

4.2.2 Single Channel Techniques

In this section, we consider the more challenging problem of single channel source separation. This is more challenging because spatial information is not available. Some of the techniques that are discussed here, have been used in multi-channel source separation as well (this is pointed out where applicable). Single channel source separation can roughly be categorized as follows:

1. Computational Auditory Scene Analysis (CASA)
2. Basis Decomposition Techniques
3. Model Based Techniques

These techniques generally operate on a time–frequency representation of the sound mixture such as a spectrogram. We discuss each of the above categories.

Computational Auditory Scene Analysis

The human auditory system has a remarkable ability to distinguish between multiple sources in a mixture. The cues that are used by the auditory system for this purpose have been termed auditory scene analysis (ASA) by Bregman [7]. Computational implementations of these ideas have been termed computation auditory scene analysis (CASA). The idea behind these methods is to create time–frequency masks by grouping time–frequency bins such that each group corresponds to a different source. This grouping is done according ASA cues such as:

1. Proximity in time and frequency
2. Synchronous changes in energy – common onset, common offset, common frequency modulation, common amplitude modulation
3. Spatial proximity (this cue can only be used with multi-channel techniques)
4. Consistent harmonic structure

Several source separation methods have been developed based on this grouping, such as the work by Parsons [45], Weintraub [77], Mellinger [39], and Brown and Cooke [8].

These methods are generally suited for harmonic signals. They have been extended to deal with a larger class of sounds such as unvoiced sounds and transients by Ellis [16] and Hu and Wang [27].

CASA methods generally tend to be heuristic implementations of the grouping cues. Smaragdis points out that this grouping can be explained in a unified framework as redundancy reduction in audio [63].

Basis Decomposition Techniques

The basic idea in these techniques is to represent a sound mixture as a linear combination of basis vectors. The hope is that the individual sources correspond to disjoint subsets of basis vectors. These techniques can be used in an unsupervised method or a supervised method (this overlaps with model based techniques).

If we resort to unsupervised separation, the idea is that the spectrogram of the sound mixture is first decomposed using some kind of factorization technique to yield a set of basis vectors. These basis vectors are then clustered into groups corresponding to the different sources. An example of this can be seen in the independent subspace analysis (ISA) work by Casey and Westner [11].

The non-negative spectrogram factorization techniques that were discussed in previous chapters come under the category of basis decomposition techniques. Particularly, the spectral components can be interpreted as basis vectors. However, these methods are generally used in a supervised or semi-supervised setup and will therefore be discussed under model based techniques.

Model Based Techniques

The basic idea in these techniques is to make use of the properties of individual sources. These properties can then be used to distinguish between the sources in a mixture. Therefore, models of individual sources and models of sound mixtures are

addressed. The proposed framework comes under this category. Supervised and semi-supervised source separation come under this category. The procedure for supervised separation is often as follows:

1. Train models of single sources from isolated training data of the sources. The training generally involves parameter estimation in a given model.

2. Combine the trained models of single sources into a model of sound mixtures. This generally involves fixing some of the parameters in the model of sound mixtures according to the estimated parameters in the single source models.

3. Estimate the parameters that are not fixed in the model of sound mixtures.

4. Perform reconstructions of the individual sources using the estimated parameters.

Semi-supervised separation is a similar procedure. The difference is that we only train models of a subset of the sources from training data. All of the parameters of the other sources are estimated directly from the mixture. This is discussed in the context of the proposed models in Sec. 5.2.2.

These ideas have been used in the context of FHMMs and non-negative spectrogram factorizations. Since these are both closely related to the proposed model, we discuss them in more detail in the following sections.

4.2.3 Factorial Hidden Markov Models

In this section, we review prior work on the application of FHMMs and its variants to model based supervised and semi-supervised source separation as discussed above. In these methods, the first step is to train individual HMMs of each source from training data. This gives us the observation model $P(f_t|q_t^{(s)})$, transition probabilities $P(q_{t+1}^{(s)}|q_t^{(s)})$, and initial state probabilities, $P(q_1^{(s)})$ for each source. We then need some way of combining the observation models of the individual sources into the interaction model of the mixture $P(f_t|q_t^{(1)}, q_t^{(2)})$. This often involves learning additional parameters from the mixture. Also, the initial state probabilities simply tell us how

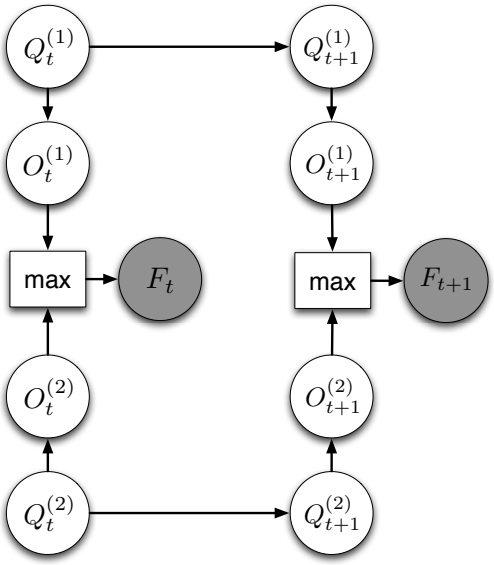


Figure 4.1: Graphical model for the factorial-max HMM [54].

likely we are to start on each state. This might be specific to the training data and therefore might need to be relearned with respect to the mixture.

As in Chapter 3, we use the example of two sources for simplicity of the explanation.

Factorial-Max HMMs

Roweis [54, 55], used the insight that spectrograms of speech tend to be sparse. He therefore developed the factorial-max HMM (Fig. 4.1). This model assigns the entire energy of a given time–frequency bin to a single source (binary mask).

In the factorial-max HMM, Roweis models the output of each individual HMM using $P(o_t^{(s)}|q_t^{(s)})$. He uses a Gaussian distribution for each individual state and uses a total of 8192 states (for each HMM). The actual output $o_t^{(s)}$ is a spectral vector that corresponds to a log magnitude spectrum. In the generative process, at each time frame, each individual HMM proposes an output according to $P(o_t^{(s)}|q_t^{(s)})$. The observation f_t is then simply the elementwise maximum of the proposed outputs along with some Gaussian noise. The interaction model is therefore given by:

$$P(f_t|q_t^{(1)}, q_t^{(2)}) \;=\; \mathcal{N}(\max[o_t^{(1)}, o_t^{(2)}], R),$$

where R is the variance of the additive noise. This noise serves as the small contribution from the non-dominating source in a given time–frequency bin.

As mentioned in Sec. 4.2.1, a binary mask can be quite rigid and can lead to artifacts such as musical noise. It can therefore beneficial to consider models in which the observation is modeled as a sum of sources.

FHMMs with Observations in the Feature Space

Virtanen [74] used FHMMs for source separation and speech recognition in the presence of non-stationary noise. The non-stationary noise is speech from another speaker so this is essentially source separation and concurrent speech recognition of multiple speakers. He used standard FHMMs (as in Fig. 3.3).

In the training of the individual source models, rather than training directly on raw spectrograms, he computes mel-frequency cepstral coefficients (MFCCs) [13] at each time frame (this is commonly done in speech recognition). MFCCs provide a summary of a magnitude spectrum in a much lower dimension (24 in this case) than the original magnitude spectrum. To compute the MFCCs, we weight the power spectrum (magnitude spectrum squared) by mel-frequency bands (effectively applying an auditory filter bank), taking the logarithm, and applying the discrete cosine transform (DCT). They tend to be more robust than the magnitude spectrum for many analysis applications such as speech recognition. This is largely due to the fact that they do not account for pitch information, which is not relevant for speech recognition.

The observation model for each source $P(f_t|q_t^{(s)})$ and $P(f_t|q_t^{(n)})$ is Gaussian (the superscripts s and n indicate speech and noise, which are both speech in practice). Therefore, once he trains individual HMMs, there is a mean vector and covariance matrix, $\mu_{\{s\}}^{\{m\}}$ and $\Sigma_{\{s\}}^{\{m\}}$ for each state of the speech and $\mu_{\{n\}}^{\{m\}}$ and $\Sigma_{\{n\}}^{\{m\}}$ for each state of the noise in the MFCC domain (which is indicated by the superscript m).

In a sound mixture, at each time frame, each source will correspond to a state with its own mean vector and covariance matrix. In order to model a mixture, he has to somehow combine the individual MFCC distributions to model MFCC distributions of mixtures $P(f_t|q_t^{(s)}, q_t^{(n)})$. Particularly, he needs an MFCC distribution with mean vector $\mu_{\{x\}}^{\{m\}}$ and covariance matrix $\Sigma_{\{x\}}^{\{m\}}$ for each possible combination of states. If each source has N possible states, he needs to have MFCC distributions of mixtures for N^2 possible state combinations.

If the sources are independent, the power spectrum of the mixture is equal to the sum of the power spectra of the individual sources. Therefore, he first estimates the means $\mu_{\{s\}}^{\{e\}}$, $\mu_{\{n\}}^{\{e\}}$ and covariance matrices $\Sigma_{\{s\}}^{\{e\}}$, $\Sigma_{\{n\}}^{\{e\}}$ of each state of each individual source in the power spectral domain using inverse DCTs and exponentiations, as done by Gales and Young [21]. He then sums them with appropriate gains $g_{\{s\}}$ and $g_{\{n\}}$ in the power spectral domain. In order to go back to the MFCC domain, he computes the parameters of the log-normal distribution as done by Gales and Young [21]. He then computes the DCT. Given the gains, he therefore has a way to compute the parameters of $P(f_t|q_t^{(s)}, q_t^{(n)})$ given the parameters of $P(f_t|q_t^{(s)})$ and $P(f_t|q_t^{(n)})$.

Using this framework, he has developed a greedy algorithm based on the Viterbi algorithm and the Nelder-Mead algorithm [34] that finds the optimal state sequence and gains of each source. These optimal state sequences yield the speech recognition result.

In order to perform source separation, he estimates a soft mask. We can interpret the mask as filtering the mixture signal by a time–varying Wiener filter (in the frequency domain), which he designs using the means of the optimal state sequence and the estimated gains. The power response W_i of the Wiener filter for the speech at mel-frequency i is as follows:

$$W_i^{\{s\}} = \frac{g_{\{s\}}\mu_{i\{s\}}^{\{e\}}}{g_{\{s\}}\mu_{i\{s\}}^{\{e\}} + g_{\{n\}}\mu_{i\{n\}}^{\{e\}}}.$$

The power response of the Wiener filter for the noise is $1 - W_i^{\{s\}}$. This time–varying Wiener filter therefore allocates some portion of each time–frequency bin to

each source.

This method has the advantage that it uses a well established speech recognition feature vector in order to perform concurrent speech recognition. However, it involves a fairly complicated interaction model.

Also, the goal of the decoding procedure is to estimate the optimal state sequence for each source as well as the gains for each source. To perform source separation, he estimates a soft mask using these things. Therefore, in a given time frame, the reconstruction of a given source depends only on the mean of the state of that source in that time frame (and the gain of the source). Therefore he reconstructs a given source, which could correspond to any vector in the distribution $P(f_t|q_t^{(s)})$ (where $q_t^{(s)}$ is the inferred optimal state of source s at time t), by approximating it with the the mean of that state. This could be a reasonable approximation as it could work well in practice. Also, inferring the optimal vector in $P(f_t|q_t^{(s)})$ is likely to be computationally intensive. However, it will still lead to some approximation error in the reconstruction.

FHMMs with Grammar Dynamics

Hershey et al. [26, 33, 51, 25, 49, 50], modeled temporal dynamics at the acoustic level as well as the grammar level, each with their own transition matrix. This has resulted in an FHMM with "dual dynamics" (Fig. 4.2). For a given source, they model the acoustics dynamics $P(s_{t+1}^{(s)}|s_t^{(s)})$ using 256 states. They model the grammar dynamics $P(v_{t+1}^{(s)}|v_t^{(s)})$ using left–right phone models according to a given vocabulary and grammar. This serves as a language model. They use 506 states for this purpose. They consider models with and without the use of the acoustics dynamics. However, they always use the grammar dynamics. They map the grammar level to the acoustic level with another transition matrix $P(s_t^{(s)}|v_t^{(s)})$ that they learn from training data. For the observation model of a given source $P(x_t^{(s)}|s_t^{(s)})$, they use a Gaussian with a diagonal covariance matrix, where $x_t^{(s)}$ is a vector in the log power spectrum domain.

The interaction model, in terms of the observation models of the individual sources

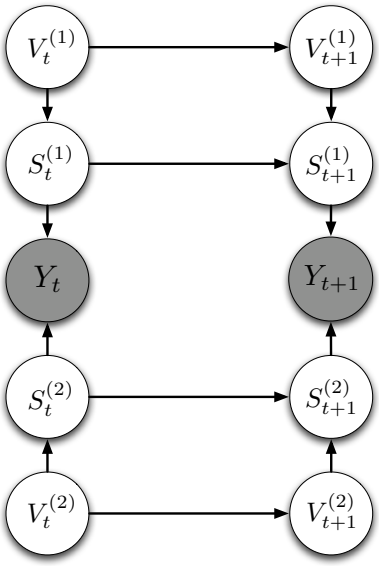


Figure 4.2: Graphical model for a FHMM with grammar dynamics [26].

and the acoustic interaction model $P(y_t|x_t^{(1)}, x_t^{(2)})$, is given by:

$$P(y_t|s_t^{(1)}, s_t^{(2)}) \;=\; \int P(y_t|x_t^{(1)}, x_t^{(2)}) P(x_t^{(1)}|s_t^{(1)}) P(x_t^{(2)}|s_t^{(2)}) \mathrm{d}x_t^{(1)} \mathrm{d}x_t^{(2)}.$$

They consider each frequency independently over here. They consider two different models for the acoustic interaction model $P(y_t|x_t^{(1)}, x_t^{(2)})$ (of each frequency). The first model is called Algonquin [20]. The idea is that the power spectrum of the mixture is approximately the sum of the power spectra of the two individual sources. Since the observations are in the log domain, the observed mixture y_t is related to the log power spectra of the individual sources as follows:

$$y_t \;\approx\; \log\left(\exp(x_t^{(1)}) + \exp(x_t^{(2)})\right).$$

Using this relation, the acoustic interaction model is given by:

$$P(y_t|x_t^{(1)}, x_t^{(2)}) \;=\; \mathcal{N}\left(y_t; \log\left(\exp(x_t^{(1)}) + \exp(x_t^{(2)})\right), \psi\right).$$

The second model is similar to the interaction model used in the factorial-max

HMM [54]. It is given by:

$$P(y_t|x_t^{(1)}, x_t^{(2)}) \quad = \quad \delta\left(y_t - \max(x_t^{(1)}, x_t^{(2)})\right).$$

They perform temporal inference using either a 2-D Viterbi algorithm [25], loopy belief propagation [49], or an approximate inference method that combines loopy belief propagation and variational inference [50].

They reconstruct the individual sources using $P(y_t|s_t^{(1)}, s_t^{(2)})$ and the posterior expected values of the sources given the acoustic states and the observed mixture. It is done as a minimum mean squared error (MMSE) or a maximum a posteriori (MAP) estimation.

They perform speech recognition on each of the separated speakers. They showed that the accuracy surpassed that of humans who attempted to transcribe the speech of the individual speakers, given the mixture. They attribute this high level of accuracy to the use of grammar dynamics.

Factorial Scaled-HMMs

Ozerov et al. [44] combined ideas from HMMs and the IS-NMF [17] to form the factorial scaled-HMM (FS-HMM) (Fig. 4.3) . The basic idea is that they model each source with a scaled-HMM (S-HMM). In a S-HMM, each state corresponds to one specific spectral component (termed sub-component here). At time frame t, source k therefore corresponds to the sub-component of one of the states. If this state is i, the contribution of source k is given by :

$$\mathbf{c}_{k,t} \quad = \quad \sum_{i=1}^{J_k} \mathbf{u}_{ki,t} \mathbf{1}(I_{k,t} = i),$$

where $\mathbf{1}()$ is the indicator function. The sub-components are zero-mean Gaussian vectors with the following covariance matrix:

$$\Sigma_{ki,t} \quad = \quad h_{ki,t} \mathrm{diag}(\mathbf{w}_{ki}),$$

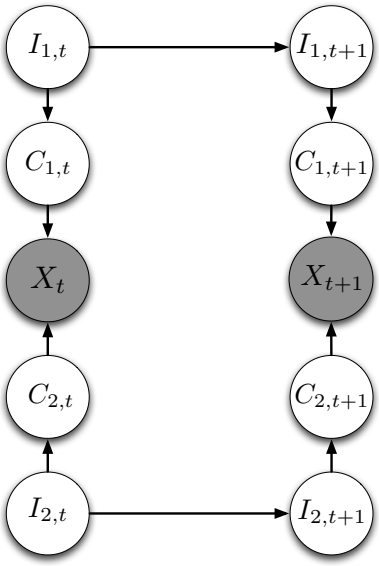


Figure 4.3: Graphical model for the FS-HMM [44].

where $\mathbf{w}_{ki}$ is a vector that corresponds to the actual spectral component and $h_{ki,t}$ corresponds to the weight of the spectral component at time t. The contribution of source k therefore corresponds to a single scaled (weighted) spectral component given by $\mathbf{c}_{k,t}$. The observation at time t is the sum of these contributions from all K sources and is given by:

$$\mathbf{x}_t = \sum_{k=1}^{K} \mathbf{c}_{k,t}.$$

$\mathbf{u}_{ki,t}$, $\mathbf{c}_{k,t}$, and $\mathbf{x}_t$ are complex vectors as they have directly modeled the complex STFT. It should however be noted that this corresponds to using the power spectrum of the mixture STFT in the EM equations that they have used to estimate the model parameters.

This model has similarities to the factorial max-HMM [54]. In both models, each source gives rise to single spectral vector. In the factorial max-HMM, the observation is the elementwise maximum of the spectral vector of each source. In the FS-HMM, the observation is a weighted summation of the spectral vectors of each source.

Like Virtanen's model [74], the FS-HMM essentially uses a single scaled spectral

	Proposed Model (N-FHMM)	Non-negative spectro-gram factoriza-tion	Factorial Max-HMM	FHMM with Ob-servations in the Feature Space	FHMM with Grammar Dynamics	FS-HMM
Reference		Raj and Smaragdis [48], Smaragdis et al. [67], etc.	Roweis [54]	Virtanen [74]	Hershey et al. [26]	Ozerov et al. [44]
Models observation of a single source as a linear combination of components from a dictionary	Yes	Yes	No	No	No	No
Models temporal dynamics	Yes	No	Yes	Yes	Yes	Yes
Language Model	No	No	No	Yes	Yes	No
Interaction Model	Additive	Additive	Max	Additive through non-linearity	Max or Additive through non-linearity	Additive

Table 4.1: Comparison of models for source separation.

component to explain each time frame of each source. This is because each state corresponds to a single spectral component. Therefore, the same scaled spectral component is essentially used to approximate every occurrence of a given state of a given source (with different scalings). In the N-FHMM however, each state corresponds to a dictionary of spectral components. Therefore, the FS-HMM models each time frame as a linear combination of spectral components, where each source corresponds to a single component. The N-FHMM also models each time frame as linear combination of spectral components. However, each source corresponds to linear combination of multiple spectral components. This is discussed in more detail below.

Discussion

We now relate the above models to the proposed model of sound mixtures (N-FHMM). A summary of the comparisons is shown in Table 4.1.

Roweis [54], Virtanen [74], Hershey et al. [26], and Ozerov et al. [44] model each time frame of each source as a single draw from a multivariate distribution. Although, this can work fairly well for monophonic sources such as speech, it has limitations with respects to polyphonic music sources. Ozerov et al. [44] have pointed this out. Certain sources of music such as the rhythm guitar are inherently polyphonic. Moreover, in certain applications, it is desirable to model multiple sources as a single source. For example, we can formulate the creation of automatic karaoke tracks as a semi-supervised source separation problem in which we model all of the instruments except for the vocals as a single instrument. We discuss this in Sec. 5.2.2.

The N-FHMM, on the other hand, models each time frame of each source as a linear combination of multiple spectral components. This is possible due to the flexible observation model and is well suited for modeling polyphonic sources. This idea of modeling a time frame as a linear combination of spectral components rather than a single component is inspired by its use in non-negative spectrogram factorizations.

The reason that using a linear combination of draws is well suited for modeling polyphonic music is that we can model different aspects of the variations of a given state by different spectral components. For example, if a given time frame of a given source corresponds to three notes, we can independently model the variations of each note and them combine them. When we model the contributions of all three notes as a single draw, each variation of each note would change the value of the single draw. This is a practical example as a given state could correspond to a chord that consists of three notes.

Although the advantages of using a dictionary of spectral components per state rather than a single spectral component per state are more obvious with polyphonic sources, it could also be useful for monophonic sources such as speech. Virtanen's [74] model and the FS-HMM [44], explain each occurrence of a given state of a given source by a scaled version of the same spectral component. Although this could be a reasonable approximation, it does not account for the variations of a given state by more than a scaling factor. The N-FHMM however accounts for these variations by using a linear combination of a dictionary of spectral components. Therefore, the N-FHMM explains different occurrences of a given state of a given source by different

linear combinations of a given dictionary rather than different scalings a of a single spectral component.

If we use a single spectral component per dictionary in the proposed model, it is similar to the FS-HMM [44]. However, in Sec. 4.3.3, we see that using multiple spectral components per dictionary rather than a single spectral component yields superior results in speech separation. Particularly, Fig. 4.8 shows the effect of using different numbers of spectral components per dictionary. If we were to use 1 spectral component per dictionary, the N-FHMM would be similar to the FS-HMM. However, we see that using 5 and 10 spectral components per dictionary yields superior results (based on three metrics that will be explained in Sec. 4.3.3).

A potential strategy of trying to model polyphonic sources with the above methods is to use an extremely large number of states. However, this could lead to problems of insufficient training data unless a very large amount of data is available (which is often not the case). Moreover, the FHMM has a combinatorial computational complexity. Particularly, the complexity is exponential in the number of sources. For example, if each source has ten possible states and we consider two sources, there are one hundred possible combinations of states to consider in each time frame. Although approximate inference algorithms can be used to alleviate some of the issues caused by using a large number of states, as done by Roweis [54], Virtanen [74], and Rennie et al. [49, 50], the complexity still increases by some amount with the number of states.

Although the need for a large number of states is more obvious for polyphonic sources, some of the above methods use a large number of states even for speech. For example, Roweis [54] used more than 8000 states per source. In the N-FHMM, we use a mere 40 states per source (as will be described below).

Another issue with the above models has to do with the interaction model. Since audio is additive, an additive interaction model is a simple model that reflects the underlying process. With the exception of the FS-HMM [44], none of the above FHMM models use a simple additive interaction model. They either use a non-linear model that has an additive step [26, 74] or they simply use a model that implies a binary mask [26, 54]. Although a binary mask could work reasonably well, it also has

chances of creating artifacts such as musical noise (as discussed earlier in Sec. 4.2.1).

Hershey et al. [26] and Virtanen [74] have used language models. As pointed out by Hershey et al. [26] this greatly improves the source separation and speech recognition performance as it highly constrains the problem. It should be noted that the use of language models necessitates the use of a large number of states due to the use of context-dependent phoneme models. If we were to use a language model in the N-FHMM, we would also need to used a large number of states.

4.2.4 Non-negative Spectrogram Factorization

One of the main attractive qualities of non-negative spectrogram factorization techniques is the ability to model rich polyphonic sources. They have been used by various researchers for source separation. In fact, almost all of the non-negative spectrogram factorization techniques that have been mentioned in Sec. 2.2 have been applied to source separation.

We discuss the application of the asymmetric factorization method, introduced by Raj and Smaragdis [48], to supervised source separation. This is the model that was discussed in Sec. 2.2.2 and Sec. 3.2. This method is representative of this class of techniques. The source separation procedure is the same as the general procedure outlined for model based techniques. Specifically, it is as follows:

1. Compute the spectrograms of the isolated training data of the individual sources and the sound mixture.

2. Train models of single sources from training data spectrograms of the sources. The models consist of the spectral components of each source $P(f|z,s)$.

3. Use the model of sound mixtures described in Sec. 3.2, keeping the spectral components of each source fixed.

4. Given the spectrogram of the sound mixture, estimate the weights of each source $P_t(z_t|s_t)$.

5. Perform reconstructions of the individual sources using:

$$P_t(f_t, s_t) = P_t(s_t) \sum_{z_t} P_t(z_t|s_t) P(f_t|z_t, s_t).$$

6. Use these reconstructions to estimate a soft mask using Eq. 4.4.

7. Obtain the estimates of the separated source spectrograms by modulating the mixture spectrogram using Eq. 4.3.

8. Obtain the time domain signals of the individual sources by taking the inverse STFT, using the separated spectrograms and phase of the mixture STFT.

It should be noted that we estimate spectrograms of the individual sources but use the phase of the mixture STFT for all sources. As discussed in Sec. 1.2, we can attribute each time–frequency bin of the mixture STFT to a single source. Therefore the magnitude and phase in that time–frequency bin will be due to mainly one source. When we use take the inverse STFT of a given reconstructed source using the mixture phase, one of two things will happen in each time–frequency bin. In the first case, that time–frequency bin will primarily correspond to the given source. The mixture phase will therefore primarily correspond to the given source and could be considered a good approximation to the true phase of that source. In the second case, that time–frequency bin will primarily not correspond to the given source. In that case, the magnitude of the given source will be very low in that time–frequency bin. Since the mixture phase will correspond primarily to the other source, the inverse STFT (with respect to that time–frequency bin) will primarily be done with the wrong phase. However, since the magnitude of that source is so low in that time–frequency bin, this error will be of minimal consequence.

In signal processing techniques that involve the estimation of a time domain signal from a spectrogram (magnitude information only), Griffin and Lim's method [24] is traditionally used. However, in the source separation community, it is common to obtain the separated time domain signals using the separated spectrograms and the mixture phase for the reasons described above. We have therefore tried this method. Specifically, we used the separated spectrograms and initialized Griffin and Lim's

method with the mixture phase. We then iteratively estimated the time domain signals. We found that the resulting time domain signals yielded higher artifacts (according to the SAR metric that will be discussed in Sec. 4.3.3) than directly using the mixture phase. We therefore report all of the metrics in Sec. 4.3.3 based on directly using the mixture phase. It should be noted that this increase in artifacts might be due to the fact that the SAR metric is computed directly on the time domain signals rather than a more perceptually based representation.

The problem with non-negative spectrogram factorization techniques is that they do not model non-stationarity or temporal dynamics. The proposed model extends these methods by modeling these things. Another recent extension was made by Ozerov et al. in which NMF was extended to multi-channel source separation [43].

4.3 Non-negative Factorial Hidden Markov Models

In this section, we describe the application of the proposed models to single channel supervised source separation. We first describe the procedure. This is followed by a few simple illustrative examples. Finally, we present experimental results.

4.3.1 Procedure

The procedure for performing supervised source separation using the N-HMM and the N-FHMM is as follows:

1. Compute the spectrograms of the isolated training data of the individual sources and the sound mixture.

2. Train N-HMMs of single sources from training data spectrograms of the sources. The model of a given source consists of the spectral components $P(f|z,s,q^{(s)})$, transition matrix $P(q_{t+1}^{(s)}|q_t^{(s)})$, energy distributions $P(v_t^{(s)}|q_t^{(s)})$, and optionally the initial state probabilities $P(q_1^{(s)})$.

3. Using an N-FHMM, fix the above parameters for each source.

4. Given the spectrogram of the sound mixture, estimate the weights $P_t(z_t, s_t|q_t^{(1)}, q_t^{(2)})$ and optionally the initial state probabilities $P(q_1^{(s)})$ of each source.

5. Perform reconstructions of the individual sources as describe in Sec. 3.4.3 using:

$$P(f_t, s_t|\overline{\mathbf{f}}, \overline{\mathbf{v}}) = \sum_{q_t^{(1)}} \sum_{q_t^{(2)}} P(q_t^{(1)}, q_t^{(2)}|\overline{\mathbf{f}}, \overline{\mathbf{v}}) \sum_{z_t} P(f_t|z_t, s_t, q_t^{(s)}) P(z_t, s_t|q_t^{(1)}, q_t^{(2)}).$$

6. Use these reconstructions to estimate a soft mask using Eq. 4.4.

$$P(s_t|f_t, \overline{\mathbf{f}}, \overline{\mathbf{v}}) = \frac{P(f_t, s_t|\overline{\mathbf{f}}, \overline{\mathbf{v}})}{\sum_{s_t} P(f_t, s_t|\overline{\mathbf{f}}, \overline{\mathbf{v}})}.$$

7. Obtain the estimates of the separated source spectrograms by modulating the mixture spectrogram using Eq. 4.3.

8. Obtain the time domain signals of the individual sources by taking the inverse STFT, using the separated spectrograms and phase of the mixture STFT. As with non-negative spectrogram factorization, using the phase of the mixture STFT is a reasonable approximation for the reasons mentioned in Sec. 4.2.4.

4.3.2 Examples

In order to illustrate the performance of the proposed models and compare them to that of non-negative spectrogram factorization, we show source separation results on a few simple music examples. These examples highlight the advantages of explicitly modeling non-stationarity and temporal dynamics. In all of the examples, we train models of single sources on isolated clips of the sources. We then artificially mix the same isolated sources and use that as the mixture on which source separation is to be performed. This means that the training data and the test data are essentially the same. We have done this for illustrative purposes.

In a real world scenario, the training data will not be the same as the test data. However, if the training data exhibits the characteristics of the test data, then the estimated parameters of the N-HMM are likely to be quite similar. Also, the purpose

of these examples is to highlight the need to explicitly model non-stationarity and temporal dynamics. We show that the proposed model exhibits superior performance due to this, when compared to non-negative spectrogram factorization. Since the purpose is a comparison, it is fair if we use the same setup to show the performance of both methods. In the following section, we present a more real world scenario in which the test data is different from the training data.

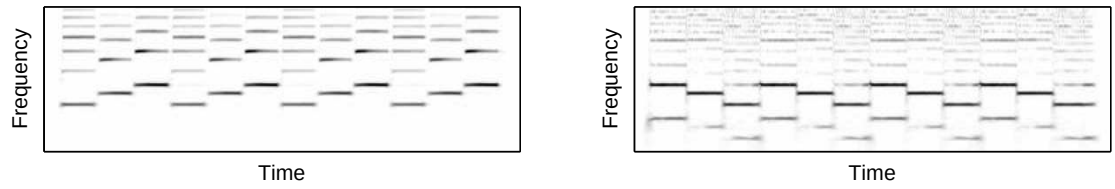
We consider three examples in increasing order of difficulty. We present purely qualitative results here as this is simply an illustration. In the first example, non-negative spectrogram factorization does well. In the second example, we can see some slight artifacts when using non-negative spectrogram factorization. In the third example, non-negative spectrogram factorization almost completely fails. The proposed models do well in all three examples.

Example 1

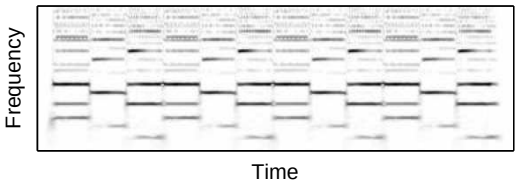
In this example, the first source is a synthesized saxophone playing ascending C major arpeggios. This is the same example that was shown in Fig. 2.13a. The second source is a synthesized electric guitar playing descending C major arpeggios in a lower octave. The results are shown in Fig. 4.4. We use the constant-Q transform [9] for display purposes (only) so that we can clearly see the fundamental frequencies and harmonic structure of the different notes. Since there is a high degree of frequency separation of the two sources, both methods perform well in this example.

Example 2

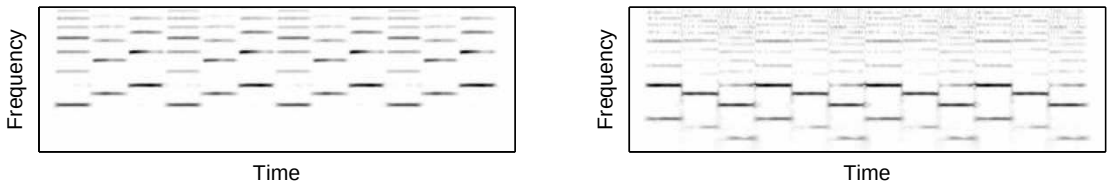
In this example, the first source is the same synthesized saxophone playing ascending C major arpeggios. The second source is a synthesized electric guitar playing descending C major arpeggios in the same octave. The results are shown in Fig. 4.5. Even though the notes are in the same octave, the timbral differences between the two instruments help differentiate the spectral patterns of the two sources. Non-negative spectrogram factorization still does fairly well. However, some erroneous harmonics show up on the third note of the separated electric guitar as seen in Fig. 4.5c. The



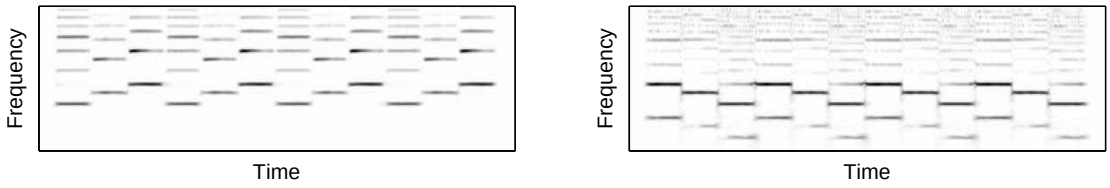
(a) Original sources



(b) Mixture

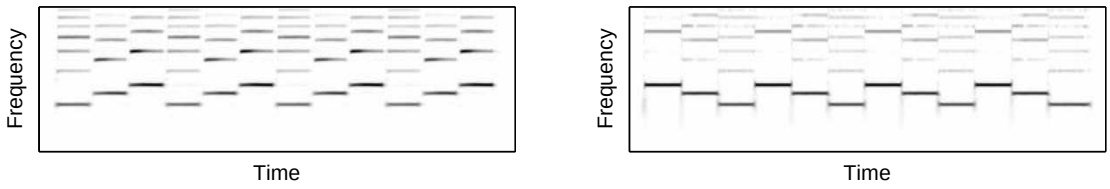


(c) Separated sources using non-negative spectrogram factorization

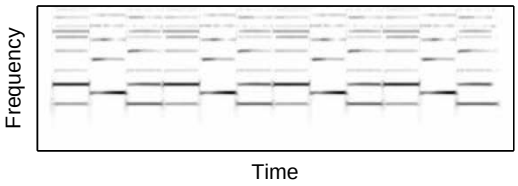


(d) Separated sources using the N-FHMM

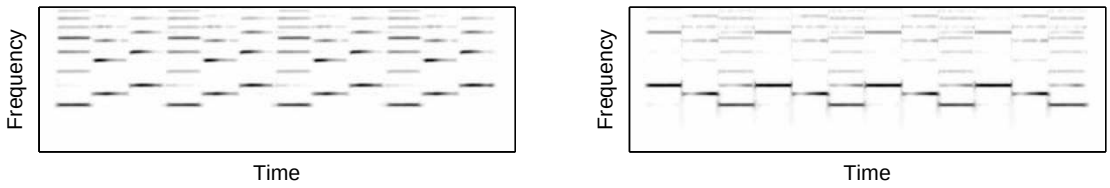
Figure 4.4: Example of source separation using the N-FHMM. The first source is an ascending arpeggio played by a synthesized saxophone. The second source is a descending arpeggio on a different octave played by a synthesized electric guitar.



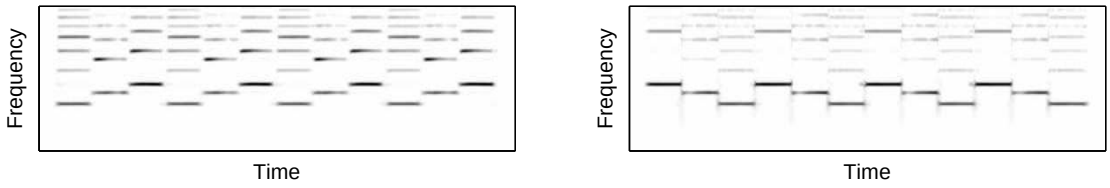
(a) Original sources



(b) Mixture



(c) Separated sources using non-negative spectrogram factorization



(d) Separated sources using the N-FHMM

Figure 4.5: Example of source separation using the N-FHMM. The first source is an ascending arpeggio played by a synthesized saxophone. The second source is a descending arpeggio on the same octave played by an synthesized electric guitar.

N-FHMM still does a good job of separation.

Example 3

In this example, the first source is the same synthesized saxophone playing ascending C major arpeggios. The second source is the same synthesized saxophone on the same octave. However it is playing descending C major arpeggios. The results are shown in Fig. 4.6. Since there are no timbral differences in the two sources, the models of the individual sources that have been learned by non-negative spectrogram factorization are almost identical [1]. Source separation therefore almost completely fails as seen in Fig. 4.6c. Both of the separated sources look very similar to the sound mixture.

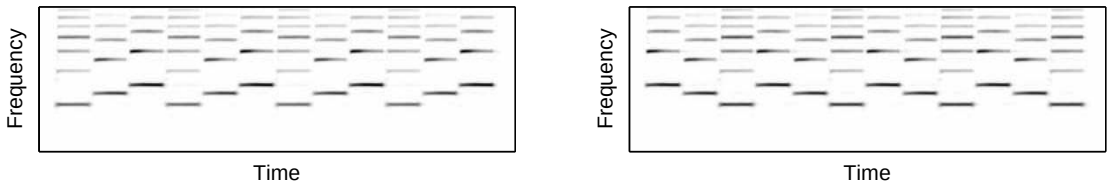
The N-FHMM on the other hand uses the temporal dynamics of the two sources. The temporal dynamics tells us that the first source corresponds to only ascending arpeggios and the second source corresponds to only descending arpeggios. This information is reflected in the transition matrices of the individual sources. The transition matrix of the first source is shown in Fig. 2.14. This has allowed the N-FHMM to still do a good job of source separation, as shown in Fig. 4.6d.

This is an extreme example that has been used to illustrate the value of modeling temporal dynamics. Although, we are not likely to commonly encounter such extreme examples in practice, it shows how temporal dynamics can be used to resolve ambiguities. Most real world examples are not as simple as the examples presented in this section and there is often a great deal of ambiguity.

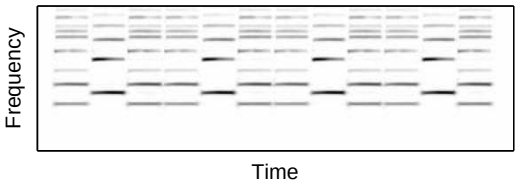
4.3.3 Experiments

We now present the results of experiments that were performed to evaluate the performance of the proposed models. The experiments are on single channel supervised speech separation of two speakers using the proposed models. In each experiment, we consider a mixture of a male and a female speaker. The data was obtained from the TIMIT database. Each experiment consists of the following:

[1] The slight difference between the models are attributed to the time frames of the training data that correspond to the transitions between notes.

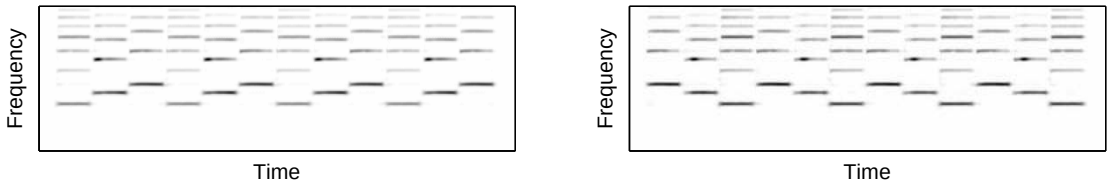


(a) Original sources



(b) Mixture

(c) Separated sources using non-negative spectrogram factorization



(d) Separated sources using the N-FHMM

Figure 4.6: Example of source separation using the N-FHMM. The first source is an ascending arpeggio played by a synthesized saxophone. The second source is a descending arpeggio on the same octave played by by the same synthesized saxophone.

1. Training data for a given speaker — Nine sentences of that speaker that have been concatenated.

2. Test data — A mixture of one sentence of each of the above speakers. The individual speech files have been mixed at 0dB gain. These sentences are different from the sentences in the training data.

The spectrogram was computed using a window size of 64ms and a hop size of 16ms. Separation was performed on eight different pairs of speakers. We report the average results.

We first discuss the metrics that we use to judge the performance. We then discuss model selection by presenting the performance of the proposed models for different configurations of user-defined parameters. Finally, we present experiments that compare the performance of the proposed model to that of non-negative spectrogram factorization.

Metrics

We use source separation metrics that were developed by Vincent et al. [73, 18]. Specifically, we use the following three metrics:

1. Signal to Interference ratio (SIR) — This is a measure of the suppression of the unwanted source.

2. Signal to Artifact ratio (SAR) — This is a measure of the artifacts (such as musical noise) that have been introduced by the separation process.

3. Signal to Distortion ratio (SDR) — This is an overall measure of performance as it accounts for both of the above criteria.

The goal is to maximize SIR (as this is the measure of the actual separation) while trying to keep SAR as high as possible (in order to prevent the introduction of artifacts).

In order to compute these metrics, the estimated time domain signals of the individual sources $x_1(t)$ and $x_2(t)$ are decomposed into three parts [73, 18]. A given estimated time domain signal $x_i(t)$ is decomposed as a sum of the following parts:

1. s_{target} — actual source estimate
2. e_{interf} — interference signal (i.e. the unwanted source)
3. e_{artif} — artifacts of the separation algorithm

The decomposition is done up to a constant scaling factor. Using these terms, the metrics are computed as follows:

$$\begin{aligned} \text{SIR} &= \frac{||s_{target}||^2}{||e_{interf}||^2}, \\ \text{SAR} &= \frac{||s_{target} + e_{interf}||^2}{||e_{artif}||^2}, \\ \text{SDR} &= \frac{||s_{target}||^2}{||e_{artif} + e_{interf}||^2}. \end{aligned}$$

Model Selection

There are two user-defined parameters for a given source when using the proposed models. They are as follows:

1. Number of dictionaries
2. Number of spectral components per dictionary

The optimal number of dictionaries really depends on the data at hand. However, we evaluated the performance using different numbers of dictionaries for each source keeping the number of spectral components per dictionary fixed at 10. The results are shown in Fig. 4.7. As mentioned above, these are the average results over the eight pairs of speakers. Therefore, they are the average results of sixteen individual speakers. As seen in the figure, the optimal number of dictionaries with respect to SIR is 40.

The optimal number of spectral components per dictionary is a property of the class of data. For example, if we were to perform speech separation on speech from a different database with different sizes of training and test data, these results are

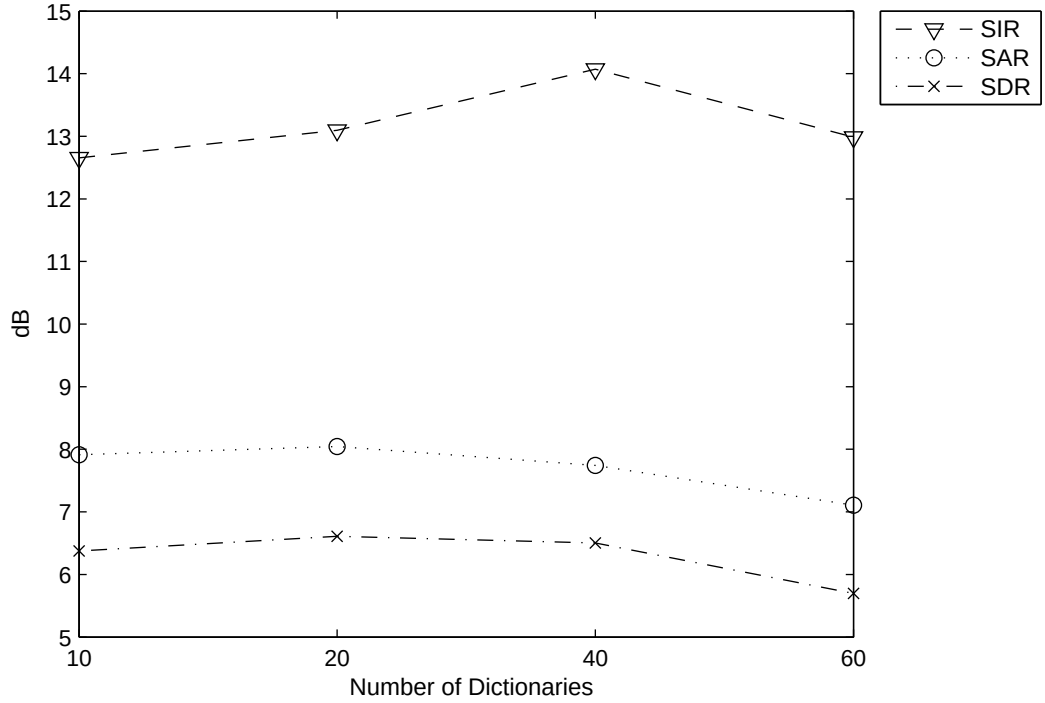


Figure 4.7: Metrics for finding the optimal number of dictionaries when performing source separation using the N-FHMM (the number of spectral components per dictionary has been fixed at 10). These metrics were computed on single channel supervised speech separation examples.

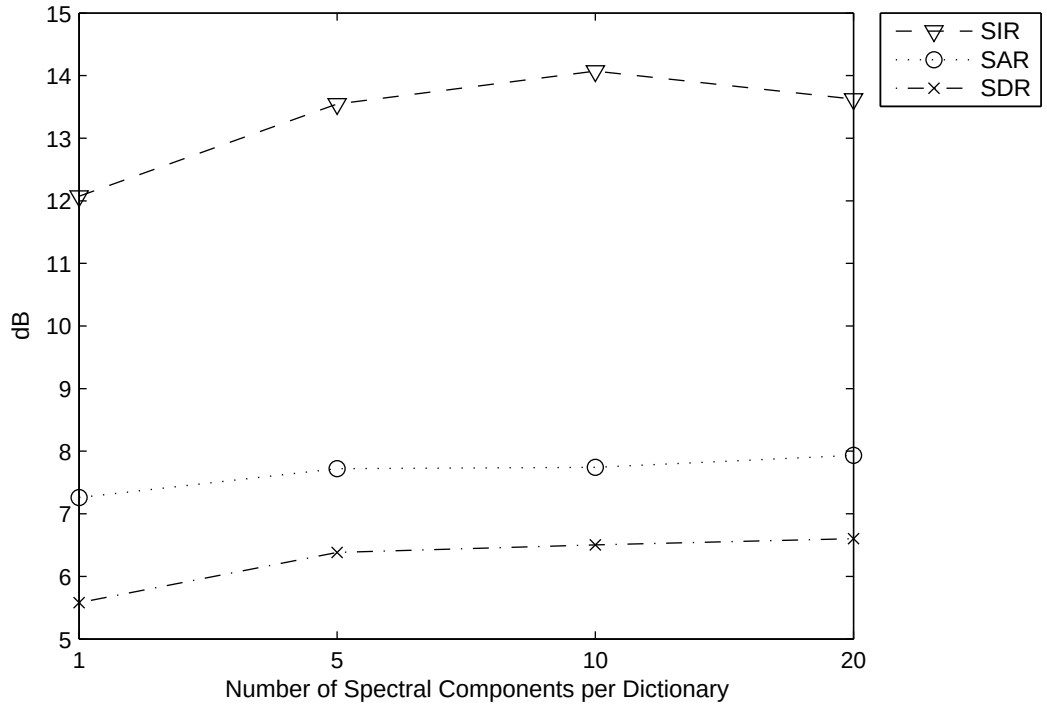


Figure 4.8: Metrics for finding the optimal number of spectral components per dictionary when performing source separation using the N-FHMM (the number of dictionaries has been fixed at 40). These metrics were computed on single channel supervised speech separation examples.

likely to generalize. Keeping the number of dictionaries fixed at 40, we evaluated the performance using different numbers of spectral components per dictionary. The results are shown in Fig. 4.8. As seen in the figure, the optimal number of spectral components per dictionary with respect to SIR is 10.

An example of the estimated parameters using 40 dictionaries and 10 spectral components per dictionary is shown in Fig. 2.12.

These results should be compared to the model selection procedure in Sec. 2.4.4, which simply indicated that we should choose the most complex models. The above results indicate that the optimal models for source separation are not the most complex models. A possible explanation for this is that if the model is too complex, then it will have a large number of spectral components. The spectral components of one source could then plausibly be used to explain the other source. Therefore,

we could say that the model will not be as discriminative if the model complexity of the individual N-HMMs are too high. It should be noted that the above results indicate optimal parameters for source separation. However, they are not necessarily the optimal parameters for other applications.

When we used a single spectral component per dictionary, this model is similar to the FS-HMM [44]. However, Fig. 4.8 indicates that more than one component per dictionary is optimal for source separation based on all three metrics. It should be noted that this is the case even though the sources are monophonic.

Comparison to Non-negative Spectrogram Factorization

As the proposed models are hypothesized to outperform non-negative spectrogram factorization due to the modeling of non-stationarity and temporal dynamics, we compared the source separation performance using the two classes of models. For the proposed models, we used the results of using 40 dictionaries with 10 spectral components each (as found to be optimal above). We performed the same experiments using non-negative spectrogram factorization. Specifically, we used the model proposed by Raj and Smaragdis [67].

In order to perform a fair comparison, we evaluated the optimal parameter configuration for non-negative spectrogram factorization. The results are shown in Fig. 4.9. The only user-defined parameter when using this technique is the number of spectral components per source. This optimal value of this parameter with respect to SIR was found to be 30. Therefore, we used this configuration for the comparison.

The comparison of the source separation performance is shown in Table 4.2. The SIR is over 5 dB higher when using the proposed model. However, there is only a slight decrease in the SAR (approximately 0.2 dB), when using the proposed model. The overall performance taking both of these criteria into account is reflected in the SDR. As seen, the N-FHMM clearly outperforms non-negative spectrogram factorization in this task.

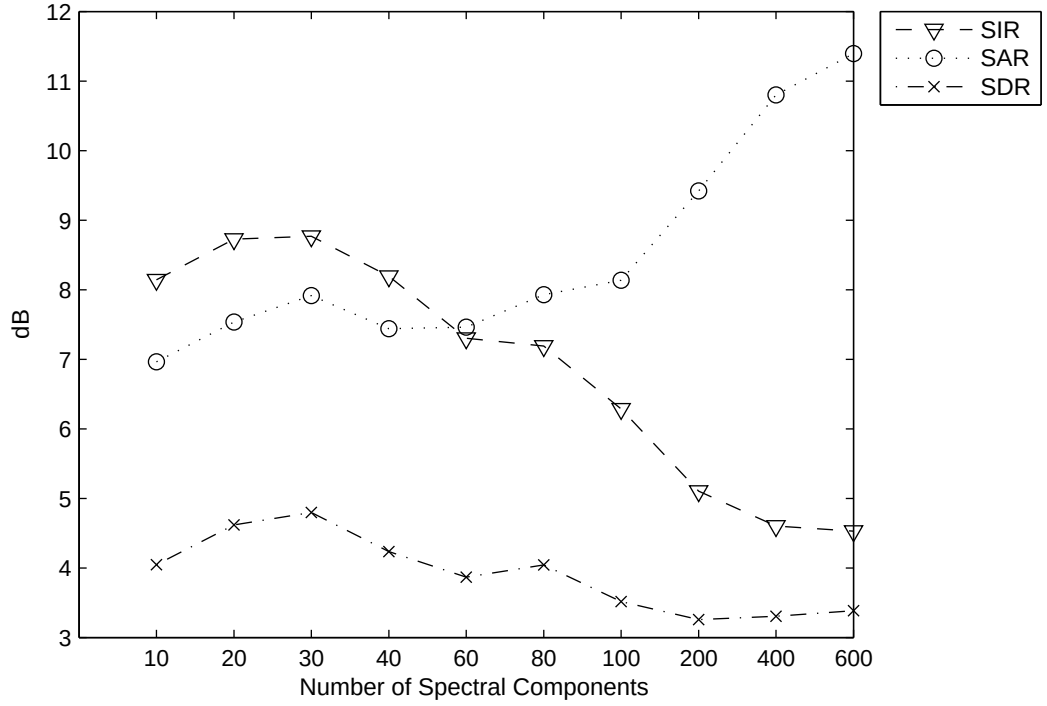


Figure 4.9: Metrics for finding the optimal number of spectral components when performing source separation using non-negative spectrogram factorization. These metrics were computed on single channel supervised speech separation examples.

	SIR (dB)	SAR (dB)	SDR (dB)
N-FHMM	14.07	7.74	6.49
Non-negative Spectrogram Factorization	8.65	7.95	4.82

Table 4.2: Comparison of single channel supervised source separation performance with the N-FHMM and non-negative spectrogram factorization.

4.4 Conclusions

In this chapter, we presented the application of the proposed models to single channel supervised source separation. We started with an overview of source separation techniques. Techniques that used FHMMs, its variants, and non-negative spectrogram factorization were described in more detail and compared to the proposed model.

We illustrated the importance of modeling temporal dynamics with a few simple examples that qualitatively compared the source separation performance of the proposed models with that of non-negative spectrogram factorization. We then presented the results of single channel supervised speech separation experiments to perform model selection in the proposed models. Finally, we presented a comparison of the proposed models and non-negative spectrogram factorization in single channel supervised speech separation.

Chapter 5

Conclusions and Future Research

5.1 Overview

In these thesis, we presented a framework for modeling sound mixtures. Particularly, we presented a new model of sound mixtures, the non-negative factorial hidden Markov model (N-FHMM). In this model, we addressed three important aspects of audio structure – spectral structure, non-stationarity, and temporal dynamics. We also presented a mechanism in which we can perform high quality reconstructions of the individual sources.

We also presented a new model of single sound sources, the non-negative hidden Markov model (N-HMM). The N-FHMM is composed of an N-HMM for each constituent source. Therefore, the N-FHMM explicitly models spectral structure and temporal dynamics of each individual source.

We presented a conceptual explanation, the probabilistic model, and a parameter estimation technique for each of the above models. We presented a method of performing content–aware audio processing with the N-HMM. We presented a method of performing single channel supervised source separation using the N-HMM to train models of individual sources and the N-FHMM to perform separation. We also compared the proposed models to other related techniques.

Although we presented two specific applications of the proposed models, they can be useful for various other applications that involve sound mixtures. That will be

discussed in this chapter.

5.2 Future Directions

In this section, we discuss various potential extensions of the proposed framework. We first discuss algorithmic extensions and then discuss specific applications. The algorithmic extensions are as follows:

1. Approximate Inference
2. Prior Distributions
3. Music Theory Rules and Language Models

The applications are as follows:

1. Semi-Supervised Source Separation
2. Denoising
3. Imputation
4. Automatic Music Transcription
5. Chord Recognition
6. Concurrent Speech Recognition

5.2.1 Algorithmic Extensions

Approximate Inference

In the N-FHMM, each source can be in any one of its N states at each time frame. This means that each time frame can correspond to a total of N^M possible combinations of states if there are M sources. This leads to a combinatorial complexity in inference (E step of the parameter estimation algorithm). Particularly, the complexity is exponential in the number of sources.

As an alternative to exact inference, approximate inference can be used for this purpose. This can considerably reduce the computational complexity. This can often be done without much difference in performance. Various approaches to approximate inference have been used in the literature. The following methods have been used for approximate inference in FHMMs:

1. Sampling Methods – Sampling methods are commonly used to perform inference in computationally expensive inference tasks. These methods offer the assurance that the sampling will converge to the true solution in the limit. The problem is that the amount of sampling required to obtain a satisfactory solution could in practice be quite high leading to a slow convergence. Ghahramani and Jordan [23] used Gibbs sampling for inference in FHMMs. They noted that variational inference was faster than Gibbs sampling without a loss in accuracy. An extensive review of different sampling methods has been made by Neal [42].

2. Variational Inference – Variational inference is a technique that is used to approximate a given distribution with a another more tractable distribution. As opposed to sampling methods, it is a deterministic technique. A tutorial on the application of this class of techniques to graphical models has been made by Jordan et al. [30]. Ghahramani and Jordan [23] presented two forms of variational inference in the context of FHMMs – completely factored variational inference and structured variational inference. They mention that structured variational inference seems to be the better choice as it retains a great deal of structure of the original distribution with a considerable decrease in computation (compared to exact inference). The original distribution in this case is the distribution over all of the random variables in the FHMM. Structured variational inference basically decouples the individual HMMs in the FHMM. A fictitious observation is attached to each constituent HMM. The forward–backward algorithm can then be performed in each constituent HMM. This has a complexity that is linear in the number of sources. This will yield another distribution that is not the original distribution. The KL divergence is then minimized between this distribution and the true distribution, yielding another

set of fictitious observations. This procedure is iterated.

3. Loopy Belief Propagation – The max-product belief propagation method is a message passing algorithm that can be used to perform inference in graphical models. This is described by Weiss and Freeman [78]. When used on tree structured graphical models, it yields the maximum a posteriori (MAP) estimate of the hidden variables given the observed variables. In graphical models with loops (such as the FHMM), this algorithm can be applied iteratively. Although, it is not guaranteed to yield the MAP estimate, if it converges, it will be a local MAP estimate over a potentially exponentially large neighborhood. This is also described by Weiss and Freeman [78]. Rennie et al. [49] have used this method for speech separation and speech recognition using a factorial HMM. They report that the algorithm is almost an order of magnitude faster than exact inference with almost the same performance.

4. Other Methods – There are certain ways of combining the above methods. For example, Rennie et al. [50] have combined variational inference and loopy belief propagation. There are also other methods that do not exactly fall into the above categories. For example, Virtanen [74] has used a greedy method as described in Sec. 4.2.3.

Prior Distributions

A number of parameters are estimated in the proposed models. Prior knowledge about the structure of these parameters can greatly improve the quality of the estimation as it can constrain the problem in a meaningful way. As many of the parameters have physical and intuitive interpretations, we can specify meaningful priors. For example, the spectral components are spectral patterns that correspond to different sources. If we know something about the source, we can specify priors on the components. This knowledge could be from musical acoustics [53, 19]. It could also be from a user input mimicking a source as described by Smaragdis and Mysore [66] or a synthesized rendition of a MIDI based score as described by Ganseman et al. [22].

It is beneficial to specify priors that are conjugate to a given distribution. Since most of the distributions in the proposed models, including the spectral components, are multinomial distributions, we can conveniently specify Dirichlet priors due to their conjugacy to multinomial distributions. The only distributions that are not multinomials are the energy distributions. These distributions are Gaussian. Since Gaussian distributions are conjugate to themselves, we can conveniently specify priors to these distributions in the same form.

Music Theory Rules and Language Models

Temporal dynamics is one of the key focus areas of this thesis. There is a great deal of known structure in the temporal dynamics of music and speech. This can be used to meaningfully constrain the given estimation problems.

Music theory rules can be encoded in a transition matrix. For example, if the given dictionaries correspond to notes (this can be forced using prior distributions or by fixing the dictionaries), the transitions between the notes can be specified using music theory rules. The same idea can be used if the dictionaries correspond to chords.

Language models are used in most practical speech recognition systems. Since the states have the same interpretation in the proposed models as in traditional speech recognition systems, language models can be used in the same way. This has shown to be effective in speech separation by Hershey et al. [26].

5.2.2 Applications

Semi-Supervised Source Separation

In Sec. 4.3, we described the use of the proposed model to perform supervised separation. We estimated the spectral components and transition matrices for each of the sources in the mixture from training data. In many cases, this training data is not easily available. If training data for a subset of the sources is available we can estimate the spectral components and transition matrices for only those sources from training data. For the remaining sources, we estimate the parameters directly

from the mixture (along with the mixture weights). This is called semi-supervised separation.

This can be quite useful for the extraction of vocals (or lead instrument) from a recording. If we can isolate a section of the recording without vocals (or lead instrument), we can consider the background music in that section as a single source. We can estimate the spectral components and transition matrix for that source from the given section. In the section which has the vocals (or lead instrument), we fix the spectral components and transition matrix that corresponds to the background music and then estimate the remaining parameters. We then reconstruct the background music and vocals separately as described in Sec. 4.3.

This can be useful of generating automatic karaoke tracks and automatic "jam tracks". It can also be useful to extract a given instrument from a mix, process it, and re-insert it into the mix.

Denoising

A noisy source can be considered as a mixture of the given source and noise. If we have a clean recording of training data for that source, we can then treat denoising as a source separation problem. If we have an isolated section of noise, we can treat it is a supervised source separation problem. If not, we can treat it as a semi-supervised source separation problem. Since we are treating the noise a source, it is likely that we would be able to handle highly non-stationary noise.

Imputation

Audio restoration from highly corrupted or clipped recordings can be a challenging problem. It is often the case that certain segments of a recording are corrupted while other segments are clean. We can consider the corrupted regions of the spectrogram as missing values and consider the restoration problem as that of imputing the missing values. Using the clean segments as observations and the corrupted segments as priors, we estimate the model parameters. We then reconstruct the missing values from the estimated parameters. Smaragdis et al. [69] have done this using ideas from

non-negative spectrogram factorization. The proposed models are likely to make the estimation more accurate as they account for non-stationarity and temporal dynamics.

Automatic Music Transcription

The main goals of automatic music transcription [32] of polyphonic music are to estimate the pitch, gain, and locations of each note of each instrument in time. As this is an analysis application, reconstruction quality is not important. We start with the estimation of the pitch of a single known monophonic instrument. Let us assume that the range of notes is known a priori and that the notes conform to the piches of a chromatic scale (slight detuning is acceptable). This is a fair assumption for a great deal of music. Moreover, if the specific range of notes is not known, we can assume that the range spans the entire range of the given instrument.

Given that we know the instrument and permissible pitches, we can synthesize samples of those pitches from any synthesizer of that instrument. Although a synthesizer of that instrument will not have the exact same timbre as the original instrument, it will maintain certain acoustical properties of the the original instrument. This idea was used by Ganseman et al. [22] for source separation based on MIDI based score information.

We separately estimate a single spectral component for each pitch using non-negative spectrogram factorization. This will capture the average spectral shape for each note. We assign each spectral component to its own dictionary. We then specify the transition matrix between the dictionaries according to music theory rules between the notes. For a given instrument, we will therefore have the spectral components and transition matrix available to us. We then perform N-HMM parameter estimation on unseen data for which the pitch is to be estimated. Keeping the spectral components and transition matrix fixed, we estimate the mixture weights. The weight of a given component at a given time frame will indicate the gain of that particular pitch at that time frame. Therefore, we will have the information of the pitch, gain, and location of each note in time, yielding music transcription of a single monophonic instrument.

In the context of a mixture, we estimate a single spectral component for each pitch of each instrument from synthesized data. We then perform N-FHMM parameter

estimation on unseen data for which the pitches are to be estimated. Keeping the spectral components and transition matrices fixed, we estimate the mixture weights. The weight of a given component at a given time frame will indicate the gain of that particular pitch for that instrument. This will therefore give us a pitch estimation for each of the given instruments. It should be noted that rather than fixing the spectral components that are learned from synthesized data, we could alternatively use them as Dirichlet priors when performing parameter estimation in the mixture.

Recordings of training data of the given instruments are not required. We simply need to know the name of the instrument. We can use readily available synthesizers to generate the data. The synthesized data does not need to timbrally correspond exactly to the recording of the instrument. However, when dealing with multiple instruments, it is important that the timbre of the synthesized instrument is closer to the given acoustical instrument than the other acoustical instruments [22]. This is a fair assumption in many cases. Also, in cases in which there is a fair amount of overlap in timbres, the temporal dynamics as specified by the music theory rules in the transition matrices, are likely to help disambiguate potential confusion.

Since we use a single spectral component per note, there could potentially be a great deal of approximation error in the reconstruction. This is however not a problem since reconstruction is not the goal. We could use more than one spectral component per dictionary. However, if the spectral shape of a given note of a given instrument does not have a great deal of variations, we hypothesize that a single spectral component per dictionary will produce better transcription results. When we use more than one spectral component per dictionary, each individual component is not as representative of the data. We therefore have the risk of a dictionary of one source explaining the data of the other source. For example, if we were to use a large number of spectral components to model a single note, a single component might model a single harmonic of the note. Multiple notes share the same harmonic. Therefore the spectral component that corresponds to a harmonic in a dictionary of a given source could potentially easily model the data of the other source.

Chord Recognition

The ideas in chord recognition are similar to that of automatic music transcription. The difference is that each dictionary would now correspond to a chord rather than a single note. The constituent spectral components of each dictionary would correspond to the constituent notes of the given chord. The transition matrix could be specified from music theory rules or learned from data as done by Lee and Slaney [37].

Since we use a separate N-HMM with its own dictionaries to model each source, we can use the chord model for some sources and the simple melody model (as described in the context of automatic music transcription) for other sources. These individual N-HMMs can be incorporated into a single N-FHMM. This corresponds well to a great deal of real music. It is often the case that certain instruments play chords while other instruments play melodies.

Concurrent Speech Recognition

The proposed models can be used to model speech in a similar fashion to the modeling of speech by traditional HMMs in speech recognition systems. The main difference is in the observation model. Also, the proposed model would ideally use speaker specific models. However, it could potentially be used for gender specific models.

The state sequences can be interpreted in a manner similar to that of traditional speech recognition systems. Therefore, the individual state sequences of the N-FHMM could be used to model the speech of different speakers. We could then use a 2D Viterbi algorithm or an approximation of this to find the optimal state sequences of the speakers. This would give us concurrent speech recognition of each of the speakers in the mixture.

FHMMs have been used for this application by Virtanen [74] and Hershey et al. [26].

5.3 Closing Remarks

Analyzing and processing individual sources within a sound mixture is a challenging problem, particularly with single channel sound mixtures. In order to attempt this problem, it is crucial to take into account the characteristics of the individual sources. Audio is a rich form of data with a great deal of information. The challenge is in correctly making use of this information. In order to do this effectively, we need to use models that conform to general structural characteristics of audio. Within such models, the characteristics of the specific source at hand can be specified by estimating parameters from data of that source. If high–quality reconstructions of individual sources is important, then the models have to be designed such that they are amenable to this.

The proposed models take into account three particular aspects of audio structure – spectral structure, non-stationarity, and temporal dynamics. We have shown that this can yield superior performance to using spectral structure alone. Also, the proposed models are amenable to high quality reconstructions of both monophonic and polyphonic sources.

This is an exciting area of research as it enables numerous useful applications. There are many other aspects of modeling the structure of audio that are waiting to be explored.

Bibliography

[1] Hirotugu Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723, 1974.

[2] Francis Bach and Michael Jordan. Discriminative training of hidden markov models for multiple pitch tracking. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Philadelphia, PA, March 2005.

[3] Leonard E. Baum, Ted Petrie, George Soules, and Norman Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. *The Annals of Mathematical Statistics*, 41(1):164–171, 1970.

[4] Anthony J. Bell and Terrence J. Sejnowski. An information maximization approach to blind separation and blind deconvolution. *Neural Computation*, 7:1129–1159, 1995.

[5] Laurent Benaroya, Frédéric Bimbot, and Rémi Gribonval. Audio source separation with a single sensor. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(1), January 2006.

[6] Michael S. Brandstein and Darren Ward. *Microphone Arrays: Signal Processing Techniques and Applications*. Springer, 2001.

[7] Albert S. Bregman. *Auditory Scene Analysis: The Perceptual Organization of Sound*. The MIT Press, 1990.

[8] Guy J. Brown and Martin Cooke. Computational auditory scene analysis. *Computer Speech and Language*, 8, 1994.

[9] Judith C. Brown. Calculation of a constant Q spectral transform. *Journal of the Acoustical Society of America*, 89:425–434, 1991.

[10] Jean-Francois Cardoso. Blind signal separation: statistical principles. *Proceedings of the IEEE*, 9(10):2009–2025, 1998.

[11] Michael A. Casey and Alex Westner. Separation of mixed audio sources by independent subspace analysis. In *Proceeding of the International Computer Music Conference (ICMC)*, Berlin, Germany, August 2000.

[12] Ali Taylan Cemgil. Bayesian inference in non-negative matrix factorisation models. *Computational Intelligence and Neuroscience*, January 2009.

[13] Steven B. Davis and Paul Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 28(4):357–366, 1980.

[14] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38, 1977.

[15] Inderjit Dhillon and Suvrit Sra. Generalized nonnegative matrix approximations with bregman divergences. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 283–290. MIT Press, Cambridge, MA, 2006.

[16] Daniel P. W. Ellis. *Prediction–driven computational auditory scene analysis.* PhD thesis, M.I.T., 1996.

[17] Cedric Fevotte, Nancy Bertin, and Jean-Louis Durrieu. Nonnegative matrix factorization with the itakura-saito divergence with application to music analysis. *Neural Computation*, 22:793–830, March 2009.

[18] Cedric Fevotte, Remi Gribonval, and Emmanuel Vincent. BSS EVAL toolbox user guide – revision 2.0. *IRISA Technical Report 1706*, April 2005.

[19] Neville H. Fletcher and Thomas D. Rossing. *The Physics of Musical Instruments*. Springer, New York, 2010.

[20] Brendan J. Frey, Li Deng, Alex Acero, and Trausti Kristjansson. Algonquin: Iterating Laplace's method to remove multiple types of acoustic distortion for robust speech recognition. In *Proceeding of Eurospeech*, Aalborg, Denmark, September 2001.

[21] Mark J. F. Gales and Steve J. Young. Cepstral parameter compensation for HMM recognition in noise. *Speech Communication, Special issue on Speech Processing in Adverse Conditions*, 12(3), 1993.

[22] Joachim Ganseman, Paul Scheunders, Gautham J. Mysore, and Jonathan S. Abel. Source separation by score synthesis. In *Proceedings of the International Computer Music Conference (ICMC)*, New York, NY, June 2010.

[23] Zoubin Ghahramani and Michael Jordan. Factorial hidden Markov models. *Machine Learning*, 29, 1997.

[24] Daniel W. Griffin and Jae S. Lim. Signal estimation from modified short-time Fourier transform. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 32(2):236–243, April 1984.

[25] John R. Hershey, Trausti Kristjansson, Steven Rennie, and Peder A. Olsen. Single channel speech separation using factorial dynamics. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 593–600. MIT Press, Cambridge, MA, 2007.

[26] John R. Hershey, Steven J. Rennie, Peder A. Olsen, and Trausti T. Kristjansson. Super-human multi-talker speech recognition: A graphical modeling approach. *Computer Speech and Language*, 24(1):45–46, 2010.

[27] Guoning Hu and DeLiang Wang. Segregation of unvoiced speech from nonspeech interference. *Journal of the Acoustical Society of America*, 124(2):1306–1319, August 2008.

[28] Aapo Hyvärinen, Juha Karhunen, and Erkki Oja. *Independent Component Analysis*. Wiley, 2001.

[29] Shiro Ikeda and Noboru Murata. A method of ica in time-frequency domain. In *Proceedings of the International Conference on Independent Component Analysis and Signal Separation (ICA)*, 1999.

[30] Michael I. Jordan, Zoubin Ghahramani, Tommi S. Jaakkola, and Lawrence Saul. An introduction to variational methods for graphical models. *Machine Learning*, 37:183–233, 1999.

[31] Alexander Jourjine, Scott Rickard, and Ozgur Yilmaz. Blind separation of disjoint orthogonal signals: Demixing N sources from 2 mixtures. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Istanbul, Turkey, June 2000.

[32] Anssi Klapuri and Manuel Davy. *Signal Processing Methods for Music Transcription*. Springer-Verlag, New York, 2006.

[33] Trausti Kristjansson, John R. Hershey, Peder A. Olsen, and Ramesh Gopinath. Super-human multi-talker speech recognition: The IBM 2006 speech separation challenge system. In *Proceedings of the Ninth International Conference on Spoken Language Processing*, 2006.

[34] Jeffrey C. Lagarias, James A. Reeds, Margaret H. Wright, and Paul E. Wright. Convergence properties of the Nelder–Mead simplex method in low dimensions simplex method in low dimensions. *SIAM Journal of Optimization*, 9(1):112–147, 1998.

[35] Daniel D. Lee and H. Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401:788–791, October 1999.

[36] Daniel D. Lee and H. Sebastian Seung. Algorithms for non-negative matrix factorization. In *Advances in Neural Information Processing Systems (NIPS)*, volume 13, pages 556–562, 2001.

[37] Kyogu Lee and Malcolm Slaney. Acoustic chord transcription and key extraction from audio using key-dependent HMMs trained on synthesized audio. *IEEE Transactions on Audio, Speech and Language Processing*, 16(2), February 2008.

[38] Aaron S. Master. *Stereo Music Source Separation via Bayesian Modeling*. PhD thesis, Stanford University, June 2006.

[39] Dave Mellinger. *Event Formation and Separation in Musical Sound*. PhD thesis, Stanford University, 1991.

[40] Ryo Mukai, Hiroshi Sawada, Shoko Araki, and Shoji Makino. Blind source separation for moving speech signals using blockwise ICA and residual crosstalk subtraction. *IEICE Transactions Fundamentals*, E-87A(8):1941–1948, August 2004.

[41] Kevin Murphy. *Dynamic Bayesian Networks: Representation, Inference and Learning*. PhD thesis, UC Berkeley, Computer Science Division, July 2002.

[42] Radford M. Neal. Probabilistic inference using markov chain monte carlo methods. Technical Report CRG-TR-93-1, Dept. of Computer Science, University of Toronto, 1993.

[43] Alexey Ozerov and Cedric Fevotte. Multichannel nonnegative matrix factorization in convolutive mixtures for audio source separation. *IEEE Transactions on Acoustics, Speech, and Language Processing. Special issue on Signal Models and Representations of Musical and Environmental Sounds*, 18(3):550–563, March 2010.

[44] Alexey Ozerov, Cedric Fevotte, and Maurice Charbit. Factorial scaled hidden markov model for polyphonic audio representation and source separation. In

Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA), New Paltz, New York, October 2009.

[45] Thomas W. Parsons. Separation of speech from interfering speech by means of harmonic selection. *Journal of the Acoustical Society of America*, 60(4):911–918, 1976.

[46] Michael Syskind Pedersen, Jan Larsen, Ulrik Kjems, and Lucas C. Parra. A survey of convolutive blind source separation methods. In *Springer Handbook of Speech Processing*. Springer, 2007.

[47] Lawrence R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[48] Bhiksha Raj and Paris Smaragdis. Latent variable decomposition of spectrograms for single channel speaker separation. In *Proceedings of the IEEE Workshop of Applications of Signal Processing to Audio and Acoustics (WASPAA)*, New Paltz, New York, October 2005.

[49] Steven J. Rennie, John R. Hershey, and Peder A. Olsen. Single-channel speech separation and recognition using loopy belief propagation. In *Proceeding of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Tapei, Taiwan, April 2009.

[50] Steven J. Rennie, John R. Hershey, and Peder A. Olsen. Variational loopy belief propagation for multi-talker speech recognition. In *Proceedings of Interspeech*, Brighton, UK, 2009.

[51] Steven J. Rennie, Peder A. Olsen, John R. Hershey, and Trausti T. Kristjansson. The iroquois model: Using temporal dynamics to separate speakers. In *Proceedings of ISCA Tutorial and Research Workshop on Statistical and Perceptual Audition (SAPA)*, Pittsburgh, PA, September 2006.

[52] Jorma Rissanen. Modeling by shortest data description. *Automatica*, 14:465–471, 1978.

[53] Thomas D. Rossing, Richard F. Moore, and Paul A. Wheeler. *Science of Sound*. Addison Wesley, 2001.

[54] Sam T. Roweis. One microphone source separation. In *Advances in Neural Information Processing Systems (NIPS)*, volume 13, pages 793–799, December 2001.

[55] Sam T. Roweis. Factorial models and refiltering for speech separation and denoising. In *Proceeding of Eurospeech*, Geneva, Switzerland, September 2003.

[56] Hiroshi Saruwatari, Satoshi Kurita, and Kazuya Takeda. Blind source separation combining frequency-domain ICA and beamforming. In *Proceeding of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2001.

[57] Mikkel N. Schmidt, Ole Winther, and Lars Kai Hansen. Bayesian non-negative matrix factorization. In *Independent Component Analysis and Signal Separation (ICA)*, LNCS, pages 540–547, 2009.

[58] Gideon Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 6(2):461–464, 1978.

[59] Madhusudana Shashanka. *Latent Variable Framework for Modeling and Separating Single-Chanel Acoustic Sources*. PhD thesis, Boston University, 2007.

[60] Madhusudana Shashanka, Bhiksha Raj, and Paris Smaragdis. Probabilistic latent variable models as non-negative factorizations. *Special issue on Advances in Non-negative Matrix and Tensor Factorization, Computational Intelligence and Neuroscience Journal*, May 2008.

[61] Madhusudana Shashanka, Bhiksha Raj, and Paris Smaragdis. Sparse overcomplete latent variable decomposition of counts data. In J.C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 1313–1320. MIT Press, Cambridge, MA, 2008.

[62] Paris Smaragdis. Blind separation of convolved mixtures in the frequency domain. *Neurocomputing*, 22(1):21–34, 1998.

[63] Paris Smaragdis. *Redundancy reduction for computational audition*. PhD thesis, M.I.T., 2001.

[64] Paris Smaragdis. Discovering auditory objects through non-negativity constraints. In *ISCA Tutorial and Research Workshop on Statistical and Perceptual Audio Processing (SAPA)*, Jeju, Korea, October 2004.

[65] Paris Smaragdis and Judith C. Brown. Non-negative matrix factorization for polyphonic music transcription. In *Proceedings of the IEEE Workshop of Applications of Signal Processing to Audio and Acoustics (WASPAA)*, New Paltz, New York, October 2003.

[66] Paris Smaragdis and Gautham J. Mysore. Separation by "humming": User-guided sound extraction from monophonic mixtures. In *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, New Paltz, New York, October 2009.

[67] Paris Smaragdis, Bhiksha Raj, and Madhusudana Shashanka. Probabilistic latent variable model for acoustic modeling. In *Advances in models for acoustic processing workshop, Neural Information Processing Systems (NIPS)*, December 2006.

[68] Paris Smaragdis, Bhiksha Raj, and Madhusudana Shashanka. Supervised and semi-supervised separation of sounds from single-channel mixtures. In *Proceedings of the seventh International Conference on Independent Component Analysis and Signal Separation (ICA)*, London, UK, September 2007.

[69] Paris Smaragdis, Bhiksha Raj, and Madhusudana Shashanka. Missing data imputation for spectral audio signals. In *Proceeding of the IEEE International Workshop on Machine Learning for Signal Processing*, Grenoble, France, September 2009.

[70] Julius O. Smith. *Spectral Audio Signal Processing, March 2010 Draft*. http://ccrma.stanford.edu/~jos/sasp/, online book, 2010.

[71] Barry D. van Veen and Kevin M. Buckley. Beamforming: a versatile approach to spatial filtering. *IEEE ASSP Magazine*, 5(2):4–24, 1988.

[72] Emmanuel Vincent. Musical source separation using time-frequency source priors. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(1):91–98, 2006.

[73] Emmanuel Vincent, Cedric Fevotte, and Remi Gribonval. Performance measurement in blind audio source separation. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(4):1462–1469, 2006.

[74] Tuomas Virtanen. Speech recognition using factorial hidden Markov models for separation in the feature space. In *Proceedings of Interspeech*, Pittsburgh, PA, 2006.

[75] Tuomas Virtanen. Monaural sound source separation by nonnegative matrix factorization with temporal continuity and sparseness criteria. *IEEE Transactions on Acoustics, Speech, and Language Processing*, 15(3):1066–1074, 2007.

[76] Tuomas Virtanen, Ali Taylan Cemgil, and Simon Godsill. Bayesian extensions to non-negative matrix factorisation. In *Proceeding of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Las Vegas, Nevada, April 2008.

[77] Mitchel Weintraub. *A theory and computational model of auditory monoaural sound separation*. PhD thesis, Stanford University, 1985.

[78] Yair Weiss and William T. Freeman. On the optimality of solutions of the max-product belief propagation algorithm in arbitrary graphs. *IEEE Transactions on Information Theory*, 47(2):723–735, 2001.

[79] Ozgur Yilmaz and Scott Rickard. Blind separation of speech mixtures via time–frequency masking. *IEEE Transactions on Signal Processing*, 52(7):1830–1847, July 2004.