Statistical Graphical Models for Scene Analysis, Source Separation and Other Audio Applications

Manuel J. Reyes Gómez

Submitted in partial fulfillment of the

requirements for the degree

of Doctor of Philosophy

under the Executive Committee

of the Graduate School of Arts and Sciences

COLUMBIA UNIVERSITY

2005

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.

_____

Daniel P.W. Ellis, Principal Advisor
(Department of Electrical Engineering)

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.

_____

Shih-Fu Chang
(Department of Electrical Engineering)

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.

_____

Tony Jebara
(Department of Computer Science)

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.

_____

Bhiksha Raj
(Mitsubishi Electric Research Laboratories)

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.

_____

Nebojsa Jojic
(Microsoft Research)

Approved for the Department of Electrical Engineering:

_____

Tony Heinz
Chair

ABSTRACT

Statistical Graphical Models for Scene Analysis, Source Separation and Other Audio Applications

Manuel J. Reyes Gómez

The problem of separating overlapping sound sources has long been a research goal in sound processing, not least because of the apparent ease with which we as listeners achieve perceptual separation and isolation of sound sources in our everyday experiences.

Human listeners use their prior knowledge of all the sound classes that they have experienced through their lives to impose constraints on the form that elements on a mixture can take. Listeners use the information obtained from partial observation of the unmixed context to disambiguate the components where the energy is locally swamped by interfering sources.

Researchers working on this problem (Ellis 1996) argue that just as human listeners have top-down knowledge, prior constraints on the form that the mixture components can take is the critical component to making source separation systems work. In this thesis, we propose to encode these contraints in the form of models which capture the statistical distributions of the features of mixture components, using the framework of statistical graphical models, and then use those models to estimate obscured or corrupted

portions from partial observations. Our overarching goal is to explain composed data as a composition of the models of the individual sources.

After reviewing the basic statistical tools, this dissertation describes three models of this kind. The first uses multiple-microphone recordings from reverberant rooms combined in a filter-and-sum setup. The filter coefficients are optimized to match system output against a model of speech taken from a speech recognizer. The second model addresses the more difficult case of a single channel recording, and handles the tractability problems of the very large number of states required by decomposing the signal into subbands. The final model provides very precise fits to source signals without an enormous dictionary of prototypes, but instead by exploiting the observation that much of a real-world signal can be described as systematic local spectral deformations of adjacent time frames; by inferring these deformations between occasional spectral templates, the entire sound is accurately described. For this last model, we show in detail how a mixture of two sources can be segmented at points where local deformations do not provide adequate explanation, to delineate regions dominated by one source. Individual sources can then be reconstructed by interpolation of the deformation parameters to reconstruct estimates of the mixture components even when they are hidden behind high-energy maskers.

Although acoustic scene analysis and source separation are used as motivating and illustrative applications throught, the intrinsic descriptions of the nature of sound sources captured by these models could have other, broader applications in signal recognition, compression and modification, and even beyond audio in other domains where signal properties have the appropriate nontrivial local structure.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The current availability of large quantities of digital audio data has created the necessity to develop an efficient way to model the content of digital audio files for a wide range of tasks including event detection, segmentation, content description/recognition, and indexing and retrieval.

However, most audio recordings are composed of mixtures of several sources (i.e. lyrics plus instruments in music, foreground plus background in outdoor recordings, etc.) a situation that greatly complicates those tasks.

Two similar but by no means equivalent areas of research address this situation: Auditory Scene Analysis and Audio Source Separation. The former refers to identifying the different sources present in the mixture as they would be perceived by a listener. The latter consists, based on a more objective definition, of separating into different audio streams the individual sources present in the mixture. Take for instance the example portrayed in figure 1.1, where the signal of interest consists of a mixture of a female voice and a male voice with traffic noise in the background. An auditory scene analysis system should be able to identify the different objects in the scene: a "woman's voice", a "man's voice" and "traffic" in the background; the analysis could even take a step forward and assign a semantic or subjective meaning to the mixture, such as "a man and

a woman chatting in the street". On the other hand, instead of an abstract description, an audio source separation system would attempt to produce three different streams of audio data corresponding to the three different sources present in the mixture. This is known as blind source separation.

The auditory scene analysis and source separation terms are often used interchangably in the literature since frequently techniques that are used for auditory scene analysis can be used for blind source separation as well and viceversa.

As humans, researchers working on source separation of audio mixtures have been particulary interested in the class of audio signals with which they are the most familiar, namely human speech. Much work has focused on developing systems capable of separating mixtures of speech from different speakers into streams containing the speech corresponding to the individual speakers. This problem is referred to in the community as the "cocktail party" problem since in resembles the problem encountered during a social event of extracting the single voice of interest from the composition of chatter and other noises.

In this thesis we place an emphasis in source separation applications such as the cocktail party problem, however the models to be presented can also be applied to other kinds of mixtures and for scene analysis applications.

The problem of separating overlapping sound sources has been a research goal in sound processing for quite some time. Previous approaches can be roughly separated into three categories: (Multimicrophone) Blind Source Separation (BSS), Computational Auditory Scene Analysis (CASA) and Time-Frequency Masking. The latter two approaches utilize, at some point during the separation process, the time-frequency representation of audio signals. A graphical display of such representation is known as the spectrogram.

Figure 1.2 shows a spectrogram representation of a speech signal, where each column depicts the energy content across frequency in a short-time window, or time-frame. The value in each cell is actually the log-magnitude of the short-time Fourier

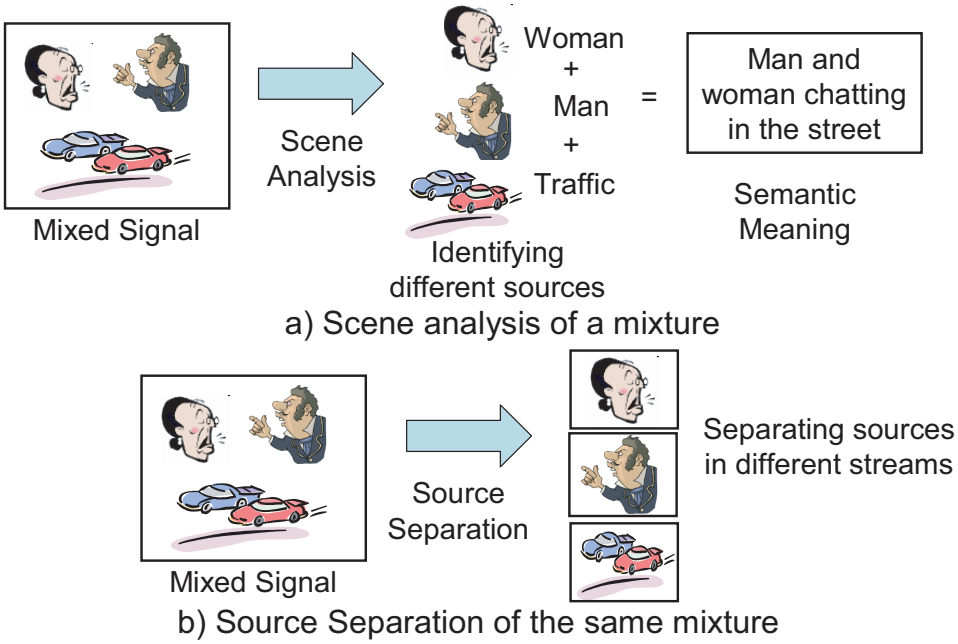

Figure 1.1: a) Scene analysis and b) source separation of the same mixed signal.

transform in decibels:

$$x_t^k = 20 \log \left(abs \left(\sum_{\tau=0}^{N_F-1} w[\tau] x[t \cdot H + \tau] e^{-j2\pi\tau k/N_F} \right) \right) \tag{1.1}$$

where t is the time-frame index, k indexes the frequency bands, N_F is the size of the discrete Fourier transform, H is the hop between successive time-frames, $w[\tau]$ is the N_F-point short-time window, and $x[\tau]$ is the original time-domain signal. The spectrogram usually only includes k up to $N_F/2 + 1$ since the remaining bins are conjugate repetitions.

We now provide a little more detail on prior approaches to source separation in order to explain how our work differs.

1.1 Blind Source Separation

Conventional blind source separation (BSS) requires, in most cases, the use of signals recorded using multiple microphones. The algorithms involved typically require at least as many microphones as the number of signal sources.

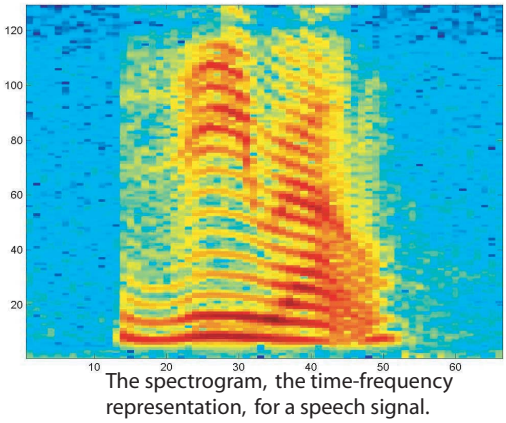

Figure 1.2: Time-frequency representation of a speech signal.

Blind Source Separation does not utilize any knowledge of the statistical characteristics of the signals to be separated, relying instead on general properties between the various signals to separate them.

Most Blind Source Separation approaches rely on the theoretical framework of Independent Component Analysis (ICA).

In this approach, no *a priori* knowledge of the signals is assumed. Instead, the component signals are estimated as a weighted combination of current and past samples from the multiple recordings of the mixed signals. The weights are estimated to optimize an objective function that measures the independence of the estimated component signals(Hyvärinen 1999).

Even though these techniques perform well for certain signal mixtures, they fail in many situations, such as when the signals are recorded in a reverberant environment, since the algorithms do not aim to dereverberate (deconvolve) the data but simply identify the sources in the scene. This arises given that the goal of the objective function is to find independent sources rather than to dereverberate the actual signals.

Convolutive ICA also performs poorly when the degree of overlap and/or the dimensionality of the recordings makes the blind inference problem intractable: if the mix-

ture components really could be anything, we have no way of getting good estimates for missing parts of the signal where the energy is locally swamped by interfering sources.

Greater detail of conventional ICA is presented in chapter 8.2.

1.2 Computational Auditory Scene Analysis (CASA)

Analyzing, modelling and finally emulating the process by which people perceive, process and convert continuous sound into distinct, interpreted abstractions using a computer is known as "Computational Auditory Scene Analysis" (CASA). The title acknowledges that the work is founded on experimental and theoretical results derived from *Psychoacoustics*, such as the ones described in Bregman's book *Auditory Scene Analysis* (Bregman 1990).

CASA systems can be classified as *data-driven* systems, also regarded as *bottom-up* systems, and as *prediction driven* systems also, regarded as top-down systems.

1.2.1 Data-Driven Systems

Several systems in this category have been focused on the problem of separating speech from interfering noise, either unwanted speech (Weintraub 1985) or more general interference (Cooke 1991)(Brown 1992)(Wang and Brown 1999). Given that data-driven systems rely solely on locally-derived features present in the input data to decompose the input sound into sensory elements, patches of time-frequency with consistent characteristics. The systems employ either computer vision techniques or complete ear models, such as a chochlea model, to segment the auditory scene into several audio elements. The systems later group those segments that are likely to have originated from the same source. The regrouping results in a time-frequency mask indicating the energy of the target based on their common underlying periodicity. The desired source is later resynthesized by filtering the original mixture according to the mask.

1.2.2 Prediction-Driven Systems

Even though *data-driven* systems can characterize several kinds of auditory phenomena they do not deal well with perceptual phenomena that involve the use of the auditory context sorrounding a local event, given that data-driven system rely solely in locally derived features.

Examples of such phenomena are the *continuity illusion* discussed in (Bregman 1990) and the *phonemic restoration* phenomenon first noted in (Warren 1970). The *continuity illusion* consists of presenting a subject with a short sequences of 130 ms of sine tone alternating with 130 ms of noise centered on the same frequency. When the energy of the noise is low relative to the energy of the tone, the signal is perceived by the subject as an alternation of the two. But if the energy levels are changed to be similar, the perception changes to a steady continuous sine tone to which short noise bursts have been *added*, rather than hearing the noise burst as *replacing* the sine tone. In the *phonemic restoration* phenomenon, a small piece of a speech recording is removed and replaced by a noisy masking signal. The listeners perceived the speech as complete without precisely locating where the noisy signal ocurred, therefore the listeners does not know which phoneme their auditory systems has inferred. Moreover, the content of the speech that is "filled in" by the auditory system depends on what would make sense in the sentence. If the deletion is left as silence, no restoration ocurrs.

These phenomena present evidence that the auditory system effectively takes account of contextual factors. As long as the local evidence (energy) does not contradict the context, the auditory system "fills in" the local information to achieve a perception that better accommodates the context. However when the local evidence does contradict the context, the auditory system disregards the context in favor of the distinct local data.

Prediction-Driven CASA systems extend *Data-Driven* systems to acommodate the influence of the context on auditory perception (Ellis 1996). They do so by including representations of generic sound elements in an internal world model, such that the

internal world model is used to predict the observed cues expected in the next time slice based on the current state of the model. This is then compared with the actual information arriving from the front end; these two are reconciled by modifying the internal state, and the process continues. Such systems have been used to separate objects in natural scenes such as a "construction site" and to separate speech from noisy backgrounds.

1.3 Time-Frequency Masking

The time-frequency representation of speech signals is very sparse: since most narrow frequency bands carry substantial energy only during a small fraction of time and therefore is rare to encounter two independent sources with large amounts of energy at the same frequency band at the same time.

The time-frequency masking approach exploits this characteristic of the time-frequency representation by assigning each time-frequency bin x_t^k (eq. 1.1) from the mixture signal to one and only one of the sources. Each source in the mixture has a correspondent binary mask with the same dimensions as the time-frequency representation of the mixed signal, where *"one"* in a given source mask signals that the correspondent time-frequency bin in the spectrogram of the mixture signal corresponds to the the correspondent speaker. The individual sources are later resynthesized using the masks and the spectrogram and phase information of the original mixed signal.

Most systems developed untill this date using this approach involve only one microphone and they are reviewed in section 4.1. A notable exception is the DUET algorithm (Yilmaz and Rickard 2004), which separates an arbitrary number of sources using two *anechoic* mixtures.

1.4 Model Based Source Separation

Human listeners use their prior knowledge of all the sounds classes that they have experienced through their lives to impose constraints on the form that mixture components can take. Listeners use the information obtained from partial observation of the unmixed context to disambiguate the components where the energy is locally swamped by interfering sources, just as the phenomena described in section 1.2.2 exemplify.

Researchers working on *Prediction-Driven CASA* systems, (Ellis 1996) argue that just as human listeners have top-down knowledge, prior constraints on the form that the mixture components can take is the critical component to making source separation systems work. Unfortunally given the *psychoacoustics* nature of CASA systems is difficult to incorporate large amounts of detailed stastical knowledge about the expected signals in such approach. Moreover, *CASA* precepts do not really contemplate a direct way to explain local composed data by a direct composition of the individual source models.

Therefore, we propose to encode these contraints in the form of models which capture the statistical distributions of the features of mixture components, using the framework of statistical graphical models and then use those models to estimate obscured or corrupted portions from partial observations, and moreover, being able to explain composed data as a composition of the models of the individual sources.

Model-based source separation approaches have indeed been developed lately, such as in (Roweis 2000)(Hershey and Casey 2001), where hidden Markov models (HMMs) are used to encode the statistical contraints of the sources and in (Kristjansson et al. 2004), where Gaussian Mixture models, which can be seen as an special case of HMMs, are used in a more simple source separation task. These three approaches utilize only one microphone and they are discussed in more detail in section 4.1.

It is not surprising that hidden Markov models have been the first choice for most existing model-based source separation approaches, since they have been extensively used to model audio data, in particular speech. However, their use in speech recognizers

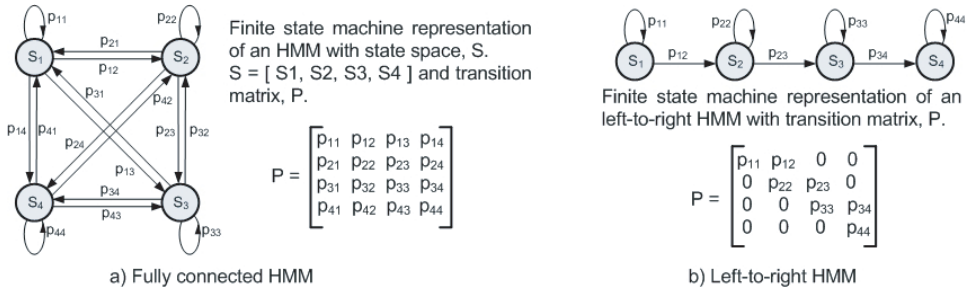

Figure 1.3: a) Finite machine state representation of a fully connected HMM. b) Finite machine state representation of a left-to-right HMM.

does not constitute a full generative model, since it is required only to discriminate between candidate words. When HMMs are used in source separation, several limitations become apparent. We quickly review these models, explaining their limitations, when used for source separation applications.

1.5 Hidden Markov Models

Audio is most commonly modeled using the standard single chain hidden Markov model (HMM), which comprises a set of states associated with representative signal feature distributions – a kind of signal codebook – and a state transition matrix describing the dynamics of the signal in terms of the probability that a certain state at a given time will be followed by a particular subsequent state.

HMMs offer a natural and flexible way to model time-sequential data because via self-loops, they easily accommodate time-warping signal variations.

They can be interpreted as a finite state machine, where each entry in the HMM's codebooks represents a state and the dynamics between states are defined by the transition matrix (figure 1.3 a)

They have been widely and successfully used in speech recognition applications where they are aimed to recover the discriminative features between phonemes in speech, information that can be covered by a reasonably-sized codebook. Given the specific

structure of speech, HMMs for speech recognition are constrained to be left-to-right chains, meaning that at any given time frame the model can only stay in the present state or transition to the following one, and the model can't return to previously visited states. The representation of a left-to-right HMM as a finite state machine is depicted in figure 1.3 b. However, HMMs are not a practical generative model of audio signals: To be able to generate a typical real-world signal at a perceptually-acceptable quality level would usually require an HMM with an impractically large number of states.

When multiple observations of the mixed signals are available, as when using a microphone array, speech-recognition-like HMMs, which represent a coarse representation of the signal, can be used to guide the separation of the signals, but the actual separation is done through another part of the source separation system, i.e. a set of filters,(Reyes-Gomez et al. 2003b),(Reyes-Gomez et al. 2003a). This approach is described in detail in chapter 8.2.

However when only one microphone is available, the models are more directly involved in the separation, and must therefore provide a greater level of detail. In the case of HMMs, this translates to hundreds or even thousands of fully connected states. Such a large number of parameters presents many challenges during both learning and inference.

1.6 Thesis Contributions

In this thesis, we explore different scenarios for the model-based source separation scheme using a variety of statistical models. Some of them are well known, such as hidden Markov models. Some others are brand new and first introduced in this thesis.

Our contributions can be divided in two kind of scenarios: the meeting recordings scenario and the single-channel recordings scenario.

1.6.1 The Meeting Recordings Scenario

This kind of the recordings are of the kind that are obtained when the audio of a typical business meeting is recorded. Meeting rooms are generally just big enough to enclose a table sorrounded by chairs. Therefore, this kind of scenario is highly reverberant.

Multi-microphone approaches fit well in meeting scenarios where the dimensions of the room are known and there is a limited number of possible positions for the speakers. This permits an optimal set up for the microphone array such that a good coverage of all the speakers could take place regardless of their actual positions. Also, since meeting scenarios are very reverberant by nature, they demand the use of multiple different observations to better cope with the situation.

We propose to adapt the conventional blind source separation multimicrophone setup to accommodate the use of statistical models. We call our approach the maximum likelihood filter-and-sum system.

1.6.1.1 Maximum Likelihood Filter-and-Sum System

We treat the signal separation problem as one of beam-forming, where each signal is extracted using a filter-and-sum array. The filters are estimated to maximize the likelihood of the summed output, measured on the statistical model for the desired signal.

The statistical models used in this approach were very coarse hidden Markov models of the source's speech, which "guide" the filters to find their optimal set of coefficients to achieve the separation. The actual separation is done by the filters and not by the models themselves. The sources' HMMs are derived from a left-to-right HMM-based speech recognizer.

Remarkable results are obtained using this approach for this scenario, where traditional ICA-based approaches fail. This is because the models were obtained from clean speech (during the training of the ASR system). Therefore, the system, unlike ICA-systems, dereverberates the source signals in addition to separating them.

1.6.1.2 The Single-Channel Recordings Scenario

The requirement for an array of microphones, makes the application of multimicrophone approaches impractical in many scenarios. Also, many commercial audio signals such as soundtracks and music are available only as single-channel signals. Therefore, there is the need to develop systems that can perform audio source separation from a single recording of a mixed signal.

Most commercial single-channel audio signals are recorded in anechoic environments. Therefore, reverberance is not a salient feature in this scenario.

From the model based source separation perspective the single source restriction imposes more demands on the audio model to be used, requiring models that represent the single sources with greater detail. The model has a greater role in the separation process itself, unlike in the multimicrophone case, where the actual separation is performed by the filters.

When the complexity and variability of the sounds are high, as in a particular speaker's voice, a model that aims to capture every single possible distinct sound might require millions of parameters to cover the full range of possibilities.

In this thesis we propose models that factor the large parameter space required in detailed audio. We introduced two new graphical models: The multiband and the deformable spectrograms models.

1.6.1.3 The Multiband Model

Representing every single instance of a particular complex sound is equivalent to representing every single possible column on the spectrogram representation (figure 1.2) of the audio class. This is the approach followed in (Roweis 2000) when building detailed models using HMMs.

Rather than using a monolithic state to represent the spectrum, We propose to divide the spectral representation into multiple frequency *bands* i.e. multiple parallel

horizontal sections of the spectrogram, as shown in figure 5.1 b, and then use separate HMMs in each band with many fewer states. Factorizing the complete spectrogram in this way, we could represent a large number of full spectral configurations with substantially fewer parameters making inference and learning more feasible. This model is described in chapter 5.

Even factorizing the spectrogram in this way, each frame in the spectrogram 1.2 is treated as an independent identity. However, speech and other natural sounds show high temporal correlation and smooth spectral evolution punctuated by a few, irregular and abrupt changes. Therefore, it would be more efficient and informative to model successive spectra as *transformations* of their immediate predecessors.

1.6.1.4 The Deformable Spectrograms Model

This model (described in chapter 6) focuses on local deformations of adjacent bins in a time-frequency surface to explain an observed sound, using explicit representation only for those bins that cannot be predicted from their context.

The model is used to segment mixtures of speech into dominant speaker regions on a unsupervised source separation task. Identifying and modeling the dynamics of the speech on regions where a given speaker is dominant are later used to ”filled in” the information masked out by the interference of another speaker.

We also present results on a speech recognition task that suggest that the model discovers a global structure on the dynamics of the signal’s energy that helps to alleviate the problems generated by noise interference.

1.6.2 Graphical Models Framework for Audio Modeling

Audio is most commonly modeled using hidden Markov models, but other instances of statistical models have not been so widely explore.

HMMs are particular instances of the graphical models framework, an intuitive and modular way to model complex systems as a graphical structure of simpler parts.

In this thesis we extend the representative power of HMMs through the theoretical framework of graphical models in order to develop richer models better suited to applications where regular HMMs perform poorly or are infeasible. At the same time, we aim to maintain the flexibility of regular HMMs for modeling time sequential data, and the tractability of their inference procedures.

Therefore, this thesis contributes in showing the applicability of the graphical models framework, a topic typically associated with pure machine learning, into audio modeling. This framework enables us to develop richer models more suitable to the applications at hand.

1.7 Source Separation Applications

There are several applications in which an automatic source separation system can be used.

1. **Automatic Meeting Transcriptions/Indexing**: Effectively separating the different speakers in meeting recording and subsequently applying a speech recognizer to each individual audio stream will result on an automatic transcription of the meeting content; The high incidence of speaker overlap in meetings is a major barrier at present (Morgan et al. 2001). Once the meetings are complete transcribed, they could be indexed by topic by processing the resulting text.

2. **Reducing Speech Interference in Hearing Aids**: Even with the restoration of lost sensitivity through amplification and dynamic range compression, hearing impaired individuals still have difficulty understanding mixtures of voices (Kollmeier et al. 1993). A portable system that can separate mixtures of voices could be built into the hearing aid to help the user in such situations.

3. **Film Soundtracks Indexing, Editing and Remixing**: Being able to automatically separate and classify the element of a movie soundtrack through an automatic sound-processing system is required for useful automatic content-based indexing of this kind of data. Once the different elements on the scene have been separated, the soundtrack could be edited by removing unwanted objects. Moreover the different streams of audio could be recombined to create new pieces from the original soundtrack.

4. **Personal Audio Editing**: Being able to separate different audio sources form the audio of our personal video recordings would permit us to edit them to eliminate the unwanted interferences, like the kid crying next to us in our trip to the zoo or the traffic noise outside of the wedding ceremony of our best friend.

5. **Surveillance Applications**: Being able to discriminate between objects in an audio scene will greatly improve the performance of surveillance applications. Moreover, being able to identify and trancribed mixtures of speech on surveillance tapes could help to identify potential dangerous situations.

1.8 Dissertation Organization

This chapter has introduced the field of audio source modeling, and the application area of source separation. The contributions of this work in terms of several new signal models, implemented in systems for source separation, have been introduced and are described in more detail in chapters 8.2, 5, 6, and 7. Further background for the later models in presented in chapter 4 which looks at the specific issues in single-channel source separation. But first, in chapter 2, we present the statistical graphical models framework including an introduction to the approximate inference procedures utilized in this thesis.

Chapter 2

Statistical Graphical Models

This chapter provides a background introduction to the statistical graphical models framework to provide the foundation required for the later chapters.

The statistical graphical models framework is an intuitive and modular way to model complex systems as a graphical structure of simpler parts. The observed variables of the system as well as the unknown or hidden variables are represented using nodes. Observed variables which have known fixed values are represented by shaded nodes, while hidden variables, modeled as random variables, are illustrated by unshaded nodes. Sets of variables that have direct interaction with each other are connected through edges, forming a graphical representation of the system. Probability theory permits us to investigate or to query the state of the unknown variables given the observed variables, a process known as inference. Graphical models are divided into two major classes: directed and undirected graphical models.

2.1 Directed Graphical Models

In directed graphical models, the edges between variables have a notion of causality and therefore are represented by edges with directions or arrows. The set of nodes (X_{π_i}) that

have arrows pointing into node X_i are referred as the parents of X_i.

In directed graphs, the causal relationship between a node and its parents is defined by conditional probabilities $p(X_i \mid X_{\pi_i})$. The joint probability $p(X_1, X_2, X_3, ..., X_n)$ between all variables (hidden and observed) in the system is defined as:

$$p(X_1, X_2, X_3, ..., X_n) = \prod_{i=1}^{n} p(X_i \mid X_{\pi_i}). \tag{2.1}$$

A directed graphical model widely used to model speech and audio is the hidden Markov model (HMM), (figure 2.1a). There, hidden nodes X = [X_0,X_1,..,X_T] represent the acoustic class at frame t, while the observed variables Y = [Y_0,Y_1,...,Y_T] represent features of the audio signal.

The conditional probabilities in this model are defined by p($X_{t+1} \mid X_t$) and p($Y_t \mid X_t$). The conditional probabilities p($X_{t+1} = j \mid X_t = i$) = $a_{(i,j)}$ =$a_{(X_t, X_{t+1})}$, are represented as entries on an $N \times N$ transition probabilities matrix A, where N is the number of different acoustic classes that X_t can take. The local likelihood conditional probabilities p($Y_t \mid X_t$) are frequently modeled with a Gaussian distribution (or a mixture of Gaussians) such that p($Y_t \mid X_t = i$) = $\mathcal{N}(Y_t; \mu_i, \Sigma_i)$. Then the N acoustic classes are defined by N sets of parameters $\theta_i = (\mu_i, \Sigma_i)$. The parameters θ that define an HMM are $\theta = \{A, \mu_1, \Sigma_1, \mu_2, \Sigma_2, ..., \mu_N, \Sigma_N\}$.

Therefore, the HMM's joint probability is given by:

$$p(X, Y \mid \theta_t) = \prod_{t=0}^{T-1} p(X_{t+1} \mid X_t) \prod_{t=0}^{T} p(Y_t \mid X_t) \tag{2.2}$$

For a given model with parameters θ and observations Y, we would like to find the best set of parameters that maximize the likelihood of the observations given the model, a process known as "maximum likelihood parameters estimation". The expectation-maximization (EM) algorithm provides a general approach to the problem of maximum

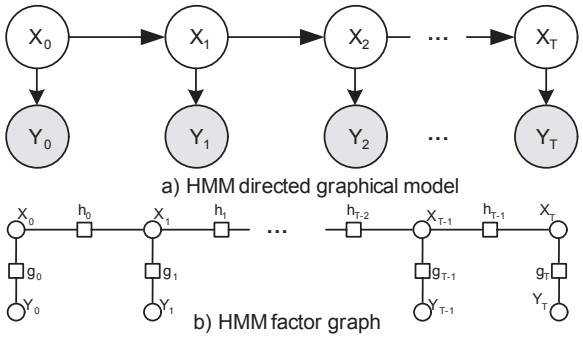

Figure 2.1: a) An HMM as a directed graphical model, b) HMM factor graph representation, where $g_t = p(Y_t \mid X_t)$ and $g_t = p(X_{t+1} \mid X_t)$

likelihood parameter estimation in statistical graphical models. In this approach the log-likelihood of the model p$(Y \mid \theta)$ is lower bounded by a auxiliary function, $\mathcal{L}(q, \theta)$, defined as:

$$\log p(Y \mid \theta) \geq \mathcal{L}(q, \theta) = \sum_X q(X \mid Y) \log \frac{p(X, Y \mid \theta)}{q(X \mid Y)} \tag{2.3}$$

where the term $q(X \mid Y)$ is regarded as the averaging function approximating the posterior (Jordan and Bishop 2004)(see below). The EM algorithm is essentially a coordinate ascent algorithm on the auxiliary function $\mathcal{L}(q, \theta)$. In the $t + 1^{th}$ iteration, q_{t+1} is found as the choice of q that maximizes $\mathcal{L}(q, \theta_t)$ given the current set of parameters θ_t. Then q_{t+1} is used to maximize $\mathcal{L}(q_{t+1}, \theta)$ with respect to θ to find θ_{t+1}. Further iterations of the algorithm are made to find q_{t+2}, θ_{t+2}, etc.

The above steps give the algorithm its name since they are regarded as:

$$\textbf{E}\text{xpectation Step} : q_{t+1} = \underset{q}{\operatorname{argmax}} \ (\mathcal{L}(q, \theta_t)) \tag{2.4}$$

$$\textbf{M}\text{aximization Step} : \theta_{t+1} = \underset{\theta}{\operatorname{argmax}} \ (\mathcal{L}(q_{t+1}, \theta)) \tag{2.5}$$

Chosing $q(X \mid Y)$ to be $p(X \mid Y)$ yields equation 2.3 to be:

$$\begin{aligned}\mathcal{L}(q,\theta)_{q=p(X|Y)} &= \sum_X p(X \mid Y) \log \frac{p(X,Y \mid \theta)}{p(X \mid Y)} \\ &= \sum_X p(X \mid Y) \log p(Y \mid \theta) = \log p(Y \mid \theta)\end{aligned} \tag{2.6}$$

Then, equation 2.4 is maximized by $q = p(X \mid Y, \theta_t)$, the posterior probability of the hidden variables given the observations and the latest estimated parameters. Therefore the E step is effectively done by estimating $p(X \mid Y, \theta_t)$. The M step is done by taking the derivatives of $\mathcal{L}(q_{t+1}, \theta)$ with respect to of each one of the parameters in θ and solving the equations. The E step is also referred to as the inference procedure since it consists of inferring the state of the model's hidden variables, by computing their joint probability given the observed variables and the model parameters. The M step is also referred to as "learning" since it is the process of estimating the best set of parameters for the model given the observations and current inferences of hidden variables.

The posterior probability of the hidden variables given the observation can be found using Bayes theorem:

$$p(X \mid Y, \theta_t) = \frac{p(X,Y \mid \theta_t)}{p(Y \mid \theta_t)} \tag{2.7}$$

Since HMMs will be a recurrent submodule in the models introduced in this thesis, we further discuss the particularities of the EM algorithm for these kind of models.

For the case of an HMM, the numerator of equation 2.7 is defined by equation 2.2. The overall likelihood of the model, $P(Y \mid \theta)$, can be obtained by marginalizing eqn. 2.2 with respect to X, resulting in:

$$p(Y \mid \theta) = \sum_X p(X,Y \mid \theta) = \sum_{X_0} \sum_{X_1} \cdots \sum_{X_T} \prod_{t=0}^{T-1} p(X_{t+1} \mid X_t) \prod_{t=0}^{T} p(Y_t \mid X_t) \tag{2.8}$$

At first sight it seems that we need to perform N^T summations since we have T variables X_t with N values each. However, the factorized form of the joint probability distribution

(eqn. 2.2) permits us to organize the summations by moving the relevant factors inside as shown in equation 2.9, reducing the total number of summations needed and revealing useful recursions.

$$p(Y) = \sum_{X_T} \cdots \sum_{X_1} p(X_2 \mid X_1) p(Y_1 \mid X_1) \sum_{X_0} p(X_1 \mid X_0) p(Y_0 \mid X_0) \tag{2.9}$$

The sum-product algorithm (described later) systematically exploits the factorization of the joint probability distribution to perform exact inference in complex graphical models. Appendix A shows the steps of the sum-product algorithm reqired to perform exact inference on HMMs.

The auxiliary function $\mathcal{L}(q, \theta)$ for an HMM has the form:

$$\begin{aligned}
\mathcal{L}(q,\theta) &= \sum_X q(X \mid Y) \log \frac{\prod_{t=0}^{T-1} p(X_{t+1} \mid X_t) \prod_{t=0}^{T} p(Y_t \mid X_t)}{q(X \mid Y)} \\
&= \sum_{t=0}^{T-1} \sum_X q(X \mid Y) \log p(X_{t+1} \mid X_t) + \sum_{t=0}^{T} \sum_X q(X \mid Y) \log p(Y_t \mid X_t) \\
&\qquad - \sum_X q(X \mid Y) \log q(X \mid Y) \\
&= \sum_{t=0}^{T-1} \sum_{X_t, X_{t+1}} q(X_t, X_{t+1} \mid Y) \log p(X_{t+1} \mid X_t) + \sum_{t=0}^{T} \sum_{X_t} q(X_t \mid Y_t) \log p(Y_t \mid X_t) \\
&\qquad - \sum_X q(X \mid Y) \log q(X \mid Y)
\end{aligned} \tag{2.10}$$

where in the last equality some elements of $X = [X_0, X_1, .., X_t, X_{t+1}, ..., X_T]$, have been marginalized out. It is important to recognize the form of the function described in equation 2.10, since it will appear later while analyzing the models introduced in this thesis.

The M-step is calculated by expressing equation 2.10 in terms of the parameters of the model and taking derivatives with respect to each one of the parameters.

$$\mathcal{L}(q,\theta) = \sum_{t=0}^{T-1}\sum_{i,j} q(X_t = i, X_{t+1} = j \mid Y) \log p(X_{t+1} = j \mid X_t = i) +$$
$$\sum_{t=0}^{T}\sum_{i} q(X_t = i \mid Y) \log p(Y_t \mid X_t = i) - \sum_{X} q(X \mid Y) \log q(X \mid Y)$$
$$= \sum_{t=0}^{T-1}\sum_{i,j} q(X_t = i, X_{t+1} = j \mid Y) \log a_{i,j} - \frac{D}{2}(T+1)\log 2\pi \tag{2.11}$$
$$-\frac{1}{2}\sum_{t=0}^{T}\sum_{i} q(X_t = i \mid Y)(\log \mid \Sigma_i \mid + (Y_t - \mu_i)'\Sigma_i^{-1}(Y_t - \mu_i))$$
$$+\lambda(\sum_{j} a_{i,j} - 1) - \sum_{X} q(X \mid Y) \log q(X \mid Y)$$

The term with the Lagrange multiplier λ is added to enforce the constraint that the rows of A sum to one. Taking the derivative of equation 2.11 with respect to parameters $a_{i,j}$, Σ_i and μ_i and solving the equations, the following update formulas are obtained:

$$a_{i,j} = \frac{\sum_{t=0}^{T-1} q(X_t = i, X_{t+1} = j \mid Y)}{\sum_{t=0}^{T} q(X_t = i \mid Y)} \tag{2.12}$$

$$\mu_i = \frac{\sum_{t=0}^{T} q(X_t = i \mid Y) Y_t}{\sum_{t=0}^{T} q(X_t = i \mid Y)} \tag{2.13}$$

$$\Sigma_i = \frac{\sum_{t=0}^{T} q(X_t = i \mid Y)(Y_t - \mu_i)(Y_t - \mu_i)'}{\sum_{t=0}^{T} q(X_t = i \mid Y)} \tag{2.14}$$

Therefore, during the E-step, it is not necessary to compute the joint posterior of all X_i random variables, i.e. $q(X \mid Y) = p(X \mid Y)$ for X = [X_0,X_1,..,X_t,X_{t+1},...,X_T], it is sufficient to compute $q(X_t, X_{t+1} \mid Y) = p(X_t, X_{t+1} \mid Y)$ and $q(X_t \mid Y) = p(X_t \mid Y)$ for all the correspondent values of t.

These posterior probabilities are efficiently calculated through a couple of equations known as the forward ($\alpha(X_t)$), backward ($\beta(X_t)$) recursions (Jordan and Bishop 2004), such that:

$$p(X_t \mid Y) = \frac{\alpha(X_t)\beta(X_t)}{\sum_{X_t} \alpha(X_t)\beta(X_t)} \tag{2.15}$$

$$p(X_t, X_{t+1} \mid Y) = \frac{\alpha(X_t)p(Y_{t+1} \mid X_{t+1})\beta(X_t)p(x_{t+1} \mid X_t)}{\sum_{X_t} \alpha(X_t)\beta(X_t)} \tag{2.16}$$

With:

$$\alpha(X_t) = \sum_{X_{t-1}} (\alpha(X_{t-1})p(X_t \mid X_{t-1}))p(Y_t \mid X_t) \tag{2.17}$$

$$\alpha(X_0) = p(X_0)p(Y_0 \mid X_0) \tag{2.18}$$

$$\beta(X_t) = \sum_{X_{t+1}} (\beta(X_{t+1})p(X_{t+1} \mid X_t)p(Y_{t+1} \mid X_{t+1})) \tag{2.19}$$

$$\beta(X_{T+1}) = 1 \tag{2.20}$$

2.2 Undirected Graphical Models

Undirected graphical models, also known as Markov random fields (MRFs) lack the notion of causality that the directed models have, eliminating the use of the directions on the edges. They are used in systems where local constraints between connected nodes can

be expressed, but where it is hard to ensure that the conditional probabilities at different nodes are consistent with each other. Local parameterization was done in directed graphs through the use of conditional probabilities; in undirected graphs such parameterization is done through the use of *potential functions*, which are structured to favor certain local configurations of variables by assigning them a larger value. They are assumed to be strictly positive, real-valued functions, but are otherwise arbitrary. In general, potential functions are neither conditional probabilities nor marginal probabilities and in this sense they do not have a local probabilistic interpretation. The product of the potential functions is, however, still required to represent the joint distribution of all the variables, hidden and observed, in the graphical model:

$$p(X) = \frac{1}{Z} \prod_{S} \psi_{X_S}(X_S) \tag{2.21}$$

where ψ_{X_S} represents the potential function defined on the subset of variables X_S, and Z is a normalization constant. Eqn. 2.21 permits the likelihood of the model to be factorized in simpler terms allowing a tractable way to perform the inference of the model. Exact inference, however, is not always possible for either type of model. This can occur when the conditional distributions or the potential functions involve a large number of variables, reducing the model factorization capabilities, or if the model has some specific topological characteristic that will be discussed later. Exact inference, when possible, for both type of models can be achieved through the use of several similar algorithms: the junction tree algorithm, Pearl's propagation algorithm, and the sum-product algorithm. In this chapter we present the sum-product algorithm since it is easily extended to perform approximate inference on intractable models. The sum-product algorithm is defined in terms of the *factor graph* representation of a graphical model.

2.3 Factor Graphs

Factor graphs have been explicitly designed to work with algorithms that exploit the factorization of a complex function p with domain X into simpler functions ψ_{X_S} defined over subsets X_S of set X, just as in eqn. 2.21, or in eqn. 2.1 with $\psi_{X_S} = p(X_i \mid X_{\pi_i})$ and $X_S = \{X_i, X_{\pi_i}\}$. **Definition:** A factor graph is a bipartite graph that expresses the structure of a factorization such as eqn. 2.21. A factor graph has a variable node for each variable X_i, a factor node for each local function ψ_{X_S}, and an edge connecting variable node X_i to factor node ψ_{X_S} if only and only if X_i is an argument of ψ_{X_S}, i.e. $X_i \in X_S$(F. Kschischang and Loeliger 2001). Variable nodes are represented with circles, while function nodes are represented with squares. Figure 2.1b shows the factor graph for an HMM. In the figure the notation of the function nodes is equivalent to: $g_k = p(Y_k \mid X_k)$ and $h_k = p(X_{k+1} \mid X_k)$.

2.4 Sum-Product Algorithm

Coming back to the likelihood $P(Y \mid \theta)$ of an HMM (eqn. 2.9), notice that the right-most summation, $\sum_{X_0} p(X_1 \mid X_0)p(Y_0 \mid X_0)$ can be seen as a function $f(X_1)$ of variable X_1. The second-rightmost summation can be expressed as: $\sum_{X_1} p(X_2 \mid X_1)p(Y_1, X_1)f(X_1)$, which in turn is a function $f(X_2)$ of variable X_2 and so on. Each one of the summations is marginalizing one of the variables in the model. The sum-product algorithm is an efficient procedure for computing marginal functions that exploits the factorization of the global function, using the distributive law to simplify the summations and reuse intermediate partial sums. The "flow" of intermediate products and summations used by the algorithm is conceptualized as a set of messages between the nodes of the factor graph representation of the model. The update rules for those messages are defined as:

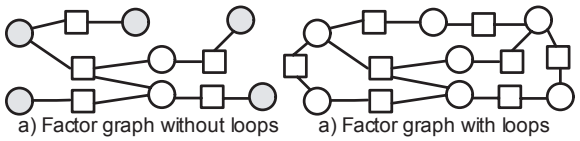

Figure 2.2: Shaded nodes represent leaf nodes

Message from variable x to local function f:

$$m_{x\to f}(x) = \prod_{h\in g_x\setminus f} m_{h\to x}(x) \tag{2.22}$$

where g_x represents all the functions that have x as one of its arguments. The messages consist of all the incoming messages into node x, except the one coming from node f.

Message from local function to variable:

$$m_{f\to x}(x) = \sum_{\sim x} f(X) \prod_{y\in n(f)\setminus x} m_{y\to f}(y) \tag{2.23}$$

where $X = n(f)$ is the set of arguments of the function f, and $\sum_{\sim x}$ represents the summations of all the arguments in X excepting x. Notice that both kind of messages are functions of variable x. Variable-to-function messages can be interpreted as the "belief" that the variable has, of itself, given to the values of all its other functions. Function-to-variable messages can be interpreted as the "belief" that the function has with respect to the variable's state, given the states of all the other variables in the function's argument.

The algorithm starts sending messages through the leaf nodes (fig. 2.2a). Here, only nodes representing hidden variables are considered. Observed variable nodes just send identity messages. *Condition one*: A node sends a message once all the incoming messages needed to send that message have been received. *Condition two*. The algorithm terminates once two messages have been passed over every edge, once in each direction. The marginal posterior probability for (hidden) variable node X_i, $p(X_i)$, is

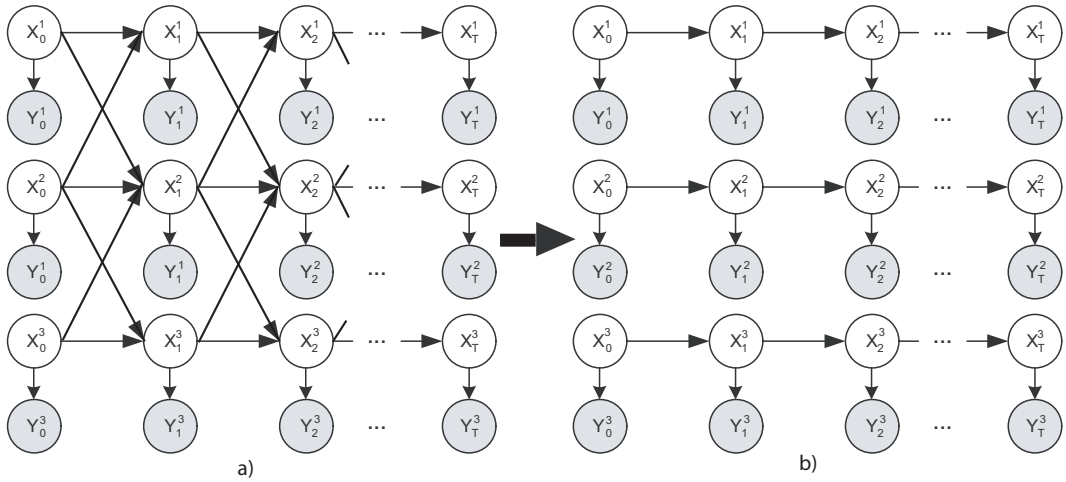

Figure 2.3: a) Complex model, b) Equivalent model with the variational approximation.

found by multiplying all the incoming messages into the node. Alternatively, it can be computed by multiplying the incoming and outgoing messages through the same edge. The operation of the algorithm is described for the case of an HMM on Appendix A.

2.5 Inference Approximation

Usually for more complex graphical models even with the factorization of the joint probability distribution, the exact posterior cannot be computed in a tractable manner. Consider for example the graphical model in figure 2.3 a).

The auxiliary function $L(q, \theta)$ for this model has the form:

$$\mathcal{L}(q,\theta) = \sum_X q(X \mid Y) \log \frac{\prod_{t=0}^{T} p(Y_t^1 \mid X_t^1) p(Y_t^2 \mid X_t^2) p(Y_t^3 \mid X_t^3)}{q(X \mid Y)}$$

$$\sum_X q(X \mid Y) \log \frac{\prod_{t=0}^{T-1} p(X_{t+1}^1 \mid X_t^1, X_t^2) p(X_{t+1}^3 \mid X_t^3, X_t^2) p(X_{t+1}^2 \mid X_t^1, X_t^2, X_t^3)}{q(X \mid Y)}$$

$$= \sum_{t=0}^{T-1} \sum_{X_t^1, X_t^2, X_{t+1}^1} q(X_t^1, X_t^2, X_{t+1}^1 \mid Y) \log p(X_{t+1}^1 \mid X_t^1, X_t^2)$$

$$+ \sum_{t=0}^{T-1} \sum_{X_t^1, X_t^2, X_t^3, X_{t+1}^2} q(X_t^1, X_t^2, X_t^3, X_{t+1}^2 \mid Y) \log p(X_{t+1}^2 \mid X_t^1, X_t^2, X_t^3)$$

$$+ \sum_{t=0}^{T-1} \sum_{X_t^2, X_t^3, X_{t+1}^3} q(X_t^2, X_t^3, X_{t+1}^3 \mid Y) \log p(X_{t+1}^3 \mid X_t^2, X_t^3)$$

$$+ \sum_{k=1}^{3} \sum_{t=0}^{T} \sum_{X_t^k} q(X_t \mid Y) \log p(Y_t^k \mid X_t^k) - \sum_X q(X \mid Y) \log q(X \mid Y) \quad (2.24)$$

Assuming that random variables X_t^k can take N discrete values. The E-step will require to compute N^4*(T-1) probability entries to account for the $q(X_t^1, X_t^2, X_t^3, X_{t+1}^2 \mid Y)$ term, which is computationally intractable for most values of N and T. The sum-product algorithm is also infeasible since the model has loops.

There are several approximation techniques that can be used to approximate inference on intractable models such as: sampling techniques, variational approximations and loopy belief propagation. In this thesis we use the latter two.

2.5.1 Variational Methods

As discussed in chapter 2, if the form of the "averaging" function $q(X \mid Y)$ is left unconstrained, the choice that maximizes the auxiliary function $L(q,\theta) = \sum_X q(X \mid Y) \log \frac{p(X,Y)}{q(X|Y)}$ is $q(X \mid Y) = p(X \mid Y)$. However, the computation of the posterior probability is not always tractable. When this is the case a tractable maximization of $L(q,\theta)$

can be achieved by imposing restrictions on the form that the "averaging" function can take and searching for the one that maximizes $L(q, \theta)$ in the restricted tractable space.

An alternative view of this approach can be found by minimizing the cost function $F(q)$ = -$L(q, \theta)$ with respect to $q(X \mid Y)$, instead of maximizing the auxiliary function. The restricted $q(X \mid Y)$ function that minimizes the cost function $F(q)$, referred as the free energy of the model, is the one that better aproximates the true posteriors $p(X \mid Y)$ relative to their Kullback-Leibler divergence, $D(q(X \mid Y), p(X \mid Y))$, given that:

$$\mathcal{F}(q) = -\sum_X q(X \mid Y) \log \frac{p(X,Y)}{q(X \mid Y)} = \sum_X q(X \mid Y) \log \frac{q(X \mid Y)}{p(X \mid Y)p(Y)}$$
$$= \sum_X q(X \mid Y) \log \frac{q(X \mid Y)}{p(X \mid Y)} - \sum_X q(X \mid Y) \log p(Y)$$
$$= D(q(X \mid Y), p(X \mid Y)) - \log p(Y) = \mathcal{F}(q,p) \quad (2.25)$$

And that $\log p(Y)$ is not a function of $q(x \mid Y)$. The choice of $q(x \mid Y)$ that minimizes the free energy, is also the choice that has the minimum Kullback-Leibler divergence with respect to the true posterior $p(X \mid Y)$

Therefore the idea behind variational methods is to estimate a "simple" and tractable function $q(x \mid Y)$ that is as close as possible to the true posterior, to be used to learn the model parameters, answer queries relative to the "state" of the model, etc.

As an example, consider again the model in figure 2.3 a) with auxiliary function (2.24). Exact inference will require an averaging function with the form $q(X) = q(X_1^1, X_2^1, ..X_T^1, X_1^2, ..X_T^3)$, which yields to an intractable E-step. However, we can use a restricted averaging function of the form $q(X) = \prod_{k=1}^{3}(q^k(X_1^k, X_2^k, ..., X_1^k, X_T^k))$

The correspondent auxiliary function has the form:

$$\mathcal{L}(q,\theta) = \sum_X q(X \mid Y) \log \frac{\prod_{t=0}^{T} p(Y_t^1 \mid X_t^1) p(Y_t^2 \mid X_t^2) p(Y_t^3 \mid X_t^3)}{q(X \mid Y)}$$

$$\sum_X q(X \mid Y) \log \frac{\prod_{t=0}^{T-1} p(X_{t+1}^1 \mid X_t^1, X_t^2) p(X_{t+1}^3 \mid X_t^3, X_t^2) p(X_{t+1}^2 \mid X_t^1, X_t^2, X_t^3)}{q(X \mid Y)}$$

$$= \sum_{t=0}^{T-1} \sum_{X_t^1, X_t^2, X_{t+1}^1} q^1(X_t^1, X_{t+1}^1 \mid Y) q^2(X_t^2 \mid Y) \log p(X_{t+1}^1 \mid X_t^1, X_t^2)$$

$$+ \sum_{t=0}^{T-1} \sum_{X_t^1, X_t^2, X_t^3, X_{t+1}^2} q^1(X_t^1 \mid Y) q^2(X_t^2, X_{t+1}^2 \mid Y) q^3(X_t^3 \mid Y) \log p(X_{t+1}^2 \mid X_t^1, X_t^2, X_t^3)$$

$$+ \sum_{t=0}^{T-1} \sum_{X_t^2, X_t^3, X_{t+1}^3} q^2(X_t^2 \mid Y) q^3(X_t^3, X_{t+1}^3 \mid Y) \log p(X_{t+1}^3 \mid X_t^2, X_t^3)$$

$$+ \sum_{k=1}^{3} \sum_{t=0}^{T} \sum_{X_t^k} q(X_t \mid Y) \log p(Y_t^k \mid X_t^k) - \sum_{k=1}^{3} \sum_{X^k} q^k(X^k \mid Y) \log q^k(X^k \mid Y) \tag{2.26}$$

The E-step for this auxiliary function will require to estimate $3 \times N^2 * T$ values (3 times as a regular HMM) instead of the intractable number $(N^4 + 2N^3)T$, needed with a unrestricted averaging function.

As hinted in the previous paragraph this variational approximation results in an "apparent" decoupling of the three chains on the model, figure 2.3 b). Later in this thesis we will show that the individual $q^k(X_k \mid Y)$ can be estimated using forward-backward recursions (eqns. 2.17-2.20) with non-stationary transition matrices on each one of the resultant "uncoupled" chains.

2.5.2 Loopy Belief Propagation

The sum-product algorithm, like the junction tree algorithm, computes exact inference in models that can be organized as trees (fig. 2.2a), i.e. models without loops. When the model structure involves loops (fig.2.2b), *condition one* for the algorithm operation

can not be met. In those situations, the sum-product algorithm can still be used although it no longer provides exact inference. There is evidence, however, that it approximates exact inference(J.S. Yedidia and Weiss 2001; Weiss and Freeman 2001). When the sum-product algorithm is used for models with loops it is called the loopy belief propagation algorithm. The lack of clear leaf nodes in loopy graphs blurs the message passing initialization process, creating the need for *ad hoc* schedules for the message passing. Whenever *condition one* cannot be met, the missing incoming messages are set to be uniform, which requires iterating the message passing procedure even after *condition two* has been met. The algorithm should finish when the inference of the variables' posteriors does not change between successive iterations (where an iteration is defined as a complete cycle of message passing rules).

2.6 Summary

This chapter intended to provide the foundation on the statistical graphical models framework required for the later chapters. We now proceed to extend the conventional multi-microphone source separation scheme to accomodate the use of statistical models.

Chapter 3

Maximum likelihood filter-and-sum system

As stated in chapter 1, the challenging environment conditions of the meeting recording scenario require the use of multiple observations to better deal with the adverse conditions. Therefore, we start briefly describing the conventional multimicrophone techniques, so that we could later emphasize the differences with our approach.

Conventional Blind Source Separation (BSS) of audio mixtures requires the use of signals recorded using multiple microphones, the problem is usually framed under the theorethical framework of Independent Component Analysis with convolutive mixtures (Convolutive ICA), which is expressed mathematically for each mixture $x_i[n]$ as:

$$x_i[n] = \sum_{j=1}^{N} a_{ij}[n] * s_j[n] = \sum_{j=1}^{N} \sum_{k=0}^{K-1} a_{ij}[k] s_j[n-k] \tag{3.1}$$

where x_i is the observed mixture at the i_{th} microphone, s_j ($j \in [1-N]$), the N assumed independent sources and a_{ij} are the K coefficients of the FIR filter that relates source s_j with mixture x_i.

To invert the convolutive mixtures (eq. 3.1), a set of similar FIR filters is typically

used.

$$y_i[n] = \sum_{j=1}^{L} w_{ij}[n] * x_j[n] = \sum_{j=1}^{L} \sum_{k=0}^{K-1} w_{ij}[k] x_j[n-k] \tag{3.2}$$

where y_i is the estimated output signal for independent source s_i. Therefore, the component signals are estimated as a weighted combination of current and past samples from the multiple recordings of the mixed signals.

Representing the separating filters as a sequence of coefficient matrices $\mathbf{W}_k$ at delay k, the separated complete output with this notation can be expressed as:

$$\mathbf{y}[n] = \sum_{k=0}^{K-1} \mathbf{W}_k \mathbf{x}[n-k] \tag{3.3}$$

Here $\mathbf{x}[n-k]$ is the L-dimensional data vector containing the values from the mixtures captured at the L microphones at time frame $n-k$, and $\mathbf{y}[n]$ is the estimated output vector whose components are the estimates of the N source signals.

Weights W_k are estimated using gradient descent to optimize an objective function that measures the independence of the estimated component signals, $\mathbf{y}$.

$$\triangle \mathbf{W}_k \propto -\frac{\partial Q(\mathbf{y})}{\partial \mathbf{W}_k} \tag{3.4}$$

where objective function $Q(\mathbf{y})$ measures the degree of independence between the individual estimated signals y_i.

Even though these techniques perform well for certain signal mixtures, they fail in many situations, such as when the signals are recorded in a reverberant environment, since the algorithms do not aim to dereverberate (deconvolve) the data but simply identify the sources in the scene. This arises given that the goal of the objective function $Q(\mathbf{y})$ is to find independent sources rather than to dereverb the actual signals. The resulting

signals would be the independent sources, as they would be captured by the sensors, if they were alone in the room.

They also usually fail when the degree of overlap and/or the dimensionality of the recordings makes the blind inference problem intractable: if the mixture components really could be anything, we have no way of getting good estimates for missing parts of the signal where the energy is locally swamped by interfering sources.

Therefore, we propose to adapt the conventional blind source separation multi-microphone setup to accommodate the use of statistical models to guide the separation. We treat the signal separation problem as one of beam-forming, where each signal is extracted using a filter-and-sum array. The filters are estimated to maximize the likelihood of the summed output, measured on the statistical model for the desired signal. We call this approach the maximum likelihood filter-and-sum system.

3.1 Filter-and-sum processing

Assume that the number of speakers is known. For each of the speakers, a separate filter-and-sum array is designed. The filter-and-sum process is depicted in figure 3.1. The signal from each microphone is filtered by a microphone-specific filter. The various filtered signals are summed to obtain the final processed signal.

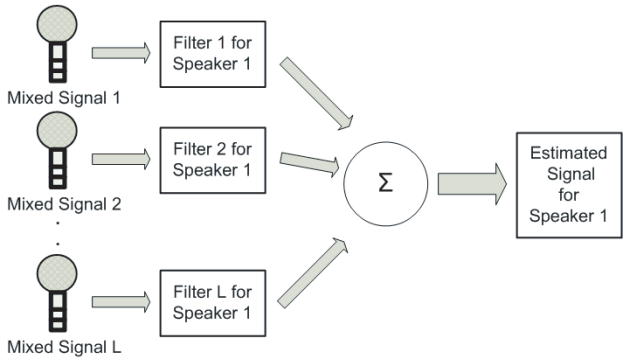

Figure 3.1: Filter-and-sum processing.

Thus, the output signal for i_{th} speaker, $y_i[n]$, is obtained as:

$$y_i[n] = \sum_{j=1}^{L} h_{ij}[n] * x_j[n] = \sum_{j=1}^{L} \sum_{k=0}^{K-1} h_{ij}[k] x_j[n-k] \tag{3.5}$$

where L is the number of microphones in the array, $x_j[n]$ is the signal at the j^{th} microphone and $h_{iv}[n]$ is the K-coefficients filter applied to the j^{th} microphone for speaker i. The filter impulse responses $h_{ij}[n]$ must be optimized such that the resultant output $y_i[n]$ is the separated signal from the i^{th} speaker.

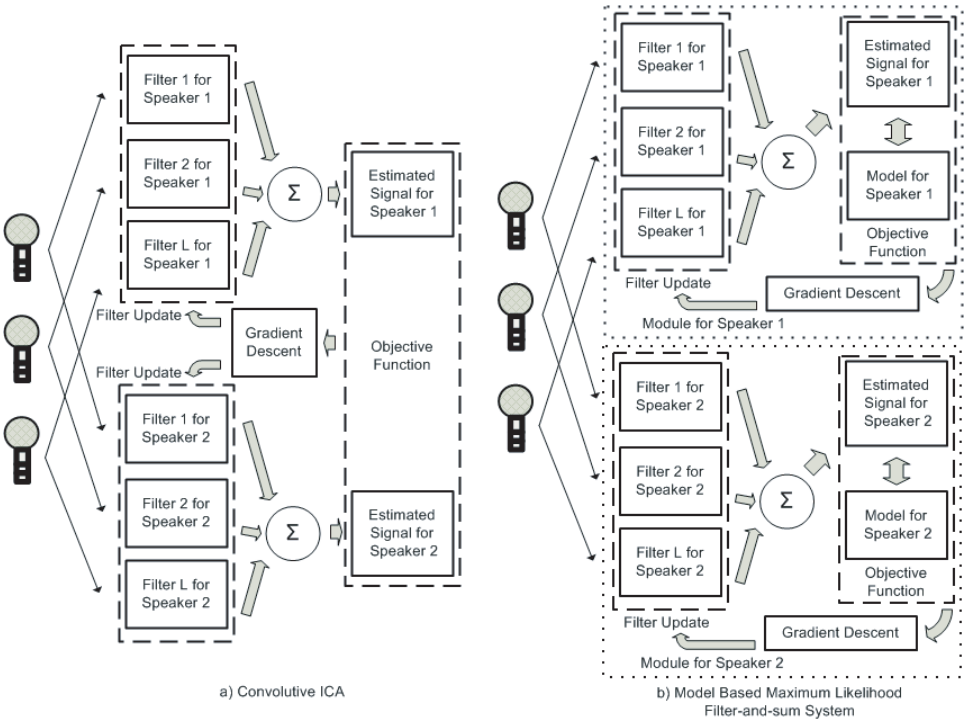

Figure 3.2: a) Filter Optimization for convolutive ICA; b) Proposed model-based optimization.

Equation 3.5 is the same as equation 3.2, but the difference in our approach lies in how we estimate the filter coefficients for each source. In the convolutive ICA approach the filters of all the speakers are jointly optimized by increasing the degree of independence between all the estimated signals for all the sources, while in our approach we optimize the filters for each source independently of the filters for the other sources.

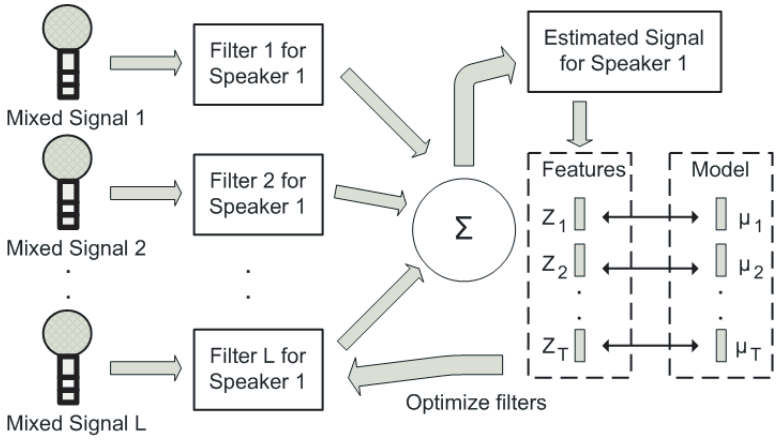

Figure 3.3: The goal is to optimize the filters such that the features derived from the estimated speaker at frame t, z_t resembles the features described by the model, μ_t

The differences in the optimization procedure between the two approaches are depicted in figure 3.2. Now we proceed to describe how we learn the filters through such an optimization scheme.

3.2 Learning the filter coefficients

We first describe the intuition behind the learning procedure, which we will further formalize as a maximum likelihood approach.

Each speaker filter-and-sum module has as input a combined speech mixture and the corresponding single speaker speech as the output. Then, if for a short segment of the combined speech we know the "expected" spectral features for the desired speech, we could "train" the filters to enhance those spectral features on the composed signal, while attenuating all other spectral components present on the composed speech that are not "required" for the targeted speaker speech. The "expected" spectral features actually consists of a speech model for each speaker. Figure 3.3 depicts the idea behind this approach.

The filters are trained using just a few seconds of a combined "training" signal

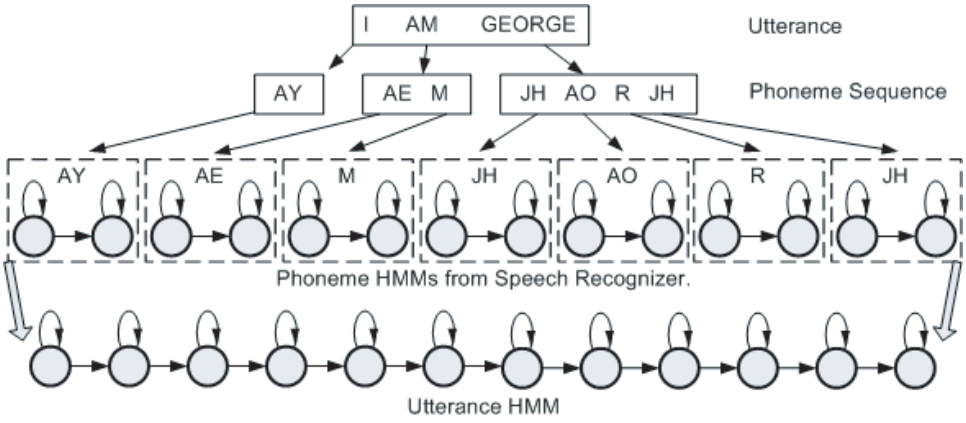

Figure 3.4: The utterance transcription is parse into the correspondent sequence of phonemes. The HMMs for each phoneme is retrieved from the speech recognizer and concatenated to from a single utterance HMM.

for which the correct speech transcriptions for each speaker present in the mixture are known. It is assumed that this training signal has the same characteristics in terms of speakers and their relative positions with respect to the microphones as the combined signals that the filters are intended to separate.

We further assume that we have access to a speaker-independent single gaussian hidden Markov model (HMM) based speech recognition system that has been trained on a 40-dimensional log-Mel-spectral representation of the speech signal. The recognition system includes HMM parameters for the various phonemes that comprise the language. The required speech models for each speaker are obtained using the phoneme HMM parameters from the speech recognizer and the known transcription for each speaker's training utterance in the following way: Each word in the transcription is decomposed into its correspondent sequence of phonemes, then the HMM parameters for each of those phonemes are retrieved from the speech recognizer and concatenated together to form a word-HMM. The word-HMMs parameters are in turn concatenated to obtained an utterance HMM, which corresponds to the desired speaker speech model. Figure 3.4 depicts this procedure.

For each speaker i, our objective is to maximize the likelihood of $Z_i = [z_{i,0}, z_{i,1}, ..z_{i,T}]$,

the sequence of 40-dimensional log-Mel-spectral vectors computed from the output of its filter-and-sum processed signal, on its utterance HMM. Even though this technique was introduced for a single speaker in a speech enhancement application in (Seltzer et al. 2002), we present it as a variation of the generalized EM algorithm.

The log-Mel-spectral representation of speech is widely used for speech recognition applications. The speech waveform, y is first windowed with analysis window w and then the discrete short time Fourier transform, y_t^k is computed:

$$y_t^k = \sum_{\tau=0}^{N-1} w[\tau] y[t \cdot H + \tau] e^{-j2\pi\tau k/N} \tag{3.6}$$

where t is the time-frame index, k indexes the frequency bands, N is the size of the discrete Fourier transform, H is the hop between successive time-frames, $w[\tau]$ is the N-point short-time window, and $y[\tau]$ is the original time-domain signal.

The magnitude of y_t^j is then weighted by a series of filter frequency responses, known as Mel-scale filters, whose center frequencies and bandwidths roughly match those of the auditory critical band filter. Finally, the spectral amplitudes are compressed by applying the $\log(\cdot)$ to the resulting Mel coefficients.

The log-Mel-spectral coefficients for the i_{th} speaker at time-frame t can be expressed as:

$$z_{i,t} = \log(\mathbf{M} diag(\mathbf{F}\hat{\mathbf{y}}_{i,t}\hat{\mathbf{y}}_{i,t}^{'}\mathbf{F}^H)) \tag{3.7}$$

where $\mathbf{M}$ is the Mel-scale filters matrix, $diag(\mathbf{X})$ is a vector equal to the diagonal of matrix $\mathbf{X}$, $\hat{\mathbf{y}}_{i,t}$ is the t frame with N-samples of the speech signal at the output of the i_{th} filter-and-sum module weighted by window w, $\mathbf{F}$ is the $N \times N$ Fourier matrix with its (τ, k) element equal to $e^{-j2\pi\tau k/N}$, $\hat{\mathbf{y}}_t^{'}$ is the transpose of vector $\hat{\mathbf{y}}_t$, and $\mathbf{F}^H$ is the Hermitian matrix of matrix $\mathbf{F}$.

From eqn. 3.5 the n_{th} value of $\hat{\mathbf{y}}_{i,t}$ can be expressed as:

$$\hat{\mathbf{y}}_{i,t}[n] = w[n]y[t \cdot H + n] = w[n] \sum_{j=1}^{L} \sum_{k=0}^{K-1} h_{ij}[k]x_j[t \cdot H + n - k] \tag{3.8}$$

Vector $\hat{\mathbf{y}}_{i,t}$, can then be expressed as:

$$\hat{\mathbf{y}}_{i,t} = \mathbf{X}_t \cdot \mathbf{h}_i = \tag{3.9}$$

$$\begin{pmatrix} w[0]x_1[tH] & w[0]x_1[tH-1] & . & w[0]x_L[tH-K+1] \\ w[1]x_1[tH+1] & w[1]x_1[tH] & . & w[1]x_L[tH-K+2] \\ & & & \\ . & . & . & \\ w[N-1]x_1[tH+N-1] & w[N-1]x_1[tH+N-2] & . & w[N-1]x_L[tH-K+N] \end{pmatrix} \begin{pmatrix} h_1[0] \\ h_1[1] \\ . \\ h_1[K-1] \\ h_2[0] \\ . \\ . \\ h_L[K-1] \end{pmatrix}$$

Therefore we can express feature vectors $z_{i,t}$ as functions of $\mathbf{h}_i$, the filter coefficients for the i_{th} filter-and-sum module, (expressed as a $L \times K$ column vector obtained by concatenating the K-point filter coefficients for each one of the L microphones) as:

$$z_{i,t}(\mathbf{h}_i) = \log(\mathbf{M}diag(\mathbf{F}\mathbf{X}_t\mathbf{h}_i\mathbf{h}_i^{'}\mathbf{X}_t^{'}\mathbf{F}^H)) \tag{3.10}$$

The filters for the i^{th} speaker are optimized, by maximizing the log-likelihood of $Z_i(\mathbf{h}_i)$, the sequence of log-Mel-spectral vectors computed from the output of the array for the i^{th} speaker, on the HMM for that speaker.

As discussed in section 2, the expectation-maximization (EM) algorithm maximizes the log-likelihood of the model through the use of auxiliary function, $\mathcal{L}(q, \theta)$, (eq. 2.3). The algorithm maximizes the auxiliary function by iteratively finding the "averaging" function q_{t+1} that maximizes $\mathcal{L}(q, \theta)$ given the current parameters, θ_t (E Step), and then estimating the parameters θ_{t+1} that best fit the model, given the state of the model described by q_{t+1}, (M Step).

However for the present task, the optimal model parameters are given by the model retrieved from the speech recognizer, therefore, we want to perform exactly the opposite of what the M step normally does, instead of finding the set of parameters that better fit the data given the present state of the model, we want to find the best "data" $Z_i(\mathbf{h}_i)$ that best fits the optimal model parameters given the present state of the model. Since the "data" is a function of filter parameters $\mathbf{h}_i$, the "inverse" M-step can be perfomed by maximizing $\mathcal{L}(q, \theta)$ with respect to $\mathbf{h}_i$, instead of, with respect to model parameters θ.

For observation variable $Z(\mathbf{h}_i) = [z_1, .., z_T]$ and HMM states $S = [s_1, .., s_T]$, (where for simplicity, we have drop the speaker indices and the direct references to $\mathbf{h}_i$), $\mathcal{L}(q, \theta, \mathbf{h}_i)$ has the form:

$$\mathcal{L}(q, \theta, \mathbf{h}_i) = \sum_{t=0}^{T} \sum_{s_t} q(s_t \mid z_t) \log p(z_t \mid s_t) + \sum_{t=0}^{T-1} \sum_{s_t, s_{t+1}} q(s_{t+1}, s_{t+1} \mid z_t) \log p(s_{t+1} \mid s_t) + H_q(S) \quad (3.11)$$

As discussed in section 2, the E-step for an HMM can be easily performed by the forward-backward recursions, (eqns. 2.17-2.20). However, in the early iterations of the algorithm, the filters have not been fully optimized and the output of the filter-and-sum array for any speaker will contain a significant fraction of the signal from other speakers as well. As a result the "data" will have serious mismatches with individual HMMs resulting in bad estimates for $q(S)$. Therefore, we need to envision a way to account for other speakers while estimating the right $q(S)$ on individual HMMs. In the next section, we present how we accomplish this task, for now we assume that reliable $q(S)$ estimates are available.

The "inverse" M-Step consist on:

"Inverse" **M**aximization Step $\quad argmax_{\mathbf{h}_i}(\mathcal{L}(q_{t+1}, \theta, \mathbf{h}_i))$ (3.12)

Then we need to compute the derivative of $\mathcal{L}(q_{t+1}, \theta, \mathbf{h}_i)$ with respect to $\mathbf{h}_i$. Since $p(z_t \mid s_t)$ is defined as a single Gaussian distribution, the component of eqn. 3.11 that involves $\mathbf{h}_i$ is:

$$\mathcal{L}^{\mathbf{h}_i}(q, \theta, \mathbf{h}_i) = \sum_{t=0}^{T} \sum_{s_t} q(s_t \mid z_t) \log p(z_t \mid s_t) = \\ -\frac{1}{2} \sum_{t=0}^{T} \sum_{s_t} q(s_t \mid z_t) \left(\log((2\pi)^K \mid \Sigma \mid) + (z_t - \mu_{s_t})' \Sigma_{s_t}^{-1} (z_t - \mu_{s_t}) \right) \quad (3.13)$$

Which can be further simplified to:

$$\mathcal{L}^{\mathbf{h}_i}(q, \theta, \mathbf{h}_i) = -\frac{1}{2} \sum_{t=0}^{T} \sum_{s_t} q(s_t \mid z_t)(z_t - \mu_{s_t})' \Sigma_{s_t}^{-1} (z_t - \mu_{s_t}) \quad (3.14)$$

Posteriors $q(s_t \mid z_t)$ are very "peaky" given that the speaker models are left-to-right HMMs, therefore, in practice we approximate equation 3.17 by only optimizing with respect to the parameters of the most likely state $\hat{s}_t$ under distribution $q(s_t \mid z_t)$.

$$\mathcal{L}^{\mathbf{h}_i}(q, \theta, \mathbf{h}_i) \approx \hat{\mathcal{L}}^{\mathbf{h}_i}(q, \theta, \mathbf{h}_i) = -\frac{1}{2} \sum_{t=0}^{T} (z_t - \mu_{\hat{s}_t})' \Sigma_{\hat{s}_t}^{-1} (z_t - \mu_{\hat{s}_t}) \quad (3.15)$$

where:

$$\hat{S} = maxarg_S(q(S \mid Z)). \quad (3.16)$$

Sequence $\hat{S}$ = $[\hat{s}_1, \hat{s}_2, .., \hat{s}_T]$ is referred from now on as the target.
Computing the derivative, we obtain:

$$\frac{\partial \hat{\mathcal{L}}^{\mathbf{h}_i}(q, \theta, \mathbf{h}_i)}{\partial \mathbf{h}_i} = -2\Re((\mathbf{F}\mathbf{X}_t)^H Diag(\mathbf{F}\mathbf{y}_{i,t}))\mathbf{M}' Diag(\mathbf{M} diag(\mathbf{F}\hat{\mathbf{y}}_{i,t}\hat{\mathbf{y}}_{i,t}^T \mathbf{F}^H)) \Sigma_{\hat{s}_t}^{-1} (z_t - \mu_{\hat{s}_t})$$

(3.17)

where $Diag(x)$ is a diagonal matrix with diagonal equal to vector x.

Direct maximization of $\hat{\mathcal{L}}^{\mathbf{h}_i}(q, \theta, \mathbf{h}_i)$ with respect to $\mathbf{h}_i$ is, however, not possible due to the highly non-linear relationship between the two. We therefore optimize $\hat{\mathcal{L}}^{\mathbf{h}_i}(q, \theta, \mathbf{h}_i)$ using the method of conjugate gradient descent, just as it is done in the "generalized" EM algorithm when direct maximization of the model parameters is not possible in the maximization step (Neal and Hinton 1998).

The filter optimization algorithm proceeds iteratively by alternately estimating the best target $\hat{S}$, and optimizing the filters.

Since the algorithm aims to minimize the distance between the output of the array and the target, the choice of a good target becomes critical to its performance.

3.3 E step: Target Estimation

Each speaker target sequence is derived from the HMM for that speaker's utterance.

Given that the speech recognizer HMMs where trained with clean speech and assuming that the optimal target is found and the filters are correctly optimized accordingly. The system should not only separate the sources but it should dereverberate them as well, since the target features are those of clean unreverberant speech. A critical difference in the objectives between our approach and ICA-based approaches.

This is done by determining the best state sequence through the HMM from the current estimate of that speaker's signal. A direct approach to obtaining the state sequence would be to directly find the most likely state sequence for the sequence of Log-Mel-spectral vectors for the signal. Unfortunately, in the early iterations of the algorithm when the filters have not yet been fully optimized, the output of the filter-and-sum array for any speaker contains a significant fraction of the signal from other speakers as well. As a result, naive alignment of the output to the HMM results in poor estimates for the

target.

Instead, we also take into consideration the fact that the array output is a mixture of signals from all the speakers. The HMM that represents this signal is a factorial HMM (FHMM) (Ghahramani and Jordan 1997) that is the cross-product of the individual HMMs for the various speakers. In an FHMM each state is a composition of one state from the HMMs for each of the speakers, reflecting the fact that the individual speakers may have been in any of their respective states, and the final output is a combination of the output from these states.

For simplicity, we focus on the two-speaker case. Extension to more speakers is straightforward.

Figure 3.5 illustrates the graphical model of an FHMM for two speakers.

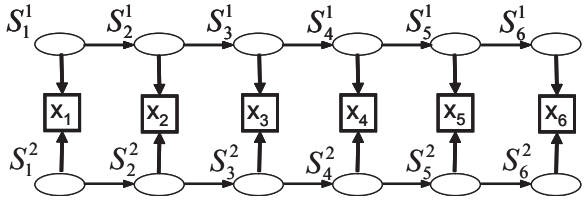

Figure 3.5: Factorial HMM for two speakers (two chains).

The complete set of hidden nodes in the model are represented by S = $[S^l, S^k]$, where S^l = $[s_1^l, .., s_T^l]$ and S^k = $[s_1^k, .., s_T^k]$ respectively correspond to the hidden states for the HMMs of speakers l and k. Observations Z = $[z_1, z_2, ..., z_T]$ correspond to the log-Mel-spectra coefficients calculated from the output of one of the filter-and-sum modules.

The joint probability for all the variables in the model is given by:

$$p(Z, S) = \prod_{t=0}^{T} p(z_t \mid s_t^l, s_t^k) \prod_{t=0}^{T-1} p(s_{t+1}^l \mid s_t^l) p(s_{t+1}^k \mid s_t^k) \tag{3.18}$$

Transition matrices $p(s_{t+1}^l \mid s_t^l)$ $p(s_{t+1}^k \mid s_t^k)$ are directly taken from the individual HMM's parameters. However, to account for the combined observations, the parameters of the output densities, $p(z_t \mid s_t^l, s_t^k)$ are a combination of the parameters of the output

densities for the individual models. Given parameters θ_i^l = $[\mu_i^l, \Sigma_i^l]$, θ_j^k = $[\mu_j^k, \Sigma_j^k]$, mean and variances for states i and j on the HMMs for speakers l and k, the factorial output probability $p(z_t \mid s_t^l = i, s_t^k = k)$ is given by:

$$p(z_t|s_t^l = i, s_t^k = k) = f(\theta_i^l, \theta_j^k) \tag{3.19}$$

The precise nature of the function $f()$ depends on the proportions to which the signals from the speakers are mixed in the current estimate of the desired speaker's signal. This in turn depends on several factors including the original signal levels of the various speakers, and the degree of separation of the desired speaker effected by the current set of filters. Since these are difficult to determine in an unsupervised manner, $f()$ cannot be precisely determined.

We do not attempt to estimate $f()$. Instead, the HMMs for the individual speakers are constructed to have simple Gaussian state output densities. We assume that the state output density for any state of the FHMM is also a Gaussian whose mean is a linear combination of the means of the state output densities of the component states. We define $\mu_{ij}^{l,k}$, the mean of the Gaussian state output density $p(z_t|s_t^l = i, s_t^k = k)$ as:

$$\mu_{ij}^{k,l} = \mathbf{A}^l \mu_i^l + \mathbf{A}^k \mu_j^k \tag{3.20}$$

where $\mathbf{A}^l$ and $\mathbf{A}^k$ are $D \times D$ weighting matrices, which we will estimate as part of the solution.

The intuition behind this approach is the following: assuming that we are estimating the target sequence for speaker l, at the early iterations of the algorithm, feature vectors Z_l will be composed by features from both speakers, thus weighting matrix $\mathbf{A}_k$ will have non-zero values to account for the presence of speaker k in speech signal, y_l, at the output of the filter-and-sum module for speaker l. However, once we start to learn the correct values for the filter coefficients for speaker l, the presence of speaker k on speech

signal y_l will decrease diminishing as well the presence of features from speaker k on feature vectors Z_l, resulting in a reduction of the magnitude of the weights of matrix $\mathbf{A}_k$, eventually driving $\mathbf{A}_k$ to zero.

As shown below when matrix $\mathbf{A}_k$ is close to zero, when signal y_l is mostly composed by speech from speaker l the factorial HMM actually behaves as an individual HMM as originally intended.

All factorial states have a common diagonal covariance matrix **C**.

The state output density of factorial state $s_{ij}^{k,l}$ = [s^l=i,s^k=j] is now given by:

$$p(z_t|s^l = i, s^k = j) = |\mathbf{C}|^{-1/2}(2\pi)^{-D/2}e^{-\frac{1}{2}(z_t-\mu_{ij}^{l,k})'\mathbf{C}^{-1}(z_t-\mu_{ij}^{l,k})} \tag{3.21}$$

The $\mathbf{A}^l$, $\mathbf{A}^k$, **C**, values are unknown and must be learned from the current estimate of the speaker's signal. Therefore the M-step is extended to estimate the degrees of mixture present in the outputs of the filter-and-sum modules through the learning of parameters $\mathbf{A}^l$, $\mathbf{A}^k$ and **C**. Their update formulas will be presented after discussing the inference of the model.

Exact inference requires an "auxiliary" function $q(S \mid Z)$ with the form:

$$q(S \mid Z) = q(s_0^l, s_1^l, ..., s_T^l, s_0^k, s_1^k, ..., s_T^k \mid Z) \tag{3.22}$$

For this choice of $q(S)$, the free energy of the model, $F(q,p)$ = -$\mathcal{L}(q,\theta)$ is given by:

$$\begin{aligned} F(q,p) = &\sum_S q(S \mid Z)\log q(S \mid Z) - \sum_{t=0}^{T}\sum_{s_t^l,s_t^k} q(s_t^l, s_t^k)\log p(z_t \mid s_t^l, s_t^k) - \\ &\sum_{t=0}^{T-1}\sum_{s_t^l,s_{t+1}^l} q(s_t^l, s_{t+1}^l)\log p(s_{t+1}^l \mid s_t^l) - \sum_{t=0}^{T-1}\sum_{s_t^k,s_{t+1}^k} q(s_t^k, s_{t+1}^k)\log p(s_{t+1}^k \mid s_t^k) \end{aligned} \tag{3.23}$$

Exact inference of a two speakers Factorial HMM can be done through factorial-forward/backward recursions: $\alpha_F(s_t^l, s_t^k)$ and $\beta_F(s_t^l, s_t^k)$. However, this approach requires the computation of $N_l \times N_k$ values for α_F, β_F at each frame t, (where N_l and N_k are the number of states for the corresponding HMMs), resulting in a total complexity of $O(TN_lN_k)$ per iteration. Even for a few seconds of "training" speech the resultant HMMs have over 500 states, which implies over 250,000 per iteration per frame, making exact inference computationally intractable.

We approximate inference through a variational "auxiliary" function $q(S \mid Z)$ with the form:

$$q(S \mid Z) = q(S^l \mid Z)q(S^k \mid Z) = q(s_0^l, ..., s_T^l \mid Z)q(s_0^k, ..., s_T^k \mid Z) \tag{3.24}$$

For this choice of $q(S)$, the free energy of the model is given by:

$$\begin{aligned} F(q,p) = &\sum_{S^l} q(S^l \mid Z) \log q(S^l \mid Z) + \sum_{S^k} q(S^k \mid Z) \log q(S^k \mid Z) \\ &- \sum_{t=0}^{T} \sum_{s_t^l} \sum_{s_t^k} q(s_t^l) q(s_t^k) \log p(z_t \mid s_t^l, s_t^k) - \sum_{t=0}^{T-1} \sum_{s_t^l, s_{t+1}^l} q(s_t^l, s_{t+1}^l) \log p(s_{t+1}^l \mid s_t^l) \\ &- \sum_{t=0}^{T-1} \sum_{s_t^k, s_{t+1}^k} q(s_t^k, s_{t+1}^k) \log p(s_{t+1}^k \mid s_t^k) \end{aligned} \tag{3.25}$$

Since the chosen $q(S)$ decouples the individual HMMs, we can maximize the free energy with respect to the variational posteriors for one chain while keeping the variational posteriors for the other one fixed. If eqn. 3.25 is to be maximized with respect to $q(S^l \mid Z)$, we will need to take its derivative with respect to $q(S^l \mid Z)$, therefore the components of $F(q,p)$ that involve $q(S^l \mid Z)$ are the only ones relevant for this maximization.

$$F^l(q^l, p) = \sum_{S^l} q(S^l \mid Z) \log q(S^l \mid Z) - \sum_{t=0}^{T} \sum_{s_t^l} \sum_{s_t^k} q(s_t^l) q(s_t^k) \log p(z_t \mid s_t^l, s_t^k) - \sum_{t=0}^{T-1} \sum_{s_t^l, s_{t+1}^l} q(s_t^l, s_{t+1}^l) \log p(s_{t+1}^l \mid s_t^l) \quad (3.26)$$

Which can be expressed as:

$$F^l(q^l, p) = \sum_{S^l} q(S^l \mid Z) \log q(S^l \mid Z) - \sum_{t=0}^{T} \sum_{s_t^l} q(s_t^l) \overline{\log}_{q^k} p(z_t \mid s_t^l) - \sum_{t=0}^{T-1} \sum_{s_t^l, , s_{t+1}^l} q(s_t^l, s_{t+1}^l) \log p(s_{t+1}^l \mid s_t^l) \quad (3.27)$$

where $\overline{\log}_{q^k} p(z_t \mid s_t^l) = \sum_{s_t^k} q(s_t^k) \log p(z_t \mid s_t^l, s_t^k)$ can be interpreted as the "expected" log-local likelihood for variable s_l^t on the HMM for speaker l given the state of the HMM for speaker k.

Comparing the function defined in eqn. 3.27 with the function defined in eqn. 2.10, the auxiliary function for an individual HMM, we can observe that they have the same form and therefore they can be maximized in the same way. Thus, we estimate optimal $q(s_t^l)$ using the forward/backward recursions define in eqns.2.15-2.19 with $\exp^{\overline{\log}_{q^k} p(z_t | s_t^l)}$ as the local likelihood, i.e. $p(Y_t \mid X_t)$, in eqns. 2.15–2.19.

Expressing eqn. 3.26 in terms of the model parameters:

$$F^l(q^l,p) = \sum_{S^l} q(S^l \mid Z)\log q(S^l \mid Z) - D(T+1)\log 2\pi$$
$$-\frac{1}{2}\sum_{t=0}^{T}\sum_{i} q(s_t^l = i)[\log \mid \mathbf{C} \mid +(z_t - \mathbf{A}^l\mu_i^l)'\mathbf{C}^{-1}(z_t - \mathbf{A}^l\mu_i^l)$$
$$+\sum_{j} q(s_t^k = j)(-2(z_t - \mathbf{A}^l\mu_i^l)' + (\mathbf{A}^k\mu_j^k)')\mathbf{C}^{-1}\mathbf{A}^k\mu_j^k]$$
$$-\sum_{t=0}^{T-1}\sum_{i,j} q(s_t^l = i, s_{t+1}^l = j)\log \pi_{i,j}^l \quad (3.28)$$

If $\mathbf{A}^k \approx 0$, eqn. 3.28 can be reduced to:

$$F^l(q^l,p) \approx \sum_{S^l} q(S^l \mid Z)\log q(S^l \mid Z) - D(T+1)\log 2\pi$$
$$-\frac{1}{2}\sum_{t=0}^{T}\sum_{i} q(s_t^l = i)[\log \mid \mathbf{C} \mid +(z_t - \mathbf{A}^l\mu_i^l)'\mathbf{C}^{-1}(z_t - \mathbf{A}^l\mu_i^l)]$$
$$-\sum_{t=0}^{T-1}\sum_{i,j} q(s_t^l = i, s_{t+1}^l = j)\log \pi_{i,j}^l \quad (3.29)$$

Eqn. 3.29 has the same form as eqn. 2.11, showing that when A_k is close to zero, the factorial HMM actually behaves as an individual HMM as originally intended.

We define $\vec{A} = [A^l, A^k]$, and $\vec{s}_t^{\,l}$ as a $(N_l \times 1)$ indicator vector used to indicate that the HMM for speaker l is at state j at frame t by having a one at its j_{th} position and zeros in all others; ($\vec{s}_t^{\,k}$ is the analogous vector for the HMM for speaker k) and $\mathbf{s}_t$ a $(N_l + N_k \times 1)$ indicator vector formed by the concatenation of vectors $\mathbf{s}_t^l$ and $\mathbf{s}_t^k$,i.e. $\mathbf{s}_t^{'}$ = $[\mathbf{s}_t^{l'}\ \mathbf{s}_t^{k'}]$, which is used to indicate the current factorial state of the HMM, i.e. $s_{ij}^{k,l}$, that the factorial HMM is at.

The update formulas for parameters $\vec{A}$ and $\vec{C}$ can be found to be:

$$\mathbf{A} = \sum_t (z_t \bar{\mathbf{s}}_t' \mathbf{M}')(\mathbf{M}\sum_t \overline{(\mathbf{s}_t \mathbf{s}_t')}\mathbf{M}')^{-1} \quad (3.30)$$

$$\mathbf{C} = \frac{1}{T+1}\sum_{t}(z_t^{'}z_t) - \frac{1}{T+1}\sum_{t}(A^l M^l(\bar{\mathbf{s}}_t^l)^{'} + A^k M^k(\bar{\mathbf{s}}_t^k)^{'})z_t \tag{3.31}$$

where $\bar{\mathbf{s}}_t^l$ is the expected value of $\mathbf{s}_t^l$ under distribution $q^l(s_t^l)$, i.e. $\bar{\mathbf{s}}_t^l = \sum_{s_t^l} q^l(s_t^l)\mathbf{s}_t^l$, $\bar{\mathbf{s}}_t^k$ is the expected value of $\mathbf{s}_t^k$ under distribution $q^k(s_t^k)$, $\bar{\mathbf{s}}_t$ is the expected value of $\mathbf{s}_t$ under distribution $q(s_t)$, i.e. $\bar{\mathbf{s}}_t = \sum_{s_t^k}\sum_{s_t^l} q^l(s_t^l)q^k(s_t^k)\mathbf{s}_t$. $\mathbf{M}^l$ is a $D \times N_l$ matrix whose columns are the means of the N_l states on the HMM for speaker l, $\mathbf{M}^k$ is the corresponding matrix for the HMM for speaker k and block matrix $\mathbf{M}$ is defined by:

$$\mathbf{M} = \begin{pmatrix} \mathbf{M}^l & 0 \\ 0 & \mathbf{M}^k \end{pmatrix}$$

The overall algorithm works as follows:

1. Initialize filter parameters to h_i[0] = 1/N, and h_i[k] = 0 for $k \neq 0$.

2. Find HMMs for each speaker using the corresponding transcriptions and compose them into a Factorial HMM.

3. For each speaker i, estimate "isolated" speech signals y_i, processing the microphone signals with the corresponding filter-and-sum module (eq. 3.5).

4. Compute feature vectors Z_i from the corresponding "isolated" speech signal.

5. For each i filter-and-sum module iteratively estimate the variational posteriors $q(s)$ and parameters $\vec{A}$ and $\vec{C}$ (eqns. 3.30 and 3.31) for the speakers factorial HMM applied on observations Z_i until the bound on the loglikelihood of Z_i under the model converges.

6. Compute the target $\hat{S}_i$ for each speaker i by finding the most likely state path on the HMM for speaker i on the factorial model learned for feature vectors Z_i.

7. For each filter-and-sum model i, find optimal filter coefficients $\mathbf{h}_i$ through conjugate gradient descent to optimize eq. 3.15.

8. If target has not changed in successive iterations go back to step 3.

9. The system is ready to separate new composed signals as long as the number of speakers, their identity and their relative positions with respect to the microphones does not differ from those implicit in the signals used to learn the filters.

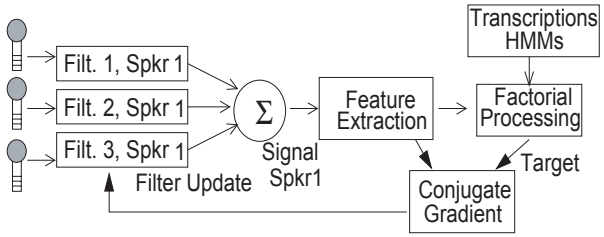

Figure 3.6: Complete system with speaker model produced by factorial processing.

A schematic of the overall system is shown in figure 3.6.

3.4 Experimental Results

Simulated mixed-speaker recordings were generated using utterances from the test set of the Wall Street Journal(WSJ0) corpus.

Room simulation impulse response filters were simulated for a room 4m $\times$ 5m $\times$ 3m with a reverberation time of 200msec. The microphone array configuration consisted of 8 microphones placed around an imaginary 0.5m $\times$ 0.3m flat panel display on one of the walls. To obtain mixed recordings, two speech sources were placed in different locations in the room. A room impulse response filter was created for each source/microphone pair. The clean speech signals for both sources were passed through each of the 8 speech source room impulse response filters and then added together.

Mixed training recordings were generated using two utterances, each from a different speaker. A different position in the room was assigned to each speaker. Filters were estimated for each of the speakers in the training mixture using the algorithm described in this chapter. For the test data, mixed recordings were generated using other

utterances, both with utterances from the same speakers as in the training recording, and with recordings from new speakers. The locations of the speakers in the test recordings were also varied, with recordings being generated both from the same locations as the training speakers, and from other locations.

Table 3.1 shows the separation results for four typical mixed recordings, obtained with filters estimated from a single training recording. The results on the separation for the training signal are also shown for comparison purposes. The table gives the ratio of the energy of the signal from the desired speaker to that from the competing speaker, measured in decibels, in the separated signals. We refer to this measurement as the "speaker-to-speaker ratio", or the SSR. The higher this value, the higher the degree of separation obtained for the desired speaker.

The first row in 3.1 (labeled Delay & Sum) shows separation results obtained with a default comparison. Since the basic approach is that of beamforming, we designated simple delay-and-sum processing (Johnson and Dugeon 1997) as the comparison. Here the signals are simply aligned to cancel out the delays from the desired speaker to the microphone (computed here with full prior knowledge of speaker and microphone positions) and added. The second row in table 3.1 shows the results for the filter-and-sum processing using the filter for the desired speaker. The columns in the table are arranged pair-wise. Each column reports the separation performance obtained for one of the two speakers. In all cases, the SSR for the desired speaker is reported.

Notice from the reported SSRs for the training signal for the simple delay and sum approach, that speaker 2 clearly dominates the mixture since its energy is 11dB over the energy for Speaker 1, even after delaying the microphone signals with respect to Speaker 1 to enhance its components in the mixtures, and summing the delayed versions. Therefore the training signal clearly consists of a situation of a dominant speaker in the foreground (speaker 2) with a weak speaker on the background.

In the experiments, we measure the similarity between training and test signals by

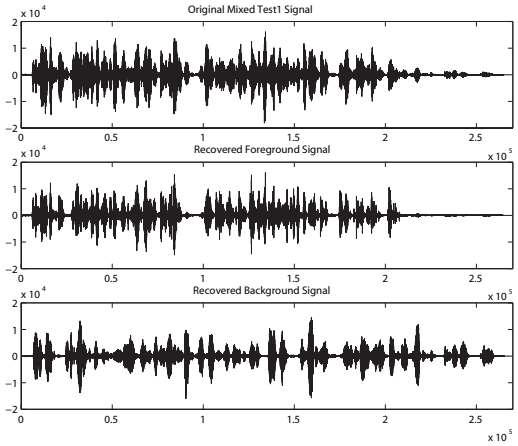

Figure 3.7: Input and output signals from complete system using factorial models and factorial processing.

two factors: relative distance to the speakers positions on the training signal and whether the test and training utterances are generated by the same speakers or not. These values are given in the table just above the “sp*/sp*” labels. The numbers correspond to the relative distance between the training and testing positions measured in meters. Labels ”‘YES/NO”’ are used to indicate if the identity of the speakers on the test mixture is the same as in the training mixture (”‘YES”’) or not. The first set of test signals has 0.0 relative distance from the locations of the speakers in the training utterances, and is generated by the same speakers as in the training utterances. Notice that for this test mixture the filter tuned to Speaker1 does a remarkable job, retrieving a weak background speaker. with a dramatic improvement of over +46dB on the estimated signal. This degree of separation is accomplished for all other tested mixtures with the same position and identity of speakers as in the training set, regardless of whatever is actuallly said in the mixture. This shows that the filters are well able to generalize to other utterances by the same speakers in the same location.

Test signal set 2 is also composed from utterances by the same speakers. However their relative position with respect to the speaker positions in the training signals has a distance of 1.48m. The separation results are still good, showing that the filters are

relatively robust to moderate fluctuations in speaker position. Test signal set 3 is also composed from utterances by the same speakers as in the training signal but the speaker positions were swapped. This had drastic influence on separation performance: here the filter sets for both speakers retrieved the signal from the foreground speaker. The last set of test signals corresponds to two different speakers placed in the same positions as in the training sequences. In this test both filter sets retrieved the signal from the background speaker.

	Training		Test1	
System Type	0.00m, yes		0.00m, yes	
	Sp1/Sp2	Sp2/Sp1	Sp1/Sp2	Sp2/Sp1
Delay&Sum	-11dB	+12dB	-11dB	+12dB
Filter&Sum	+36dB	+24dB	+35dB	+23dB

Test2		Test3		Test4	
1.48m, yes		2.54m, yes		0.00m, no	
Sp1/Sp2	Sp2/Sp1	Sp1/Sp2	Sp2/Sp1	Sp1/Sp2	Sp2/Sp1
-12dB	+13dB	-12dB	+14dB	+2dB	+1dB
+34dB	+18dB	-40dB	+29dB	+46dB	-8dB

Table 3.1: SSRs obtained for different test signals. For the training signal, sp1 represents the background speaker, and sp2 the foreground speaker.

	Training		Test	
System Type	0.00m, yes		0.00m, yes	
	Sp1/Sp2	Sp2/Sp1	Sp1/Sp2	Sp2/Sp1
Delay&Sum	+1dB	+1dB	+1dB	+1dB
Filter&Sum	+22dB	+18dB	+21dB	+18dB

Table 3.2: SSRs obtained for a mixture with similar energies and similar characteristics

The results suggest that the filters learn both speaker specific frequency characteristics, as well as the spatial characteristics of the speakers. Also, for a given set of speakers, the estimated filters are relatively robust to small variations in speaker location.

Table 3.2 shows the results for another set of speakers. Here the energy of the speakers in the original mixtures is very similar (sp1/sp2 = 1dB and sp2/sp1 = 1dB for

the delay and sum approach). Moreover the characteristics of the speakers are very similar as well. (Both are male speakers with very similar pitch). The system is still able to separate the speakers since the constraints imposed by the speech models are focused in what its being said rather than in the characterisitcs of the individual speakers and because the generic speech recognizer encodes speaker independent features.

3.5 Summary and Conclusions

We have presented a model-based extension of the conventional multimicrophone Source Separation of audio mixtures. The system works well under reverberant conditions, given that the objective function of our approach involves both the separation and the dereverberation of the individual sources. A critical difference when compared with the objective function of ICA approaches that only involves the independence factor, which factors in the poor performance of those systems under those conditions.

It is important to mention that even though the generic speech recognizer encodes speaker independent features, which will constitute a very coarse model of a given person's speech, the constraints imposed by the model are enough to guide the system towards the right set of filter coefficients to achieve separation.

It is also worth mentioning that the system is able to separate mixtures from very similar speakers, in part precisely because of the coarseness of the models using during training, given that the models are tunned by the speech content rather than by subtle differences in the spaeker voices or speech styles.

As with any other EM learning approach, inference and learning of this system will face problems with local minima, which for some mixtures may result in suboptimal values for the filter coefficients resulting in a suboptimal separation of the sources. Standard techniques to avoid local minima such as annealing or several random initializations can be applied on this system as well.

The need of an array of microphones, make the applicability of multimicrophone approaches in more natural scenarios very impractical. Also, many commercial audio signals such as soundtracks and music are available only as single-channel signals. Therefore, there is the need to develop systems that can perform audio source separation from a single recording of a mixed signal. The rest of the thesis is devoted to this goal. The following chapter presents a brief discussion of the previous related work in the area.

Chapter 4

Introduction to Single-Channel Source Separation

Multi-microphone approaches fit well in meeting scenarios where the dimensions of the room are well known and there is a limited number of possible positions for the speakers. This permits an optimal set up for the microphone array such that a good coverage of all the speakers could take place regardless of their actual positions. Moreover, since the speakers rarely move widely around the room, the scenario is a good match to the spatial contraints imposed by microphone-speaker specific filters.

Meeting scenarios are reverberant by nature, since conference rooms are generally just big enough to enclose a table sorrounded by chairs, demanding then, the use of multiple different observations to better cope with the situation.

However, the need for an array of microphones, makes the applicability of multimicrophone approaches in more natural scenarios very impractical. Also, many commercial audio signals such as soundtracks and music are available only as single-channel signals. Therefore, there is the need to develop systems that can perform audio source separation from a single recording of a mixed signal.

For model-based source separation, the restriction to a single channel translates to

a demand for greater detail in the model than was required for model-guided separation from multiple channels. This is because in the single channel case the model has a greater role in the separation process itself, rather than simply identifying a good output from a distinct filter-and-sum separation engine. In the previous chapter, the speech model, taken from a speaker-independent speech recognizer, constituted a very coarse model of any particular individual's speech, yet it was sufficient to provide training targets to learn the filter coefficients for the filter-and-sum array of each independent source. Once those coefficients are learned, the filter-and-sum arrays, and not the models, are the ones that perform the separation of new mixtures.

In single-channel source separation systems (e.g. Roweis (2000)), separation is typically achieved by selecting individual time-frequency cells that are dominated by the target source (described below), meaning that the model must pinpoint the target signal down to this level of detail.

Also, since most commercial single-channel audio signals are recorded in anechoic environments reverberance is not a salient feature in this scenario.

In the next section we quickly review recent work done on Single-Channel Source Separation. There we also stress some of the challenges encountered in this task.

4.1 Related Previous Work

In (Roweis 2000), detailed log-spectral models of speech are used to separate combined speech signals using the "refiltering" and "log-max" techniques using only one microphone. The idea behind this approach is that when two clean speech signals are mixed additively in the time domain, the log-spectrogram of the mixture is almost exactly the maximum of the individual log-spectrograms (Roweis 2003), i.e. given speech signals $x_1(t)$ and $x_2(t)$ with log-spectra X_1 and X_2 respectively, the mixed source $x_s(t)$ = $x_1(t)$ + $x_2(t)$ has a log spectrum approximated as $X_s \approx M \cdot X_1 + (1 - M) \cdot X_2$ where M

comes from the element-wise maximum-indicator operator applied to the individual log-spectrograms, $M = maxind(\mid X_1 \mid, \mid X_2 \mid)$, where

$$maxind(a, b) = \begin{cases} 1 & \text{when } a > b \\ 0 & \text{otherwise} \end{cases} \tag{4.1}$$

Refiltering recovery consists of estimating a mask M_{est} from the composed log-spectrogram and recovering the individual sources from a composed audio signal by assigning a weight to each time-frequency bin of the composed signal spectrogram, i.e.

$\hat{X_1} = M_{est} \cdot X_s$ and $\hat{X_2} = (1 - M_{est}) \cdot X_s$.

HMMs with 8000 full spectra states were built for two different speakers.

To analyze an unseen composed signal, the speaker models were combined into a two-chain factorial HMM (fig. 3.5), in which composed observations are formed from a combination of the individual states of each HMM. The emission probabilities are given by:

$$p(x_t \mid s_t^1 = i, s_t^2 = j) = \mathcal{N}(x, max(\mu_i^1, \mu_j^2), C) \tag{4.2}$$

where s_t^1 is the state of HMM 1, s_t^2 is the state of HMM 2, x_t is the composed observation, and $max(\cdot, \cdot)$ represents an element-wise-maximum operator.

Ideally, refiltering would be done by finding the factorial Viterbi path for the composed signals, which consists, at each frame, of the pair of states (one for each chain) that maximize the likelihood of the entire composed sequence. Given the pair of states $s_t^1 = i, s_t^2 = j$ from the factorial Viterbi path at time frame t, the mask m_t is found by applying the bitwise maximum operator to the means of the Viterbi states, $m_t = maxind(\mu_i, \mu_j)$. In practice, the true Viterbi path cannot be calculated due to the combinatorial explosion in the size of the factorial state space $N^2 = 8000^2 = 6.4 \times 10^7$. In (Roweis 2000), a limited set of factorial states with the highest observation likelihood at each time frame are used to perform Viterbi decoding on a limited grid. This approach,

however, does not guarantee that the solution found has the highest likelihood on the complete composed chain since it has a strong bias toward the observation probability.

In this approach, the HMMs are intended to cover the full range of different distinct short-term spectra with adequate resolution, a situation that requires the use of a large number of states. A more practical approach would be to design models that, as in this proposal and the following discussed previous work, explicitly factorize the sources of variability such that fewer parameters are required to model the signal with adequate resolution.

Kristjansson et al. (Kristjansson et al. 2004) also used pretrained models of the expected speakers in the form of gaussian mixtures models to perform the separation using a more complex composed speech model.

In (Hershey and Casey 2001), the large state space needed by full log spectra models is factored in wide and narrow-band components. The wide-band component is derived by low-pass filtering the log-spectra, and the narrow-band component is derived by high pass filtering the log spectra. The full log spectrogram is the sum of the two.

The wide and narrow-band components are represented in submodels with independent dynamics. The submodels are implemented using HMMs which are trained independently. The full log spectra models are then defined as a two-chain factorial hidden Markov model.

The composed speech of two speakers is modeled by combining the wide/narrow band models of each speaker, as illustrated in figure 4.1.

Researchers who have implemented this approach have commented that independent training of the narrow and wide-band components, and then combining them to produce the full spectra representation of the person's speech, frequently creates a problem in which the combination of the components result in non-natural, non representative full spectra representations of the person's speech. Introducing dependencies between the two components may reduce the combinatorial problem.

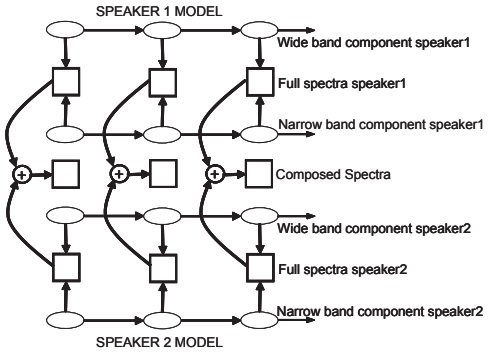

Figure 4.1: Composed speech of two speakers using wide/narrow band models.

In the following chapters the speech models are composed in the spectra feature domain (rather than in the log spectra one), where the relationship between single and composed models is approximately linear. The refiltering mask for speaker 1, M_t^1 in this case is estimated by:

$$M_t^1 = \frac{E[S_t^1]}{E[S_t^1] + E[S_t^2]} \tag{4.3}$$

where $E[S_t^1]$ and $E[S_t^2]$ are the expected value of the amplitude of each model at frame t. In the cases where $E(S_t^1)$ is either much larger or much smaller than $E(S_t^2)$, this reduces to the binary masks of eqn. 4.3.

A completely different approach was presented in (Bach and Jordan 2004a), where the blind source separation with a single microphone is framed as a spectral clustering problem. Each time frequency bin in the log-spectrogram representation of the mixture is assigned to a particular cluster. Thereby segmenting the spectrogram into clusters. The resultant clusters correspond to the individual log-spectrograms of sources in the mixtures.

A training session is required to choose the right parameters for the spectral clustering algorithm. Finding clusters among the set of all time-frequency bins requires huge matrices that pose significant numerical problems. Hence, the algorithm requires a great deal of time to separate short mixtures.

4.2 Summary

All the previous work reviewed in this section contitute valuable contributions to the problem solution. All of them have their pros and cons but they provide a good source of encouragement for the developement of this area of research.

Chapter 5

Multiband Model

Detailed models that capture the constraints implicit in a particular sound can be used to estimate obscured or corrupted portions from partial observations. This situation is encountered when trying to identify multiple, overlapping sounds.

The perceptual quality of sound sources may be broken down into two aspects. The first depends on the short-term spectral content of the sound, i.e. the energy in different frequency bands, as extracted by the resonant structures in the cochlea. This local spectrum is captured by the short-time Fourier transform (STFT) magnitude, the absolute value of the Fourier Transform of segments of the original sound localized in time with a short, shifting window, as captured in a regular spectrogram.

The second aspect of sound quality, visible in the spectrogram, is the evolution of this spectral structure in *time*. Hidden Markov models (HMMs) can compactly represent sequential constraints. In its simplest form, an HMM sound model defines a vocabulary of discrete states to approximate all the possible signal characteristics within a frame. A state-to-state transition matrix completes the model definition.

Given enough states, a sound model of arbitrary accuracy can, in theory, be produced.

These detailed HMMs of audio signals can be used to separate acoustic mixtures

between the sources by searching for combinations of state sequences that give the greatest agreement with combined observations, as described in chapter 4.

Good separation, however, requires detailed source models — for instance, a model of a particular speaker's voice might require several thousand states to cover the full range of different short-term spectra with adequate resolution. To address the tractability problems of such large models, we break the source signals into multiple frequency bands, and build separate, but coupled, HMMs for each band, requiring many fewer states per model. We show that these grid-like models allow a more effective deployment of model parameters, and for comparable computation expense, can achieve more accurate signal separations than full-band models, for instance by improving the SNR of mask-based resynthesis. Subband modeling also presents an interesting basis for learning source models directly from mixed signals, since there are more opportunities for unobstructed views of individual subbands than of the complete spectrum.

5.1 The Graphical Model

The principal limitation for the implementation of detailed audio model with traditional HMMs is the need of a large number of spectral parameters. For instance, in Roweis (2000) HMMs are used to build detailed audio models requiring 8000 states of dimension 513 to accurately represent the speech signals. The total number of spectral parameters used in such models is $8000 \times 513 = 4,104,000$. Such a large number of parameters presents many challenges during both learning and inference.

Rather than using a monolithic state to represent the spectrum, we propose dividing the spectral representation into multiple frequency *bands* i.e. multiple parallel horizontal sections of the spectrogram, as shown in figure 5.1 b, and then use separate HMMs in each band with many fewer states. Factorizing the complete spectrogram in this way, we could represent a large number of full spectral configurations with substan-

tially fewer parameters making inference and learning more feasible. For instance, if we divide a 513 dimension full spectrum into 19 equal bands of dimension 27, with 30 states per band. We can potentially represent $30^{19} = 1.16 \times 10^{28}$ full spectrum states, using only 15,390 spectral parameters. Training each band model independently without any constraints between bands (as in the multiband speech models used in Mirghafori. (1998) and Bourlard and Dupont (1997)), will result in many frames with unnatural combinations of band states that are not representative of the speaker.

To prevent this and to enforce consistency within and between bands, we couple adjacent bands in such a way that at any given frame the state in each band is determined by the previous states in that band as well as the two adjacent bands. In other words, the correlation between adjacent frequency bands is replaced by the dependency between their states.

Therefore, the proposed model consists of a series of HMMs, one for each frequency subband. The HMM for the k^{th} subband is coupled with HMMs for the $k - 1^{th}$ and $k + 1^{th}$ subbands to form a grid-like model, as illustrated in figure 5.1 (c).

The model can be used with different feature spaces and with different ways to partition the full band domain. For example, we could use a two chain version of the multiband model using the features and the partition proposed in (Hershey and Casey 2001). (Figure 5.2).

The model hidden variables are defined as $S = (s_1^1, ., s_T^1, s_1^2, ., s_T^2, ., s_1^K, ., s_T^K)$ and

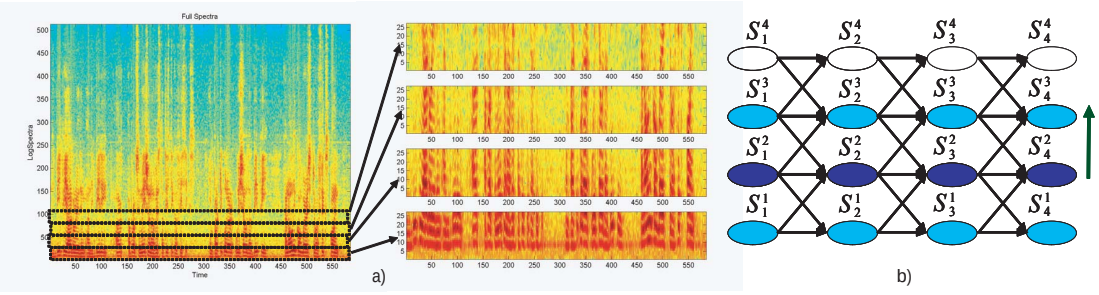

Figure 5.1: a) Full Spectrogram b) Spectrogram Partition and c) multiband model.

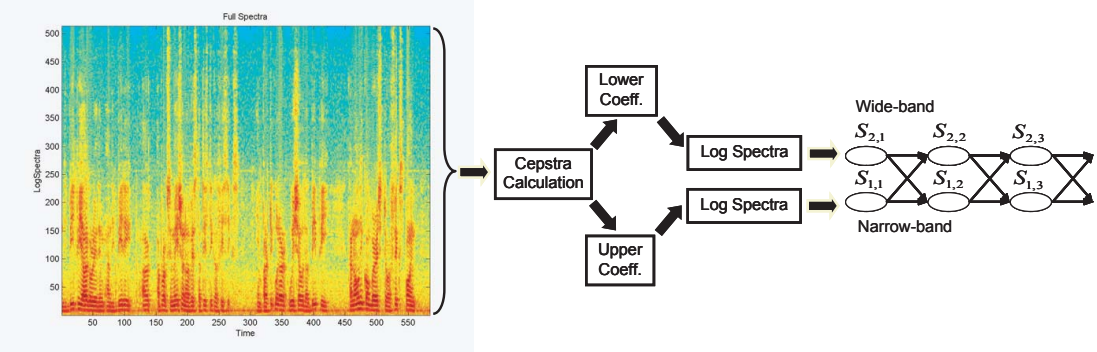

Figure 5.2: Multiband model for the wide/narrow band partition.

the observations (not shown) are defined as $X = (x_1^1, ., x_T^1, x_2^2, ., x_T^2, ., x_1^K, ., x_T^K)$ where s_t^k and x_t^k, represent the state and the observation at time t at frequency band k.

The joint probability for this model is given by:

$$p(S, X) = \prod_{k=1}^{K} \prod_{t=1}^{T-1} p(s_t^k \mid s_{t-1}^k, s_{t-1}^{k-1}, s_{t-1}^{k+1}) \prod_{k=1}^{K} \prod_{t=1}^{T} p(x_t^k \mid s_t^k) \tag{5.1}$$

Its parameters θ_k for each band k are defined by the transition probabilities $p(s_{t+1}^k \mid s_t^k, s_t^{k-1}, s_t^{k+1})$, and the means μ_j^k and variances Σ_j^k for single gaussian local output probabilities:

$$p(x_t^k \mid s_t^k = j) = \mathcal{N}(x_t^k, \mu_j^k, \Sigma_j^k) \tag{5.2}$$

5.1.1 Inference

Hidden variables S form a graphical model with high treewidth, therefore finding the exact posterior probability $p(S \mid X)$ requires an "averaging" function with the form:

$$q(S \mid X) = q(s_t^k \mid s_{t-1}^k, s_{t-1}^{k-1}, s_{t-1}^{k+1} \mid X) \tag{5.3}$$

and the minimization of the corresponding free energy:

$$\mathcal{F}(q,p) = \sum_{S} q(S)\log q(S) - \sum_{k=1}^{K}\sum_{t=1}^{T}\sum_{s_t^k} q(s_t^k)\log p(x_t^k \mid s_t^k)$$
$$- \sum_{k=1}^{K}\sum_{t=1}^{T-1}\sum_{s_t^{k-1},s_t^k,s_t^{k+1},s_{t+1}^k} q(s_t^{k-1}, s_t^k, s_t^{k+1}, s_{t+1}^k)\log p(s_{t+1}^k \mid s_t^k, s_t^{k-1}, s_t^{k+1}) \quad (5.4)$$

Optimizing eqn. (5.4) with respect to $q(S)$ is computationally intractable due to the large variable space, since exact inference will require forward/backward recursions with a variable space of N^K values per frame. (Assuming that all HMMs have N states).

Therefore we approximate the inference procedure by restricting the auxiliary function to have the following form:

$$q(S \mid X) = \prod_{k=1}^{K} q^k(S^k \mid X) = \prod_{k=1}^{K} q^k(s_1^k, .., s_T^k \mid X) \quad (5.5)$$

The free energy for this choice of "averaging" function is:

$$\mathcal{F}(q,p) = \sum_{k=1}^{K}\sum_{S^k} q(S^k)\log q(S^k) - \sum_{k=1}^{K}\sum_{t=1}^{T}\sum_{s_t^k} q^k(s_t^k)\log p(x_t^k \mid s_t^k)$$
$$- \sum_{k=1}^{K}\sum_{t=1}^{T-1}\sum_{s_t^{k-1},s_t^k,s_t^{k+1},s_{t+1}^k} q(s_t^{k-1})q(s_t^{k+1})q(s_t^k, s_{t+1}^k)\log p(s_{t+1}^k \mid s_t^k, s_t^{k-1}, s_t^{k+1}) \quad (5.6)$$

The variational parameters q^k, eliminate the vertical dependencies, while the influence of adjacent bands is retained through the optimization of the parameters, as is evident from eqn. (5.6).

In order to treat each band as a conventional HMM, we aim to optimize q^k and θ_k for the k_{th} band while keeping the parameters of the all other bands fixed. To isolate the terms in eqn. (5.6) that contain parameters for the k_{th} subband, we can write $\mathcal{F}(q,p)$

as the sum of two mutually exclusive terms $\mathcal{F}^k(q,p)$ and $\mathcal{F}^{k^{\neq}}(q,p)$, where the first term contains all the terms from (5.6) relevant to the k_{th} band. Noticing that state s_t^k is a variable in functions $p(s_{t+1}^{k+1} \mid s_t^{k+1}, s_t^k, s_t^{k+2})$ and $p(s_{t-1}^{k+1} \mid s_t^{k-1}, s_t^{k-2}, s_t^k)$.

$\mathcal{F}^k(q,p)$ can be expressed as:

$$\begin{aligned}
\mathcal{F}^k(q,p) &= \sum_{S^k} q(S^k)\log q(S^k) - \sum_{t=1}^{T}\sum_{s_t^k} q^k(s_t^k)\log p(x_t^k \mid s_t^k) \\
&- \sum_{t=1}^{T-1}\sum_{s_t^{k-2},s_t^{k-1},s_t^k,s_{t+1}^{k-1}} q(s_t^{k-2})q(s_t^k)q(s_t^{k-1},s_{t+1}^{k-1})\log p(s_{t+1}^{k-1} \mid s_t^{k-1}, s_t^{k-2}, s_t^k) \\
&- \sum_{t=1}^{T-1}\sum_{s_t^{k-1},s_t^k,s_t^{k+1},s_{t+1}^k} q(s_t^{k-1})q(s_t^{k+1})q(s_t^k,s_{t+1}^k)\log p(s_{t+1}^k \mid s_t^k, s_t^{k-1}, s_t^{k+1}) \\
&- \sum_{t=1}^{T-1}\sum_{s_t^k,s_t^{k+1},s_t^{k+2},s_{t+1}^{k+1}} q(s_t^k)q(s_t^{k+2})q(s_t^{k+1},s_{t+1}^{k+1})\log p(s_{t+1}^{k+1} \mid s_t^{k+1}, s_t^k, s_t^{k+2}) \quad (5.7)
\end{aligned}$$

Which we further express as:

$$\begin{aligned}
\mathcal{F}^k(q,p) &= \sum_{S^k} q(S^k)\log q(S^k) - \sum_{t=1}^{T-1}\sum_{s_t^k,s_{t+1}^k} q(s_t^k,s_{t+1}^k)\overline{\log}p(s_{t+1}^k \mid s_t^k, s_t^{k-1}, s_t^{k+1}) \\
&-\sum_{t=1}^{T}\sum_{s_t^k} q^k(s_t^k)\left[\log p(x_t^k \mid s_t^k) + \overline{\log}p(s_{t+1}^{k-1} \mid s_t^{k-1}, s_t^{k-2}, s_t^k) + \overline{\log}p(s_{t+1}^{k+1} \mid s_t^{k+1}, s_t^k, s_t^{k+2})\right]
\end{aligned} \quad (5.8)$$

and

$$\begin{aligned}
\mathcal{F}^k(q,p) &= \sum_{S^k} q(S^k)\log q(S^k) - \sum_{t=1}^{T-1}\sum_{s_t^k,s_{t+1}^k} q(s_t^k,s_{t+1}^k)\overline{\log}p(s_{t+1}^k \mid s_t^k, s_t^{k-1}, s_t^{k+1}) \\
&\qquad - \sum_{t=1}^{T}\sum_{s_t^k} q^k(s_t^k)\hat{\log}p(x_t^k \mid s_t^k) \quad (5.9)
\end{aligned}$$

where

$$\overline{\log}p(s_{t+1}^{k} \mid s_t^k, s_t^{k-1}, s_t^{k+1}) = \sum_{t=1}^{T-1} \sum_{s_t^{k-1}, s_t^{k+1}} q(s_t^{k-1}) q(s_t^{k+1}) \log p(s_{t+1}^{k} \mid s_t^k, s_t^{k-1}, s_t^{k+1}) \tag{5.10}$$

$$\overline{\log}p(s_{t+1}^{k-1} \mid s_t^{k-1}, s_t^{k-2}, s_t^{k}) = \sum_{t=1}^{T-1} \sum_{s_t^{k-2}, s_t^{k-1}, s_{t+1}^{k-1}} q(s_t^{k-2}) q(s_t^{k-1}, s_{t+1}^{k-1}) \log p(s_{t+1}^{k-1} \mid s_t^{k-1}, s_t^{k-2}, s_t^{k}) \tag{5.11}$$

$$\overline{\log}p(s_{t+1}^{k+1} \mid s_t^{k+1}, s_t^{k}, s_t^{k+2}) = \sum_{t=1}^{T-1} \sum_{s_t^{k+1}, s_t^{k+2}, s_{t+1}^{k+1}} q(s_t^{k+2}) q(s_t^{k+1}, s_{t+1}^{k+1}) \log p(s_{t+1}^{k+1} \mid s_t^{k+1}, s_t^{k}, s_t^{k+2}) \tag{5.12}$$

$$\hat{\log}p(x_t^k \mid s_t^k) = \log p(x_t^k \mid s_t^k) + \overline{\log}p(s_{t+1}^{k-1} \mid s_t^{k-1}, s_t^{k-2}, s_t^{k}) + \overline{\log}p(s_{t+1}^{k+1} \mid s_t^{k+1}, s_t^{k}, s_t^{k+2}) \tag{5.13}$$

The term $\overline{\log}p(s_{t+1}^{k} \mid s_t^k, s_t^{k-1}, s_t^{k+1})$ (eq. 5.10) can be interpreted as the "expected" log-transition matrix at frame t for the HMM for band k given the "state" of the HMMs of adjacent bands, $k-1$ and $k+1$.

Term $\overline{\log}p(s_{t+1}^{k-1} \mid s_t^{k-1}, s_t^{k-2}, s_t^{k})$ (eq. 5.11) can be interpreted as the "expected" log-state probability at frame t for the HMM for band k given the "state" of the HMMs of its two closest lower bands, $k-1$ and $k-2$.

Term $\overline{\log}p(s_{t+1}^{k+1} \mid s_t^{k+1}, s_t^{k}, s_t^{k+2})$ (eq. 5.12) can be interpreted as the "expected" log-state probability at frame t for the HMM for band k given the "state" of the HMMs of its two closest upper bands, $k+1$ and $k+2$.

Term $\hat{\log}p(x_t^k \mid s_t^k)$ (eq. 5.13) can be conceptualized as the output log-probability for observation x_t^k weigthed by the "state" of the four most adjacent bands, $k-2, k-1, k+1$ and $k+2$.

Comparing the function defined in eqn. 5.9,with the function defined in eqn. 2.10, the auxiliary function for an individual HMM, we can observe that they have the same form an therefore they can be maximized in the same way.

Thus, we estimate optimal $q^k(S^k)$ probabilities using the forward/backward recursions defined in eqns. 2.15-2.19 with $\exp\hat{\log}p(x_t^k \mid s_t^k)$ as the local output probabilities and $\exp^{\overline{\log}p(s_{t+1}^k|s_t^k,s_t^{k-1},s_t^{k+1})}$ as the transition probability. i.e. $p(Y_t \mid X_t)$ and $p(X_{t+1} \mid X_t)$ in eqns. 2.15–2.19.

As it is evident from the previous equations, even though the variational approximation decouples the posteriors of the individual band HMMs and that the posteriors of each band are estimated using the forward/backward recursions as in a regular HMM. The HMMs in adjacent bands are coupled in different ways.

First, notice that unlike regular HMMs, where the system is regarded as *stationary*, the transition matrices used on each band HMM to estimate its posteriors are *not stationary*, due to the influence of the "state" of the adjacent bands.

Moreover, the four most adjacent bands have a direct say on the output log-probability of the observations for a given band by means of eq. 5.13.

5.1.2 Learning

The M step consist in maximizing eq. (5.6), with respect to the model parameters, resulting in the following update formulas for the parameters in each band.

$$p(s_{t+1}^k \mid s_t^k, s_t^{k-1}, s_t^{k+1}) = \frac{\sum_{t=1}^{T-1} q^k(s_{t+1}^k, s_t^k) q^{k-1}(s_t^{k-1}) q^{k+1}(s_t^{k+1})}{\sum_{t=1}^{T-1} q^k(s_t^k) q^{k-1}(s_t^{k-1}) q^{k+1}(s_t^{k+1})} \tag{5.14}$$

$$\mu_i^k = \frac{\sum_{t=0}^{T} q^k(s_t^k = i) x_t^k}{\sum_{t=0}^{T} q^k(s_t^k = i)} \tag{5.15}$$

$$\Sigma_i^k = \frac{\sum_{t=0}^{T} q^k(x_t^k = i)(x_t^k - \mu_i^k)(x_t^k - \mu_i^k)'}{\sum_{t=0}^{T} q^k(s_t^k = i)} \tag{5.16}$$

5.1.3 Training Procedure

Each band's HMM is trained as a single HMM once eqns. (5.10-5.13) have been calculated. To calculate these terms, however, we need to know or have 'observed' the variational posteriors of the adjacent bands. For instance, if we want to train the HMM of the darkest nodes in fig. 5.1b, we need to know the variational posteriors for the lightly-shaded nodes. Once we finish the training of the dark band, we can use its variational posteriors to estimate the variational posteriors of any of its adjacent bands. Therefore we need to establish a "schedule" for the inference/learning procedures of the different bands. Given that often in audio signals more information is contained in the lower frequencies than in the higher ones, we would like to emphasize the influence of the lowest bands on the rest of the spectrogram. Therefore we establish a schedule, such that, the variational parameters for the HMM corresponding to the lowest band is estimated first and progressively estimating the parameters of the adjacent upper band. The recently estimated variational posteriors of the lower bands are used to estimated the variational posteriors of the higher ones. For instance, once the variational posteriors for the HMM of the darkest nodes in fig. 5.1b have been estimated, we proceeed up in the model to train the next upper band, using the variational posteriors of the just-trained band.

We continue this procedure until we reach the highest band; a complete pass from the lowest to the highest is called an iteration of the multiband model.

For the first iteration, all the variational parameters are uniformly initialized, meaning that the two upper bands (only) in each HMM training in the first iteration

are being very poorly approximated as uniform.

We continue this process until the free energy of the complete model (eq. 5.6), stops decreasing.

The initialization of the model parameters is done by estimating regular independent HMMs for each band. The mean and variances for the coupled HMMs are intialized to the values obtained from those independent HMMs. Parameters $p(s_{t+1}^k \mid s_t^k, s_t^{k-1}, s_t^{k+1})$ are obtained by replicating the independent HMMs transition matrices, π^k; .i.e. $\forall\, k, t, s_t^{k-1}, s_t^{k+1}\; p(s_{t+1}^k = j \mid s_t^k = i, s_t^{k-1}, s_t^{k+1}) = \pi_{ij}^k$

The procedure is summarized in the following algorithm.

Speaker Multiband Models Training Algorithm

1. Estimated regular independent HMMs for each band.
2. Initialized multiband model parameters using the parameters from the independent HMMs.
3. $\tau = 0$; τ = number of complete iterations.
4. $\forall\, k, t$ Initialize variational posteriors $q_\tau^k(s_t^k)$ = 1/N;
5. Compute $\mathcal{F}_\tau(q, p)$ through eqn. (5.6).
6. $k = 1$; Using terms defined on eqns. (5.10-5.13), (excluding terms referring to bands $k-2$ and $k-1$) and the variational posteriors $q_\tau^{k+1}(S^k)$ and $q_\tau^{k+2}(S^k)$. Run forward/backward recursions to estimate $q_{\tau+1}^k(S_t^k)$.
7. $k = 2$; Using terms defined on eqns. (5.10-5.13), (excluding terms referring to band $k-1$) and the variational posteriors $q_{\tau+1}^{k-1}(S^k)$, $q_\tau^{k+1}(S^k)$ and $q_\tau^{k+2}(S^k)$. Run forward/backward recursions to estimate $q_{\tau+1}^k(S^k)$.
8. For $k = 2 : K$;

Using terms defined on eqns. (5.10-5.13), and the variational posteriors $q_{\tau+1}^{k-2}(S^k)$, $q_{\tau+1}^{k-1}(S^k)$,$q_{\tau}^{k+1}(S^k)$ and $q_{\tau}^{k+2}(S^k)$.

Run forward/backward recursions to estimate $q_{\tau+1}^{k}(S^k)$.

9. Update model parameters through update formulas (5.14-5.16).

10. $\tau = \tau + 1$; Compute $\mathcal{F}_\tau(q,p)$ through eqn. (5.6).

11. if $\mathcal{F}_\tau(q,p)$ - $\mathcal{F}_{\tau+1}(q,p) > \epsilon$. Return to step 6

5.2 Factorial Multiband Model

The multiband model introduced in section 5 can be used to implement detailed models of audio sources with a relatively small number of spectral parameters. The idea is to use these models to "guide" the separation of audio mixtures of the independant sources. Therefore, we need a way to compose the individual models to account for the mixed signals. As before we use a factorial structure to compose the multiband models, resulting in a factorial multiband model, (figure 5.3).

For simplicity we refer for the case of two speakers, m and n.

Now, the model hidden variables are defined as $S = [s_1^{m,1}, ., s_{m,T}^K, s_1^{n,1}, ., s_{n,T}^K]$ and the observations that account for the composed speech are defined as: $X = [x_1^1, ., x_T^1, x_2^2, ., x_T^2, ., x_1^K, ., x_T^K$

The joint probability for this model is given by:

$$p(S,X) = \prod_{r=m,n}\prod_{k=1}^{K}\prod_{t=1}^{T-1} p(s_{r,t}^k \mid s_{r,t-1}^k, s_{r,t-1}^{k-1}, s_{r,t-1}^{k+1}) \prod_{k=1}^{K}\prod_{t=1}^{T} p(x_t^k \mid s_{m,t}^k, s_{n,t}^k) \quad (5.17)$$

The goal of this model is to find those individual states in each band for each speaker that, when composed, better fit the mixed signal. Therefore we are not interested in optimizing any of the models parameters, we are just interested in finding the most likely states for the model, i.e. $[\hat{s}_{m,t}^k, \hat{s}_{n,t}^k]$ (eqn. 5.18) given the composed data.

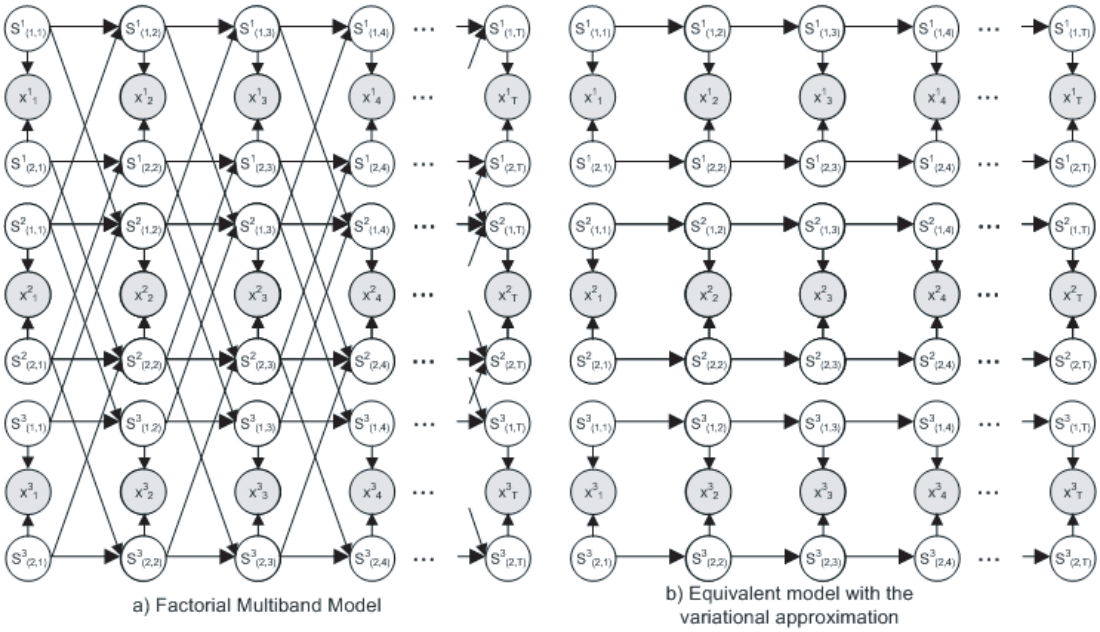

Figure 5.3: a) Factorial Multiband Model b) The proposed variational approximation decouples the factorial HMMs (FHMM) into individual FHMMs per band.

$$[\hat{s}^k_{m,t}, \hat{s}^k_{n,t}] = maxarg_{s^k_{m,t}, s^k_{n,t}} p(s^k_{m,t}, s^k_{n,t} \mid X)) \tag{5.18}$$

But exact inference on the model is again computationally intractable. Therefore we approximate the inference procedure by restricting the auxiliary function to have the following form:

$$q(S \mid X) = \prod_{k=1}^{K} q^k(S^k \mid X) = \prod_{k=1}^{K} q^k(s^k_{m,1}, .., s^{m,k}_T, s^k_{n,1}, .., s^{n,k}_{n,T} \mid X) \tag{5.19}$$

Which is equivalent to decoupling the model in figure 5.3 a) into a series of factorial HMMs, one per band. 5.3 b).

The free energy for this choice of $q(S)$ is

$$\mathcal{F}(q,p) = \sum_{k=1}^{K}\sum_{S^k} q(S^k)\log q(S^k) - \sum_{k=1}^{K}\sum_{t=1}^{T}\sum_{s^k_{m,t},s^k_{n,t}} q^k(s^k_{n,t},s^k_{n,t})\log p(x^k_t \mid s^k_{m,t},s^k_{n,t})$$
$$- \sum_{r=m,n}\sum_{k=1}^{K}\sum_{t=1}^{T-1}\sum_{s^{k-1}_{r,t},s^k_{r,t},s^{k+1}_{r,t},s^k_{r,t+1}} q(s^{k-1}_{r,t})q(s^{k+1}_{r,t})q(s^k_{r,t},s^k_{r,t+1})\log p(s^k_{r,t+1} \mid s^k_{r,t},s^{k-1}_{r,t},s^{k+1}_{r,t}) \tag{5.20}$$

In order to treat each band as a factorial HMM, we aim to compute q^k for the k_{th} band while keeping the variational posteriors for the all other bands fixed. The procedure is again to express $\mathcal{F}(q,p)$ as the sum of two mutually exclusive terms $\mathcal{F}^k(q,p)$ and $\mathcal{F}^{k^{\neq}}(q,p)$ where the first term contains all the terms from (5.20) relevant to the k_{th} band.

Manipulating term $\mathcal{F}^k(q,p)$ in a similar way as in the single source case, we found:

$$\mathcal{F}^k(q,p) = \sum_{S^k} q(S^k)\log q(S^k) - \sum_{t=1}^{T}\sum_{s^k_{m,t},s^k_{n,t}} q^k(s^k_{m,t},s^k_{n,t})\hat{\log p}(x^k_t \mid s^k_{m,t},s^k_{n,t})$$
$$- \sum_{r=m,n}\sum_{t=1}^{T-1}\sum_{s^k_{r,t},s^k_{r,t+1}} q(s^k_{r,t},s^k_{r,t+1})\overline{\log p}(s^k_{r,t+1} \mid s^k_{r,t},s^{k-1}_{r,t},s^{k+1}_{r,t}) \tag{5.21}$$

The terms $\overline{\log p}(s^k_{r,t+1} \mid s^k_{r,t},s^{k-1}_{r,t},s^{k+1}_{r,t})$ for each speaker are calculated using eqn. (5.10), with the parameters and variational posteriors correspondent to each speaker.

Term $\hat{\log p}(x^k_t \mid s^k_{m,t},s^k_{n,t})$ now corresponds to a "weighted" output factorial log-probability where factors from each speaker like the ones on eqns. (5.11,5.12) are added to the regular factorial log-probability, $\log p(x^k_t \mid s^k_{m,t},s^k_{n,t})$.

$$\hat{\log p}(x^k_t \mid s^k_{m,t},s^k_{n,t}) = \log p(x^k_t \mid s^k_{m,t},s^k_{n,t}) +$$
$$\sum_{r=m,n}\left[\overline{\log p}(s^{k-1}_{r,t+1} \mid s^{k-1}_{r,t},s^{k-2}_{r,t},s^k_{r,t}) + \overline{\log p}(s^{k+1}_{r,t+1} \mid s^{k+1}_{r,t},s^k_{r,t},s^{k+2}_{r,t})\right] \tag{5.22}$$

Comparing the function defined in eqn. 5.21, with the function defined in eqn. 3.23, the free energy for a factorial HMM, we can observe that they have the same form.

As mentioned before, exact inference of a two-speakers Factorial HMM can be done through factorial-forward/backward recursions: $\alpha_F(s^k_{m,t}, s^k_{n,t})$ and $\beta_F(s^k_{m,t}, s^k_{n,t})$. This approach requires computing N^2 values for α_F, β_F at each frame t. In our previous use of a factorial HMM, the values of N were prohibitively large and therefore inference was approximated. However, given that in this case N is a small number since we are factorizing the state space into subbands, exact inference is feasible.

$$q(s^k_{m,t}, s^k_{n,t} \mid X^k) = \frac{\alpha(s^k_{m,t}, s^k_{n,t})\beta(s^k_{m,t}, s^k_{n,t})}{\sum_{i,j} \alpha(s^k_{m,t} = i, s^k_{n,t} = j)\beta(s^k_{m,t} = i, s^k_{n,t} = j)} \tag{5.23}$$

With:

$$\alpha(s^k_{m,t}, s^k_{n,t}) = \sum_{s^k_{m,t-1}, s^k_{n,t-1}} \left[\alpha(s^k_{m,t-1}, s^k_{n,t-1}) p(s^k_{m,t} \mid s^k_{m,t-1}) p(s^k_{n,t} \mid s^k_{n,t-1})\right] p(x^k_t \mid s^k_{m,t-1}, s^k_{n,t-1}) \tag{5.24}$$

$$\beta(s^k_{m,t}, s^k_{n,t}) = \sum_{s^k_{m,t+1}, s^k_{n,t+1}} \left[\beta(s^k_{m,t+1}, s^k_{n,t+1}) p(s^k_{m,t+1} \mid s^k_{m,t}) p(s^k_{n,t+1} \mid s^k_{n,t}) p(x^k_{t+1} \mid s^k_{m,t+1}, s^k_{n,t+1})\right] \tag{5.25}$$

As in the case of single source multiband models, computing the variational posteriors for a given band requires the knowledge of the variational posteriors of the adjacent bands. Therefore, here there is also the need for setting a schedule for the calculation of the variational posteriors on the different bands. And a few iterations through the entire factorial multiband model may be neccesary.

The estimation of the most likely sequence of band-factorial states, which we refer to as the Viterbi sequence is summarized in the following algorithm.

Algorithm for the Viterbi Sequence Estimation for the Factorial Mutiband Model

1. $\tau = 0$; τ = number of complete iterations.
2. $\forall\ k, t$ Initialize variational posteriors $q_\tau^k(s_{m,t}^k, s_{n,t}^k)$ = $1/N^2$;
3. Compute $\mathcal{F}_\tau(q,p)$ through eqn. (5.20).
4. $k = 1$; Using terms defined on eqns. (5.10-5.12) (one per speaker), the term defined on eqn. (5.22) and the variational posteriors $q_\tau^{k+1}(S^k)$ and $q_\tau^{k+2}(S^k)$. Run factorial forward/backward recursions (eqns 5.23-5.25 to estimate $q_{\tau+1}^k(S_t^k)$.
5. $k = 2$; Using terms defined on eqns. (5.10-5.12) (one per speaker), the term defined on eqn. (5.22) and the variational posteriors $q_{\tau+1}^{k-1}(S^k)$, $q_\tau^{k+1}(S^k)$ and $q_\tau^{k+2}(S^k)$. Run forward/backward recursions to estimate $q_{\tau+1}^k(S^k)$.
6. For $k = 2 : K$;

   Using terms defined on eqns. (5.10-5.12) (one per speaker), the term defined on eqn. (5.22) and the variational posteriors

   $q_{\tau+1}^{k-2}(S^k)$, $q_{\tau+1}^{k-1}(S^k)$,$q_\tau^{k+1}(S^k)$ and $q_\tau^{k+2}(S^k)$.

   Run forward/backward recursions to estimate $q_{\tau+1}^k(S^k)$.
7. $\tau = \tau + 1$; Compute $\mathcal{F}_\tau(q,p)$ through eqn. (5.20).
8. if $\mathcal{F}_\tau(q,p)$ - $\mathcal{F}_{\tau+1}(q,p) > \epsilon$. Return to step 4
9. Find $[\hat{s}_{m,t}^k, \hat{s}_{n,t}^k] = maxarg_{s_{m,t}^k, s_{n,t}^k} q(s_{m,t}^k, s_{n,t}^k \mid X)$

5.3 Experimental Results

We have applied the multiband model in a source separation application and we compared its performance against the one obtained with full band models.

We built multiband models for two speakers and combined them into a factorial model to explain new composed signals. The training procedure is done using a full set of factorial emission probabilities (4.2) since in each band our state space is considerably smaller than when using a single 'full spectrum' HMM. This makes the combinatorial problem less daunting, and variational inference in the complete factorial state space can be performed. Refiltering is done by estimating the mask for band k as the maximum-indicator between the ' expectations' of the state means for each chain under the variational parameters taken as posteriors, i.e.

$$M_t^k = maxind\left(\sum_j Q(S_t^{1k}=j)\cdot\mu_i^{1k}, \sum_j Q(S_t^{2k}=j)\cdot\mu_i^{2k}\right) \tag{5.26}$$

The variational parameters are obtained in a iterative process that may involve a few passes over the entire multiband model.

We built full-spectral HMMs with 1000 states to compare with multiband models with varying numbers of bands and coefficients per band. Subband models train much faster: Three EM iterations of the full-spectral HMMs for each speaker took over two weeks using the HTK software tools, whereas 20 iterations of the multiband model took in average 3 to 4 days using Matlab. The speaker models were tested in a refiltering source-separation task, where test samples of the two speakers were added together, and state sequences for each speaker were estimated via factorial HMM inference.

We quantify the degree of separation obtained by a given estimated mask, M_{est}, by measuring the Signal-to-Noise Ratio (SNR) of the resultant "separated" signals. The SNR for a given speaker measures the ratio of the content of the desired speaker versus the other speaker on the desired speaker "separated" output. When test signals are con-

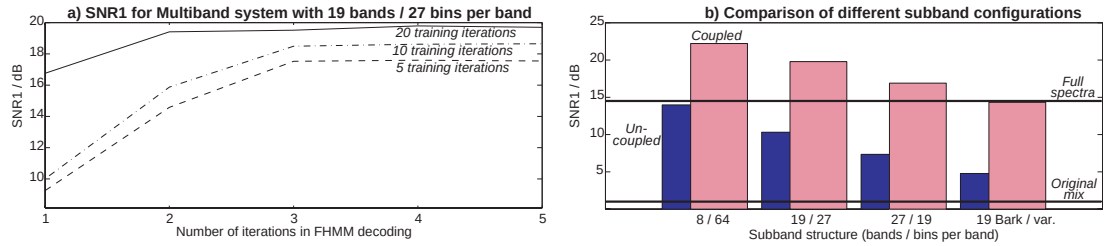

Figure 5.4: a) SNR1 for a 19-band system versus iterations in recognition and training. b) SNR1 for different structures of independent and coupled multiband systems, where the first bar in each pair corresponds to the independent model and the second to the proposed coupled model.

structed by artificially summing individual source signals, it is possible to identify the portions of the final filtered mixture that properly originate in each source by passing the individual sources through the same time-varying filter. Using these separately-filtered components, the signal-to-noise (SNR) ratio is obtained by treating each reconstruction as a corrupt version of the original target signal, i.e. for speaker 1:

$$SNR_1 = 10 \cdot log_{10} \frac{\sum_{t,k} |X_1^*|^2}{\sum_{t,k} M_{est} \cdot |X_2^*|^2 + (1 - M_{est}) \cdot |X_1^*|^2} \tag{5.27}$$

The noise denominator is obtained by direct subtraction of the original source from the refiltered output. This penalizes both inclusion of energy from the interference $(M_{est} \cdot |X_2^*|^2)$ as well as deletion of target energy, $((1 - M_{est}) \cdot |X_1^*|^2)$.

Examining equation 5.27 with more detail. Using variables $M_{est,t}^k$, $X_{1,t}^{*k}$ and $X_{2,t}^{*k}$ to represent the elements for the k_{th} frequency at frame t for variables M_{est}, X_1^* and X_2^*. For a particular time frequency bin the corresponding element on the denominator of eq. 5.27 is given by:

$$e_t^k = M_{est,t}^k \cdot |X_{2,t}^{*k}|^2 + (1 - M_{est,t}^k) \cdot |X_{1,t}^{*k}|^2 = M_{est,t}^k \cdot (|X_{2,t}^{*k}|^2 - |X_{1,t}^{*k}|^2) + |X_{1,t}^{*k}|^2 \tag{5.28}$$

The optimal value of $M_{est,t}^k$ under the model is one when $X_{1,t}^{*k} \leq X_{2,t}^{*k}$, and zero otherwise. If $M_{est,t}^k$ is equal to one, then e_t^k on eq. 5.28 is equal to $|X_{2,t}^{*k}|^2$. If $M_{est,t}^k$ is

equal to zero, then e_t^k on eq. 5.28 is equal to $|X_{1,t}^{*k}|^2$. If the value of $M_{est,t}^k$ is equal to its optimal value, the energy captured on e_t^k will correspond to the energy of the weaker speaker. In the other hand, if the value of $M_{est,t}^k$ is not equal to its optimal value, the energy captured on e_t^k will correspond to the energy of the dominant speaker punishing greatly then the SNR value computed on eq. 5.27.

X_1^* can be either X_1 (the magnitude-spectra of $x_s(t)$), or $M_{opt} \cdot X_s$, the magnitude-spectra obtained from the optimal mask $M_{opt} = maxind(X_1, X_2)$. We have observed that SNRs computed with the latter have a higher correlation with the perceptual quality of the separated signals. (Since M_{opt} is the best mask we can achieve under the model, in the sense of giving the best SNR against the original target, it also measures how close a given solution is to the best possible solution.)

Fig. 5.4a depicts SNR_1 values for 19 bands with 27 coefficients per band and 30 states per band, the vertical axis corresponds to the obtained SNR in dB, while the horizontal axis corresponds to the number of iterations performed on the factorial multi-band model. There are three traces, corresponding to the SNR values obtained using speaker models trained for 5 (dashed line), 10 (dotted-dashed line), or 20 (solid line) iterations. We obtained a higher SNR when using parameters trained with more iterations, showing the benefits of the coupling. Fig. 5.4b shows four pairs of bars, each pair corresponds to a multiband model with a different structure (8 bands/64 coefficients, 19/27, 27/19 and 19 Bark-spaced bands with between 6 and 128 bins). The first bar in each couple corresponds to the SNR obtained when the bands are trained independently, the second band corresponds to the SNR obtained by the proposed model. The higher horizontal black line (around 14 dB) corresponds to the SNR obtained by the 1000-state full-spectrum model with a 100-state limited Viterbi grid. The lower line (1.2 dB) show the SNR calculated for the original mixture. We note that training the coupled models gives a consistent SNR improvement of around 10 dB in all models, with fewer, larger subbands (e.g. 8 bands of 64 bins) performing best.

5.4 Summary and Conclusions

The model is capable of factorizing the large variability encountered in the full spectral representation of a person's speech into substates each with a substantially lower variability than the whole. There is a clear trade off in the partition of the full spectra into subbands between the amount of variability captured in each band and the permutation dilemma when the substates are regrouped to form a valid full spectral state. The subbands have to be large enough to capture a few harmonics, to alleviate the permutation problem, but small enough to keep the variability within the subband relatively low. Regardless of the nature of the partitions, coupling the subbands is a critical step to achieving the best performance given the partition.

Even factorizing the spectrogram in this way, each frame in the spectrogram is treated as an independent identity. However, speech and other natural sounds show high temporal correlation and smooth spectral evolution punctuated by a few, irregular and abrupt changes. Therefore, it would be more efficient and informative to model successive spectra as *transformations* of their immediate predecessors. The following chapter presents a model that exploits the correlation and self-similarity of those kind of signals to model them with detail with a relatively small number of parameters.

Chapter 6

The Deformable Spectrograms Model

Hidden Markov Models (HMMs) work best when only a limited set of distinct states need to be modeled, as in the case of speech recognition where the models need only be able to discriminate between phone classes. When HMMs are used with the express purpose of accurately modeling the full detail of a rich signal such as speech, they require a large number of states. In (Roweis 2000), HMMs with 8,000 states were required to accurately represent one person's speech for a source separation task. The large state space is required because it attempts to capture every possible instance of the signal. If the state space is not large enough, the HMM will not be a good generative model since it will end up with a "blurry" set of states which represent an average of the features of different segments of the signal, and cannot be used in turn to "generate" the signal.

In many audio signals including speech and musical instruments, there is a high correlation between adjacent frames of their spectral representation. Our approach consists of exploiting this correlation so that explicit models are required only for those frames that cannot be accurately predicted from their context. In (Bilmes 1998), context is used to increase the modeling power of HMMs, while keeping a reasonable size of parameter space, however the correlation between adjacent frames is not explicity modeled. Our model captures the general properties of such audio sources by model-

ing the evolution of their harmonic components. Based on the widely-used source-filter model for such signals, we devise a layered generative graphical model that describes these two components in separate layers: one for the excitation harmonics, and another for resonances such as vocal tract formants. This layered approach draws on successful applications in computer vision that use layers to account for different sources of variability Jojic and Frey (2001); N. Jojic and Kannan (2003); A. Levin and Weiss (2003). Our approach explicitly models the self-similarity and dynamics of each layer by fitting the log-spectral representation of the signal in frame t with a set of transformations of the log-spectra in frame $t-1$. As a result, we do not require separate states for every possible spectral configuration, but only a limited set of "sharp" (not blurry) states that can still cover the full spectral variety of a source via such transformations. This approach is thus suitable for any time series data with high correlation between adjacent observations.

6.1 Spectral Deformation Model

Many audio signals, including speech and musical instruments, have short-time spectral representations that show great similarity between temporally-adjacent frames over much of the signal. We propose a model that discovers and tracks the nature of such correlation by finding how the patterns of energy are transformed between adjacent frames and how those transformations evolve over time.

Figure 6.1 shows a narrow-band spectrogram representation of a speech signal, where each column depicts the energy content across frequency in a short-time window, or time-frame. The value in each cell is actually the log-magnitude of the short-time Fourier transform in decibels:

$$x_t^k = 20 \log \left(abs \left(\sum_{\tau=0}^{N_F-1} w[\tau] x[t \cdot H + \tau] e^{-j2\pi\tau k/N_F} \right) \right) \tag{6.1}$$

where t is the time-frame index, k indexes the frequency bands, N_F is the size of the discrete Fourier transform, H is the hop between successive time-frames, $w[\tau]$ is the

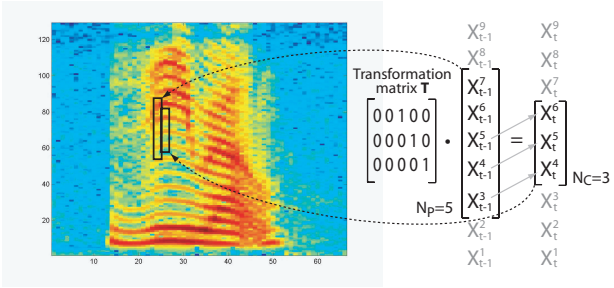

Figure 6.1: The $N_C = 3$ patch of time-frequency bins outlined in the spectrogram can be seen as an "upward" version of the marked $N_P = 5$ patch in the previous frame. This relationship can be described using the matrix shown.

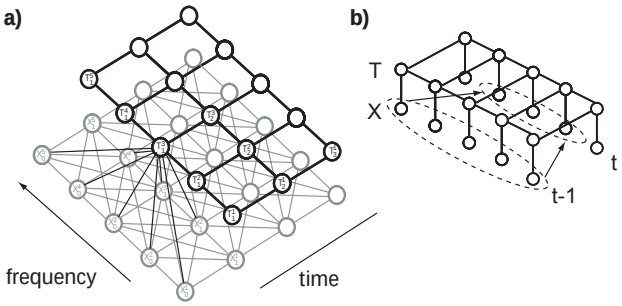

Figure 6.2: a) Graphical model b) Graphical simplification.

N_F-point short-time window, and $x[\tau]$ is the original time-domain signal. We use 32 ms windows with 16 ms hops. Using subscript C to designate current and P to indicate previous, the model predicts a patch of N_C time-frequency bins centered at the k^{th} frequency bin of frame t as a "transformation" of a patch of N_P bins around the k^{th} bin of frame $t-1$, i.e.

$$\vec{X}_t^{[k-n_C,k+n_C]} \approx \vec{T}_t^k \cdot \vec{X}_{t-1}^{[k-n_P,k+n_P]} \tag{6.2}$$

where $n_C = (N_C - 1)/2$, $n_P = (N_P - 1)/2$, and $\vec{T}_t^k$ is the particular $N_C \times N_P$ transformation matrix employed at that point on the time-frequency plane. Figure 6.1 shows an example with $N_C = 3$ and $N_P = 5$ to illustrate the intuition behind this approach. The selected patch in frame t can be seen as a close replica of an upward shift of part of the patch highlighted in frame $t-1$. This "upward" relationship can be captured by a transformation matrix such as the one shown in the figure. The patch in

frame $t-1$ is larger than the patch in frame t to permit both upward and downward motions. The proposed model selects, from a discrete set, the particular transformation that better describes the evolution of the energy from frame $t-1$ to frame t around every one of the time frequency bins x_t^k in the spectrogram. The patches used between adjacent time frequency bins overlap, which promotes transformation consistency (N. Jojic and Kannan 2003). The model also tracks the structure of the transformations throughout the whole signal to find useful patterns of transformation.

The generative graphical model is depicted in figure 6.2. Nodes $\mathcal{X} = \{x_1^1, x_1^2, ..., x_t^k, ..., x_T^K\}$ represent all the time-frequency bins in the spectrogram. For now, we consider the continuous nodes $\mathcal{X}$ as observed, although below we will allow some of them to be hidden when analyzing the missing-data scenario. Discrete nodes $\mathcal{T} = \{T_1^1, T_1^2, ..., T_t^k, ..., T_T^K\}$ index the set of transformation matrices used to model the dynamics of the signal. Each $N_C \times N_P$ transformation matrix $\vec{T}$ is of the form:

$$\begin{pmatrix} \vec{w} & 0\ 0 \\ 0 & \vec{w} & 0 \\ 0\ 0 & \vec{w} \end{pmatrix} \tag{6.3}$$

i.e. each of the N_C cells at time t predicted by this matrix is based on the same transformation of cells from $t-1$, translated to retain the same relative relationship. This approach enforces global consistency which is consistent with the strong spectral correlation shown between adjacent time frequency bins and prevents local minima issues.

Here, $N_C = 3$ and $\vec{w}$ is a row vector with length $N_W = N_P - 2$; using $\vec{w} = (0\ 0\ 1)$ yields the transformation matrix shown in figure 6.1. To ensure symmetry along the frequency axis, we constrain N_C, N_P and N_W to be odd. The complete set of $\vec{w}$ vectors include upward/downward shifts by whole bins as well as fractional shifts. An example

set of 13 vectors, containing each $\vec{w}$ vector as a row, is:

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & .25 & .75 \\ 0 & 0 & 0 & .75 & .25 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & .25 & .75 & 0 \\ 0 & 0 & .75 & .25 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & .25 & .75 & 0 & 0 \\ 0 & .75 & .25 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ .25 & .75 & 0 & 0 & 0 \\ .75 & .25 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{6.4}$$

The length N_W of the transformation vectors defines the supporting coefficients from the previous frame $\vec{X}_{t-1}^{[k-n_W,k+n_W]}$ (where $n_W = (N_W - 1)/2$) that can "explain" x_t^k.

We experimented with a wide range of transformations and we found that any set of transformations that includes "'pure shifts"' (i.e. rows 1,3,6,etc on matrix 6.1) as well as "'fractional shifts"' (i.e. rows 1,3,6,etc on matrix **??**) has a really similar performance as the one obtained using a set with only "'pure shifts"'. For instance the representational capabilities of a model with a set of transformations like the one in matrix is virtually the same as the one obtained with the set described by matrix . This is due the probabilistic nature of the model. For example a transformation like the one described by [0 0 0 .25 .75] with probabilty one is equivalent to a trasformation described by [0 0 0 1 0] and [0 0 0 0 1] with probabilities .25 and .75 respectively.

Therefore simple sets of transformation describing only "'pure shifts"' are enough to represent a wide variety of transformations.

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{6.5}$$

For harmonic speech signals sampled at 16 kHz and analyzed over 1024-point short-time windows (15 Hz bin resolution), we have found that a model using the above set of $\vec{w}$ vectors with parameters $N_W = 5$, $N_P = 9$ and $N_C = 5$ is very successful at capturing the self-similarity and dynamics of the harmonic structure.

The results presented in this paper are obtained using the *fixed* set of transformations described by the matrix in eqn. 6.1.

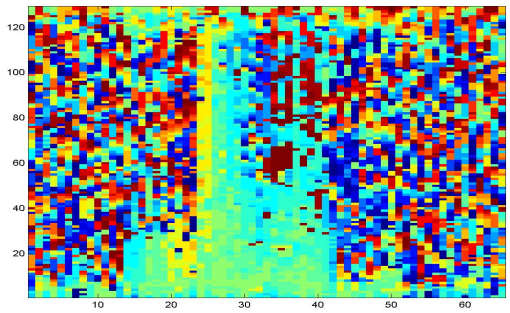

Figure 6.3: Transformations that naively maximize the likelihood potentials. Each color represents a different transformation matrix from the set of 13.

Since we want to capture spectral transformations that can be described in the form of eqn. 6.2, we need to select potentials that impose such restrictions on the data. Therefore, the "local-likelihood" potential between the time-frequency bin x_t^k, its relevant neighbors in frame t, its relevant neighbors in frame $t-1$, and its transformation node T_t^k has the following form:

$$\psi\left(\vec{X}_t^{[k-n_C,k+n_C]}, \vec{X}_{t-1}^{[k-n_P,k+n_P]}, T_t^k\right) = \mathcal{N}\left(\vec{X}_t^{[k-n_C,k+n_C]}; \vec{T}_t^k \vec{X}_{t-1}^{[k-n_P,k+n_P]}, \Sigma^{[k-n_C,k+n_C]}\right) \quad (6.6)$$

The diagonal matrix $\Sigma^{[k-n_C,k+n_C]}$, which is learned, has different values for each frequency band to account for the variability of noise across frequency bands. Local constraints between adjacent transformation nodes are modeled by horizontal and vertical transition potentials $\psi_{hor}(T_t^k, T_{t+1}^k)$ and $\psi_{ver}(T_t^k, T_t^{k+1})$, which are represented by transition matrices.

A naive approach for finding the best set of transformations $\mathcal{T}$ that better describe the data would be to choose the transformations T_t^k that maximize the local potentials, eqn. 6.6. Figure 6.3a shows an example of such transformations for the spectrogram on the left in figure 6.5. In the figure each color indexes a different transformation matrix.

There is little visible structure since the transformation choices capture only local information. If we require transformations with more global consistency we have to perform inference on the model.

When nodes $\mathcal{X}$ are fully observed, inference consists of finding probabilities for each transformation index at each time-frequency bin. Exact inference is intractable given that our model is quite "loopy", and it is approximated using Loopy Belief Propagation(J.S. Yedidia and Weiss 2001; Weiss and Freeman 2001). Figure 6.4 shows the factor graph representation of a section of our model. For now, variable nodes x_t^k are observed, therefore all messages $m_{x_k^t \to g_i^j}$ consist of trivial identity messages; the only messages we need to compute are the ones that go through the T_t^k variable nodes.

Our schedule for the "belief propagation" is as follows: We first run messages through the vertical chains, i.e. all the bins in a given frame t. Next, we run messages through all the horizontal chains (constant frequency index). We choose this order because there is more correlation between the frequency bins in a given frame than between the bins at the same frequency across all time frames. Applying the "belief propagation" formulas on the chains results in forward/backward, upward/downward recursions similar to the ones obtained in HMMs. But unlike HMMs where the posterior at each point is determined by the local likelihood and by the neighbors in the chain, here the equations result in a weighted local likelihood that takes into account the match to the local observation as well as the "beliefs" from the neighboring chains. (This derivation is presented in Appendix B). The use of HMM-like recursions make the inference procedure relatively fast.

We consider a full iteration of the model as a full pass of messages in both directions for all the vertical and horizontal chains. Given that the graph has loops we typically find it takes around five iterations before the transformations posteriors converge.

Once converged, we can find the transformation map, a graphical representation of the *expected* transformation node indices across time-frequency, which provides an ap-

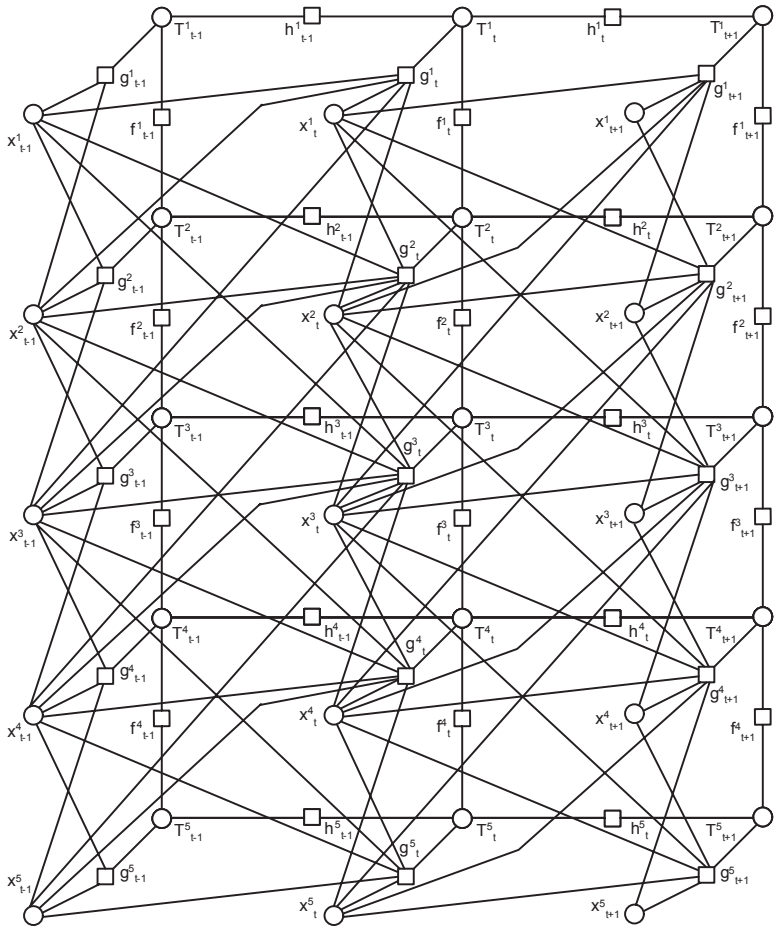

Figure 6.4: Factor Graph for the relationships between spectrogram bins x_t^k and transformation nodes T_t^k. Function nodes g_t^k correspond to the "local-likelihood" potentials (eq. 6.6). Function nodes h_t^k and f_t^k correspond to the horizontal and vertical potentials.

pealing description of the harmonics' dynamics as shown in figure 6.5. In these panels, the links between three specific time-frequency bins and their corresponding transformations on the map are highlighted. Bin 1 is described by a steep downward transformation, while bin 3 also has a downward motion but is described by a less steep transformation, consistent with the dynamics visible in the spectrogram. Bin 2, on the other hand, is described by a steep upwards transformation. Notice how the transformation map emphasizes the global structure of the signal that the naive approach fails to reflect. Also, since the transformation maps follow global consistencies they can be robust to noise

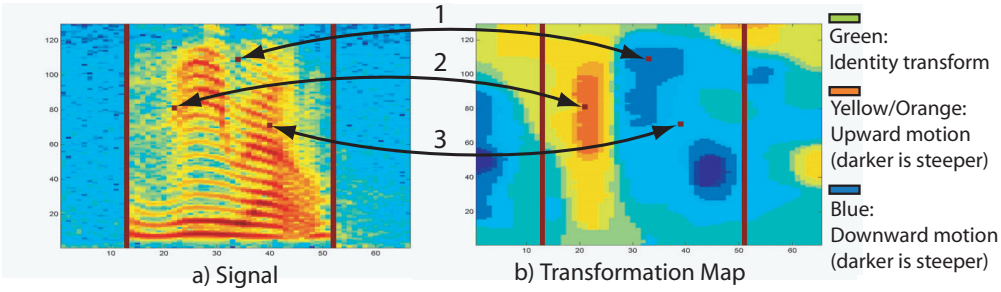

Figure 6.5: Example transformation map showing corresponding points on the original signal.

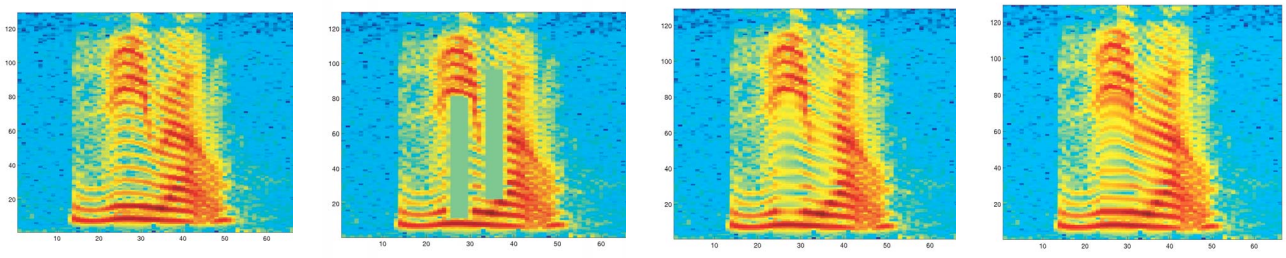

Figure 6.6: Missing data interpolation example a) Original, b) Incomplete, c) After 10 iterations, d) After 30 iterations.

(see fig. 6.7), potentially making them a valuable representation in their own right, as investigated in section 6.6 for speech recognition.

6.2 Inferring Missing Data

If a certain region of cells in the spectrogram is missing, as in the case of corrupted data, the corresponding nodes in the model become hidden. This is illustrated in figure 6.6, where regions of the spectrogram have been removed and tagged as missing. Inference of the missing values is performed again using belief propagation, although the update equations are more complex since there is the need to deal with continuous messages.

The posteriors of the hidden continuous nodes are represented using Gaussian distributions, and the missing sections on figure 6.6b are filled in with the means of their

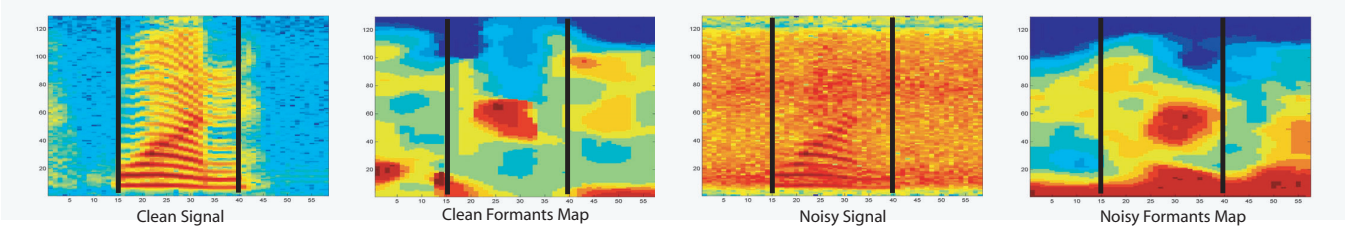

Figure 6.7: Formant tracking map for clean speech (left panels) and speech in noise (right panels).

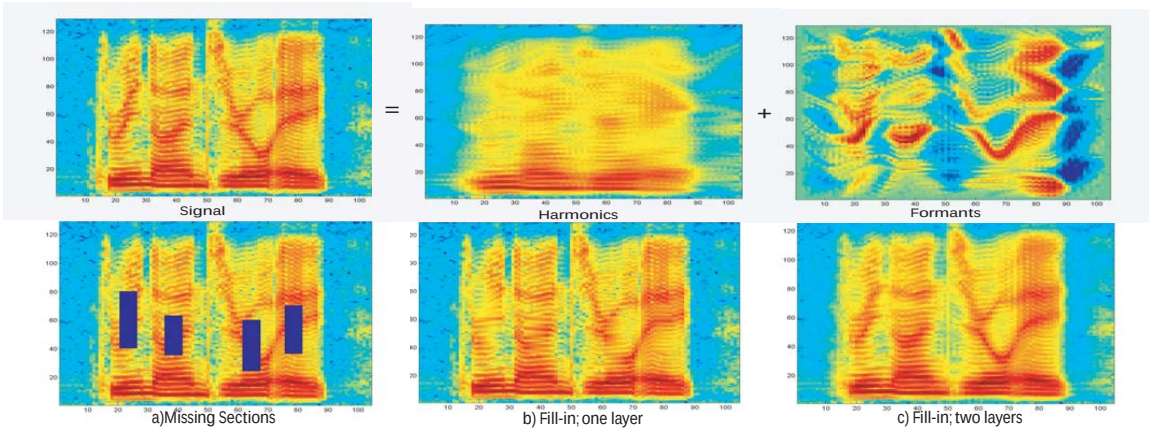

Figure 6.8: First Row: Harmonics/Formants decomposition (posterior distribution means). Second Row: (a) Spectrogram with deleted (missing) regions. (b) Filling in using a single-layer transformation model. (c) Results from the two-layer model.

inferred posteriors as shown in figure 6.6 parts c and d.

The transformation node posteriors for the missing region are also estimated. In the early stages of the "fill-in" procedure the transformation "belief" ($m_{g_j^i \to T_i^j}$) from the local likelihood potential g_i^j to the transformation nodes T_i^j that interact with 'missing" nodes $\tilde{x}_t^k$ are set to uniform so that their transformation posteriors are driven only by the reliable observed neighbors. Messages $m_{\tilde{x}_k^t \to g_i^j} = \delta(\tilde{x}_t^k - \mu)$ are initialized with the global mean μ of the observed data.

The fill-in process starts with the missing values that have reliable immediate neighbors. Once those missing values have been filled-in with estimated data (i.e. using the mean μ_t^k of their Gaussian distributions) the process continues to their immediate

"missed" neighbors and so on. Full details including the equations used in this scenario are given in Appendix C.

Here, an iteration is defined as each time the complete set of missing values is estimated. Define N_{width} as the maximum number of consecutive corrupted bins in any direction: The transformation posteriors are re-estimated every N_{width}/3 iterations, and at each re-estimation those transformation "beliefs" from the local likelihood potential g_i^j to the transformation nodes T_i^j that interact with 'missing" nodes x_t^k are recomputed using the newly-estimated values for the missing variables. This is important to ensure that the estimated data from different directions of the missing region agree. The algorithm normally converges after $N_{width} \times 3$ iterations.

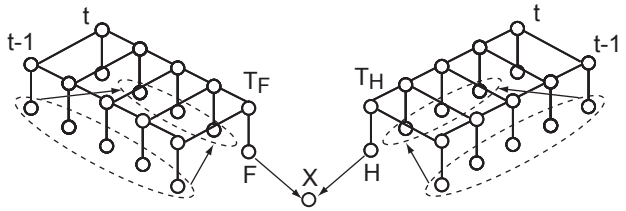

Figure 6.9: Graphical representation of the two-layer source-filter transformation model.

6.3 Two Layer Source-Filter Transformations

Many sound sources, including voiced speech, can be successfully modeled as the convolution of a broadband *source excitation*, such as the pseudo-periodic glottal flow, and a time-varying resonant *filter*, such as the vocal tract, that 'colors' the excitation to produce speech sounds or other timbres. When the excitation has a spectrum consisting of well-defined harmonics, the resulting spectrum is in essence the resonant frequency response sampled at the frequencies of the harmonics, since convolution of the source with the filter in the time domain corresponds to multiplying their spectra in the Fourier domain, or adding in the log-spectral domain. Hence, we can model the log-spectra X as the sum of variables F and H, intended to be explicit models of the formants and harmonics of

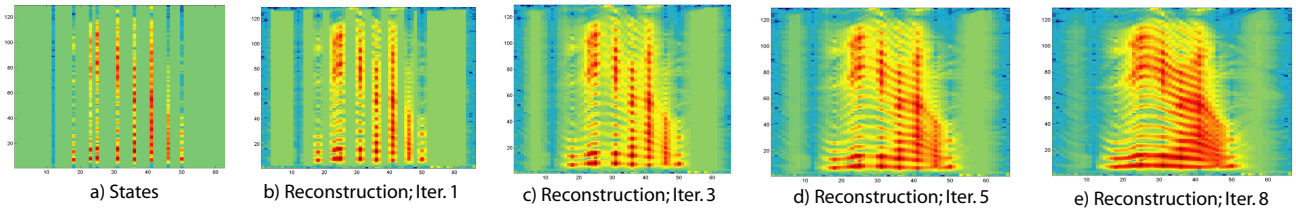

Figure 6.10: Reconstruction from the matching-tracking representation, starting with just the explicitly-modeled states, then progressively filling in the transformed intermediate states.

the speech signal. The source-filter transformation model is based on two additive layers of the deformation model described above, as illustrated in figure 6.9.

Variables F and H in the model are hidden, while X can be observed or hidden, as before. The symmetry between the two layers is broken by using different parameters in each, chosen to suit the particular dynamics of each component. We use transformations with a larger support in the formant layer ($N_W = 9$) compared to the harmonics layer ($N_W = 5$). Since all harmonics tend to move in the same direction, we enforce smoother transformation maps on the harmonics layer by using potential transition matrices with higher self-loop probabilities.

An example of the transformation map for the formant layer is shown in figure 6.7, which also illustrates how these maps can retain key features in the face of high levels of signal corruption; belief propagation searches for a consistent dynamic structure within the signal, and since noise is less likely to have a well-organized structure, it is properties of the speech component that are extracted. Inference in this model is more complex, but the actual form of the continuous messages is essentially the same as in the one layer case, with the addition of the potential function relating the signal x_t^k with its transformation components at each time-frequency bin h_t^k (not to be confused with the factor graph nodes) and f_t^k:

$$\psi(x_t^k, h_t^k, f_t^k) = \mathcal{N}(x_t^k; h_t^k + f_t^k, \sigma^k) \tag{6.7}$$

The two layers are iteratively estimated as described on Appendix D. An iteration

is defined as one estimation of the harmonic layer followed by one estimation of the formants layer. The model usually converges after 10 iterations.

The first row of figure 6.8 shows the decomposition of a speech signal into harmonics and formants components, illustrated as the means of the posteriors of the continuous hidden variables in each layer. The decomposition is not perfect, since we separate the components in terms of differences in dynamics; this criteria becomes insufficient when both layers have similar motion. However, separation improves modeling precisely when each component has a different motion, and when the motions coincide, it is not really important in which layer the source is actually captured. In the second row of fig. 6.8, panel (a) shows spectrogram from the top row with deleted regions; notice that the two layers have distinctly different motions. In panel (b) the regions have been filled via inference in a single-layer model; notice that since the formant motion does not follow the harmonics the formants are not captured in the reconstruction. In panel (c) the two layers are first decomposed and then each layer is filled in; the figure shows the addition of the filled-in reconstructions from each layer.

6.4 Matching-Tracking Model

Prediction of frames from their context is not always possible, for instance when there are transitions between silence and speech or transitions between voiced and unvoiced speech. As a result, we need a set of states to represent these unpredictable frames explicitly. We will also need a second "switch" variable that will decide when to "track" (transform) and when to "match" the observation with a state. Figure 6.11 shows a graphical representation of this model. At each time frame, discrete variables S_t and C_t are connected to all frequency bins in that frame. S_t is a uniformly-weighted Gaussian Mixture Model containing the means and the variances of each of the explicitly-modeled states, μ_j and ϕ_j. Variable C_t takes two values: when it is equal to 0, the model is in

"tracking mode"; a value of 1 designates "matching mode".

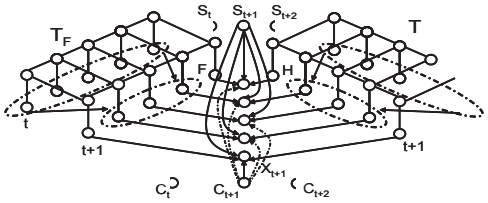

Figure 6.11: Graphic model of the matching-tracking model

The potentials between observations x_t^k, harmonics and formants hidden nodes h_t^k and f_t^k respectively, and variables S_t and C_t, are given by:

$$\psi\left(x_t^k, h_t^k, f_t^k, S_t, C_t = 0\right) = \mathcal{N}\left(x_t^k; h_t^k + f_t^k, \sigma^k\right) \tag{6.8}$$

$$\psi\left(x_t^k, h_t^k, f_t^k, S_t = j, C_t = 1\right) = \mathcal{N}\left(x_t^k; \mu_j^k, \phi_j^k\right) \tag{6.9}$$

Inference is done again using loopy belief propagation. Defining ϕ as a diagonal matrix, the M-Step is given by:

$$\begin{aligned}
\mu_j &= \frac{\sum_t Q(S_t = j)Q(C_t = 1)X_t}{\sum_t Q(S_t = j)Q(C_t = 1)} \\
\sigma_k &= \frac{\sum_t Q(C_t = 1)(x_t^k - (f_t^k + h_t^k))^2}{\sum_t Q(C_t = 1)} \\
\phi_j &= \frac{\sum_t Q(S_t = j)Q(C_t = 1)(X_t - \mu_j)^2}{\sum_t Q(S_t = j)Q(C_t = 1)}
\end{aligned} \tag{6.10}$$

$Q(S_t)$ and $Q(C_t)$ are obtained using the belief propagation rules. $Q(C_t = 0)$ is large if eqn. 6.8 is larger than eqn. 6.9 for several time frequency bins at frame t. In early iterations when the means are still quite random, eqn. 6.8 is quite large, making $Q(C_t = 0)$ large with the result that the explicit states are never used. To prevent this we start the model applying large thresholds to variances ϕ and σ, which will result in non-zero values for $Q(C_t = 1)$, and hence the explicit states will tend to be learned.

As we progress, we start to learn the variances by annealing the thresholds i.e. reducing them at each iteration. We start with a relatively large number of means, but

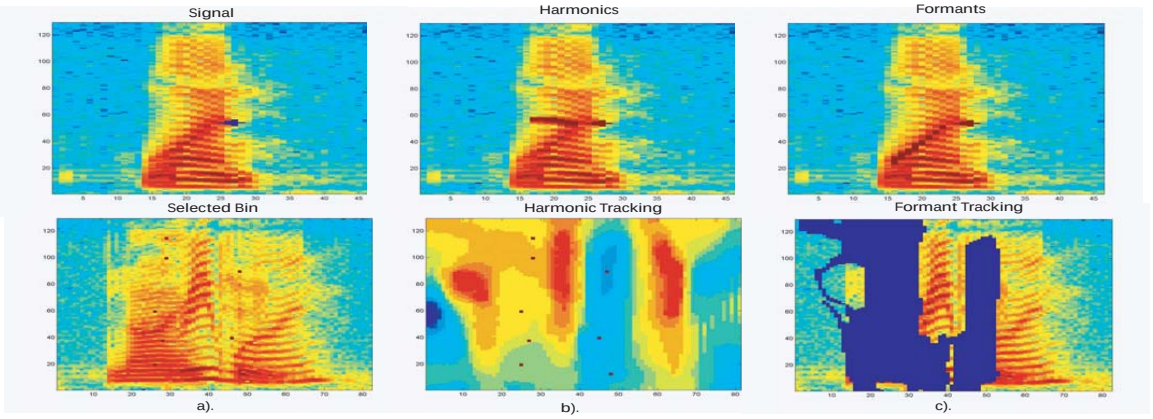

Figure 6.12: Row 1: Harmonics/formants tracking example. The transformation maps on both layers are used to find the ancestors a given time-frequency bin (shown by the dark patches). Row 2: Semi-supervised two speaker separation. (a) The user selects bins on the spectrogram that she believes correspond to one speaker. (b) The system finds the corresponding bins on the transformation map. (c) The system selects and removes all bins whose transformations match the ones chosen; the remaining bins are assumed to correspond to the other speaker.

this becomes much smaller once the variances are reduced; the lower thresholds then control the number of states used in the model. The resulting states typically consists of single frames at discontinuities as intended. An iteration for this model consist of finding the posteriors for each one of the layers then applying the belief propagation rules to nodes S_t and C_t. Finally, the means and variances are learned through eqns. 6.10.

Figure 6.10a shows the frames chosen for a short speech segment whose spectrogram is shown in figure 6.6. The following panes show, at various iterations, how the signal can be regenerated from the model using the states and the two estimated motion fields. This reconstruction is another instance of inferring missing values, but in this case the motion fields are not re-estimated since we have the true ones.

Figure 6.13, shows the states chosen $Q(C_t \approx 1)$ for a single source signal with 15 initial states. The states find those frames where there are transitions between silence and speech and between voiced and unvoiced speech as well as frames where there are

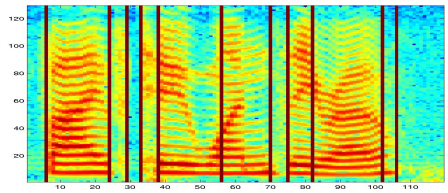

Figure 6.13: Frames selected by the matching-tracking model for a single source signal.

"great" motion changes in both layers. The later frames are chosen because the belief propagation enforces smooth changes in the kind of transformations, the "great" changes in the transformation pattern are not inmediatly followed by the belief propagation algorithm which results in a not so good prediction from the context at those frames hence triggering the switch variable.

6.5 Model Demonstration

We have built an interactive demo that implements formant and harmonics tracking, missing data interpolation, formant/harmonics decomposition, and semi-supervised source separation of two speakers. Videos illustrating the use of this demo are available at: `http://www.ee.columbia.edu/~mjr59/def_spec.html`.

Formants and Harmonics Tracking: Analyzing a signal with the two-layer model permits separate tracking of the harmonic and formant 'ancestors' of any given point. The user clicks on the spectrogram to select a bin, and the system reveals the harmonics and formant transformation "history" of that bin, by plotting the probability that the bin energy came from any of the previous frame "'tranformed"' bins. An example is illustrated in the first row of figure 6.12.

Semi-Supervised Source Separation: After modeling the input signal, the user clicks on time-frequency bins that appear to belong to a certain speaker. The program then selects all neighboring bins with the same value in the transformation map; the remaining bins should belong to the other speaker. This application works under the

assumption that the speakers have different dynamic patterns. The second row of figure 6.12 depicts an example with the resultant mask and the "clicks" that generated it. Although far from perfect, the separation is good enough to perceive each speaker in relative isolation.

The demo also includes the missing data interpolation and harmonics/formants separation as described in the earlier sections.

Features	CLEAN	SNR20	SNR15	SNR10	SNR5
PLP12+delta	.94	2.3	4.1	7.9	12.2
PLP12+delta+dct8(FTM1)	.98	2.3	3.4	6.8	11.1
PLP12+delta+dct10(FTM1)	.99	2.3	3.3	7.4	11.3
PLP12+delta+dct8(FTM2)	1.3	2.5	4.2	9.7	12.5
PLP12+delta+dct10(FTM2)	1.3	2.5	4.2	9.4	12.7

Table 6.1: Word Error Rate percentages obtained with different sets of features as a function of signal-to-noise ratio in dB.

6.6 Speech Recognition Results:

The phonetic distinctions at the basis of speech recognition reflect vocal tract filtering of glottal excitation. In particular, the dynamics of formants (vocal tract resonances) are known to be powerful "information-bearing elements" in speech.

The formant transformation maps capture information about the global dynamics of the formants since the belief propagation algorithm searches for consistent structure in the energy evolution across both time and frequency. The delta and double-delta features of commonly-used features such as Perceptual Linear Prediction (PLP) or Mel Frequency Cepstrum Coefficients (MFCC) similarly capture local dynamics of energy, but they describe the frame-to-frame changes only within each frequency band. The transition maps can capture how the energy is moving *across* frequencies. Since the formant transformation maps consist in the $expected$ transformations on the formant layer, they are composed by continuous values.

We computed two sets of transformation maps: one using formants obtained with our model as described above, and another with formants obtained using conventional cepstral smoothing. For the latter we only require a single layer model to compute the transformations maps. We then use features derived from these maps in combination with standard features in a speech recognizer to test if the maps can contribute new information not captured by the regular features. We chose PLP coefficients plus deltas as the baseline features.

To convert the formant transformation maps into features suitable for the recognizer, we applied mel-scale filtering to the maps then used a discrete cosine transform to decorrelate and further reduce the dimensionality of the final feature vectors.

We used the Aurora-2 noisy digits database for our experiments (Hirsch and Pearce 2000). We trained on the complete "multicondition" training set and tested the recognizer using test set C (mismatched noises and channel). Results at different SNR levels are shown in table 6.1. Features derived from formant transformation maps obtained using two layers are referred to as "FTM2" and the ones obtained from the single layer model are denoted "FTM1"; we tried using both 8 and 10 coefficients from the DCT ("dct8" and "dct10").

Looking at the results obtained using PLP features combined with FTM1 features (second and third rows in table 6.1), we see that recognizer performance remains about the same as the standard features alone (first row) when the signal has high SNR values, but when the SNR decreases the new features improve the word error rate (WER) by as much as 19.5% relative for the 15 dB SNR ("SNR15") condition. We interpret these results as follows: when the signals are relatively clean, a local analysis of the energy dynamics, as performed by conventional features, is sufficient to effectively disambiguate the words. However as the interference becomes larger a more global model of the energy dynamics, such as the formants transition maps, can reduce the influence of local energy variations due to the noise. The belief propagation process searches for a consistent

dynamic structure within the signal, and since noise is less likely to have a well-organized structure, it is the properties of the speech component that are extracted.

The table also shows that FTM2 features derived from the two-layer version of the model do not improve the performance of the recognizer. This may be because the layers cannot be separated when the two layers have parallel dynamics, as mentioned above: When the formants and the harmonics are well modeled by the same transformation, the formants are usually captured and modeled in the harmonics layer. Independent modeling of transformation maps for both layers may be more important for other applications such as the missing data inference in section 6.2, as well as the source separation approach described in the following chapter.

6.7 Summary and Conclusions.

In this chapter we introduced a novel statistical graphical model that exploits the the correlation and self-similarity encountered in speech and other audio signals to effectively model the signal dynamics. Efficient algorithms based on the belief propagation algorithm were derived for the inference procedures of the model.

Unlike the delta and double-delta features of commonly-used features such as Perceptual Linear Prediction (PLP) or Mel Frequency Cepstrum Coefficients (MFCC) that capture local dynamics of energy, as the frame-to-frame changes only within each frequency band, our model can capture how the energy is moving *across* frequencies. The results presented on the speech recognition task, suggest that the model discovers a global structure on the dynamics of the signal's energy that helps to alleviate the problems generated by noise interferences. In the next chapter, the model is used to segment mixtures of speech into dominant speaker regions on an unsupervised source separation task. By identifying and modeling the dynamics of the speech on regions where a given speaker is dominant and later using that information to "fill in" the speaker's data masked

out by the interference of another speaker.

Chapter 7

Unsupervised Dominant Speaker Source Separation Using Deformable Spectrograms.

The separation of speech mixtures into its individual sources using a single microphone is a very hard problem that has generated considerable interest in the research community. Current approaches include attempts to segregate a time-frequency representation (spectrogram) on a bin-by-bin basis, sometimes called time-frequency masking. Each bin is subjected to analysis and tagged as belonging to one of the individual sources. The large combinatorial space created by the analysis of the signal at such a fine resolution poses a great challenge to systems attempting to do such a separation. In (Roweis 2003) the combinatorial search is restricted by the use of pretrained speaker models, which limits the applicability of the approach to mixtures of sources whose individual properties are known in great detail. In (Bach and Jordan 2004a), a training session is required to choose the right parameters for a spectral clustering algorithm. Finding clusters among the set of all time-frequency bins requires huge matrices that pose significant numerical problems. Hence, the algorithm requires a great deal of time to separate short mixtures.

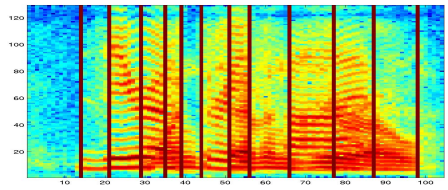

Figure 7.1: Frames selected by the matching-tracking model for a composed signal.

On the other hand, other research had shown that an intelligible separation can be done by grouping those regions of the spectrogram where a given speaker is more dominant than the others Cooke (2004). The problem is how to find those speaker-dominant regions.

A subband version of our matching-and-tracking model has been used to segment such regions, which can be further clustered to separate the sources. Since the number of these regions is significantly smaller than the total number of time-frequency bins in the spectrogram, the clustering problem is several orders of magnitude less complex.

7.1 Subband Matching and Tracking Model

The matching and tracking model segments the spectrogram in regions where the energy of the signal can be described by "smooth" transformations of its temporal context. Therefore, one could envision the use of such a model on a signal with multiple sources to perform scene analysis, by tracking smooth regions of a single source while detecting disruptions of the energy patterns due to interference from a new source. Figure 7.1 shows the segmentation of a mixture of two speakers given the choice of frames taken by the model. Even though the model does find the frames where a "new source enters" the scene or when an "old one leaves" it, in general the segmentation does not produce regions belonging to a single source. This is so, because the magnitude of the interference is not uniform across all the spectrum. Then we require a model that can "track" in some sections of the spectra while "matching" in others.

Our goal is to find regions where a single source dominates the mixture by finding

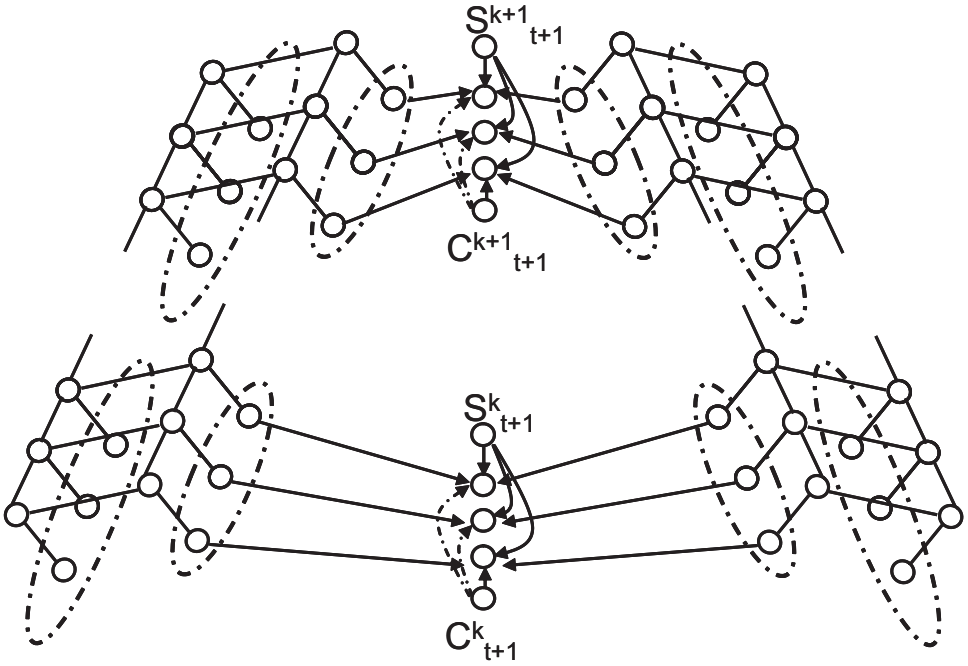

Figure 7.2: Subband version of the match-and-track model. Each subband r with $\mathcal{K}_r$ spectral coefficients has has its own state, S_t^r, and switching, C_t^r, variables

switches of the dominant source. We then extend our matching and tracking model conceived as a single source model to a subband version to accomodate the modelling of signals with multiple sources.

Figure 7.2, shows the graphical representation of the subband version. The tracking part of the model is done as in its full spectra version, the matching part is divided in R subbands. Each subband has its own S_t^r and C_t^r variables, where r indexes the subband number. Defining $\mathcal{K}_r$ = $[k_{min}^r, k_{max}^r]$ as the range of frequencies encompassed by subband r. Potentials between observations x_t^k (providing that $k \in \mathcal{K}_r$), harmonics and formants hidden nodes h_t^k and f_t^k respectively, and variables S_t^r and C_t^r are now given by:

$$\psi\left(x_t^k, h_t^k, f_t^k, S_t^r, C_t^r = 0\right) = \mathcal{N}\left(x_t^k; h_t^k + f_t^k, \sigma^k\right) \tag{7.1}$$

$$\psi\left(x_t^k, h_t^k, f_t^k, S_t^r = j, C_t^r = 1\right) = \mathcal{N}\left(x_t^k; \mu_j^{r,k}, \phi_j^{r,k}\right) \tag{7.2}$$

Providing that $k \in \mathcal{K}_r$, the M-step produces the following equations:

$$\begin{aligned}
\mu_j^r &= \frac{\sum_t (Q(S_t^r = j) Q(C_t^r = 0) X_t^{[k_{min}^r, k_{max}^r]})}{\sum_t (Q(S_t^r = j) Q(C_t^r = 0))} \\
\sigma^k &= \frac{\sum_t (Q(C_t^r = 1)(x_t^k - (f_t^k + h_t^k))^2)}{\sum_t (Q(C_t^r = 1))} \\
\phi_j^r &= \frac{\sum_t (Q(S_t^r = j) Q(C_t^r = 0)(X_t^{[k_{min}^r, k_{max}^r]} - \mu_j^r)^2)}{\sum_t (Q(S_t^r = j) Q(C_t^r = 0))}
\end{aligned} \tag{7.3}$$

$Q(C_t^r = 0)$ is large if eqn. 7.1 is larger than eqn. 7.2 for several time frequency bins x_t^k on the range of subband r, i.e $k \in \mathcal{K}_r$, regardless of the values of the time frequency bins in the other subbands, isolating in this way the switching decisions within the subband boundaries.

Means Initialization

The choice for the number of states for the S_t^r variables is application dependant, just as in the case of the number of states used with a hidden Markov model. Since we want to capture the frames in the boundary between dominant speaker regions in each subband, we will require at least M_r+1 states, where M_r is the different number of regions dominated by a particular source in the r_{th} subband. Random initialization works fine for signals with a single source, since the changes on the energy pattern encountered in the transitions between silence and speech or transitions between voiced and unvoiced speech are quite large and the model does not have problems finding them. However, this is not the case for composed signals, specially if they are analyzed in subbands, since the transitions between the dominant speaker regions are not always very abrupt and the model might miss them.

We choose the number of states for each subband and the initialization values for their means by finding regions with great mismatches on the spectrogram between subsequent frames. As mentioned before an iteration of the matching and tracking models consist of estimating the posteriors of both layers first and then learning the posteriors

for variables S_t^r and C_t^r, where the means are used for the first time, since the means are parameters of those variables potentials. We initialize the means right after estimating the harmonics and formants layer, we then find the euclidian distance between the spectrogram and the summation of the two estimated layers. Taking the summation on the distance matrix over the bins corresponding to each subband at each frame we find a sequence Seq_r measuring the distance between adjacent frames in each subband. We define the number of means per band as the number of peaks above the empirical mean for all the distances on Seq_r. Each mean is initialized as the mean of a set of time fequency bins around the peak location plus some noise. We add the noise since we are "suggesting" possible switching locations rather than choosing them.

7.2 Segmentation Results

We ran experiments on 200 artificially mixed mixtures of two speakers, divided in four categories: 50 female-female, 50 male-male, 50 male-female and 50 with the same speaker voicing different utterances. To measure the performance of the model we have to define "ground truth" for our experiments.

Ground Truth

We artificially generate mixtures $y^s(t)$ by adding in time the correspondent single source signals. $y^s(t)$ = $y^1(t)$ + $y^2(t)$. Then we can use the spectral energy of the single source signals to find the regions that each speaker dominates. Since we are detecting regions by subbands, the ground truth has to be measured by subbands as well. We find the spectral energy of each signal, i.e. $y_{k,t}^i = (abs(\sum_{\tau=0}^{N_F-1} w[\tau]y^i[\tau - t \cdot H]e^{-j2\pi\tau k/N_F}))^2$, where t is the time-frame index, k indexes the frequency bands, N_F is the size of the discrete Fourier transform, H is the hop between successive time-frames, $w[\tau]$ is the N_F-point short-time window, and $y^i[\tau]$ is the original time-domain signal. Computing, $Y_{r,t}^i$, the energy of i speaker at subband r at frame t as: $Y_{r,t}^i$ = $\sum_{k \in \mathcal{K}_r} y_{k,t}^i$. We then

find three regions for the composed spectrogram, $\mathcal{R}_1$ = $y^s_{k,t}\forall(k,t)$ such that $k \in \mathcal{K}_r$ and 10*log10($Y^1_{r,t}$/$Y^2_{r,t}$) $\geq$ 3db, $\mathcal{R}_2$ = $y^s_{k,t}\forall(k,t)$ such that $k \in \mathcal{K}_r$ and 10*log10($Y^1_{r,t}$/$Y^2_{r,t}$) $\leq$ -3db and $\mathcal{R}_0$ = $y^s_{k,t}\forall(k,t)$ such that $k \in \mathcal{K}_r$ and -3db $\geq$ 10*log10($Y^1_{r,t}$/$Y^2_{r,t}$) $\leq$ 3db; $\mathcal{R}_1$ defines those regions where speaker 1 clearly "dominates" speaker 2 for a margin of at least 3db, $\mathcal{R}_2$ is the correspondent region for speaker 2. $\mathcal{R}_0$ defines those regions where there is no clear dominant speaker.

We then define two types of dominant speaker boundaries, hard and soft boundaries. Hard boundaries correspond to the boundaries between regions $\mathcal{R}_1$ and $\mathcal{R}_2$. They correspond to abrupt transitions of dominant speaker, soft boundaries correspond to regions $\mathcal{R}_0$ found between $\mathcal{R}_1$,$\mathcal{R}_2$ regions, which corresponds to regions rather than boundaries where the transition of dominant speakers are more subtle. We require our model to detect a switch in either of the two frames bordering the hard edges and to detect a switch anywhere on the regions defined by the soft edges.

We measure the effectiveness of our search for the boundaries where changes between dominant speakers ocurr by the use of the basic measures used in evaluating search strategies: precision and recall.

Recall is the ratio of the number of relevant records, in this case the ground truth boundaries, retrieved to the total number of records on the database. It is normally presented as a percentage.

Precision is the ratio of the number of relevant records retrieved to the total number of records retrieved, both relevant and irrelevant.

Then the recall percentage indicates how many "'ground truth"' boundaries where actually detected by the model. The precison percentage indicates the number of false positives found by the model, a small precision percentage, lower than 50%, indicates that the system is detecting more false positives than ground truth boundaries.

The segmentation results using the subband deformable spectrograms segmentation can be observed in the first part of table 7.1:

Type of Mixture	Female Female	Male Male	Female Male	Same Speaker
Deformable Spectrograms Segmentation				
Recall	96.64	97.94	97.51	96.88
Precision	62.80	62.37	61.14	69.18
Log Spectrogram-BIC Segmentation				
Recall	68.67	73.34	70.48	62.63
Precision	40.64	43.25	41.86	36.29
Pitch-Bic Segmentation				
Recall	68.47	66.19	71.46	61.49
Precision	39.94	38.92	42.04	36.55

Table 7.1: Recall/Precision results

As can be observed in the table 7.1, the recall values are pretty high without substantial differences between the different kind of mixtures, even for the ones with the same speaker. The model does well regardless of the nature of the speakers because it discovers interruptions in the energy pattern of the signal without relying on any source dependant features. On the other hand, our precision results are not as good. This is because transitions between voiced and unvoiced data for the same speaker are also detected but they are not considered a "ground truth" dominant speaker change boundary. These transitions, however, are very important for the subsequent dominant speaker re-segmentation and shouldn't be considered a false positive.

False positives do indeed happen, when the model finds mismatches within the same speaker like when there are abrupt variations in the motion of both layers and when there are local variations on the formants energy. These local variations are averaged out when working with the full spectra, but they become important while working with subbands. One way to deal with these local variations is to add a gain constant α_t^k to the model so that we use Eq. 7.4 as the local likelihood potential, instead of Eq. 6.6. We experiment with this new variable and we indeed improve significantly the precision

results, unfortunately with a serious decrease on the recall results.

$$\psi\left(\vec{X}_t^{[k-n_C,k+n_C]}, \vec{X}_{t-1}^{[k-n_P,k+n_P]}, T_t^k, \alpha_t^k\right) = \mathcal{N}\left(\vec{X}_t^{[k-n_C,k+n_C]}; \vec{T}_t^k \vec{X}_{t-1}^{[k-n_P,k+n_P]} + \alpha_t^k \vec{e}, \Sigma^{[k-n_C,k+n_C]}\right) \quad (7.4)$$

where $\vec{e}$ is a $N_c \times 1$ column vector of ones.

Table 7.1, also shows the segmentations results obtained using the Bayesian Information Criteria (*BIC*) procedure originally proposed for speaker segmentation in brodcast news speech recognition Chen and Gopalakrishnan (1998) , also used more recently to detect boundaries in personal audio archives (Ellis and Lee 2005). The Bayesian Information Criterion (BIC) provides a principled way to compare the likelihood performance of models with different numbers of parameters and explaining different amounts of data. The speaker segmentation algorithm presented in Chen and Gopalakrishnan (1998) uses BIC to compare every possible segmentation of a window that is expanded until a valid boundary is found, meaning that the decisions are based on the broadest possible time windows.

The BIC is a likelihood criterion penalized by model complexity as measured by the number of model parameters. Let $\chi = \{x_i : i = 1, ..., N\}$ be the data set we are modeling and $\mathcal{M} = \{m_i : i = 1, ..., K\}$ be the candidate models we wish to choose between. Let $\#(M_i)$ be the number of parameters in model M_i, and $\mathcal{L}(\chi, M_i)$ be the total likelihood of χ under the optimal parameterization of M_i. BIC is defined as:

$$BIC(M) = log\mathcal{L}(\chi, M) - \frac{\lambda}{2}\#(M) \cdot log(N) \quad (7.5)$$

where λ is a weighting term for the model complexity penalty which should be 1 according to theory. By balancing the expected improvement in likelihood for more complex models by the penalty term, choosing the model with the highest BIC score is, by this measure, the most appropriate fit to the data.

The BIC-based segmentation procedure described in Chen and Gopalakrishnan (1998) proceeds as follows. We consider a sequence of d-dimensional audio feature vectors $\chi = \{x_i \in R^d : i = 1, ..., N\}$ covering a portion of the whole signal as independent draws from one or two multivariate Gaussian processes. Specifically, the null hypothesis is that the entire sequence is drawn from a single distribution:

$$H_0 : \{x_1, ..., x_N\} \sim N(\mu_0, \Sigma_0) \tag{7.6}$$

which is compared to the hypothesis that the first i points are drawn from one distribution and that the remaining points come from a different distribution i.e. there is a segment boundary after sample t:

$$H_1 : \{x_1, ..., x_t\} \sim N(\mu_1, \Sigma_1), \{x_{t+1}, ..., x_N\} \sim N(\mu_2, \Sigma_2) \tag{7.7}$$

where $N(\mu, \Sigma)$ denotes a multivariate Gaussian distribution with mean vector μ and full covariance matrix Σ.

The difference in BIC scores between these two models is a function of the candidate boundary position t:

$$BIC(t) = log\left(\frac{\mathcal{L}(\chi|H_0)}{\mathcal{L}(\chi|H_1)}\right) - \frac{\lambda}{2}\left(d + \frac{d(d+1)}{2}\right) log(N) \tag{7.8}$$

where $\mathcal{L}(\chi|H_0)$ is the likelihood of χ under hypothesis H_0 etc., and $d + d(d+1)/2$ is the number of extra parameters in the two-model hypothesis H_1. When $BIC(t) > 0$, we place a segment boundary at time t, and then begin searching again to the right of this boundary, and the search window size N is reset. If no candidate boundary t meets this criteria, the search window N is increased, and the search across all possible boundaries t is repeated. This continues until the end of the signal is reached. This procedure is applied independently in each subband meaning that only the features from the correspondent subband are used to find the boundaries in the correspondent subband. Figure 7.3 depicts a block module with exemplifying the procedure.

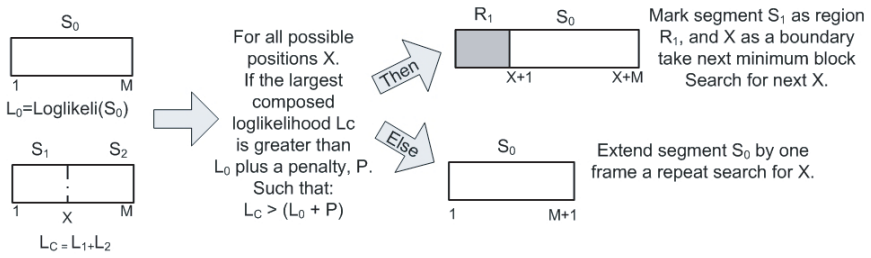

Figure 7.3: Schematic of the BIC segmentation procedure

The weighting parameter λ provides a 'sensitivity' control which can be adjusted to make the overall procedure generate a larger or smaller number of boundaries for a given signal.

We use the BIC segmentation approach using two sets of features, the set of log spectral coefficients from each subband and a subband pitch estimation feature. It is not very clear how to compare the segmentation derived from the different approaches. Since our goal is to compare the two BIC segmentations with the segmentation derived from the subband deformable spectrogram for each composed signal, the weighting parameter λ is fixed such that the number of resulting segments would be equivalent to the number of segments obtained through the subband deformable spectrogram model.

The precision/recall percentages for both BIC approaches are also presented in table 7.1. It is not surprising that the BIC approaches perform significantly lower than the deformable spectrograms one. This is because the BIC approaches make subband segmentation decisions solely on the data corresponding to the given subband, which does not account for "the interferences" from the features on the adjacent subbands. On the other hand, the subband version of the deformable spectrogram model tracks the dynamics of the signal globally, activating a subband "switch" variable when the global context cannot account for an interference in the subband of interest. Also notice that segmentation obtained using the BIC with the pitch feature has a very poor performance on mixtures of signals from the same speaker. This is so because pitch is a speaker dependant feature.

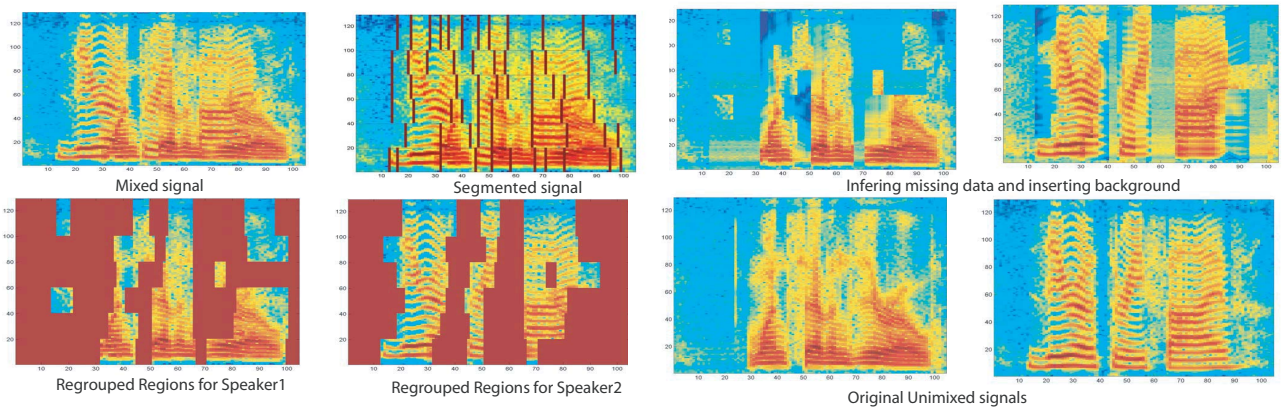

Figure 7.4: The first row shows the original mixed signal. Second row shows the regrouped signals by the spectral clustering algorithm. The third rows shows the reconstructions of the "'missing"' regions. Fourth row shows the original signals prior to the mixing.

Since the deformable spectrograms based segmentation has high recall values we can be pretty certain that the signal is segmented in dominant speaker regions. The number of regions is higher than the optimal since we have a few false positives in the detection of the dominant speaker transitions. However, clustering these regions is a task several degrees simpler than clustering individual time frequency bins.

Figure 7.4, shows an example of the segmentation results on a composed signals.

Clustering these regions is beyond the scope of the model. However, we present two alternatives to do such clustering.

7.3 Clustering Regions

We propose two alternatives to perform the clustering of the regions, one uses similarities between the features on the different regions to do the clustering, while the second relies on "a semantic grouping" given by a higher model to perform the clustering.

7.3.1 Spectral Clustering

Spectral clustering refers to a set of algorithms that rely on the eigen-structure of a similarity matrix between the points to be clustered to partition the points in such a way that points in the same cluster have high similarity while points in different clusters have low similarity. (Bach and Jordan 2004b). It has been used in a wide range of different applications such as source separation, computer vision and VLSI design. In this paper, we adapt the algorithm presented in (Ng et al. 2002) to perform the clustering of the dominant speaker regions. The general algorithm presented at (Ng et al. 2002) is the following:

Given a set of point S=$s_1, ..., s_n \in \Re^l$ that we want to cluster into k subsets:

• Form the affinity matrix $A \in \Re^{n \times n}$ defined by A_{ij} = exp (-$\|s_i - s_j\|^2/2\sigma^2$) if $i \neq j$, and A_{ii} = 0.

• Define D to be the diagonal matrix whose (i,i) element is the sum of A's i_{th} row, and construct the matrix $L = D^{-1/2}AD^{-1/2}$.

• Find x_1,x_2,...,x_k, the k largest eigenvectors of L, and form the matrix X = [x_1,x_2,...,x_k] $\in \Re^{nxk}$ by stacking the eigenvectors in columns.

• Form the matrix Y from the X by renormalizing each of X's rows to have unit length.

• Treating each row of Y as a point in $\Re^k$, cluster them into k clusters via K-means.

• Assign the original point s_i to cluster j if and only if row i of the matrix Y was assigned to cluster j.

We cluster regions first by subbands and later we group clusters between subbands. Since we are clustering regions rather than points we need to adapt the definition of affinity matrix A_{ij} to measure similarity between regions rather than distances between points. Defining M_s as the number of regions on subband s, we need to define a $M_s \times M_s$ affinity matrix $A^s_{r^s_i, r^s_j}$, where r^s_i and r^s_j are two different regions in subband s.

Which we define as:

Figure 7.4 shows the results after reconstructing the missing regions.

$$A_{ij} = exp(-\|PEDi,j\|^2/2\sigma^2) \tag{7.9}$$

where $PED_{i,j}$ is the summation of the n time-frequency patches taken from regions i and j with the minimum distances divided by n. When clustering within subbands we used $n = 3$, when clustering between bands we used $n = 10$; This similarity matrix does not depend on pitch, therefore even regions with similar pitch can be clustered if they show other sources of dissimilarity like prosody or style.

Even though, A^s diagonal entries (i.e. $A^s_{r_i^s,r_i^s}$) are in general non zero, we equate them to zero to satisfy the requisites of the algorithm.

We compute affinity matrices A^s for each subband, then we run the above algorithm with k=3, (i.e. two speakers plus background) to cluster the regions in the band. Then we cluster the clusters in adjacent bands defining 6×6 similarity matrices and running the above algorithm again with k=3. We complete the clustering by propagating the cluster labels across all subbands.

Second row of figure 7.4 shows the clustering results of the segmented signal on the first row We use this mixture as an illustrative example of the significant reduction obtained when clustering dominant speaker regions rather than clustering every single time-frequency bin. The spectral clustering algorithms used per subband have in average a 15×15 similarity matrix, whereareas the time-frequency bin approach will have a $13,416 \times 13,416$ simmilarity matrix.

Even though the similarity matrix proposed in this section do not required speaker dependant features such as pitch. When the the speakers are to similar in several respects the similarity matrix would not be able to capture sufficient differences for the algorithm to work.

Therefore we require another to regroup the signals without specifically counting

in the direct similarity between the regions. We propose "a semantic grouping" given by a speech recognizer to perform the clustering.

7.3.2 Grouping using a speech model.

When the different sound sources have distinctive, low-level properties (such as widely separated pitch ranges), it can be relatively straightforward to identify the correct grouping of regions. If, however, these gross differences are not available – for instance, if two relatively similar voices are interfering – a more complex set of constraints need to be employed. As an extreme example, if the different groupings of cells lead to reconstructed voices, it may be that certain groupings give rise to clearly intelligible speech, whereas incorrect groupings that mix up energy from multiple sources resynthesize to gibberish. Although this seems like a sophisticated judgement, we can in fact use the relatively strong model of likely speech signals implicit within a traditional speech recognition system, to distinguish these cases. This is part of the idea behind the 'speech fragment decoder' Barker et al. (2005), which aims to recognize speech that has had portions of its time-frequency surface corrupted by interference. The speech fragment decoder uses missing-data recognition – integration of likelihood values over the possible ranges of unknown or distorted dimensions – to do a joint search for both the most likely utterance (the conventional speech recognition problem) and the most likely 'missing data mask'. These likelihoods are easily defined in terms of the distribution models (probability of observations given the underlying state) at the heart of speech recognition, but comparing all possible missing-data masks can quickly become intractible. If, however, the set of alternative data masks can be drastically cut down by dividing time-frequency into large regions, and requiring that all cells in a given region receive the same label, recognition again becomes feasible. This can be used both to recognize the speech and to find a likely grouping of the cells; the grouping may be correct even when the word recognition makes errors. In Barker et al. (2005), a simple scheme for dividing the time-

frequency surface into larger patches was described, based on looking at energy peaks above a background noise estimate. The segmentation derived above, based on a much more detailed model of the fine structure of the signal, can identify boundaries much more precisely and will be a good match to the speech fragment decoder approach to speech recognition and source organization.

7.4 Inference of Missing Data

Once we have cluster the segments, we can use the model to infer the masked sections. Figure 7.4 shows an example of the reconstruction.

Here we keep the transformation maps of both layers for the regions that the desired speaker dominates, while relearning the transformation maps for the regions that wered masked by the other speaker. The reconstruction here is not freely done as in the missing information examples shown before, since we do have constraints of what the data can be in the missing regions, given that we can observe the mixed signal on those regions. Moreover restrictions on the structure that the reconstructed signal may take have to be enforced to prevent the reconstruction to follow the structure of the competing speaker.

7.5 Summary and Conclusion.

The subband version of the model is able to detect those regions where a speaker clearly dominates the mixture by tracking its dynamics, while detecting interferences from a new dominant speaker, when interruptions in the dynamics of the combined speech occur. Those regions can be clustered to group together the corresponding dominant regions for each of the sources. Even though clustering these regions is outside of the scope of the model, the clustering task is several times less complex than clustering each time-frequency bin independently. Even more, given that we are clustering regions rather than

bins. “Semantic” clustering using higher level models such as a speech recognizer are also feasible, a situation that would be impossible on the time-frequency bin resolution. As in the multiband model we have a trade off in the resolution to be used for the sub-band version of the model. Going to one extreme, we end up with the time-frequecy bin resolution case, which is impractical for many reasons. But in other hand a coarse subband division will result in “blurring” the dominant speaker regions in such a degree that the separated signals will have significant portions of the other sources.

In any case, once the regions are clustered the dynamics learned by the model can be used to ”fill in” the information masked out by the interference of another speaker.

Chapter 8

Summary and Conclusion.

8.1 Summary.

The problem of separating overlapping sound sources, in particular human speech, has long been a research goal in sound processing. Most research in this area can be classified in two very broad categories:

1. Data-driven approaches.

2. Model-based approaches

The first category corresponds to those approaches were previous knowledge of the signals content is not required and just general characterisitcs or features of the input signals are used to perform the separation. Among these approaches we can find: conventional convolutive independent component analysis (ICA), a multimicrophone approach, which relies mainly on the independence between the various signals to separate them; and the Data-driven Computational Auditory Scene Analysis (CASA) approach. which relies in locally-derived features present in the input data to decompose the input sound into sensory elements. Even though *data-driven* aproaches can separate/characterize several kinds of audio mixtures, they also failed in many scenarios like when the degree of

overlap and/or the dimensionality of the recordings makes the blind inference problem intractable in the case of convolutive ICA or when dealing with perceptual phenomena that involve the use of the auditory context sorrounding a local event in the case of Data-driven CASA systems.

Model-based approaches, on the other hand, are inspired by the apparent ease with which we as listeners achieve perceptual separation and isolation of sound sources in our everyday experiences. They emulate the experience of human listeners who use their prior knowledge of all the sound classes that they have experienced through their lives to impose constraints on the form that the elements of a mixture can take by utilizing models of the individual sources. Applying with these models similar constraints on the characteristics that the mixture components can have. However the large variability between and within sound classes presents a large challenge for the implementation of such models.

The first approaches in this direction, the *prediction-driven CASA* systems extend *data-driven* systems to acommodate the influence of the context on auditory perception (Ellis 1996). They do so by including representations of generic sound elements in an internal world model, such that the internal world model is used to predict the observed cues expected in the next time slice based on the current state of the model. This is then compared with the actual information arriving from the front end; these two are reconciled by modifying the internal state, and the process continues. Such systems have been used to separate objects in natural scenes such as a “construction site” and to separate speech from noisy backgrounds. Unfortunally given the *psychoacoustics* nature of CASA systems is difficult to incorporate large amounts of detailed stastical knowledge about the expected signals in such approach. Moreover, *CASA* precepts do not really contemplate a direct way to explain local composed data by a direct composition of the individual source models.

Later research in this direction focused their attention on the statistical model of choice for audio signals, the hidden Markov models (HMMs), given that they had been widely and successfully used in speech recognition applications. However HMMs are not good generative models for audio signals, they work best when only a limited set of distinct states need to be modeled, as in the case of speech recognition where the models need only be able to discriminate between phone classes. When HMMs are used with the express purpose of accurately modeling the full detail of a rich signal such as speech, they require a large number of states. In (Roweis 2000), HMMs with 8,000 states were required to accurately represent one person's speech for a source separation task. The large state space is required because it attempts to capture every possible instance of the signal. If the state space is not large enough, the HMM will not be a good generative model since it will end up with a "blurry" set of states which represent an average of the features of different segments of the signal, and cannot be used in turn to "generate" the signal.

From the model-based point of view, the work presented in this thesis can be divided in two:

1. 1.- How to accomodate the use of a coarse model such as speech-recognition-like HMMs into a model-based framework to perform source separation

2. 2.- To come up with alternative model to HMMs that could accurately model a large space of audio instances while keeping the number of parameters within reasonable size.

8.2 What has been presented.

In chapter 8.2, we introduced the Maximum likelihood filter-and-sum system, which incorporates HMMs from a previously trained speaker-independent speech recognizer into

the learning of the filters coefficients for the filter-and-sum arrays on a conventional convolutive ICA-like multimicrophone framework. Unlike convolutive ICA where the filter coefficients are estimated to optimize an objective function that measures the independence of the estimated component signals(Hyvärinen 1999); in our approach the filters are estimated to maximize the likelihood of the summed output, measured on the statistical model for the desired signal. The mentioned statistical models are obtained from the transcriptions of the content of each individual source for a very short training signals by constructing an HMM for each source by concatenating the speech recognizer HMMs for the sequence of phonemes encountered in the transcriptions.

The filters are learned using the EM algorithm, but unlike a regular implementation of the EM algorithm for the parameters estimation of an HMM, where the parameters are estimated to maximize the likelihood of the model given the observations and the expected state of the model (M-Step). Here, the observations are reestimated during the M step at each iteration to maximize the likelihood of the observations given the optimal parameters and the expected state of the model. The observations are reestimated at each iteration by the reestimation of the filter coefficient at each filter-and-sum array.

To account for the mixed signal at the output of each filter-andsum array, the individual HMMs are combined into a composed model known as the factorial HMM (FHMM), that is the cross-product of the individual HMMs for the various speakers.

In an FHMM each state is a composition of one state from the HMMs for each of the speakers, reflecting the fact that the individual speakers may have been in any of their respective states, and the final output is a combination of the output from these states. The state output density for any state of the FHMM was assumed to be a Gaussian whose mean is a linear combination of the means of the state output densities of the component states. As the learning of the filters progresses and the presence of the interferring speakers for a given speaker output signal decreases, so do their components in the FHMM state. Given that the factorial composition of individual HMMs with a

large number of states is intractable inference of the factorial model, which is required to estimate the expected states for each individual HMM, is done efficiently through a variational approximation.

Once the filters are learned during the filter coefficient learning phase, the system keeps separating the individual sources as long as the identity of the speaker does not change. Moreover, our results showed that for a given set of speakers, the estimated filters are relatively robust to small variations in speaker location.

These results suggest that the filters learn both speaker specific frequency characteristics, as well as the spatial characteristics of the speakers.

Remarkable results are obtained using this approach for highly reverberant enviroments, situations where traditional ICA-based approaches usually fail. This is because the models were obtained from clean speech (during the training of the ASR system). Therefore, the system, unlike ICA-systems, dereverberates the source signals in addition to separating them.

It is important to mention that even though the generic speech recognizer encodes speaker independent features, which will constitute a very coarse model of a given person's speech, the constraints imposed by the model are enough to guide the system towards the right set of filter coefficients to achieve separation.

It is also worth mentioning that the system is able to separate mixtures from very similar speakers, in part precisely because of the coarseness of the models using during training, given that the models are tunned by the speech content rather than by subtle differences in the spaeker voices or speech styles.

The Maximum likelihood filter-and-sum system was introduced in the framework of the meeting recordings scenario, which are the kind that are obtained when the audio of a typical business meeting is recorded. Multi-microphone approaches fit well in meeting scenarios where the dimensions of the room are known and there is a limited number of possible positions for the speakers. This permits an optimal set up for the microphone

array such that a good coverage of all the speakers could take place regardless of their actual positions. Also, since meeting scenarios are very reverberant by nature, they demand the use of multiple different observations to better cope with the situation. However, the requirement for an array of microphones, makes the application of multimicrophone approaches impractical in many scenarios. Also, many commercial audio signals such as soundtracks and music are available only as single-channel signals. Therefore, there is the need to develop systems that can perform audio source separation from a single recording of a mixed signal.

From the model based source separation perspective the single source restriction imposes more demands on the audio model to be used, requiring models that represent the single sources with greater detail. The model has a greater role in the separation process itself, unlike in the multimicrophone case, where the actual separation is performed by the filters.

When the complexity and variability of the sounds are high, as in a particular speaker's voice, a model that aims to capture every single possible distinct sound might require millions of parameters to cover the full range of possibilities.

In this thesis we proposed two brand new models that factor the large parameter space required in detailed audio. Those two models are: The multiband and the deformable spectrograms models.

Representing every single instance of a particular complex sound is equivalent to representing every single possible column on the spectrogram representation of the audio class. Rather than using a monolithic state to represent the spectrum, the multiband model divides the spectral representation into multiple frequency *bands* and then use separate HMMs in each band with many fewer states. Factorizing the complete spectrogram in this way, large number of full spectral configurations can be represented with substantially fewer parameters making inference and learning more feasible. The adjacent bands are coupled in such a way that at any given frame the state in each band is

determined by the previous states in that band as well as the two adjacent bands, this is done to prevent the formation of frames with unnatural combinations of band states that are not representative of the speaker.

Multiband models are learned for each speaker in the mixture using clean speech signals from the corresponding speaker. The coupling between adjacent bands makes the model intractable and therefore the parameters of the models are learned through a variational approximation of the EM algorithm. The selected variational parameters eliminate the vertical dependencies on the model, decoupling the posteriors of the individual band HMMs permitting then to infer the posteriors in each band using the forward/backward recursions as in a regular HMM. However the HMMs in adjacent bands are coupled in different ways.

First, unlike regular HMMs, where the system is regarded as *stationary*, the transition matrices used on each band HMM to estimate its posteriors are *not stationary*, due to the influence of the "state" of the adjacent bands, (eq. 5.10). Moreover, the four most adjacent bands have a direct say on the output log-probability of the observations for a given band by means of eq. 5.13.

Composed signals are then modeled using the factorial version of the multiband model. The composed model is again intractable and a variational approximation is used as before. Given that the individual models do not require a large number of states per band, the composed output local likelihoods of the factorial HMMs at each band are tractable and therefore exact inference within each factorial HMM is feasible.

The composed factorial likelihoods are modeled using the "log-max" approximation. The idea behind this approach is that when two clean speech signals are mixed additively in the time domain, the log-spectrogram of the mixture is almost exactly the maximum of the individual log-spectrograms (Roweis 2003),

Once inference has been done in the composed model, the signals are recovered through the use of a time-fequency mask. If for a given time-fequency bin in the com-

posed log-spectra spectrogram: the expected mean for a given speaker, at that particular bin, has a larger magnitude than the expected mean, also at tha particular bin, for all other speakers. The bin is classified as belonging to that speaker. The individual signals for each speaker are then regenerated using their assigned log-spectra bins and the original composed phase signal.

The presented results showed that the model is capable to factorize the large variability encountered in the full spectra representation of a person's speech into substates each with a substantially lower variability than the whole. Factorizing the signal in this way results in a substantial reduction of the number of parameters needed to build accurate models for the signals, when compared with the number of parameters needed in a regular full spectra model. This situation results in substantial gains in training and testing times for the multiband model, when compared again with the corresponding times for full spectra models.

In the presented results the multiband model outperformed full spectra models in a speaker separation task.

There is a clear trade off in the partition of the full spectra into subbands between the amount of variability captured in each band and the permutation dilemma when the substates are regrouped to form a valid full spectra state. The subbands have to be large enough to capture a few harmonics, to alleviate the permutation problem, but small enough to keep the variability within the subband relatively low. Regardless of the nature of the partitions, coupling the subbands is a critical step to achieving the best performance given the partition.

Even factorizing the spectrogram in this way, each frame in the spectrogram is treated as an independent identity. However, speech and other natural sounds show high temporal correlation and smooth spectral evolution punctuated by a few, irregular and abrupt changes.

In chapter 6, we introduced the deformable spectrogram model, a model that dis-

covers and tracks the nature of such correlation by finding how the patterns of energy are transformed between adjacent frames and how those transformations evolve over time.

Based on the widely-used source-filter model for such audio signals, we devised a layered generative graphical model that describes these two components in separate layers: one for the excitation harmonics, and another for resonances such as vocal tract formants.

This approach explicitly models the self-similarity and dynamics of each layer by fitting the log-spectral representation of the signal in frame t with a set of transformations of the log-spectra in frame $t-1$.

The proposed model selects, from a discrete set, the particular transformation that better describes the evolution of the energy from frame $t-1$ to frame t around every one of the time frequency bins in the spectrogram.

The model not only can capture the dynamics of the audio signal through the inference of the transformation variables, it can also infer the values of missing portions of the spectrogram by propagating the expected energy profiles from the "'observed"' context into the missing regions.

Inference of the model is not tractable and it is efficiently approximated using the loopy belief propagation algorithm.

Prediction of frames from their context is not always possible, like when there are transitions between silence and speech or transitions between voiced and unvoiced speech. To account for this, the model is extended with set of states to represent these unpredictable frames explicitly.

As a result, we do not require separate states for every possible spectral configuration, but only a limited set of "sharp" (not blurry) states that can still cover the full spectral variety of a source via such transformations.

We showed that the model can capture and "'regenerate"' the modeled signal using only a very limited number of parameters, something that will require a large

number of parameters when using for instance a regular HMM.

We presented results where the model was used to segment mixtures of speech into dominant speaker regions on a unsupervised source separation task. Identifying and modeling the dynamics of the speech on regions where a given speaker is dominant are later used to ”filled in” the information masked out by the interference of another speaker.

We also present results on a speech recognition task that suggest that the model discovers a global structure on the dynamics of the signal's energy that helps to alleviate the problems generated by noise interference.

8.3 Possible modifications, alternatives, problems and improvements to what was presented.

8.3.1 Maximum likelihood filter-and-sum system

The main limitation for an automatic implementation of this approach is the need for the speech transcriptions for each speaker on the training composed speech signal.

The transcriptions are required to come out with the individual speaker models needed during the learning of the filter coefficients. Therefore, we could eliminate the need for the transcriptions if the speaker models could be obtained in an alternative way.

If we were dealing with a single source speech signal, a model of the signal could be obtained by decoding the signal with a speech recognizer and then concatenating the HMMs of the sequence of decoded phonemes. Similarly for a composed signal, a speech raconizer could be used to decode the content of each source, however the decoding would have to be done on factorial compositions of the regular phoneme HMMs from the speech recognizer, which would be a fairly complex but not imposible task. As before, once the speech from each source is decoded, models for each speaker would be obtained by concatenating the HMMs of the sequence of decoded phonemes for the

corresponding speaker.

Another drawback of this approach is that once the filter coefficient are learned, the system can only separate composed signals, if the individual speakers remain somewhat static. To deal with scenarios with changes in the spatial composition of the speakers, an time-adaptive would be necessary. This could be done by using the sppech recognizer to decode the individual speaker outputs to construct new speaker models, which could be used as before to update, in parallel, the filter coefficients while the system is separating composed signals. This adaptive learning approach should work providing that the speakers do not change their position too rapidly.

8.3.2 Multiband Model

Training an accurate speaker dependant model requires a fair amount of data. Often when analyzing a mixture, the identities of the participants are not known, let alone having enough data before hand to train a model for each participant. Therefore the need for pretrained speaker models is the main limitation for a practical implementation of this approach on a real scenario.

A plausible alternative would be to learn the speaker models from the composed data, by directly learning the parameters of a factorial HMM from the composed data and then treating the parameters of each HMM as the desired speaker models. A potential problem with this approach would be that we could end up with HMMs each modeling sections of the speech for the different speakers. Therefore special care should be taken during training to ensure that each speaker is modeled in different branches of the factorial HMM. Also as we will further discussed in section **??** real life mixtures are not really a composition of all the sources at all times, they r4esemble more the idea of a "'scene"', where sources enter and leave the "'scene"' with occasional overlaps between them, given then opportunity to learn the individual source models from those frames on the mixture where only one source in present.

Even when the models are trained from clean speech for each of the speakers, modifications in the way the models are trained could result in an increase in the separation performance. In the current implementation, the individual speaker's HMMs are trained to maximize the likelihood of the clean speech for the corresponding speaker given the model parameters. However, the models are used to explain composed data when combined into a factorial HMMs, which could also be interpreted as the model discriminating which parts of the combined signal belong to each ones of the speakers.

Therefore, the models may be able to do a better job when used to explain composed data, if they are trained in a discriminative fashion on the first place. Meaning that the model parameters for each speaker are trained to maximize the likelihood of the speech for the corresponding speaker while minimizing the likelihood for the speech of the competing speakers.

Finally, even though theoretically the extension of this approach to more than two speakers is straight forward, it is not so much in practice given that the complexity of the computations required do not increase linearly with the number of speakers. We will further discuss this aspect in more detail in the discussion and conclusion section **??**.

8.3.3 Deformable Spectrogram Model

Inference of the transformations for the model with a single layer is very fast and it can be done almost in real time, that is not the case when using the two layers model, this is due to the fact that in order to infer the transformation of the formants layer, the model has to infer the actual energy content in each one of the layers. The same applies for the tracking-and-matching model, the one layer version is much faster than the two layer version. Therefore it would be more time efficient to separate the two layer by other means, i.e. by ceptra analysis, and then apply separately the one layer version to each one of the resultant layers.

In other words, the two layer model is necessary when representing speech bea-

cuse there are two well known sources of variability. However, there is not any requirement to do the separation directly by the model. if the two sources of variability could be factorized by other means, then a one layer model is sufficient to analyze the dynamics of each of the sources of variability.

The full potential of this model for unsupervised source separation applications has not been reached yet. The subband version of the tracking and-matching system is a first coarse approach to it. In the next

Appendix A

Sum-Product Algorithm on HMMs

Referring to figure 2.1 b), where $g_t = p(Y_t \mid X_t)$ and $h_t = p(X_{t+1} \mid X_t)$, then $m_{g_t \to X_t} = g_t$. The algorithm starts at the leaf nodes X_0 and X_T.

$m_{X_0 \to h_0} = m_{g_0 \to X_0} = g_0$, $m_{h_0 \to X_1} = \sum_{X_0} h_0 m_{X_0 \to h_0}$.

$m_{X_1 \to h_1} = m_{h_0 \to X_1} g_1 = \sum_{X_0} (h_0 m_{X_0 \to h_0}) g_1$.

Continuing forward the following recursion formula can be obtained:

$m_{X_t \to h_t} = \sum_{X_{t-1}} (h_{t-1} m_{X_{t-1} \to h_{t-1}}) g_t$

$m_{X_t \to h_t} = \sum_{X_{t-1}} (p(X_t \mid X_{t-1}) m_{X_{t-1} \to h_{t-1}}) p(Y_t \mid X_t)$.

which is the conventional forward recursion for HMMs, (α_t). From the other end:

$m_{X_T \to h_{T-1}} = g_T$, $m_{h_{T-1} \to X_{T-1}} = \sum_{X_T} (h_{T-1} m_{X_T \to h_{T-1}})$.

$m_{X_{T-1} \to h_{T-2}} = m_{h_{T-1} \to X_{T-1}} g_{T-1}$.

$m_{h_{T-2} \to X_{T-2}} = \sum_{X_{T-1}} (h_{T-2} m_{X_{T-1} \to h_{T-2}})$

$m_{h_{T-2} \to X_{T-2}} = \sum_{X_{T-1}} (h_{T-2} m_{h_{T-1} \to X_{T-1}} g_{T-1})$.

$m_{h_{T-2} \to X_{T-2}} = \sum_{X_{T-1}} p(X_{T-1} \mid X_{T-2}) p(Y_{T-1} \mid X_{T-1}) m_{h_{T-1} \to X_{T-1}}$.

The last recursion corresponds to the conventional backwards recursion, (β_t). Computing p(X_t) as the multiplication of the messages on the edge to h_t. p(X_t) = $m_{X_t \to h_t} m_{h_t \to X_t} = \alpha_t \beta_t$.

Appendix B

Messages for the Spectral Deformation Model with Fully Observed Spectrogram

Referring to figure 6.4, variables x_t^k are observed so they only send identity messages, i.e. $m_{x_t^k \to g_l^i} = \delta(x_t^k - \hat{x}_t^k)$, where $\hat{x}_t^k$ is the actual observations at that time-frequency bin. Function nodes g_t^k represents the likelihood potential (Eq. 6.6), $h_t^k = \psi_{hor}(T_t^k, T_{t+1}^k)$ and $f_t^k = \psi_{ver}(T_t^k, T_t^{k-1})$. Working on the vertical chain at frame t, from variable T_t^1 to variable T_t^K for a spectrogram with K coefficients.

$m_{g_t^1 \to T_t^1} = g_t^1$, $m_{T_t^1 \to f_t^1} = g_t^1 m_{h_{t-1}^1 \to T_t^1} m_{h_t^1 \to T_t^1}$.

$m_{f_t^1 \to T_t^2} = \sum_{T_t^1} (f_t^1 m_{T_t^1 \to f_t^1})$.

$m_{T_t^2 \to f_t^2} = m_{f_t^1 \to T_t^2} g_t^2 m_{h_{t-1}^2 \to T_t^2} m_{h_t^2 \to T_t^2}$.

Making $g_t^{'k} = g_t^k m_{h_{t-1}^k \to T_t^k} m_{h_t^k \to T_t^k}$.

$m_{T_t^2 \to f_t^2} = \sum_{T_t^1} (f_t^1 m_{T_t^1 \to f_t^1}) g_t^{'2}$.

This corresponds to an upward (in frequency) recursion $\alpha_t^{'k}$ on the vertical chain at frame t using a "weighted" local likelihood function $g_t^{'k}$, which corresponds to the regular local likelihood function weighted by the "belief" of the adjacent vertical chains. A sim-

ilar "weighted" downward recursion can be found examining the sequence of messages from variable T_t^T to variable T_t^1. Analogous "weighted" forward/backward recursions can be found while working with the horizontal chains.

Appendix C

Continuous messages for the missing data scenario

The local likelihood messages, i.e. $m_{g_t^k \to T_t^k}$, of the function nodes g_t^k that have any of the missing time frequency bins as arguments are initially set to uniform. For all others, $m_{g_t^k \to T_t^k} = g_t^k$. Once this initialization is done, the messages involving the transformation nodes are estimated as above, so that the "transformation beliefs" for the missing time frequency bins are driven only by the "beliefs" of the surrounding reliable neighbors and not by the unreliable local likelihood potentials.

Messages from the local likelihood potential functions g_t^k to missing time frequency bin x_j^i are functions in term of x_j^i. Therefore to facilitate the manipulation of the messages we need to express local likelihoods g_t^k as functions of an individual time frequency bin x_j^i.

The local likelihood potentials are defined as:

$$g_t^k = \mathcal{N}\left(\vec{X}_t^{[k-n_C,k+n_C]}; \vec{T}_t^k \vec{X}_{t-1}^{[k-n_P,k+n_P]}, \Sigma^{[k-n_C,k+n_C]}\right) \tag{C.1}$$

That can be rewritten as:

g_t^k=$\frac{1}{\sqrt{2\pi\Sigma}} exp^{-\frac{1}{2}(Z_t^k)' \Sigma^{-1}(Z_t^k)}$, where

Z_t^k = $X_t^{[k-n_C,k+n_C]} - T_t^k X_{t-1}^{[k-n_P,k+n_P]}$, is a function of variables
$(x_t^{k-n_C}, x_t^{k-n_C+1}, .., x_t^{k+n_C}, x_{t-1}^{k-n_P}, .., x_{t-1}^{k+n_P}, T_t^k)$;

Since column vectors $X_t^{[k-n_C,k+n_C]}$ and $X_{t-1}^{[k-n_P,k+n_P]}$ are concatenations of individual bins. We can express Z_t^k either as:

Z_t^k = $a^i x_t^i + X_t^{[k-nc,k+nc]}. * (1_{N1} - a^i) - T_t^k X_{t-1}^{[k-np,k+np]}$ or

Z_t^k = $-T_t^k b^i x_{t-1}^i - T_t^k (X_{t-1}^{[k-n_P,k+n_P]}.*(1_{N2} - b^i)) + X_t^{[k-n_C,k+n_C]}$ where a^i and b^i are N_1 and N_2 column vectors with zeros in all positions excepting the one corresponding to x_t^i and x_{t-1}^i relative to vectors $X_t^{[k-n_C,k+n_C]}$ and $X_{t-1}^{[k-n_P,k+n_P]}$; 1_{N1} and 1_{N2} are N1 and N2 column vectors of ones and (.*) is the matlab pointwise vector multiplication.

Generalizing even further we can express Z_t^k as:

Z_t^k = $\alpha_j^i x_j^i - \beta_j^i(\mathcal{X}_{t,t-1}^{[k]}, T_t^k)$

where $\mathcal{X}_{t,t-1}^{[k]} = [[X_t^{[k-n_C,k+n_C]}, X_{t-1}^{[k-n_P,k+n_P]}] \setminus x_j^i]$.

$$\alpha_j^i = \begin{cases} a^i & : \quad j = t \\ -T_t^k b^i & : \quad j = t-1 \end{cases}$$

and

$$\beta_j^i(\mathcal{X}_{t,t-1}^{[k]}, T_t^k = \begin{cases} X_t^{[k-nc,k+nc]}. * (1_{N1} - a^i) - T_t^k X_{t-1}^{[k-np,k+np]} & : \quad j = t \\ -T_t^k (X_{t-1}^{[k-n_P,k+n_P]}. * (1_{N2} - b^i)) + X_t^{[k-n_C,k+n_C]} & : \quad j = t-1 \end{cases}$$

The fill-in process starts with the missing values that have reliable neighbors. In general, a given missing bin x_j^i will exchange messages with Np function nodes g_{t-1}^l at frame $j+1$ and with Nc function nodes g_j^r at frame j. If g_t^k is one of such function nodes. Then the message from function node g_t^k to variable x_j^i has the form.

$$m_{g_t^k \to x_j^i} = [\sum_{T_t^k} \int_{\mathcal{X}_{t,t-1}^{[k]}} \frac{1}{C} exp^{\frac{1}{2}(\alpha_j^i x_j^i - \beta_j^i(\mathcal{X}_{t,t-1}^{[k]}, T_t^k))' \Sigma^{-1} (\alpha_j^i x_j^i - \beta_j^i(\mathcal{X}_{t,t-1}^{[k]}, T_t^k))} m_{T_t^k \to g_t^k} \prod_{y \in \mathcal{X}_{t,t-1}^{[k]}} m_{y \to g_t^k} dy] \quad \text{(C.2)}$$

where j is either $t-1$ or t and $i \in [k-n_P, k+n_P]$ if j = $t-1$ or $i \in [k-n_C, k+n_C]$ if j = t. The individual time frequency bins y that belong to the set $\mathcal{X}_{t,t-1}^{[k]}$ are a collection

of missing and observed variables. The ones that are observed have identity messages $m_{y \to g_t^k} = \delta(y - \hat{y})$, where $\hat{y}$ is the actual observed value, while the ones that are missing should have messages $m_{y \to g_t^k}$ equal to the multiplication of their own (N_C + N_P - 1) $m_{g_s^r \to y}$ messages (Eq. C.2) coming from all the other function nodes g_s^r connected to variable y. Given the exponential complexity of such $m_{y \to g_t^k}$ messages, we approximate them by delta functions, i.e. $m_{y \to g_t^k} = \delta(y - \mu_y)$. Parameters μ_y are initially set to the mean of the observed data, these parameters are eventually estimated as explained below.

$m_{T_t^k \to g_t^k} = m_{h_{t-1}^k \to T_t^k} m_{h_{t+1}^k \to T_t^k}\ m_{f_t^{k-1} \to T_t^k} m_{f_t^{k+1} \to T_t^k}$ is simplified by making one the position of the "most-likely" transformation, $\hat{T}_t^k$, and zero all the others. Then Eq. C.2 reduces to:

$m_{g_t^k \to x_j^i} = \frac{1}{C} exp^{\frac{1}{2}(\alpha_j^i x_j^i - \hat{\beta}_j^i)' \Sigma^{-1} (\alpha_j^i x_j^i - \hat{\beta}_j^i)}$,

where $\hat{\beta}_j^i = \beta_j^i(\hat{\mathcal{X}}_{t,t-1}^{[k]}, \hat{T}_t^k)$. $\hat{\mathcal{X}}_{t,t-1}^{[k]}$ is formed by the concatenation of the relevant $\hat{y}$s and μ_ys.

The posterior probability of node x_i^j, $p(x_t^k)$, is equal to the multiplication of all its incoming messages. We approximate this multiplication with a Gaussian distribution, $q'(x_i^j) = \mathcal{N}(x_i^j; \mu_{x_i^j}, \phi_{x_i^j})$. Minimizing their KL divergence we find:

$$\mu_{x_i^j} = \frac{\sum_{l=1}^{N_C+N_P} \alpha_l' \Sigma_l^{-1} \hat{\beta}_l}{\sum_{i=1}^{N_C+N_P} \alpha_l' \Sigma_l^{-1} \alpha_l^{-1}} \tag{C.3}$$

The values displayed by the missing data application are these mean values. The mean of the variable to local function nodes messages, ($m_{y \to g_t^k} = \delta(y - \mu_y)$ for missing variables y in Eq. C.2), have the same form as in equation C.3, just subtracting the numerator and denominator factor corresponding to the incoming message from the corresponding function.

Appendix D

Two layers decomposition

We first estimate the harmonics layer. Initializing message $m_{l_k^t \to har_t^k}$ as $\mathcal{N}(l_k^t, x_t^k, \sigma^k)$, (figure D.1 where l_k^t is the function node for the two layers local likelihood potential (6.7) and messages $m_{har_t^k \to ghar_k^t}$ = $\delta(l_k^t - x_t^k)$ with the actual values of the spectrogram. Then we estimate the posterior probabilities of the harmonics layer q(har_k^t) and their means μhar_t^k as in the one layer case using Eq. C.3, adding to both, the denominator and numerator the corresponding terms from $m_{l_k^t \to har_t^k}$.

Messages $m_{har_t^k \to ghar_k^t}$ are also recomputed. We then proceed to estimate the formants layer, initializing message $m_{l_k^t \to for_t^k}$ as $\mathcal{N}(l_k^t, x_t^k - \mu har_t^k, \sigma k)$, and messages

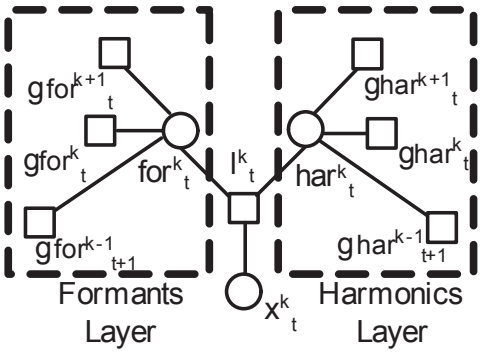

Figure D.1: Factor graph of a section of the two-layers model

$m_{for_k^t \rightarrow gfor_t^k}$ = $\delta(l_k^t - (x_t^k - \mu har_t^k)$ with the subtraction of the estimated harmonics layer from the observed data. The idea here, is to model in this layer all the data that was not captured by the harmonic layer given the restrictions on its parameters. We then estimate the posterior probabilities of the formants layer q(for_k^t) and their means μfor_t^k as in the case of the harmonics layer. Messages $m_{for_k^t \rightarrow gfor_t^k}$ are also recomputed. We then go back to re-estimate the harmonics layer, since all messages have been computed at least once, we just update the messages and recompute the harmonics layer means. Then the formants layer is re-estimated. We keep iterating back and forth until convergence.

Bibliography

A. Levin, A. Z. and Y. Weiss (2003). Learning to perceive transparency from the statistics of natural scenes. In *NIPS*, 2002.

Bach, F. and M. Jordan (2004a). Blind one-microphone speech separation: A spectral learning approach. In *NIPS*, Vancouver.

Bach, F. R. and M. I. Jordan (2004b). Learning spectral clustering. In *Advances in Neural Information Processing Systems (NIPS) 16*.

Barker, J., M. Cooke, and D. Ellis (2005). Decoding speech in the presence of other sources. *Speech Communication*.

Bilmes, J. (1998). Data-driven extensions to hmm statistical dependencies. In *ICSLP*.

Bourlard, H. and S. Dupont (1997). Subband-based speech recognition. In *ICASSP*.

Bregman, A. (1990). *Auditory Scene Analysis*. MIT Press.

Brown, G. (1992). Computational auditory scene analysis:a representational approach. In *Ph.D. thesis*. Univ. of Sheffield: Dept of Comp. Sci.

Chen, S. and P. Gopalakrishnan (1998). Speaker, enviroment and channel detection and clustering, via the bayesian information criterion. In *Proc. Darpa Broadcast News and Understanding Workshop*.

Cooke, M. (1991). Modelling auditory processing and organisation. In *Ph.D. thesis*. Univ. of Sheffield: Dept of Comp. Sci.

Cooke, M. (2004, November). 10+1 perspectives on speech separation and identification in listeners and machines. Presentation at the AFOSR/NSF Workshop on Speech Separation and Comprehension in Complex Acoustic Environments.

Ellis, D. and K. Lee (2005). Minimal-impact audio-based personal archives. In *SAPA2004*, Korea.

Ellis, D. P. (June 1996). *Prediction-driven computational auditory scene analysis*. Dept. of Elec. Eng and Comp. Sci., M.I.T.

F. Kschischang, B. F. and H.-A. Loeliger (2001). Factor graphs and the sum-product algorithm. *IEEE Transactions on information theory 47*.

Ghahramani, Z. and M. Jordan (1997). Factorial hidden markov models. In *Machine Learning*, Boston. Kluwer Academic Publishers.

Hershey, J. and M. Casey (2001). Audio visual sound separation via hidden markov models. Neural Information Processing Systems.

Hirsch, H. and D. Pearce (2000, September). The AURORA experimental framework for the performance evaluations of speech recognition systems under noisy conditions. In *Proc. ISCA ITRW ASR2000*, Paris.

Hyvärinen, A. (1999). Survey on independent component analysis. In *Neural Computing Surveys*.

Johnson, D. and D. Dugeon (1997). Array signal processing. In *Signal Processing Series*. Prentice Hall.

Jojic, N. and B. Frey (2001). Learning flexible sprites in video layers. In *CVPR*.

Jordan, M. I. and C. Bishop (2004). *Introduction to Graphical Models*. In progress.

J.S. Yedidia, W. F. and Y. Weiss (2001). Understanding belief propagation and its generalizations. In *Exploring Artificial Intelligence in the New Millennium*.

Kollmeier, B., J. Peissig, and V. Hohmann (1993). Real-time multiband dynamic compression and noise reduction for binaural hearing aids. *J. Rehab. Res. and Dev. 30*(1), 82–94.

Kristjansson, T., J. Hershey, and H. Attias (2004). Single microphone source separation using high resolution signal reconstruction. In *ICASSP*, Montreal.

Mirghafori., N. (1998). A multiband approach to atomatic speech recognition. In *Ph. D. Thesis*. Dept. of EECS, UC Berkeley.

Morgan, N., D. Baron, J. Edwards, D. Ellis, D. Gelbart, A. Janin, T. Pfau, E. Shriberg, and A. Stolcke (2001). The meeting project at ICSI. In *Proc. Human Lang. Tech. Conf.*, pp. 246–252.

N. Jojic, B. F. and A. Kannan (2003). Epitomic analysis of appearance and shape. In *ICCV*.

Neal, R. and G. Hinton (1998). A view of the em algorithm that justifies incremental, sparse, and other variants. In *Learning in Graphical Models*. Kluwer.

Ng, A. Y., M. Jordan, and Y. Weiss (2002). On spectral clustering: Analysis and an algorithm. In *NIPS 14,, 2002*.

Reyes-Gomez, M., B. Raj, and D. Ellis (April,2003b). Multi-channel source separation by hmm factorialization.

Reyes-Gomez, M., B. Raj, and D. Ellis (October,2003a). Multi-channel source separation by beamforming trained with factorial hmms. In *WASPAA 2003*, New Paltz, NY.

Roweis, S. (2000). One-microphone source separation. In *Advances in NIPS*, pp. 609–616. Cambridge MA: MIT Press.

Roweis, S. (2003). Factorial models and refiltering for speech separation and denoising. In *Proc. EuroSpeech*, Geneva.

Seltzer, M., B. Raj, and R. Stern (2002). Speech recognizer-based microphone array processing for robust hands-free speech recognition. In *ICASSP*.

Wang, D. L. and G. J. Brown (1999). Separation of speech from interfering sounds based on oscillatory correlation. *IEEE Trans. Neural Networks 10*, 684–697.

Warren, R. (1970). Perceptual restoration of missing speech sounds. *Science*.

Weintraub, M. (1985). A theory and computational model of auditory monoaural sound separation. In *Pd.D. thesis*. Stanford University: Dept. of Elect. Eng.

Weiss, Y. and W. Freeman (2001). Correctness of belief propagation in gaussian graphical models of arbitrary topology. *Neural Computation 13*, 2173–2200.

Yilmaz, O. and S. Rickard (July 2004). Blind separation of speech mixtures via time-frequency masking. *IEEE Transactions on Signal Processing 52*(7), 1830–1847.