Approximate Bayesian Inference for Robust Speech Processing

A Thesis

Submitted to the Faculty

of

Drexel University

by

Ciira wa Maina

in partial fulfillment of the

requirements for the degree

of

Doctor of Philosophy in Electrical and Computer Engineering

June 2011

Abstract

Approximate Bayesian Inference for Robust Speech Processing

Ciira wa Maina

Advisor: John MacLaren Walsh, Ph.D.

Speech processing applications such as speech enhancement and speaker identification rely on the estimation of relevant parameters from the speech signal. These parameters must often be estimated from noisy observations since speech signals are rarely obtained in 'clean' acoustic environments in the real world. As a result, the parameter estimation algorithms we employ must be robust to environmental factors such as additive noise and reverberation. In this work we derive and evaluate approximate Bayesian algorithms for the following speech processing tasks: 1) speech enhancement 2) speaker identification 3) speaker verification and 4) voice activity detection.

Building on previous work in the field of statistical model based speech enhancement, we derive speech enhancement algorithms that rely on speaker dependent priors over linear prediction parameters. These speaker dependent priors allow us to handle speech enhancement and speaker identification in a joint framework. Furthermore, we show how these priors allow voice activity detection to be performed in a robust manner.

We also develop algorithms in the log spectral domain with applications in robust speaker verification. The use of speaker dependent priors in the log spectral domain is shown to improve equal error rates in noisy environments and to compensate for mismatch between training and testing conditions.

Dedications

This work is dedicated to the memory of my father, the late Professor Godfrey Maina and to my mother, Mrs. Jane Wambui Maina.

Acknowledgments

Muti uyo mukwona, wahandirwo ni awa[1]
Joseph Kamaru, *Muti Uyu*.

I was guided. . . to get here
Gil Scott-Heron

Traveller you must set forth
At dawn
Wole Soyinka, "Death in the dawn"

This thesis is the culmination of a long journey that I would have been unable to complete were it not for the help of many people.

First of all I would like to thank my advisor, Prof. John MacLaren Walsh, for his guidance and his patience. He has been an excellent sounding board for the last four years.

I would also like to thank all the members of the ASPITRG group and Ramanan in particular for his friendship.

My life in Philly would have been very dull were it not for my friends Waquar Ahmad and Fela. Our shared love for cold beverages made many difficult periods bearable.

I can honestly say that I would not have been able to complete this work if my mother had not constantly reminded me that I was capable. For this *maitu* I say thank you. *Ni gatho, na Ngai akorathime*.

Finally I would like to thank Akanke for her love.

[1]This tree was planted by my ancestors

Contents

List of Tables

List of Figures

1. Introduction

One of the features that distinguishes human beings from other species is the ability to communicate using speech. Speech is arguably the most important means of human communication. With it we are able to convey information and a wide range of emotions. Coupled with the human ability to speak is the ability to understand what is said in a wide range of acoustic environments. We have evolved the ability to understand speech in noisy environments such as train stations and in crowded locations with several competing speakers.

The ability of humans to understand speech in noisy scenarios has motivated researchers for decades to replicate this human performance using computers. The motivation for this lies in the wealth of information we can extract from the speech signal. From it we can determine both what was said and who said it leading to applications in speech recognition and speaker identification respectively. However for these applications to be reliable, we must be able to deal with noisy conditions likely to be encountered in operation.

In this thesis, we explore the use of approximate Bayesian inference in order to improve the performance of speaker recognition systems in noise. These systems rely on the robust estimation of features from the speech signal. The Bayesian approaches we develop are shown to improve the reliability of the estimates leading to better recognition performance.

The work in this thesis emerged from the recognition that the performance of speech enhancement and speaker recognition systems can be improved if they are viewed as closely related systems. Intuitively, if we can enhance noisy speech or relevant features obtained from noisy speech, then any speaker recognition system making use of the enhanced speech would exhibit performance gains. Furthermore, if

we can construct rich models with speaker dependence, then the relationship between speech enhancement and speaker recognition can be captured elegantly in a Bayesian inference algorithm which treats the exchange of information between the two systems as message passing between nodes in a graphical model (see figure 1.1).

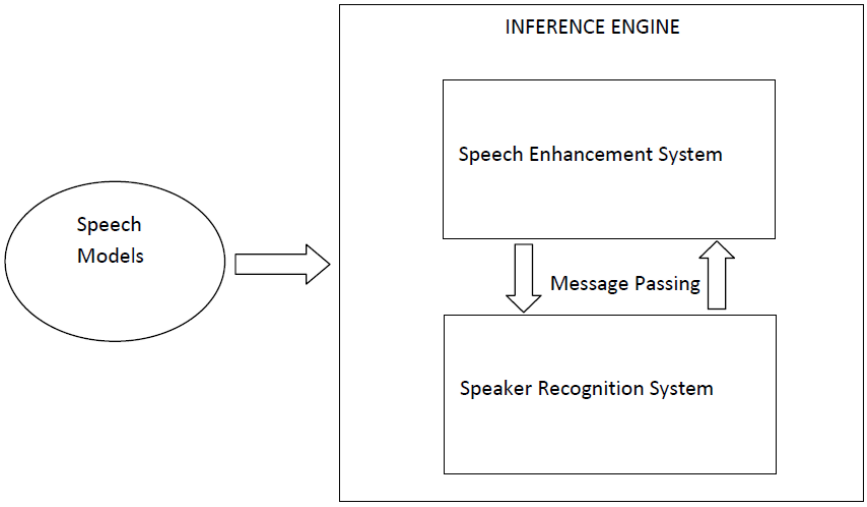


Figure 1.1: The exchange of information between the speech enhancement and speaker recognition systems viewed as message passing.

When considering speech enhancement and speaker recognition, we must decide on the domain in which to model the speech. For robust enhancement, a natural choice would be a model in the acoustic domain as 'close' as possible to the speech samples. For example one may consider autoregressive models. However, speaker models in several speaker recognition systems are in the spectral domain which captures speaker dependent variation in a robust manner. There is therefore a tradeoff between the system performance and the model domain in which we chose to work (see figure 1.2). This is borne out by the results presented in chapters 4 and 5.

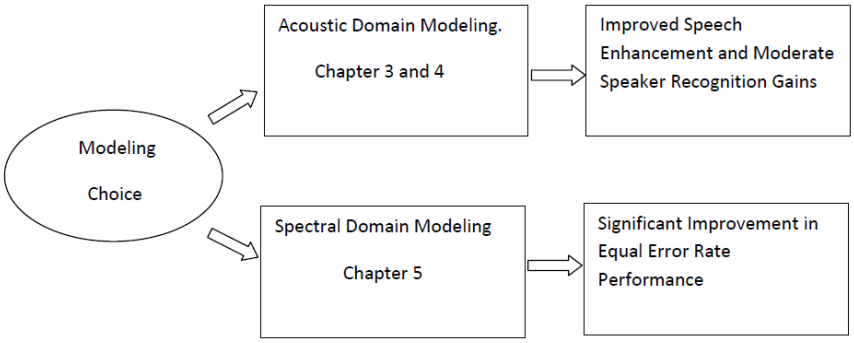


Figure 1.2: The influence of model domain on performance and relevant chapters in the thesis in which this relationship is discussed.

1.1 Thesis Contributions

This thesis makes contributions to a number of areas in speech processing. These include

1. We derive a joint speech enhancement and speaker identification algorithm that takes advantage of the fact that speech enhancement and speaker identification are inextricably linked. With enhanced speech, speaker identification decisions are more accurate and conversely with accurate speaker identification we can use speaker dependent priors over the speech parameters to improve speech enhancement. This relationship is captured in the variational Bayesian (VB) algorithm derived in chapter 4. The experimental results presented in this chapter show that significant SNR improvement is obtained by the VB algorithm with a maximum SNR improvement of approximately 10dB. Also, we achieve SNR improvements within 1 dB of the performance obtained by the theoretical upper limit. Furthermore, the VB algorithm outperforms the Ephraim-Malah algorithm which is a standard baseline in both SNR improvement and perceptual quality as measured using the PESQ score.

2. In addition to performing joint speech enhancement and speaker identification,

the algorithm presented in chapter 4 is capable of performing robust voice activity detection (VAD). VAD is an important speech processing application and the algorithm presented makes use of priors over linear prediction coefficients in silence dominated regions to accurately classify speech segments as either speech or non-speech. The experimental results show that the VB algorithm outperforms the ITU-G.729 algorithm which is the international telecommunications union standard.

3. In chapter 5 we present a VB algorithm for the enhancement of log spectral features and show how this algorithm can be applied to speaker verification to improve equal error rate performance. Once again we make use of speaker dependent priors over the speech features which in this case are log spectral features. Here the VB algorithm is able to significantly improve the equal error rate (EER) performance. In both additive Gaussian white noise and realistic noise such as factory noise, we are able to reduce the EER by up to 50% when we compare our system to a standard baseline.

1.2 Thesis Overview

This thesis is organized as follows. Chapter 2 presents the background necessary for the main areas of the thesis. This includes material on speech enhancement, speaker recognition and Bayesian inference. In chapter 3 we present preliminary work on variational Bayesian inference for speech enhancement. This work is aimed at illustrating the modelling steps necessary to make VB inference possible. We employ a generalized autoregressive model for speech and attempt to mitigate convolutive distortion by incorporating a channel model. However, due to the nature of the approximate posterior over the clean speech, we are forced to make further approximations to allow for inference. This complications arise due to the nature of the speech

model and the attempt to mitigate both additive and convolutive distortion. With this in mind, we extend this VB work in chapter 4 where we concentrate on additive distortion and enrich our speech prior by making it speaker dependent. This allows us to develop a joint speech enhancement and speaker identification algorithm that uses speaker dependent priors over the linear prediction coefficients. This algorithm is also capable of performing voice activity detection.

Encouraged by the success of speaker dependent modelling in the acoustic domain, we present a VB algorithm for the enhancement of log spectral features with the aim of improving speaker verification performance in chapter 5. Working in the log spectral domain offers an advantage over the acoustic domain in the speaker verification setting because we can easily derive Mel frequency cepstral coefficients (MFCCs) from the enhanced log spectra. MFCCs, which are discussed further in the background chapter, are features which have been successfully used in speaker recognition. Chapter 6 presents a summary of the thesis.

2. Background

In this chapter we intend to provide the background necessary for the algorithms developed in the thesis. As stated in chapter 1, we seek to develop approximate Bayesian algorithms for robust speech processing and to demonstrate the application of these algorithms. In this chapter we first discuss parameter inference and in particular we contrast maximum likelihood inference and Bayesian inference. We also discuss the following speech processing applications.

- Speech Enhancement
- Speaker Recognition
- Voice activity detection

2.1 Parameter Inference

Parameter inference is a central problem in signal processing applications. In several situations the observed data $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ are characterized by a generative probabilistic model $p(\mathbf{X}; \theta)$ where θ denotes the parameters of the probabilistic model. Given $\mathbf{X}$, we aim to estimate θ.

If θ is assumed to be an unknown constant then we can obtain the maximum likelihood (ML) estimate of θ as follows:

$$\theta_{ML} = \arg\max_{\theta} p(\mathbf{X}; \theta)$$

or equivalently

$$\theta_{ML} = \arg\max_{\theta} \underbrace{\log p(\mathbf{X}; \theta)}_{\ell(\theta)}. \tag{2.1}$$

ML estimation has been successfully used in several signal processing applications. However, it has a number of drawbacks which stem from the fact that ML estimation does not adequately take into account parameter and model uncertainty. ML estimates are subject to overfitting problems and if the wrong models are assumed parameter estimates will be erroneous.

The Bayesian framework allows us to handle both parameter and model uncertainty. In the Bayesian framework, the parameters of our probabilistic model are treated as random variables governed by a prior $p(\theta)$. We can write the joint distribution $p(\mathbf{X}, \theta)$ as a product of the likelihood and the prior, that is $p(\mathbf{X}, \theta) = p(\mathbf{X}|\theta)p(\theta)$. The posterior $p(\theta|\mathbf{X})$, which is a central quantity in Bayesian inference, is given by [4]

$$p(\theta|\mathbf{X}) = \frac{p(\mathbf{X}|\theta)p(\theta)}{\int p(\mathbf{X}|\theta)p(\theta)d\theta}$$

Using this posterior, estimates of θ are obtained that minimize approriate cost functions. For example the minimum mean square errror estimate is obtained as follows [5]

$$\begin{aligned} \hat{\theta}_{\text{MMSE}} &= \arg\min_{\hat{\theta}} \int \|\theta - \hat{\theta}\|^2 p(\theta|\mathbf{X})d\theta, \\ &= \mathbb{E}\{\theta|\mathbf{X}\}. \end{aligned}$$

The main drawback in the application of Bayesian methods is computational complexity. For example the computation of the evidence $p(\mathbf{X}) = \int p(\mathbf{X}|\theta)p(\theta)d\theta$ is often intractable.

There are a number of ways to deal with the intractability of computations arising in Bayesian inference. In this work we consider two main approaches. The first involves replacing the intractable posterior with a tractable approximation. Variational Bayesian inference and expectation propagation (EP) fall in this category. The sec-

ond approach involves sampling from the intractable posterior and using the samples obtained for inference.

2.1.1 Maximum Likelihood Inference and the EM Algorithm

Consider a sequence of N i.i.d observations $\mathbf{X} = [x_0, \ldots, x_{N-1}]^T$ with likelihood given by $p(\mathbf{X};\theta) = \prod_{n=0}^{N-1} p(\mathbf{x}_n;\theta)$ where the parameter(s) θ are unknown. The maximum likelihood estimate of θ is given by (2.1).

Consider a probabilistic model that includes hidden variables in addition to observed data. In such cases, the 'complete' likelihood is $p(\mathbf{X},\mathbf{S};\theta)$ where $\mathbf{X}$ are the observations and $\mathbf{S}$ are the hidden variables. The data likelihood is given by

$$\begin{aligned} p(\mathbf{X};\theta) &= \int p(\mathbf{X},\mathbf{S};\theta)d\mathbf{S} \\ &= \int p(\mathbf{X}|\mathbf{S};\theta)p(\mathbf{S};\theta)d\mathbf{S}. \end{aligned}$$

In order to obtain the ML parameter estimate we must maximize $\log \int p(\mathbf{X}|\mathbf{S};\theta)p(\mathbf{S};\theta)d\mathbf{S}$ which may involve intractable integrals therefore rendering ML estimation via (2.1) intractable. Expectation maximization provides an alternative framework for computing ML estimates in models with hidden variables [6]. The key idea is to introduce a surrogate quantity that can be maximized in place of the true log-likelihood.

Consider the quantity

$$\begin{aligned} \mathcal{Q}(\theta,\theta') &= \int \log\{p(\mathbf{X},\mathbf{S};\theta)\}p(\mathbf{S}|\mathbf{X};\theta')d\mathbf{S} \\ &= \int \log\{p(\mathbf{S}|\mathbf{X};\theta)p(\mathbf{X};\theta)\}p(\mathbf{S}|\mathbf{X};\theta')d\mathbf{S} \\ &= \int \log\{p(\mathbf{S}|\mathbf{X};\theta)\}p(\mathbf{S}|\mathbf{X};\theta')d\mathbf{S} + \underbrace{\log p(\mathbf{X};\theta)}_{\ell(\theta)}. \end{aligned}$$

$\mathcal{Q}(\theta,\theta')$ is the surrogate quantity of EM and it can be shown that if we can find a

value θ of the parameters such that $\mathcal{Q}(\theta, \theta') \geq \mathcal{Q}(\theta', \theta')$ where θ' is some initial value then [7]

$$\ell(\theta) - \ell(\theta') \geq \mathcal{Q}(\theta, \theta') - \mathcal{Q}(\theta', \theta'). \tag{2.2}$$

The EM algorithm consisits of two steps

1. The E step: Given θ^i compute $\mathcal{Q}(\theta, \theta^i)$ which is the expectation of $\log\{p(\mathbf{X}, \mathbf{S}; \theta)\}$ under $p(\mathbf{S}|\mathbf{X}; \theta^i)$.

2. The M step: Maximize $\mathcal{Q}(\theta, \theta^i)$ as a function of θ to obtain θ^{i+1}. That is

$$\theta^{i+1} = \arg\max_{\theta} \mathcal{Q}(\theta, \theta^i).$$

2.1.2 Variational Bayesian Inference

In variational Bayesian inference, we seek an approximation $q(\Theta)$ to the intractable posterior $p(\Theta|\mathbf{X})$ which minimizes the Kullback-Leibler (KL) divergence between $q(\Theta)$ and $p(\Theta|\mathbf{X})$ with $q(\Theta)$ constrained to lie within a tractable approximating family. The KL divergence $D(q||p)$ is a measure of the distance between two distributions and is defined by [8]

$$D(q||p) = \int q(\Theta) \log \frac{q(\Theta)}{p(\Theta|\mathbf{X})} d\Theta.$$

To ensure tractability we assume that the posterior can be written as a product of factors depending on disjoint subsets of $\Theta = \{\theta_1, \ldots, \theta_M\}$ [9; 10]. Assuming that each factor depends on a single element of Θ then

$$q(\Theta) = \prod_{i=1}^{M} q_i(\theta_i). \tag{2.3}$$

It can be shown that the optimal form of $q_j(\theta_j)$ denoted by $q_j^*(\theta_j)$ that minimizes

$D(q||p)$ is given by [10]

$$\log q_j^*(\theta_j) = \mathbb{E}\{\log p(\mathbf{X}, \Theta)\}_{q(\Theta^{\setminus j})} + const. \tag{2.4}$$

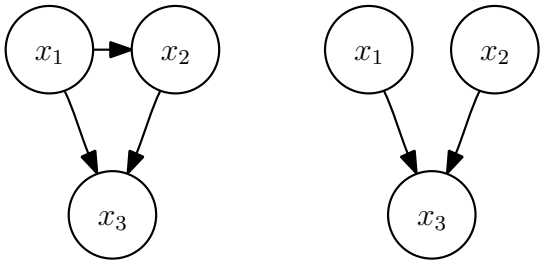
We use the notation $q(\Theta^{\setminus j})$ to denote the approximate posterior of all the elements of Θ except θ_j. We obtain a set of coupled equations relating the optimal form of a given factor to the other factors. To solve these equations, we initialize all the factors and iteratively refine them one at a time using (2.4).

The use of graphical models allows a powerful interpretation of variational techniques as message passing algorithms [11]. That is, the inference step consists of messages being passed between nodes in the graph with each node performing local computations. This allows the global inference problem to be decomposed into local computations [12].

Graphical Models

The use of probability theory to handle uncertainty lies at the heart of statistical signal processing. The probabilistic formulation of a problem is represented by the joint distribution of the parameters of the model and the observations and based on this distribution inference is performed. Graphical models allow us to capture the relationship between the random variables in our problem. That is, the graph associated with a given joint distribution describes how the joint distribution factorizes [10]. This is illustrated in figure 2.1.

A graph $G = (V, E)$ consists of a set of vertices (nodes) V and a set of edges (links) between pairs of vertices. In directed graphs, the edges have an associated direction from the 'parent' node to the 'child' node. Consider a probability distribution $p(\mathbf{x})$ $\mathbf{x} = \{x_1, \ldots, x_N\}$ whose factorization is captured by a directed graph. Each node is



(a) Source Model (b) Observation Model

Figure 2.1: Directed probabilistic graphs illustrating the factorization of $p(x_1, x_2, x_3)$ as (a) $p(x_3|x_1, x_2)p(x_2|x_1)p(x_1)$ and (b) $p(x_3|x_1, x_2)p(x_1)p(x_2)$.

associated with a random variable and we can write [10; 13]

$$p(\mathbf{x}) = \prod_{i=1}^{N} p(x_i|pa_i)$$

where pa_i is the set of random variables associated with the parent nodes of x_i.

Hierarchical models play a central role in Bayesian inference and they can be represented by directed graphical models [14]. These models allow complicated distributions to be built up from simpler components.

For undirected graphical models the factorization of the joint distribution is given in terms of maximal cliques of the graph [10; 12]. With each maximal clique is associated a potential function $\psi_C(\mathbf{x}_C)$ where $\mathbf{x}_C$ are the random variables associated with nodes in the clique. We have

$$p(\mathbf{x}) \propto \prod_{C} \psi_C(\mathbf{x}_C).$$

Given the joint distribution relating the random variables in a particular model our aim is to perform inference. For example in a signal denoising application we aim to recover the unobserved clean signal using the noisy observations. Inference in

graphical models has been applied to various applications such as speech recognition using hidden markov models [15].

The complexity of the inference step is related to the nature of the graphical model and the probability distributions associated with the random variables. If the underlying graph is a *tree* and the nodes are associated with discrete or Gaussian random variables then belief propagation (BP) computes exact marginals [10; 16]. The junction tree algorithm [12] provides a framework for exact inference in arbitrary graphical models. However, in most practical models the computational complexity of this algorithm makes it impractical. In these situations approximate inference techniques must be used.

Loopy belief propagation applies BP to graphs with loops. Even though there is no guarantee of convergence useful results have been obtained in important cases [17]. The convergence of this algorithm has been investigated by a number of authors (for example see [18; 19]). Other approximate inference techniques that can be applied to intractable graphical models include Markov chain Monte Carlo (MCMC) methods [20]. However these methods are computationaly intensive and may be too slow for most practical applications.

2.1.3 Markov Chain Monte Carlo Methods

As described in the introduction, the posterior $p(\theta|\mathbf{X})$ (where θ represents the parameters and $\mathbf{X}$ denotes the observed data) is a central quantity in Bayesian inference. If $p(\theta|\mathbf{X})$ is known we can obtain parameter estimates such as the MMSE estimate given by

$$\hat{\theta}_{\text{MMSE}} = \mathbb{E}\{\theta|\mathbf{X}\}. \tag{2.5}$$

Markov chain Monte Carlo methods are useful in evaluating expectations such as (2.5) [20; 14].

If we can draw independent samples from $p(\theta|\mathbf{X})$ then

$$\begin{aligned}\mathbb{E}\{f(\theta)|\mathbf{X}\} &= \int f(\theta)p(\theta|\mathbf{X})d\theta \\ &\simeq \frac{1}{N}\sum_{n=1}^{N} f(\theta^n)\end{aligned}$$

where $\theta^n \sim p(\theta|\mathbf{X})$. However it may not be possible to draw independent samples from $p(\theta|\mathbf{X})$. In this case we may be able to draw a sequence of samples $\theta^0, \theta^1, \theta^2, \ldots$ such that the sequence forms a Markov chain. That is for any $n \geq 0$ $p(\theta^{n+1}|\theta^n, \ldots, \theta^0, \mathbf{X}) = p(\theta^{n+1}|\theta^n, \mathbf{X})$. Subject to certain regularity conditions to be discussed later in this section the distribution $p(\theta^n|\theta^0, \mathbf{X})$ converges to a unique stationary distribution $\pi(\theta|\mathbf{X})$. If this stationary distribution is equal to $p(\theta|\mathbf{X})$ then we can estimate $\mathbb{E}\{f(\theta)|\mathbf{X}\}$ as

$$\mathbb{E}\{f(\theta)|\mathbf{X}\} \simeq \frac{1}{N - N_{burnin}} \sum_{n=N_{burnin}+1}^{N} f(\theta^n)$$

where N_{burnin} is the number of samples that must be drawn before the distribution converges to the stationary distribution.

There are a number of techniques to draw samples from a Markov chain whose stationary distribution is the target distribution $p(\theta|\mathbf{X})$. Here we will present the *Gibbs sampler.*

The Gibbs Sampler

If $\theta = \{\theta_1, \ldots, \theta_m\}$ we can draw samples from $p(\theta|\mathbf{X})$ by drawing samples from the full conditional distributions of the individual elements of θ. The Gibbs sampler draws samples from $p(\theta|\mathbf{X})$ as follows

Initialize $\theta^0 = \{\theta_1^0, \ldots, \theta_m^0\}$;
for $n = 1$ **to** N **do**
  $\theta_1^n \sim p(\theta_1|\theta_2^{n-1}, \ldots, \theta_m^{n-1}, \mathbf{X})$;
  $\theta_2^n \sim p(\theta_2|\theta_1^n, \theta_3^{n-1}, \ldots, \theta_m^{n-1}, \mathbf{X})$;
  $\theta_3^n \sim p(\theta_3|\theta_1^n, \theta_2^n, \theta_4^{n-1}, \ldots, \theta_m^{n-1}, \mathbf{X})$;
  $\vdots$
  $\theta_m^n \sim p(\theta_m|\theta_1^n, \ldots, \theta_{m-1}^n, \mathbf{X})$;
end

Algorithm 1: The Gibbs Sampler

Convergence Issues

The distribution $p(\theta^n|\theta^0, \mathbf{X})$ converges to a stationary distribution $\pi(\theta|\mathbf{X})$ if

1. The Markov chain is irreducible, that is one can reach any state with positive probability from any other state.

2. The Markov chain is aperiodic. This prevents the chain from being trapped in cycles.

3. The Markov chain is positive recurrent. That is if the initial sample is drawn from the stationary distribution then all other samples are drawn from the stationary distribution as well.

If the above conditions are satisfied then for a given target distribution $p(\theta|\mathbf{X})$ we must show that $\pi(\theta|\mathbf{X}) = p(\theta|\mathbf{X})$.

In practice convergence of the Markov chain is determined by the visual inspection of plots and by using convergence diagnostics [20, chapter 8]. This is the approach employed in [21].

2.2 Speech Enhancement

In real world acoustic environments, speech quality and intelligibility are affected by noise which may come from various sources depending on the environment. Speech enhancement algorithms are aimed at improving the perceptual quality of speech for human listeners or improving the performance of speech based applications such as speaker recognition. Given input speech which is corrupted by noise, the speech enhancement algorithm exploits the characteristics of both the speech and noise in order to mitigate the effects of noise. The output of the algorithm is 'cleaner' speech with improved perceptual quality. It is also important that the algorithm does not introduce any distortions which may in some cases be more annoying to human listeners than the original noise itself.

2.2.1 Types of Noise

A number of speech enhancement algorithms including the ones discussed in this thesis exploit the statistical properties of noise. Broadly speaking noise can be classified as white or non-white (colored). White noise is spectrally flat while non-white noise is not. Furthermore noise can either be stationary or non-stationary. In environments such as an office, the noise sources such as computer fans result in noise that is largely stationary. In a restaurant on the other hand, the noise is non-stationary. The nature of noise influences the difficulty of speech enhancement, in general it is easier to enhance speech in stationary noise as compared to non-stationary environments. However the most robust algorithms should be able to adjust to varying noise conditions.

To further illustrate the nature of noise types encountered in typical speech enhancement applications, we present time waveforms and spectrograms of factory and speech babble noise. This noise is obtained from the NOISEX 92 data set [22]. Figure

2.2 shows a time waveform of factory noise with corresponding spectra estimated from two distinct frames shown in figures 2.3(a) and 2.3(b). These spectra are estimated using the magnitude of the short time Fourier transform (STFT) computed using a 32ms window. From these spectra the non-stationarity of the noise is clear. Similarly, figure 2.4 shows a time waveform of speech babble noise with corresponding spectra estimated from two distinct frames shown in figures 2.5(a) and 2.5(b). The speech babble corresponds to overlapped speech from several speakers and is a good model for noise encountered in a restaurant for example.

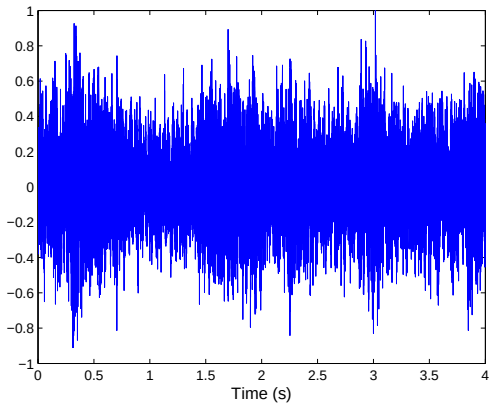
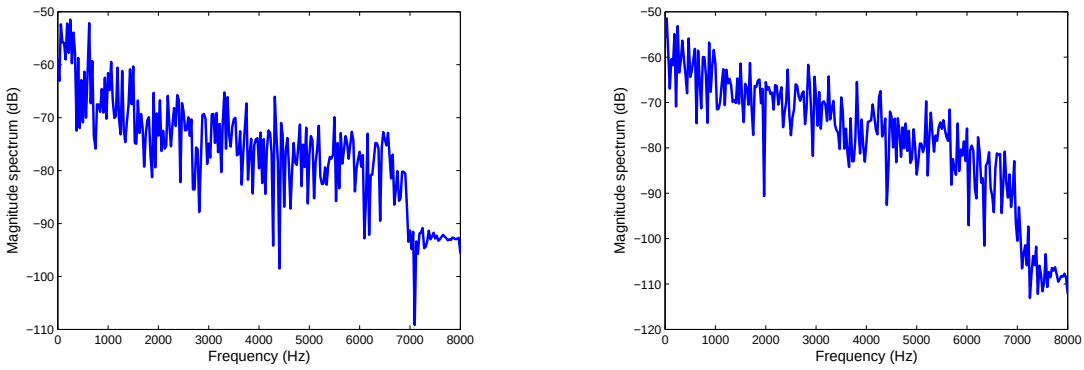


Figure 2.2: Time waveform of factory noise.

2.2.2 Effects of Noise

Human speakers are affected by noise in a number of ways. When talking in crowded restaurants for example, it may be difficult to understand the people one is talking to. Also, it may be difficult to recognize peoples voices when talking over a noisy telephone connection. These difficulties encountered by human beings are also encountered by computers. Applications such as speaker recognition and speech



(a) (b)

Figure 2.3: Approximate spectra of factory noise at two different times.

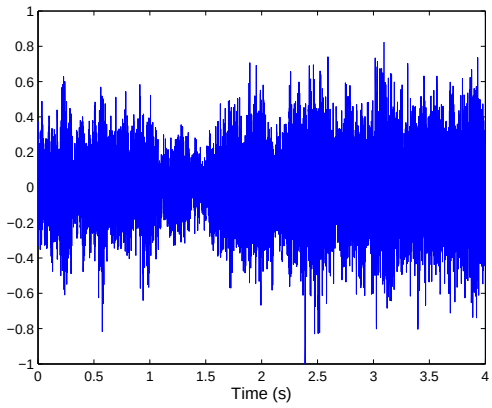
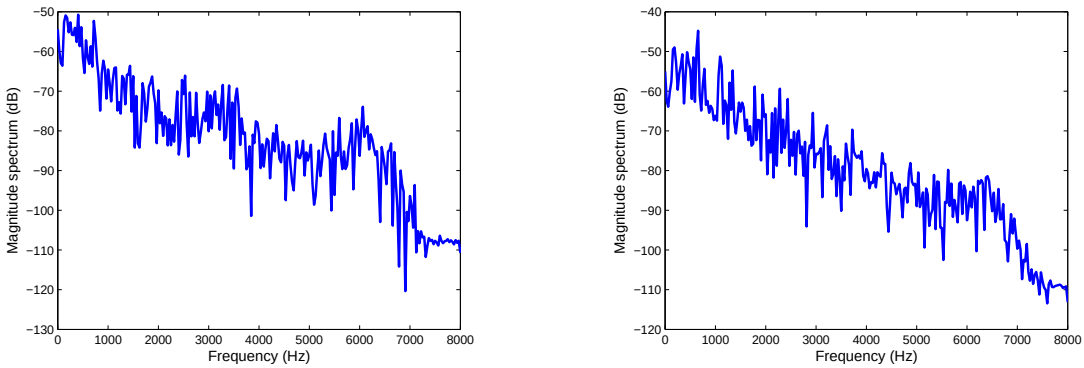


Figure 2.4: Time waveform of speech babble.

recognition are adversely affected by noise. To illustrate this, figure 2.6 shows the recognition rate of a simple speaker identification system in the presence of additive white Gaussian noise as a function of signal to noise ratio (SNR). Here, identification experiments were performed using a 4 speaker library drawn from the TIMIT data set. The test utterances were corrupted using additive white Gaussian noise before identification was done. It can be seen that the performance is worst at high noise levels.



(a) (b)

Figure 2.5: Approximate spectra of speech babble noise at two different times.

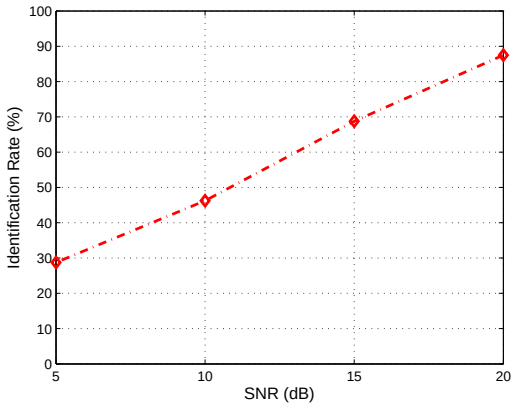


Figure 2.6: Speaker identification performance as a function of SNR.

2.2.3 Speech Enhancement Algorithms

Speech enhancement remains an active area of research (see [23] for a recent review). Speech enhancement algorithms can be broadly classified as spectral-subtractive, subspace or statistical-model based [23]. The algorithms developed in this thesis fall in the statistical-model based category. Spectral-subtractive algorithms are possibly the simplest. They rely on the assumption that the noise is additive. An estimate

of the noise spectrum is subtracted from the observed speech spectrum to obtain an estimate of the clean speech spectrum [24; 25]. Spectral subtractive algorithms are plagued by a number of drawbacks the most severe of which is the introduction of "musical" noise [23, chapter 5]. In some cases, the magnitude of the estimated noise spectrum may exceed the value of the observed speech spectrum resulting in a negative estimate of the clean speech spectrum. This negative values are processed non-linearly by setting them to zero. This leads to peaks in the clean speech spectrum at random frequencies which appear as tones at random frequencies in the time domain [23, chapter 5].

Subspace algorithms rely on the decomposition of the noisy signal vector space into a speech signal subspace and a noise subspace and enhancing the observed signal by projecting it onto the speech signal subspace [26]. Similar ideas are present in the speaker recognition literature and will be discussed further in section 2.3.5.

Statistical Speech Enhancement Algorithms

Statistical-model based algorithms employ probabilistic models for both the speech and noise. The Ephraim-Malah enhancement algorithm [27] and its extensions [28; 29] provide excellent examples of statistical-model based algorithms. Here, the DFT coeffcients of the clean speech and noise are assumed to be Gaussian distributed and a MMSE estimator for the spectral amplitude is derived. A major advantage of the Ephraim-Malah enhancement algorithm is that it does not suffer from the "musical noise" artifact [30].

In [31] the author derives a MMSE estimator for the spectral amplitude using the assumption that the spectral coefficients have super-Gaussian priors. In [32] the author proposes alternatives to the squared error distortion to derive perceptually motivated Bayesian estimators for the spectral amplitude starting with the assumption

that the spectral coefficients of the clean speech are Gaussian distributed.

2.3 Speaker Recognition

In addition to conveying information regarding what a speaker is saying, the speech signal also contains information that can be used to determine who is speaking. This is because the spectrum of the speech signal is influenced by the vocal tract during speech production [33]. The aim of speaker recognition algorithms is to be able to identify speakers from their speech signals using computers. To this end, information relevant to speaker classification must be extracted from the speech signal. Pattern recognition techniques can then be applied to identify the speaker [34].

Speaker recognition can be classified as either speaker identification or speaker verification [35; 1]. In speaker identification, the speech signal is assigned to one of the speakers in a library of known speakers. In speaker verification the input to the system is a speech utterance and a claimed identity, the aim to determine whether the given speech signal was produced by the person claiming to have produced the utterance. Before discussing speaker recognition in greater detail, we will discuss feature extraction and speaker modeling which are key steps in any speaker recognition system.

2.3.1 Feature Extraction

It has been mentioned that the speech signal contains information we can use to identify speakers. However, an important question is how do we obtain this information? What signal processing algorithms will we apply to obtain useful features for speaker recognition? A good starting point in our search for features for speaker identification is the speech spectrum. Speech is highly non-stationary, however, over intervals of 10-30ms we can approximate speech as being stationary. Given a short

speech segment we can then use the speech spectrum as a feature for speaker identification. The speech spectrum can be estimated by taking the magnitude of the FFT of the speech segment.

If we use the magnitude of the FFT as a feature for speaker recognition we can easily run into problems due to the dimension of the feature. For example if our speech signal is sampled at 16kHz and we divide the utterance into 20ms frames, the size of the FFT is 512. This results in features of dimension 257. Learning accurate models of this size is not easy and storing these models is also problematic. We are forced to consider features which compress the relevant information in each speech frame into a feature of reasonable dimension.

Linear Prediction Coefficients

Linear prediction (LP) coefficients provide a good and analytically tractable model for speech [15]. The idea behind LP coefficients is that a given speech sample can be accurately approximated using a linear combination of P previous samples. That is

$$s_n \approx a_1 s_{n-1} + \ldots + a_P s_{n-P} \tag{2.6}$$

The coefficients $a_1, \ldots, a_P$ are constant for a given speech frame. The speech model is given by

$$s_n = \sum_{p=1}^{P} a_p s_{n-p} + \epsilon_n \quad \epsilon_n \sim \mathcal{N}(\epsilon_n; 0, \sigma^2)$$

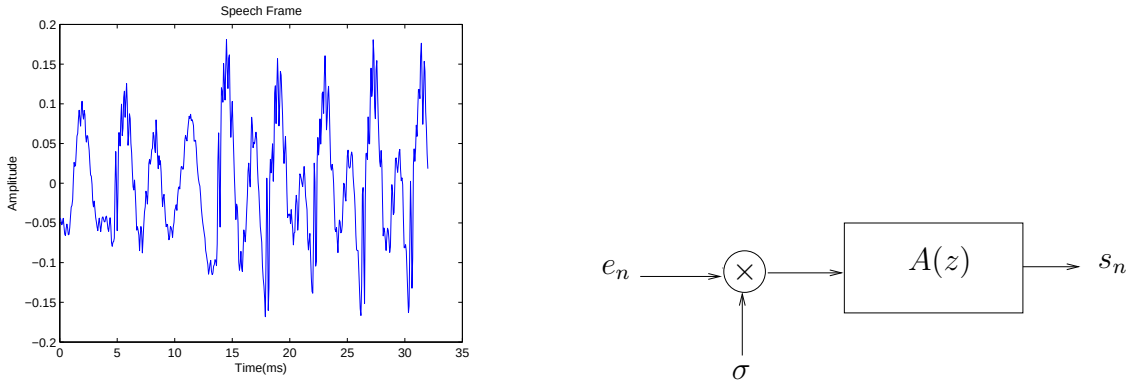
Where we have turned equation (2.6) into an equality by adding the excitation term. In the z-transform domain we have

$$S(z) = \frac{E(z)}{1 - \sum_{p=1}^{P} a_p z^{-p}} = E(z)A(z).$$

where

$$A(z) = \frac{1}{1 - \sum_{p=1}^{P} a_p z^{-p}}.$$

Figure 2.7(b) shows a system diagram representing the LP speech model. In this model the excitation is spectrally shaped by a filter $A(z)$ to produce the speech output. The LP coefficients represent the spectral shaping of the vocal tract and can therefore be used as speaker identification features. Also since the value of P is typically between 8 and 12, this feature is of sufficiently low dimension. Figure 2.8 shows a typical linear prediction spectrum of a speech frame and compares it to a periodogram.



(a) Speech Frame

(b) Linear predictive coding of speech

Figure 2.7: A speech frame (left) and a system diagram representing the LP speech model (right).

Mel Frequency Cepstral Coefficients (MFCCs)

The estimation of LP coefficients is sensitive to noise and these features do not take into account the non-linear processing of sound in the ear. Therefore, other spectral representations of speech are widely used in speech processing. One of the most

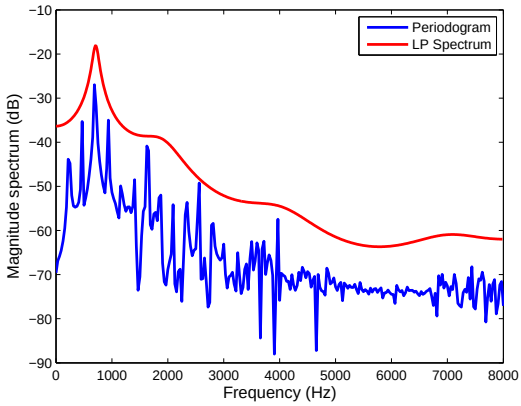


Figure 2.8: Typical Linear Prediction Spectrum

popular spectral parameterizations are Mel Frequency cepstral coefficients (MFCCs) which attempt to capture perceptually relevant features present in the speech signal in a manner similar to the human ear. Figure 2.9 shows how MFCCs are computed from the speech samples. After preemphasis, which amplifies the low frequency components, and windowing, the FFT of the speech frame is computed. Cepstral coefficients are then computed by multiplying the magnitude of the FFT by the triangular filters shown in figure 2.10. The human ear resolves frequencies non-linearly with a finer resolution in the low frequencies. The filters in the lower frequencies have lower bandwidths and are closer together to mimic the way the human ear resolves lower frequencies. The output of the filterbank is decorrelated using the discrete cosine transform to obtain the MFCCs.

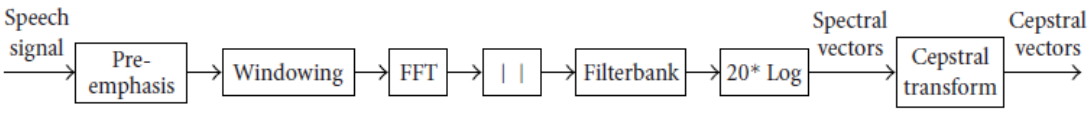


Figure 2.9: System diagram showing how MFCCs are computed from the speech samples (after [1]).

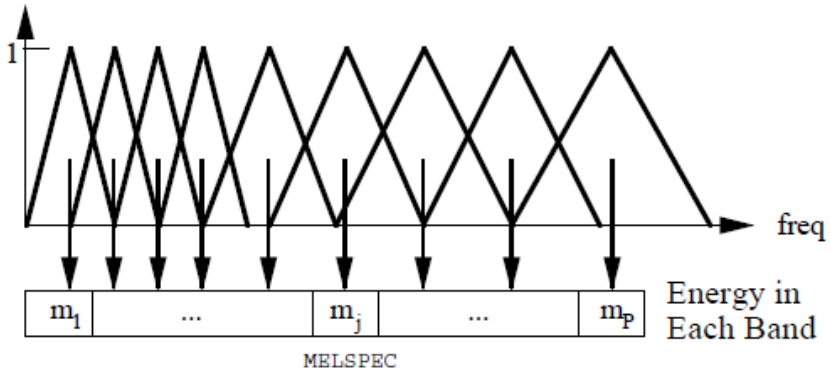


Figure 2.10: The triangular filters in the Mel filter bank (after [36]).

2.3.2 Speaker Modeling

Statistical speaker recognition relies on generative probabilistic models for the features derived from utterances. Gaussian mixture models (GMMs) have proved to be reliable models for speaker recognition and are widely used [35; 37]. GMMs are multivariate generative models that can reliably approximate complicated distributions. Analytically a GMM is given by

$$p(\mathbf{x}) = \sum_{m=1}^{M} \pi_m \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m).$$

Where the mixture coefficients π_m satisfy the conditions

$$\sum_{m=1}^{M} \pi_m = 1, \qquad \pi_m \geq 0.$$

An attractive feature of GMMs is that an efficient algorithm for estimation of the parameters of the distribution given training data exists. Given a training sample of N features $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ we can estimate the parameters $\{\pi_m, \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m\}$ of the distri-

bution using the expectation maximization algorithm discussed in section 2.1.1 [10, chapter 9]. Figure 2.11 shows the use of GMMs to model the real part of the DFT

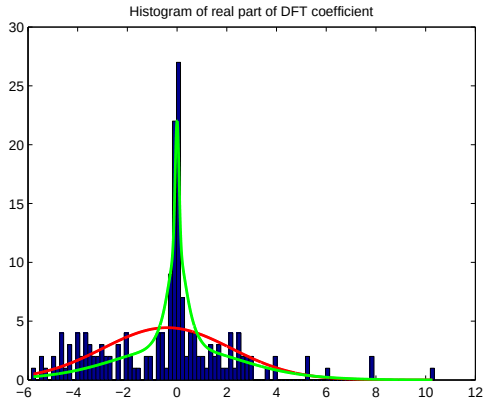


Figure 2.11: Gaussian mixture model for DFT coefficients.

coefficients derived from a speech utterance. Also shown on the figure is the Gaussian distribution with the same mean and variance as the GMM. We see that the GMM captures the peaked nature of the distribution better. In this case a GMM with two mixture coefficients was used.

As already mentioned, to obtain accurate GMMs we must have access to enough training data. In speaker recognition applications, we must have models for all speakers and this means having training data for each speaker. In some cases, the data are inadequate to learn GMMs with an adequate number of mixture coefficients. In this case we can used adapted GMMs [37]. A universal background model (UBM) is trained using data from several speakers and it is then fine tuned using individual data to produce individual speaker models.

Starting with a UBM whose parameters are $\{\pi_m^U, \boldsymbol{\mu}_m^U, \boldsymbol{\Sigma}_m^U\}$ and training data for a given speaker $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ we adapt the means of the UBM by first computing

the alignment of the training data with the UBM distribution. For each mixture component we compute

$$p(m|\mathbf{x}_n) = \frac{\pi_m p_m(\mathbf{x}_n)}{\sum_{m=1}^{M} \pi_m p_m(\mathbf{x}_n)}$$

where

$$p_m(\mathbf{x}_n) = \mathcal{N}(\mathbf{x}_n; \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m).$$

We then compute the following statistics

$$N_m = \sum_{n=1}^{N} p(m|\mathbf{x}_n)$$

$$E_m(\mathbf{x}) = \frac{1}{N_m} \sum_{n=1}^{N} p(m|\mathbf{x}_n)\mathbf{x}_n.$$

The adapted means are given by

$$\boldsymbol{\mu}_m^s = \alpha_m E_m(\mathbf{x}) + (1 - \alpha_m)\boldsymbol{\mu}_m^U$$

where

$$\alpha_m = \frac{N_m}{N_m + r}.$$

r is a relevance factor chosen empirically. The individual speaker model is then given by $\{\pi_m^U, \boldsymbol{\mu}_m^s, \boldsymbol{\Sigma}_m^U\}$ where the mixture coefficients and covariances are the same as the UBM.

2.3.3 Speaker Identification

In speaker identification, the task is to determine the speaker responsible for generating a given utterance. Let us denote the library of known speakers by $\mathcal{L}$. Given a test utterance, we determine which of the $|\mathcal{L}|$ speakers generated the utterance. This is accomplished by deriving features from the utterance and using statistical models

of the speakers to decide on who is speaking. In most systems the features used are MFCCs and the statistical models are GMMs.

The most common decision criterion is the ML criterion. That is once we obtain relevant features from the utterance $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$. The likelihood for each speaker $\ell \in \mathcal{L}$ is computed using

$$\prod_{n=1}^{N} p_\ell(\mathbf{x}_n) = \prod_{n=1}^{N} \sum_{m=1}^{M} \pi_m^\ell \mathcal{N}(\mathbf{x}_n; \boldsymbol{\mu}_m^\ell, \boldsymbol{\Sigma}_m^\ell)$$

And the estimated speaker $\hat{\ell}$ is given by

$$\hat{\ell} = \arg\max_{\ell} \prod_{n=1}^{N} p_\ell(\mathbf{x}_n)$$

Figure 2.12 shows the main components of the speaker identification system.

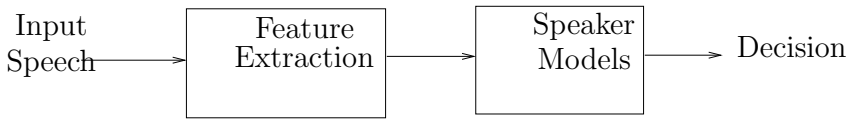


Figure 2.12: Speaker identification system diagram

2.3.4 Speaker Verification

In speaker verification the basic task is to determine whether a given target speaker is speaking in a particular speech segment. Thus given a speech segment X we test the following hypotheses

- H0: X is from speaker S
- H1: X is not from speaker S

Here the target speakers are modelled using speaker specific GMMs and a universal background model (UBM) is used to test the alternate hypothesis H1. The likelihood ratio is compared to a threshold in order to determine which hypothesis is correct. For each trial we compute the score

$$\text{Score} = \log p(\mathbf{X}|\text{TargetModel}) - \log p(\mathbf{X}|\text{UBM}). \tag{2.7}$$

where $\mathbf{X}$ are the features computed from the test utterance. Figure 2.13 shows the main components of the speaker verification system.

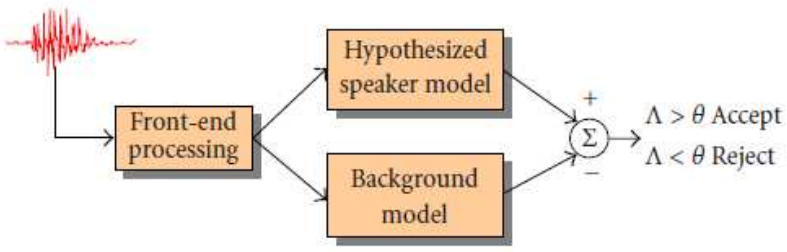


Figure 2.13: Speaker verification system diagram (after [1]).

2.3.5 Robust Speaker Recognition

Current speaker recognition systems are adversely affected by environmental noise and mismatch between training and operation conditions. As a result a significant amount of research continues to focus on improving the performance of speaker identification and verification systems in real world environments where noise and mismatch are unavoidable (for example see [3]).

There are two main approaches to noise robust speaker recognition namely the

model-domain approach and the feature-domain approach [38]. In the model-domain approach, speaker models are adapted to account for the various acoustic environments in which the system will be used [39]. Another model-domain approach involves training different models for different acoustic conditions. In [3] the authors present a system based on multicondition training where the speaker models are derived from speech distorted by different types of noise at various signal-to-noise ratios (SNRs).

In the feature domain approach, the speech or features derived from the speech such as log spectral parameters are enhanced to mitigate the effects of noise on the features. As we have already discussed in section 2.2, speech enhancement is an important area of research and there are a number of techniques such as spectral subtraction and statistical model based speech enhancement algorithms [23]. Cepstral mean subtraction (CMS) and RASTA processing are frequently used to mitigate channel effects in the log spectral domain [40]. However, these techniques fail to exploit any prior information about the features. Recently, methods that rely on prior speech and interference models have been proposed [41; 42]. Using these priors, the clean speech features are estimated using Bayesian techniques. The Algonquin speech enhancement algorithm [43; 44] and some extensions [45; 46; 47; 38] apply a variational inference technique to enhance noisy reverberant speech using a speaker independent mixture of Gaussians speech prior in the log spectral domain.

Another feature domain approach that has recently received significant attention is nuissance attribute projection (NAP) which was originally developed for use in support vector machines[48; 49]. Recent work has extended NAP for use in feature compensation [50]. Here, the space in which the features live is assumed to contain a smaller subspace of nuissance attributes due to noise and channel distortion. A projection matrix applied to the observations can zero components in the direction of the nuissance space. This is similar to the approach introduced by Kenny *et*

al. [51; 52] which is a model-domain technique. Here the means of a background Gaussian mixture model are adapted at enrollment time to determine the speaker dependent means. The technique is similar to the classical maximum *a posteriori* (MAP) adaptation technique used in state of the art speaker verification systems and is known as eigenvoice MAP. In eigenvoice MAP, the background model means are modified using a linear combination of the eigenvoice vectors which span the speaker space.

Cepstral mean subtraction

The idea behind cepstral mean subtraction is that convolution distortion in the time domain becomes additive in the log spectral domain. Thus if we assume that the channel is unchanged during an utterance, the mean of the spectral features will capture the spectra of the channel. Subtracting this mean from all the features corresponding to the utterance compensates for the distortion introduced by the channel.

Compensation of Nuisance Factors

In this section we briefly describe the feature domain intersession compensation (FDIC) technique presented in [50] to compensate for nuisance factors in speaker verification. Speaker models adapted from universal background models are widely used in speaker verification systems [37]. In most cases only the mean vectors of the UBM are adapted leaving the mixture coefficients and variances the same for all models. Therefore each speaker model can be represented by a supervector formed by concatenating all the means. If there are M mixture coefficients and the feature vectors are d dimensional, then the supervector is $M \times d$ elements long. In [50; 53] adaptation of the speaker means is performed in a smaller subspace that captures most of the interspeaker variation and compensates for nuisance variations resulting

from mismatch. We have

$$\boldsymbol{\mu}_s = \mathbf{U}\mathbf{x} + \boldsymbol{\mu}_w \tag{2.8}$$

where $\boldsymbol{\mu}_w$ is the supervector of the UBM model resulting from concatenation of the UBM means, $\boldsymbol{\mu}_s$ is the supervector of the speaker model, $\mathbf{U}$ is a $(M \times d)$ by K low rank projection matrix and $\mathbf{x}$ is a vector of channel factors within the smaller subspace. Equation (2.8) describes how to obtain speaker models that are adapted from the UBM to compensate for mismatch in the model domain. $\mathbf{x}$ is obtained from the observation vectors $\{\mathbf{o}_1, \ldots, \mathbf{o}_T\}$ as follows [50; 53]

$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{b} \tag{2.9}$$

where the elements of $\mathbf{A}$ are given by

$$a_{k,j} = \sum_{m=1}^{M} \Big(\sum_{t=1}^{T} \gamma_m(\mathbf{o}_t) \Big) \mathbf{u}_{k,m}^T \boldsymbol{\Sigma}_m^{-1} \mathbf{u}_{j,m}$$

where $\gamma_m(\mathbf{o}_t)$ is the posterior probability of the mth Gaussian component at the tth observation, $\boldsymbol{\Sigma}_m$ is the covariance matrix of the mth Gaussian component, and $\mathbf{u}_{k,m}$ is the subvector of the kth column of the matrix $\mathbf{U}$ corresponding to the mth Gaussian coefficient.

The elements of $\mathbf{b}$ are given by

$$b_k = \sum_{m=1}^{M} \sum_{t=1}^{T} \gamma_m(\mathbf{o}_t) \mathbf{u}_{k,m}^T \boldsymbol{\Sigma}_m^{-1} (\mathbf{o}_t - \boldsymbol{\mu}_m)$$

where $\boldsymbol{\mu}_m$ is the mean of the mth Gaussian component.

As decribed so far (equation (2.8)), the technique compensates for mismatch in the model domain. To perform feature domain compensation, the observed features are projected to the session independent subspace. Given a set of feature vectors

$\{\mathbf{o}_1, \ldots, \mathbf{o}_T\}$ corresponding to an utterance we have

$$\hat{\mathbf{o}_t} = \mathbf{o}_t - \sum_{m=1}^{M} \gamma_m(\mathbf{o}_t)\mathbf{U}_m\mathbf{x} \tag{2.10}$$

where $\mathbf{U}_m$ is the submatrix of $\mathbf{U}$ obtained by extracting the rows corresponding to the mth mixture component.

In order to obtain the projection matrix $\mathbf{U}$, we require training speech from several speakers recorded under various conditions. For each speaker, we obtain speaker models corresponding to different acoustic conditions via MAP adaptation. For each speaker we then compute the difference between the supervectors from the different conditions. Using these difference supervectors as training data, a K dimensional subspace is learned using probabilistic principal component analysis (PCA) [10].

2.4 Voice Activity Detection

Normal conversational speech contains silent regions and voice activity detection refers to the process of determining the regions of the speech signal that correspond to speech and those that correspond to silent periods. These silent regions are dominated by environmental noise. VAD is important in several speech processing applications such speech recognition, speech enhancement and the transmission of voice over communication channels. In speech recognition, VAD prevents insertion errors which would result if we attempt to recognize words in speech frames dominated by noise. In speech enhancement, several algorithms such as spectral subtraction and the Ephraim-Malah algorithm require an accurate estimate of the noise spectrum. Using the output of the VAD, the noise spectrum is estimated in the noise dominated silence regions [54; 55].

VAD is also very important in the transmission of speech over communication

networks [56]. Communication resources come at a premium and must be conserved. Since the most useful information in a conversation is obtained during the speech dominated regions, during silence, we can transmit information at a lower rate over the network leading to the conservation of vital network capacity.

2.4.1 VAD Algorithms

VAD is a binary classification problem. Given a particular speech frame, the output is a decision classifying the frame as either speech or silence. Thus most algorithms operate on a similar principle: given a speech frame, compute a given parameter and compare this parameter with a threshold. If the parameter corresponding to a given frame is greater than the threshold, classify the frame as speech. Otherwise classify the frame as silence.

Energy Detection

In high SNR conditions, energy thresholding provides a good and simple algorithm for voice activity detection. For the input speech signal frame energy is computed by summing the squares of the sample values. Frames with an energy value x dB lower than the maximum frame energy of the utterance are then classified as silence. The value of x is set empirically. Figure 2.14 shows the VAD result for an utterance drawn from the TIMIT data set in clean conditions and when the utterance is corrupted by additive white Gaussian noise at 0dB. From visual inspection of the results we see that the algorithm works well in clean conditions. However, in noisy conditions, several classification errors occur.

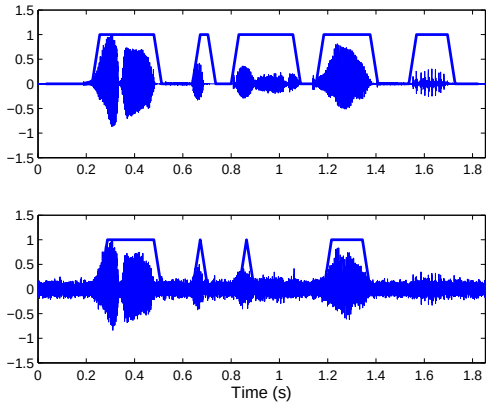


Figure 2.14: Voice activity detection results in clean conditions (top) and at 0dB (bottom) using energy thresholding.

The ITU G.729 Algorithm

Given the important role VAD plays in communication, the International Telecommunications Union (ITU) has adopted a robust VAD algorithm for use in conjunction with voice coding algorithms. We have seen that simple energy thresholding does not work well in noisy conditions which are likely to be encountered in communication scenarios. To improve performance, the ITU G.729 algorithm uses a set of features to classify speech frames. These features are

- The linear prediction spectrum
- Full-band energy
- Low-band (0 to 1KHz) energy
- The zero-crossing rate

The details of the algorithm are presented in [56].

2.5 Data Sets

The experiments reported in this thesis make use of a number of data sets. In this section we briefly describe each of them.

2.5.1 TIMIT

The TIMIT data set consists of broadband recordings of 630 speakers from 8 dialect regions of the United States [57]. Each speaker records 10 phonetically rich sentences. The sampling rate is 16kHz and the resolution is 16 bits per sample. This dataset has been widely used in both speaker and speech recognition experiments. The utterances are of short duration, generally between 3 and 6 seconds each.

2.5.2 MIT Mobile Device Speaker Verification Corpus (MDSVC)

In the MDSVC data set [58], each speaker records 54 utterances in two sessions, one for training and the other for testing. The 54 utterances are recorded in three conditions: in an office, a hallway and a noisy street intersection. 18 utterances are recorded in each environment. Each utterance is approximately two seconds long. Since the data set is designed for speaker verification, the data set includes both target and impostor speakers. There are 48 target speakers with 22 female speakers and 26 male speakers. There are 40 impostors with 23 male and 17 female.

2.5.3 GRID

The GRID corpus [59]: This database was used in the 2006 Interspeech speech separation challenge and it consists of single channel mixtures of simultaneous speech of two speakers at different SNRs with reference to a target speaker. This data set is ideal for simple speech and speaker recognition experiments.

2.5.4 Speaker Recognition Evaluations Data (SRE)

The training speech segments in this data set are continuous conversational excerpts of telephone speech with no silence removal. All the training data is telephone speech with test data from a limited number of speakers being microphone data.

Different training and test conditions differ in the duration of segments (10 sec, 30 sec, 1 side, 3 sides, 8 sides, 16 sides) and whether or not the segment consists of summed channels. Each conversation 'side' is approximately five minutes in length yielding approximately 2.5 minutes of speech from the target speaker. The core condition uses 1 side for both training and testing. [60]

2.5.5 NOIZEUS data set

This data set contains 30 IEEE sentences corrupted by real world noises at various SNRs [23]. The data set includes the clean recordings and the corrupted sentences at 0, 5, 10 and 15dB. The noise types available include train noise, car noise and airport noise.

2.5.6 NOISEX 92

This is a data set of realistic noise sources [22]. The data set includes recordings of speech babble, factory noise and car noise. The sampling rate is 19.98 KHz and the samples are encoded using 16 bit resolution.

2.6 SRE systems and Baseline

In this section we describe the baseline system used in our SRE-2004 evaluation system. We also briefly describe other systems developed by different authors. The basic task is to determine whether a given speaker is speaking in a particular speech segment. The purpose of the NIST SREs is to determine how speaker verification

performance varies as we vary the duration of training and test speech segments. Here, experimental results are reported using data from the 2004 core condition which uses one conversation side for both training and testing. Each conversation side is approximately 5 minutes long [60].

2.6.1 SRE Systems

MIT System

This system consists of seven core systems making use of short term acoustic information, pitch duration, prosodic behaviour , phoneme and word usage. Modeling uses GMMs, SVMs and N-gram language models. The development data consists of Switchboard II phase 1-5 with data from Switchboard II phase 1, 4 and OGI National Cellular Database being used to train UBMs. The baseline system consists of a GMM/UBM system using 19 dimensional MFCCs derived every 10ms using a 20ms window with the frequency band of interest 300-3138Hz. RASTA processing is performed and delta features are computed at +/- two frames. Low energy features are discarded and feature mapping and normalization are performed.

Target speaker models are derived via Bayesian adaptation with only the means being adapted (a relevance factor of 16 is used).

For the 1 side core condition, an EER of 10% is achieved with no gain observed from fusing higher level information to the baseline GMM/UBM system. [61]

SRI System

This system was aimed at incorporating long range stylistic features to improve recognition performance. The development data sets used to train UBMs are Switchboard and Fisher. A GMM/UBM system is used as a baseline with 13 dimensional MFCCs augmented with delta and delta-delta features. For the 1-side training 1-side

testing condition, the baseline achieves an EER of 11.61%. When fused with word N-gram language modeling the EER is 11.44%. Duration features reduce the EER to 8.27%. [62]

LIA System

This system was developed using the ALIZE toolkit. The system uses 16 dimensional Linear Frequency Cepstral Coefficients (LFCCs) derived every 10ms using a 20ms window. The bandwidth is 300-3400Hz. Low energy frames are discarded. Parameters are normalized to zero mean and unit variance. The baseline system uses data from the 2001 and 2002 SREs to train the UBMs. With 128 mixture coefficients, an EER of 11.2% is achieved, this reduces to approximately 10% when 2048 coefficients are used [63; 64]. When the 2004 SRE data are used, the performance degrades slightly and the EER is approximately 13% [63, figure 7].

The TNO system

This system uses perceptual linear prediction coefficients (PLPs) as features. A GMM/UBM system with 512 mixture coefficients achieves an EER of 14.8% on the 1side-1side condition using SRE 2004 data [65].

SRE Baseline System

In our system, the speaker models are GMMs with 512 mixtures and the features are 18 dimensional MFCCs with delta features. We also make use of gender dependent UBMs. Figure 2.15 shows the speaker verification performance for SRE data when the feature domain intersession compensation (FDIC) technique introduced in section 2.3.5 is applied in the feature domain. The intersession subspace has a dimension of 10.

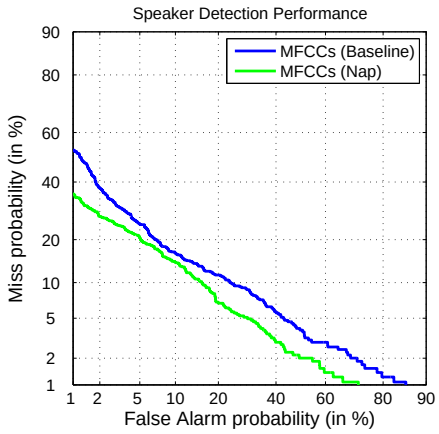


Figure 2.15: Speaker verification system performance for SRE 2004 data

Table 2.1 shows the EER performance of the FDIC system as a function of inter-session subspace dimension. The EER of the Baseline GMM/UBM system is 13.89% which compares favorably with the performance of the TNO system which is 14.8% and the LIA system which is approximately 13%.

Table 2.1: Speaker verification EER (%) for the SRE data set

System	Dimension	EER
MFCCs (Baseline)	-	13.89
FDIC	10	12.04
FDIC	20	12.81

2.6.2 UBM Training

When training universal background models in speaker recognition applications it is important to use an appropriate amount of data and select an appropriate model size. Questions about how many mixture coefficients to use, how many speakers

should provide the training data and how much data to use need to be answered in order to train up effective models. The experiments reported here aim to answer these questions.

The UBMs are gaussian mixture models trained using the EM algorithm. In the experiments, the effectiveness of the UBMs is measured by computing the log likelihood of test data under the trained model. We also measure the log likelihood of the training data at the final and intermediate iterations. The difference between the final log likelihood of the training data and the log likelihood of test data is an important metric which serves as an indicator for overfitting.

In our initial experiment, training data drawn from 40-200 speakers was used to train UBMs of varying size using varying amounts of data. This allows us to determine the optimum number of speakers to draw a certain amount of data for model training. The log likelihood of training and test data was computed every 5 iterations and the EM algorithm was run for 20 iterations. Figure 2.16(a) shows the log likelihood of the training data at the final EM iteration as a function of number of mixture coefficients and amount of training data with the data drawn from 200 speakers. Figure 2.16(b) shows the log likelihood of the test data with the two plots superimposed for comparison in figure 2.16(c). Figure 2.16(d) shows a plot of the histogram of frame scores at the final EM iteration for the training data when the number of mixture coefficients is 512. A gaussian with the same mean and variance is shown for comparison.

Similarity of UBMs

We would like to determine the similarity of the UBMs trained using varying amounts of data and of different size. Figure 2.17 shows a plot of the test loglikelihood for UBMs with 1024 mixture coefficients as a function of amount of training data

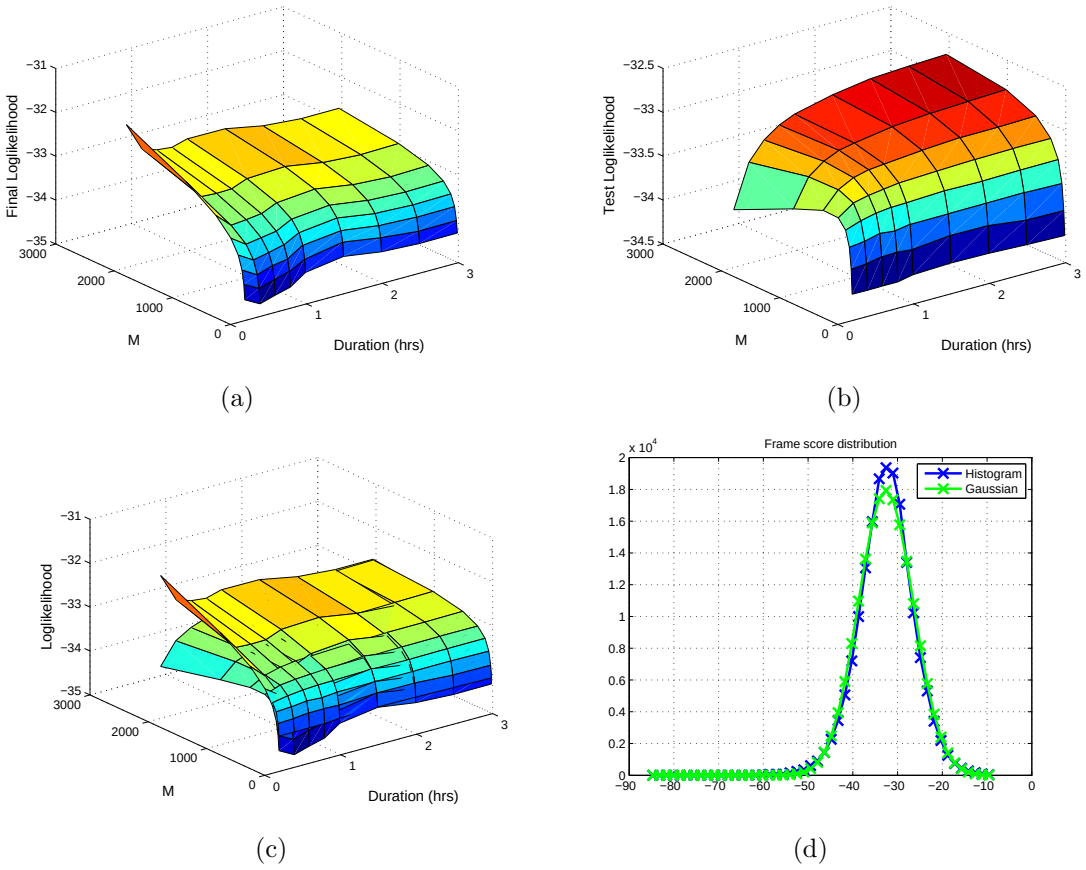


Figure 2.16

drawn from 200 speakers. From the plot we can see that the UBMs obtained using 1.5 - 3.0 hours of data give similar values for the loglikelihood. We would like to know if this means that the UBMs have 'similar' parameters. As an initial metric, we could examine the squared error between the sequence of UBM means. Here, one of the UBMs is taken as the reference and its mixture coefficients ordered from the largest to the smallest. The hypothesis is that mixture components corresponding to the largest mixture coefficients are the best trained and therefore more likely to exhibit consistency between models. Once this ordering is achieved we can determine the minimum squared error between a paricular component mean from the reference

model and the means from the other UBM. Figure 2.18 shows a plot of the minimum squared error as a function of mixture index between the UBM means of models obtained using 2.5 and 3.0 hours of data. As expected the general trend in the plot shows that the mixture components means corresponding to large mixture coefficients are closer in terms of squared distance.

A more reliable metric to measure the similarity of UBMs is the Kullback-Leibler (KL) divergence between the two models. The KL divergence between two distribution $p_1(x)$ and $p_2(x)$ ($D(p_1||p_2)$ is a measure of the distance between two distributions and is defined by [8]

$$D(p_1||p_2) = \int p_1(x) \log \frac{p_1(x)}{p_2(x)} dx.$$

Unfortunately when the two distributions concerned are GMMs, no closed form expression exists for the K-L divergence. However as an initial approximation we can measure the K-L divergence between the individual Gaussian components of the GMMs and determine the minimum divergence between a paricular component mean from the reference model and the means from the other UBM. If

$$p_i(\mathbf{x}) = \frac{1}{2\pi^{N/2}} \frac{1}{|\mathbf{\Sigma}_i|^{1/2}} \exp\{-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_i)^T \mathbf{\Sigma}_i^{-1} (\mathbf{x} - \boldsymbol{\mu}_i)\},$$

then

$$D(p_1||p_2) = \frac{1}{2}\Big\{ \log \frac{|\mathbf{\Sigma}_2|}{|\mathbf{\Sigma}_1|} + \mathsf{Tr}(\mathbf{\Sigma}_2^{-1}\mathbf{\Sigma}_1) - N + (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T \mathbf{\Sigma}_2^{-1} (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) \Big\}.$$

Figure 2.19 shows a plot of the minimum K-L divergence as a function of mixture index between the UBM means of models obtained using 2.5 and 3.0 hours of data. As expected the general trend in the plot shows that the mixture components means corresponding to large mixture coefficients are closer in terms of K-L divergence. However both figure 2.18 and 2.19 would lead us to the conclusion that there still

exist significant difference between the two UBMs despite the fact that they give similar values for the loglikelihood (figure 2.17).

Based on the loglikelihood of the test data alone we can conclude that the model obtained using 3 hours of speech drawn from 200 speakers produces the best model. Using techniques developed in [66] to approximate the K-L divergence between GMMs, we can approximate the K-L divergence between the model obtained using 3 hours of speech and models obtained using 0.2-2.5 hours of speech. We can then explore how the K-L divergence relates to speaker verification performance.

In [66] a variational approximation of the K-L divergence between two GMMs is presented. It is based on maximizing a tractable lower bound on the K-L divergence. Based on this aproach, a closed form expression for the approximate K-L divegence is derived. If

$$p_a(\mathbf{x}) = \sum_{i=1}^{M_a} \pi_i^a \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_i^a, \boldsymbol{\Sigma}_i^a), p_b(\mathbf{x}) = \sum_{i=1}^{M_b} \pi_i^b \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_i^b, \boldsymbol{\Sigma}_i^b)$$

then

$$D_{variational}(p_a||p_b) = \sum_{i=1}^{M_a} \pi_i^a \log \frac{\sum_{j=1}^{M_a} \pi_j^a \exp(-D(p_{a,i}||p_{a,j}))}{\sum_{k=1}^{M_b} \pi_k^b \exp(-D(p_{a,i}||p_{b,k}))}.$$

where

$$p_{a,i} = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_i^a, \boldsymbol{\Sigma}_i^a).$$

Figure 2.20 shows the K-L divergence between the model obtained using 3 hours of speech and models obtained using 0.2-2.5 hours of speech drawn from 200 speakers with 1024 mixture coefficients. Based on this plot we expect the speaker verification perfomance difference to be greatest between the model obtained using 3 hours and the model obtained using 0.2 hours. To test this we performed speaker verification experiments using NIST 2004 speaker recognition evaluation (SRE) data.The SRE data consists of conversational telephone speech.

Speaker models were obtained using MAP adaptation of the UBM models with only the means of the UBM being adapted. We use 13 dimensional MFCCs extracted using a 20ms window with 50% overlap. RASTA processing and CMS is performed. Also, an energy detector is used to discard low energy features. We report results on the core test of the 2004 evaluation where one conversation side is used for both training and testing (1side-1side). For each verification trial, we compute the loglikelihood ratio

$$\mathsf{Score} = \log p(\mathbf{X}|\text{TargetModel}) - \log p(\mathbf{X}|\text{UBM}).$$

where $\mathbf{X}$ are the features. Depending on the score, and the value of a threshold, we will either accept or reject the hypothesis that the test speech was produced by the target speech. As a performance measure we report the Equal error rates obtained by the systems derived from UBMs trained using various amounts of training data. Table 2.2 shows the EER as a function of amount of training data obtained from 200 speakers with 1024 mixture coefficients. As expected the model trained using the most data performs best. However the link between the K-L divergence between the models and the difference in performance of those models in speaker verification is interesting to observe.

Table 2.2: Speaker verification EER (%) for different amounts of training data

Duration (hrs)	0.2	0.6	1.0	2.0	2.5	3.0
EER (%)	27.78	20.68	16.98	16.51	16.05	14.97

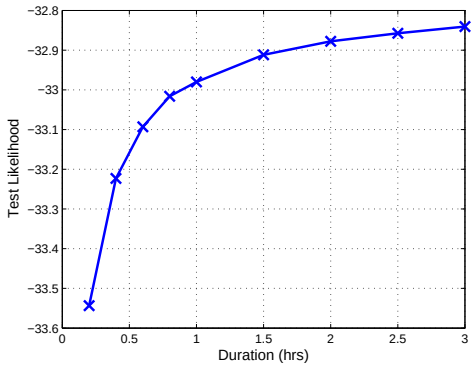


Figure 2.17: Test loglikelihood for UBMs with 1024 mixture coefficients as a function of amount of training data drawn from 200 speakers.

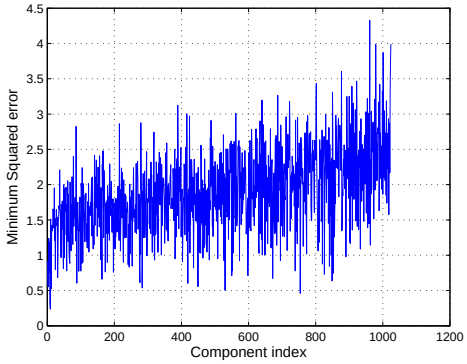


Figure 2.18: The minimum squared error as a function of mixture index between the UBM means of models obtained using 2.5 and 3.0 hours of data. There are 1024 coefficients and the data is drawn from 200 speakers

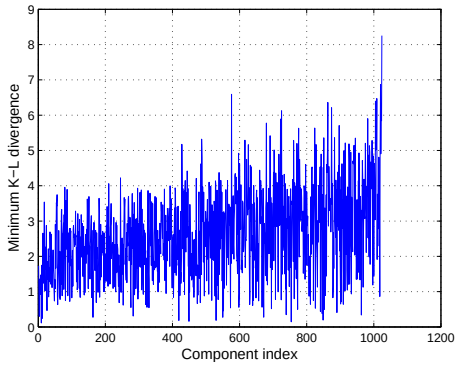


Figure 2.19: The minimum K-L divergence as a function of mixture index between the UBM means of models obtained using 2.5 and 3.0 hours of data. There are 1024 coefficients and the data is drawn from 200 speakers

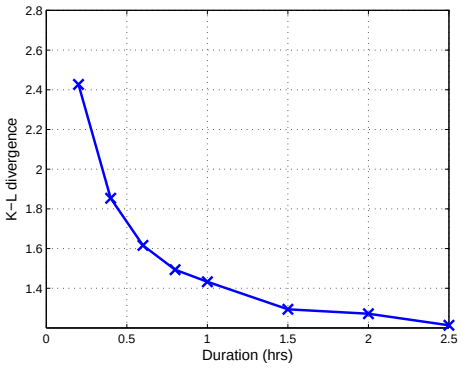


Figure 2.20: K-L divergence between the model obtained using 3 hours of speech and models obtained using 0.2-2.5 hours of speech drawn from 200 speakers with 1024 mixture coefficients.

3. Preliminary Work: A Variational Bayesian Approach to Speech Enhancement

In this chapter we describe our initial attempt at applying variational Bayesian inference to the problem of speech enhancement. This chapter is aimed at illustrating the modelling steps necessary to make VB inference possible. We employ a generalized autoregressive model for speech and attempt to mitigate convolutive distortion by incorporating a channel model. However, due to the nature of the approximate posterior over the clean speech, we are forced to make further approximations to allow for inference. This complications arise due to the nature of the speech model and the attempt to mitigate both additive and convolutive distortion. This motivates the work in chapter 4 where we concentrate on additive distortion and enrich our speech prior by making it speaker dependent. This allows us to develop a joint speech enhancement and speaker identification algorithm that uses speaker dependent priors over the linear prediction coefficients. This algorithm has the added benefit of performing voice activity detection.

3.1 Problem Formulation

Consider a single speech source $\{s_n\}$ observed at a microphone located in a room subject to reverberation as illustrated in figure 3.1. The signal observed at the microphone $\{x_n\}$ is given by

$$x_n = \sum_{k=0}^{L_h-1} h_k s_{n-k} + \eta_n \tag{3.1}$$

where $\mathbf{h} = [h_0, \ldots, h_{L_h-1}]^T$ is the impulse response of the room and $\eta_n \sim \mathcal{N}(\eta_n; 0, \tau_\eta^{-1})$ is additive white Gaussian noise with precision (inverse variance) τ_η . We can write (3.1) compactly as $x_n = \mathbf{h}^T \mathbf{s}_n + \eta_n$ where $\mathbf{s}_n = [s_n, s_{n-1}, \ldots, s_{n-L_h+1}]^T$.

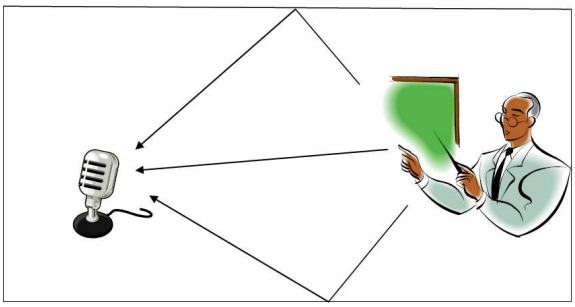


Figure 3.1: Speaker in a reverberent room

3.2 Speech Model

Speech exhibits both temporal correlation and nongaussianity. We attempt to capture these properties by modeling speech as a generalized autoregressive process (GAR) [67; 68]. We have

$$s_n = \sum_{p=1}^{P} a_p s_{n-p} + \epsilon_n = \mathbf{a}^T \mathbf{s}^*_{n-1} + \epsilon_n \tag{3.2}$$

where $\mathbf{a} = [a_1, \ldots, a_P]^T$, $\mathbf{s}^*_{n-1} = [s_{n-1}, \ldots, s_{n-P}]^T$ and the innovations process is modeled as a mixture of Gaussians

$$\epsilon_n \sim \sum_{m=1}^{M} \pi_m \mathcal{N}(\epsilon_n; 0, \tau_m^{-1}). \tag{3.3}$$

Let $\boldsymbol{\pi} = [\pi_1, \ldots, \pi_M]^T$ and $\boldsymbol{\tau} = [\tau_1, \ldots, \tau_M]^T$ then using (3.2) and (3.3) we can write

$$p(s_n | \mathbf{s}^*_{n-1}, \mathbf{a}, \boldsymbol{\pi}, \boldsymbol{\tau}) = \sum_{m=1}^{M} \pi_m \mathcal{N}(s_n; \mathbf{a}^T \mathbf{s}^*_{n-1}, \tau_m^{-1}). \tag{3.4}$$

Following [10, p. 430] we introduce a latent variable $\mathbf{z}_n = [z_{n1}, \ldots, z_{nM}]^T$ which is an $M \times 1$ vector given by the mth column of the identity matrix with probability π_m. That is $\Pr\{z_{nm} = 1\} = \pi_m$. Also $p(\epsilon_n | z_{nm} = 1) = \mathcal{N}(\epsilon_n; 0, \tau_m^{-1})$. We can write

$p(s_n, \mathbf{z}_n|\mathbf{s}^*_{n-1}, \mathbf{a}, \boldsymbol{\pi}, \boldsymbol{\tau}) = p(s_n|\mathbf{z}_n, \mathbf{s}^*_{n-1}, \mathbf{a}, \boldsymbol{\tau})p(\mathbf{z}_n|\boldsymbol{\pi})$ with

$$p(s_n|\mathbf{z}_n, \mathbf{s}^*_{n-1}, \mathbf{a}, \boldsymbol{\tau}) = \prod_{m=1}^{M} \mathcal{N}(s_n; \mathbf{a}^T\mathbf{s}^*_{n-1}, \tau_m^{-1})^{z_{nm}} \tag{3.5}$$

and

$$p(\mathbf{z}_n|\boldsymbol{\pi}) = \prod_{m=1}^{M} \pi_m^{z_{nm}}.$$

If we consider a frame of N source samples $\mathbf{S} = [s_0, \ldots, s_{N-1}]^T$ and the corresponding latent variables $\mathbf{Z} = [\mathbf{z}_0, \ldots, \mathbf{z}_{N-1}]^T$ then

$$p(\mathbf{S}|\mathbf{Z}, \mathbf{a}, \boldsymbol{\tau}) = \prod_{n=0}^{N-1} p(s_n|\mathbf{z}_n, \mathbf{s}^*_{n-1}, \mathbf{a}, \boldsymbol{\tau}). \tag{3.6}$$

Also

$$p(\mathbf{Z}|\boldsymbol{\pi}) = \prod_{n=0}^{N-1} \prod_{m=1}^{M} \pi_m^{z_{nm}}.$$

3.3 Observation Model

From (3.1) we can write $p(x_n|\mathbf{s}_n, \mathbf{h}, \tau_\eta) = \mathcal{N}(x_n; \mathbf{h}^T\mathbf{s}_n, \tau_\eta^{-1})$. Let $\mathbf{X} = [x_0, \ldots, x_{N-1}]^T$ be the observations corresponding to the source samples $\mathbf{S} = [s_0, \ldots, s_{N-1}]^T$. The observation probability model is given by

$$p(\mathbf{X}|\mathbf{S}, \mathbf{h}, \tau_\eta) = \prod_{n=0}^{N-1} p(x_n|\mathbf{s}_n, \mathbf{h}, \tau_\eta). \tag{3.7}$$

3.4 Channel Model

The channel model aims to capture prior knowledge about the room impulse response (RIR). There are a number of techniques used to model room acoustics. In

[2] the authors propose a three parameter model that takes into account the direct path delay Δ, direct path attenuation α and exponential decay time constant τ of the acoustic setting. The coefficients of the RIR are modeled as a Gaussian random vector with zero mean and covariance matrix

$$\mathbf{\Sigma_h} = \alpha \mathsf{diag}\Big(\underbrace{\epsilon, \ldots, \epsilon}_{\Delta \quad \text{terms}}, 1, e^{-\frac{2}{\tau}}, \ldots, e^{-\frac{2(L_h-\Delta-1)}{\tau}} \Big)$$

where ϵ is an appropriate small number.

In this work we find it convenient to work with the precision matrix $\mathbf{\Lambda_h} = \mathsf{diag}(\boldsymbol{\lambda}) = \mathbf{\Sigma_h}^{-1}$ and we write

$$p(\mathbf{h}|\boldsymbol{\lambda}) = \frac{(\prod_{i=0}^{L_h-1} \lambda_i)^{\frac{1}{2}}}{(2\pi)^{\frac{L_h}{2}}} \exp\Big[-\frac{1}{2} \sum_{i=0}^{L_h-1} \lambda_i h_i^2 \Big]. \tag{3.8}$$

Figure 3.2 shows a simulated RIR with $\Delta = 50, \alpha = 1, \tau = 100$, and $\epsilon = 10^{-6}$. The sampling frequency is 16kHz.

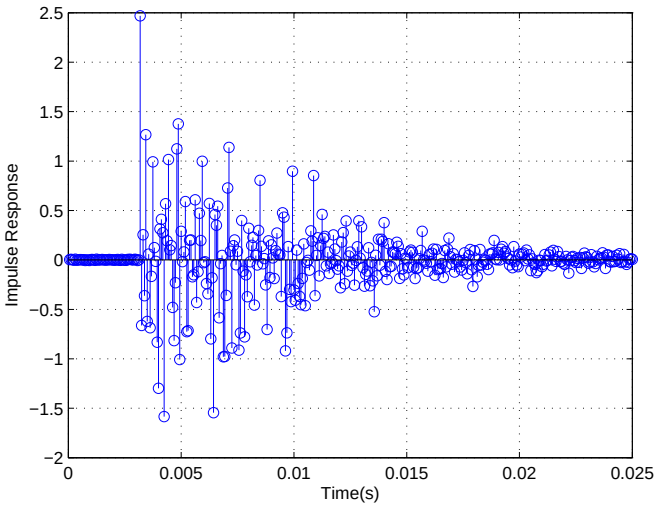


Figure 3.2: Simulated RIR using the three parameter model of [2] with $\Delta = 50, \alpha = 1, \tau = 100$, and $\epsilon = 10^{-6}$.

3.5 Prior Distributions

We now introduce the prior distributions over the parameters $\mathbf{a}, \boldsymbol{\pi}, \boldsymbol{\tau}, \tau_\eta$, and $\boldsymbol{\lambda}$. Where possible we make use of conjugate priors. We define a symmetric Dirichlet prior over $\boldsymbol{\pi}$ that is

$$p(\boldsymbol{\pi}) = \mathsf{Dir}(\boldsymbol{\pi}|\alpha_0) = \frac{\Gamma(M\alpha_0)}{\Gamma(\alpha_0)^M} \prod_{m=1}^{M} \pi_m^{\alpha_0 - 1}$$

where $\Gamma(.)$ is the Gamma function and α_0 is a hyperparameter.

The prior of each precision in $\boldsymbol{\tau}$ is a Gamma distribution with hyperparameters a_0 and b_0. That is

$$p(\tau_m) = \mathsf{Gam}(\tau_m|a_0, b_0) = \frac{1}{\Gamma(a_0)} b_0^{a_0} \tau_m^{a_0 - 1} e^{-b_0 \tau_m}.$$

Following [68] we define the prior over $\mathbf{a}$ to be a zero mean Gaussian with precision matrix given by $\mathsf{diag}([\beta, \ldots, \beta])$. That is

$$p(\mathbf{a}|\beta) = (\frac{\beta}{2\pi})^{\frac{P}{2}} \exp\Big[-\frac{\beta}{2} \mathbf{a}^T \mathbf{a} \Big]$$

with β governed by a Gamma prior $\mathsf{Gam}(\beta|a_\beta, b_\beta)$. Finally we choose Gamma priors over τ_η and each of the entries in $\boldsymbol{\lambda}$ (where we assume $p(\mathbf{h}|\boldsymbol{\lambda})$ is given by (3.8)) with hyperparameters a_η, b_η and a_λ, b_λ respectively.

3.6 VB for Speech Enhancement

In our Bayesian framework the parameters are viewed as realizations of random variables governed by prior distributions. The joint distribution of all random vari-

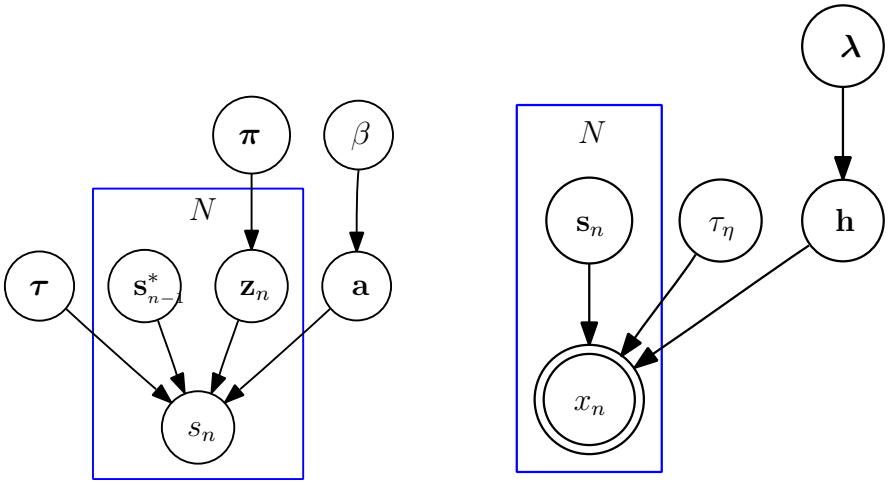
ables in our model is

$$p(\mathbf{X}, \mathbf{S}, \mathbf{Z}, \mathbf{h}, \mathbf{a}, \boldsymbol{\pi}, \boldsymbol{\tau}, \tau_\eta, \boldsymbol{\lambda}, \beta)$$
$$= p(\mathbf{X}|\mathbf{S}, \mathbf{h}, \tau_\eta)p(\mathbf{S}|\mathbf{Z}, \mathbf{a}, \boldsymbol{\tau})p(\mathbf{Z}|\boldsymbol{\pi})p(\mathbf{h}|\boldsymbol{\lambda})p(\mathbf{a}|\beta)p(\boldsymbol{\pi})p(\boldsymbol{\tau})p(\tau_\eta)p(\boldsymbol{\lambda})p(\beta). \quad (3.9)$$

For compactness we represent all the parameters and latent variables as

$$\Theta \stackrel{\text{def}}{=} \{\mathbf{S}, \mathbf{Z}, \mathbf{h}, \mathbf{a}, \boldsymbol{\pi}, \boldsymbol{\tau}, \tau_\eta, \boldsymbol{\lambda}, \beta\}.$$

Figure 3.3 shows directed acyclic graphs illustrating the source and observation models decribed by equation (3.9).



(a) Source Model (b) Observation Model

Figure 3.3: Directed acyclic graphs illustrating the source and observation probabilistic models discussed in section 3.2 and 3.3 respectively.

Our goal is to compute the posterior $p(\Theta|\mathbf{X})$ and in particular $p(\mathbf{S}|\mathbf{X})$ but due to the intractability of this posterior we are forced to consider an approximate Bayesian

technique. Here we consider the application of VB to this problem.

3.6.1 Approximate Posterior

We assume an approximate posterior $q(\Theta)$ that factorizes completely over the parameters and latent variables. That is

$$q(\Theta) = q(\mathbf{S})q(\mathbf{Z})q(\mathbf{h})q(\mathbf{a})q(\boldsymbol{\pi})q(\boldsymbol{\tau})q(\tau_\eta)q(\boldsymbol{\lambda})q(\beta).$$

The dependence of the posterior on the observations $\mathbf{X}$ is implicit. Using (2.4) we obtain expressions for the optimal form of the factors.

We have (see appendix A for details.)

1. $q^*(\tau_\eta) = \mathsf{Gam}(\tau_\eta | a^*_\eta, b^*_\eta)$.
2. $q^*(\beta) = \mathsf{Gam}(\beta | a^*_\beta, b^*_\beta)$.
3. $q^*(\boldsymbol{\tau}) = \prod_{m=1}^{M} \mathsf{Gam}(\tau_m | a^*_m, b^*_m)$.
4. $q^*(\boldsymbol{\lambda}) = \prod_{i=0}^{L_h - 1} \mathsf{Gam}(\lambda_i | a^*_{\lambda i}, b^*_{\lambda i})$.
5. $q^*(\boldsymbol{\pi}) = \mathsf{Dir}(\boldsymbol{\pi} | \boldsymbol{\alpha}^*)$.
6. $q^*(\mathbf{a}) = \mathcal{N}(\mathbf{a}; \boldsymbol{\mu}^*_{\mathbf{a}}, \boldsymbol{\Sigma}^*_{\mathbf{a}})$.
7. $q^*(\mathbf{h}) = \mathcal{N}(\mathbf{h}; \boldsymbol{\mu}^*_{\mathbf{h}}, \boldsymbol{\Sigma}^*_{\mathbf{h}})$.
8. $q^*(\mathbf{Z}) = \prod_{n=0}^{N-1} \prod_{m=1}^{M} \gamma_{nm}^{z_{nm}}$.

We can show that

$$\begin{aligned}\log q^*(\mathbf{S}) &= -\frac{1}{2}\mathbb{E}_{\mathbf{h},\tau_\eta}\Big\{\sum_{n=0}^{N-1}\tau_\eta(x_n-\mathbf{h}^T\mathbf{s}_n)^2\Big\} \\ &- \frac{1}{2}\sum_{n=0}^{N-1}\underbrace{\Big(\sum_{m=1}^{M}\mathbb{E}_{\mathbf{Z}}\{z_{nm}\}\mathbb{E}_{\boldsymbol{\tau}}\{\tau_m\}\Big)}_{\zeta_n^*} \\ &\times \mathbb{E}_{\mathbf{a}}\{(s_n-\mathbf{a}^T\mathbf{s}_{n-1}^*)^2\}+const. \end{aligned} \tag{3.10}$$

If we assume that the posterior distributions $q^*(\mathbf{h})$ and $q^*(\mathbf{a})$ are well approximated by point masses $\delta(\mathbf{h}-\boldsymbol{\mu}_{\mathbf{h}}^*)$ and $\delta(\mathbf{a}-\boldsymbol{\mu}_{\mathbf{a}}^*)$ respectively then we can compute estimates of the sources using a Kalman filter and Rauch-Tung-Striebel (RTS) smoother [7] applied to the observations generated by the following Gaussian linear state space model (GLSSM):

$$\mathbf{s}_n = \mathbf{A}\mathbf{s}_{n-1}+\mathbf{e}_1\epsilon_n \quad \epsilon_n\sim\mathcal{N}(\epsilon_n;0,\zeta_n^{*-1}), \tag{3.11}$$

$$x_n = \mathbf{H}\mathbf{s}_n+\eta_n \quad \eta_n\sim\mathcal{N}(\eta_n;0,\mathbb{E}_{\tau_\eta}\{\tau_\eta\}^{-1}). \tag{3.12}$$

Where $\mathbf{A}$ is the $L_h\times L_h$ state transition matrix, $\mathbf{H}$ is the $L_h\times 1$ observation matrix and $\mathbf{e}_1$ is the first column of the $L_h\times L_h$ identity matrix. We assume that $L_h\geq P$ which is a reasonable assumption in acoustic applications. $\mathbf{A}$ is given by

$$\mathbf{A}=\begin{bmatrix} \mu_{a1}^* & \mu_{a2}^* & \dots & \mu_{aP}^* & 0 & \dots & 0 \\ 1 & 0 & \dots & 0 & 0 & \dots & 0 \\ 0 & 1 & \dots & & & \dots & 0 \\ \vdots & & \ddots & & & & \vdots \\ & & & & & & \\ 0 & \dots & & & & 1 & 0 \end{bmatrix} \tag{3.13}$$

and $\mathbf{H} = [\mu^*_{h0}, \mu^*_{h2}, \ldots, \mu^*_{h(L_h-1)}]$.

3.6.2 Computation of required expectations

Now that we have determined the form of each of the factors in $q(\Theta)$ we can compute the expectations necessary in order to completely characterize the parameters of these factors. We need to compute:

1. $$\mathbb{E}_{\mathbf{a}}\{\mathbf{a}^T\mathbf{a}\} = \mathsf{Tr}(\boldsymbol{\Sigma}^*_{\mathbf{a}}) + \boldsymbol{\mu}^{*T}_{\mathbf{a}}\boldsymbol{\mu}^*_{\mathbf{a}}.$$

   $\mathsf{Tr}(.)$ refers to the trace of the matrix argument. This follows from the expectation of a quadratic form of a Gaussian random vector.

2. $\mathbb{E}_{\mathbf{Z}}\{z_{nm}\} = \gamma_{nm}$ where γ_{nm} is given by (A.10). This follows from the properties of the multinomial distribution [10, Appendix B].

3. $$\begin{aligned}
   \mathbb{E}_{\boldsymbol{\tau}}\{\tau_m\} &= \frac{a^*_m}{b^*_m}. \\
   \mathbb{E}_{\boldsymbol{\tau}}\{\log \tau_m\} &= \psi(a^*_m) - \log b^*_m \\
   \mathbb{E}_{\tau_\eta}\{\tau_\eta\} &= \frac{a^*_\eta}{b^*_\eta} \\
   \mathbb{E}_{\beta}\{\beta\} &= \frac{a^*_\beta}{b^*_\beta} \\
   \mathbb{E}_{\boldsymbol{\lambda}}\{\boldsymbol{\Lambda}\} &= \mathsf{diag}\Big(\frac{a^*_{\lambda 0}}{b^*_{\lambda 0}}, \ldots, \frac{a^*_{\lambda(L_h-1)}}{b^*_{\lambda(L_h-1)}}\Big)
   \end{aligned}$$

   where $\psi(.)$ is the digamma function. These follow from the properties of the Gamma distribution [10, Appendix B].

4. $$\mathbb{E}_{\boldsymbol{\pi}}\{\log \pi_m\} = \psi(a^*_0) - \psi(Ma^*_0)$$

where $\psi(.)$ is the digamma function. This follows from the properties of the Dirichlet distribution [10, Appendix B].

5.
$$\mathbb{E}_{\mathbf{h}}\{h_i^2\} = [\boldsymbol{\Sigma}_{\mathbf{h}}^*]_{ii} + [\boldsymbol{\mu}_{\mathbf{h}}^*]_i^2$$

6. We also require

$$\begin{aligned}\mathbb{E}_{\mathbf{S},\mathbf{h}}\{(x_n - \mathbf{h}^T\mathbf{s}_n)^2\} &= x_n^2 - 2x_n\boldsymbol{\mu}_{\mathbf{h}}^*\mathbb{E}_{\mathbf{S}}\{\mathbf{s}_n\} \\ &+ \boldsymbol{\mu}_{\mathbf{h}}^{*T}\mathbb{E}_{\mathbf{S}}\{\mathbf{s}_n\mathbf{s}_n^T\}\boldsymbol{\mu}_{\mathbf{h}}^{*T} + \mathsf{Tr}(\mathbb{E}_{\mathbf{S}}\{\mathbf{s}_n\mathbf{s}_n^T\}\boldsymbol{\Sigma}_{\mathbf{h}}^*)\end{aligned}$$

and

$$\begin{aligned}\mathbb{E}_{\mathbf{S},\mathbf{a}}\{(s_n - \mathbf{a}^T\mathbf{s}_{n-1}^*)^2\} &= \mathbb{E}_{\mathbf{S}}\{s_n^2\} - 2\boldsymbol{\mu}_{\mathbf{a}}^*\mathbb{E}_{\mathbf{S}}\{s_n\mathbf{s}_{n-1}^*\} \\ &+ \boldsymbol{\mu}_{\mathbf{a}}^{*T}\mathbb{E}_{\mathbf{S}}\{\mathbf{s}_{n-1}^*\mathbf{s}_{n-1}^{*T}\}\boldsymbol{\mu}_{\mathbf{a}}^* + \mathsf{Tr}(\mathbb{E}_{\mathbf{S}}\{\mathbf{s}_{n-1}^*\mathbf{s}_{n-1}^{*T}\}\boldsymbol{\Sigma}_{\mathbf{a}}^*)\end{aligned}$$

If we assume $d = L_h = P$ then $\mathbf{s}_{n-1}^* = \mathbf{s}_{n-1}$. The first and second order moments of $\mathbf{s}_n$ for $n = 0, 1, \ldots, N-1$ can be determined using a Kalman filter using the GLSSM formulation decribed earlier in this section. The Kalman filtering algorithm is presented in algorithm 2 for reference [7, p. 142].

3.7 Experimental Results

In order to test the performance of our algorithm on real speech we use the data set provided for the interspeech 2006 speech separation challenge [59]. In the simulation the clean speech corresponds to the utterance "bin green at a six now". We divide the speech into 20ms frames and assume an AR order of eight and that the number of mixture coefficients is two. The observations were generated by convolving the source with a channel of length 16 and adding white Gaussian noise so that the

for $n = 0, 1, \ldots, N-1$ **do**
  if $n = 0$ **then**
    Initialization;
    $\hat{\mathbf{s}}_{0|-1} = \mathbf{0}_{d\times 1}$;
    $\mathbf{P}_{0|-1} = \mathbf{0}_{d\times d}$;
  else
    Prediction;
    $\hat{\mathbf{s}}_{n|n-1} = \mathbf{A}\hat{\mathbf{s}}_{n-1}$;
    $\mathbf{P}_{n|n-1} = \mathbf{A}\mathbf{P}_{n-1}\mathbf{A}^T + \mathbf{e}_1\tau_{\epsilon,n}^{-1}\mathbf{e}_1^T$;
  end
  $e_n = x_n - \mathbf{h}^T\hat{\mathbf{s}}_{n|n-1}$ Innovation;
  $\mathbf{\Gamma}_n = \mathbf{h}^T\mathbf{P}_{n|n-1}\mathbf{h} + \tau_{\eta,n}^{-1}$ Innovation covariance ;
  $\mathbf{K}_n = \mathbf{P}_{n|n-1}\mathbf{h}\mathbf{\Gamma}_n^{-1}$ Kalman gain;
  $\hat{\mathbf{s}}_n = \hat{\mathbf{s}}_{n|n-1} + \mathbf{K}_n e_n$ State mean estimate;
  $\mathbf{P}_n = (\mathbf{I} - \mathbf{K}_n\mathbf{h}^T)\mathbf{P}_{n|n-1}$ State covariance estimate;
end
with $\mathbf{A}$, $\mathbf{h}$, $\tau_{\eta,n}$, and $\tau_{\epsilon,n}$ as given in section 3.6.1;

Algorithm 2: Kalman Filtering

input SNR was $-2dB$. We use uninformative priors for the Gamma distributions by setting $a = b = 10^{-3}$. We set $\alpha_0 = 10$. We initialize the posterior mean of the AR coefficients to the zero vector and the covariance matrix to the identity matrix. In our initial experiments we assume that the channel is known. Figure 3.4 shows the clean speech segment corresponding to the word "bin" (top), the observed segment (middle) and the enhanced segment (bottom). The SNR of the enhanced signal was $4dB$ after 20 iterations of our algorithm while the SNR was $2.4dB$ after the first iteration. If we use an RTS smoother to enhance the signal assuming the source is i.i.d according to a Gaussian distribution the SNR of the enhanced signal is $2.7dB$. We see that significant SNR improvement is obtained using our algorithm and that the AR coefficient estimates are useful. Also the harmonic structure of the clean speech is clearly visible in the enhanced signal. However, the algorithm fails to recover the utterance at the end of the segment (see figure 3.4).

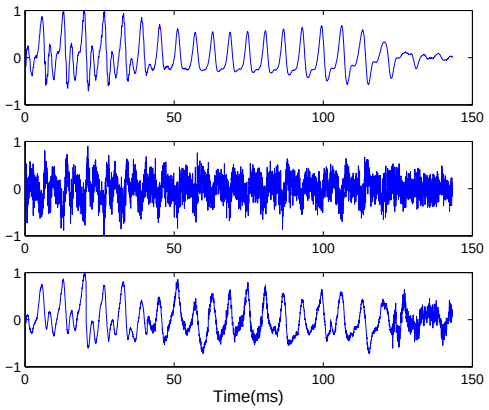


Figure 3.4: The clean speech segment (top), the observed segment (middle) and the enhanced segment (bottom).

This can be explained by observing that the end of the segment corresponds to a silent region of the utterance. We can detect this region by computing the prediction error $\hat{e}_n$ using our AR coefficient estimate $\hat{\mathbf{a}}$. We have

$$\hat{e}_n = \hat{s}_n - \hat{\mathbf{a}}^T \hat{\mathbf{s}}_{n-1}$$

where $\hat{\mathbf{s}}$ is the enhanced signal obtained from the RTS smoother. We can use the following quantity (which we call the normalized mean square error (NMSE)) as a metric to determine silent regions.

$$\mathsf{NMSE} = \frac{\frac{1}{N}\sum_{n=0}^{N-1} \hat{e}_n^2}{\mathsf{Var}(\mathbf{X})}.$$

where $\mathsf{Var}(\mathbf{X})$ is the variance of the noisy observations. Figure 3.5 shows the blockwise variation of the NMSE for the speech segment corresponding to the word 'bin'. We can see that the NMSE peaks in the silent regions at the begining and end of the utterance.

Figure 3.5 suggests a method to enhance the perceptual quality of the enhanced

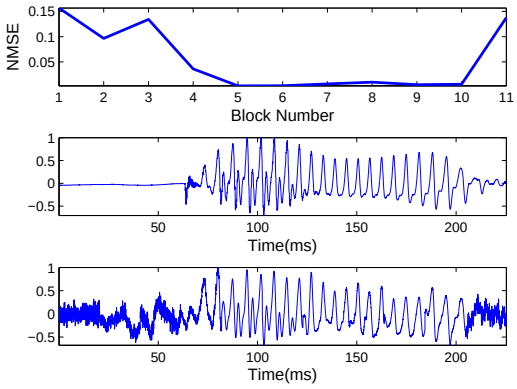


Figure 3.5: The Blockwise NMSE (top), clean speech segment (middle) and the enhanced segment (bottom).

signal. If the NMSE is above a given threshold, we can detect these silent regions and drive the output to zero in these regions. We set the threshold γ_{th} such that the probability that the NMSE is less than or equal to γ_{th} is a given value δ. To determine δ we experimented with sentences from two speakers in the interspeech data set. Listening experiments were performed on the test sentences for $\delta = 0.6, 0.7$, and 0.8. It was observed that with $\delta = 0.7$ and 0.8 the perceptual quality of the enhanced signal was improved. However with $\delta = 0.6$ performance degradation occurred.

3.8 Conclusions

We presented a variational Bayesian algorithm for speech enhancement where we model the speech as a GAR process. Our experimental results verify the appropriateness of our modeling assumptions and we are able to obtain significant SNR improvement when we apply our algorithm to noisy speech. However the algorithm does not enhance the signal in noise dominated silent regions. This problem is addressed by using the estimated AR coefficients and enhanced signal to determine the blockwise prediction error. This quantity is high in the noise dominated silent regions.

Driving the output to zero in these sections results in improved perceptual quality.

4. Joint Speech Enhancement and Speaker Identification Using Variational Bayesian Inference

In the previous chapter, we presented an initial attempt at applying variational Bayesian (VB) inference to the problem of speech enhancement. However, due to the nature of the speech model, which was a generalized autoregressive model, we were forced to make further approximations to the approximate posterior over the clean speech.

In this chapter we extend the work of chapter 3 by using a speaker dependent prior over the linear prediction coefficients which allows us to derive an algorithm for joint speech enhancement and speaker identification. In this case, the computations in the VB algorithm are exact and no futher approximations to the already approximate posterior are necessary.

Our work is built on the intuition that speaker dependent priors would work better than priors that attempt to capture global speech properties. We derive an iterative variational Bayesian algorithm that exchanges information between the speech enhancement and speaker identification tasks. With cleaner speech we are able to make better identification decisions and with the speaker dependent priors we are able to improve speech enhancement performance. We present experimental results using the TIMIT data set which confirm the speech enhancement performance of the algorithm by measuring signal-to-noise (SNR) ratio improvement and perceptual quality improvement via the PESQ score. We also demonstrate the ability of the algorithm to perform voice activity detection (VAD). The experimental results also demonstrate that speaker identification accuracy is improved.

4.1 Problem Formulation

In this work we use a source prior that takes into account the temporal correlation and nongaussianity of speech. Using single channel observations of the noisy speech, the aim is to perform speech enhancement and speaker identification jointly.

We model speech as a time varying autoregressive (AR) process of order P. For a given block k of speech samples $\mathbf{s}^k = [s_1^k, \ldots, s_N^k]^T$ we have (the speech signal is divided into K segments)

$$s_n^k = \sum_{p=1}^{P} a_p^k s_{n-p}^k + \epsilon_n^k = (\mathbf{a}^k)^T \mathbf{s}_{n-1}^k + \epsilon_n^k \tag{4.1}$$

where $\mathbf{s}_n^k = [s_n^k, \ldots, s_{n-P+1}^k]^T$, $\mathbf{a}^k = [a_1^k, \ldots, a_P^k]^T$ and $\epsilon_n^k \sim \mathcal{N}(\epsilon_n^k; 0, (\tau_\epsilon^k)^{-1})$. The signal observed at the microphone is given by

$$r_n^k = s_n^k + \eta_n^k \tag{4.2}$$

where $\eta_n^k \sim \mathcal{N}(\eta_n^k; 0, (\tau_\eta^k)^{-1})$ is additive white Gaussian noise with precision (inverse variance) τ_η^k.

From (4.1) we have

$$\begin{aligned} p(\mathbf{s}^k|\mathbf{a}^k, \tau_\epsilon^k) &= \prod_{n=1}^{N} p(s_n^k|\mathbf{s}_{n-1}^k, \mathbf{a}^k, \tau_\epsilon^k) \\ &= \prod_{n=1}^{N} \mathcal{N}(s_n^k; (\mathbf{a}^k)^T \mathbf{s}_{n-1}^k, (\tau_\epsilon^k)^{-1}). \end{aligned} \tag{4.3}$$

From (4.2) we can write $p(r_n^k|s_n^k, \tau_\eta^k) = \mathcal{N}(r_n^k; s_n^k, \tau_\eta^k)$. If $\mathbf{r}^k = [r_1^k, \ldots, r_N^k]^T$ is the block of noisy observations corresponding to the source samples $\mathbf{s}^k$ the data likelihood is

$$p(\mathbf{r}^k|\mathbf{s}^k, \tau_\eta^k) = \prod_{n=1}^{N} p(r_n^k|s_n^k, \tau_\eta^k) = \prod_{n=1}^{N} \mathcal{N}(r_n^k; s_n^k, \tau_\eta^k). \tag{4.4}$$

To complete the probabilistic formulation we require priors over $\mathbf{a}^k$, τ_ϵ^k, and τ_η^k. The speaker dependence is introduced by the prior over $\mathbf{a}^k$. We model the prior over $\mathbf{a}^k$ for speaker ℓ as a Gaussian mixture model (GMM)

$$p(\mathbf{a}^k|\ell) = \sum_{m=1}^{M_a} \pi_{\ell m}^a \mathcal{N}(\mathbf{a}^k; \boldsymbol{\mu}_{\ell m}^a, \boldsymbol{\Sigma}_{\ell m}^a) \tag{4.5}$$

where $\ell \in \mathcal{L} = \{1, 2, \ldots, |\mathcal{L}|\}$ with $\mathcal{L}$ being the library of known speakers. The parameters $\{\boldsymbol{\mu}_{\ell m}^a, \boldsymbol{\Sigma}_{\ell m}^a, \pi_{\ell m}^a\}$ for the distribution $p(\mathbf{a}^k|\ell)$ are obtained in advance from a corpus of clean speech.

We find it analytically convenient to introduce an indicator variable $\mathbf{z}_a^k$ that is a $M_a|\mathcal{L}| \times 1$ random binary vector that captures both the identity of the speaker and the mixture coefficient 'active' over a given frame. We have

$$p(\mathbf{a}^k|\mathbf{z}_a^k) = \prod_{i=1}^{M_a|\mathcal{L}|} \left[\mathcal{N}(\mathbf{a}^k; \boldsymbol{\mu}_i^a, \boldsymbol{\Sigma}_i^a)\right]^{z_{a,i}^k}. \tag{4.6}$$

The precisions τ_ϵ^k and τ_η^k are assumed to have Gamma priors, that is

$$\begin{aligned} p(\tau_\epsilon^k) &= \text{Gam}(\tau_\epsilon^k; a_\epsilon, b_\epsilon), \\ p(\tau_\eta^k) &= \text{Gam}(\tau_\eta^k; a_\eta, b_\eta). \end{aligned}$$

Now that we have the priors for all the random variables in our model we can write the joint distribution of the observations and parameters. We assume the joint distribution factors as shown in (4.7). We use the notation $\mathbf{x}^{1:K}$ to denote the set $\{\mathbf{x}_1, \ldots, \mathbf{x}_K\}$.

$$\begin{aligned} p(\mathbf{r}^{1:K}, \mathbf{s}^{1:K}, \mathbf{a}^{1:K}, \mathbf{z}_a^{1:K}, \tau_\epsilon^{1:K}, \tau_\eta^{1:K}) = \prod_k \Big\{ & p(\mathbf{r}^k|\mathbf{s}^k, \tau_\eta^k) \\ & \times p(\mathbf{s}^k|\mathbf{a}^k, \tau_\epsilon^k) p(\mathbf{a}^k|\mathbf{z}_a^k) p(\tau_\epsilon^k) p(\tau_\eta^k) \Big\} p(\mathbf{z}_a^{1:K}). \end{aligned} \tag{4.7}$$

The prior $p(\mathbf{z}_a^{1:K})$ is assumed to factor as follows

$$p(\mathbf{z}_a^{1:K}) = p(\mathbf{z}_a^1) \prod_{k=2}^{K} p(\mathbf{z}_a^k | \mathbf{z}_a^{k-1}). \tag{4.8}$$

This allows us to take into account the fact that adjacent speech blocks are likely to originate from the same speaker. In order to completely characterize (4.8) we need to know the speaker transition matrix $\mathbf{A} = [a_{ij}]$ with $a_{ij} = p(\ell^k = i | \ell^{k-1} = j)$ where ℓ^k is the speaker responsible for the kth block and the mixture coefficients $\boldsymbol{\pi}_\ell^a = [\pi_{\ell,1}, \ldots, \pi_{\ell,M_a}]^T$ for all the speakers in the library. The distribution $p(\mathbf{z}_a^k | \mathbf{z}_a^{k-1})$ is then characterized by the $M_a|\mathcal{L}| \times M_a|\mathcal{L}|$ matrix given by

$$\mathbf{T} = \begin{bmatrix} \mathbf{a}_1 \otimes (\boldsymbol{\pi}_\ell^a \mathbf{1}^T) \\ \vdots \\ \mathbf{a}_{|\mathcal{L}|} \otimes (\boldsymbol{\pi}_{|\mathcal{L}|}^a \mathbf{1}^T) \end{bmatrix} \tag{4.9}$$

where $\mathbf{a}_\ell$ is the ℓth row of $\mathbf{A}$, $\mathbf{1}$ is a $M_a \times 1$ vector of all ones, and $\otimes$ represents the Kronecker product. We can now write

$$p(\mathbf{z}_a^k | \mathbf{z}_a^{k-1}) = \prod_{i=1}^{M_a|\mathcal{L}|} \prod_{j=1}^{M_a|\mathcal{L}|} t_{ij}^{z_{a,i}^k z_{a,j}^{k-1}} \tag{4.10}$$

where $\mathbf{T} = [t_{ij}]$. For compactness we represent all the parameters and latent variables as

$$\Theta \stackrel{\text{def}}{=} \{\mathbf{s}^{1:K}, \mathbf{a}^{1:K}, \mathbf{z}_a^{1:K}, \tau_\epsilon^{1:K}, \tau_\eta^{1:K}\}.$$

Figure 4.1 shows a Bayesian network that captures the conditional dependencies between the random variables in our model.

Given the noisy observations, we would like to compute the posterior $p(\mathbf{z}_a^{1:K} | \mathbf{r}^{1:K})$ in order to determine the identity of the speaker responsible for generating the ob-

served speech and the posterior $p(\mathbf{s}^{1:K}|\mathbf{r}^{1:K})$ in order to estimate the clean speech. However due to the intractability of these posteriors we employ approximate Bayesian inference techniques to compute them. The intractability results from the fact that we cannot compute expectations with respect to these posteriors.

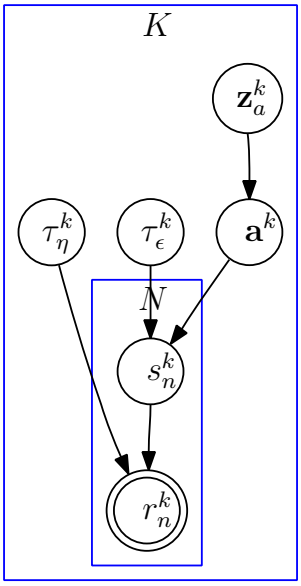


Figure 4.1: Bayesian network showing the conditional dependencies between the random variables in our model.

4.2 Approximate Posterior

Returning to the context of our joint speech enhancement and speaker ID model, we assume an approximate posterior $q(\Theta)$ that factorizes as follows

$$q(\Theta) = \prod_k q(\mathbf{s}^k)q(\mathbf{a}^k)q(\mathbf{z}_a^k)q(\tau_\epsilon^k)q(\tau_\eta^k)$$

The dependence of the posterior on the observations $\mathbf{r}^{1:K}$ is implicit. Using (2.4) we obtain expressions for the optimal form of the factors. We obtain (see appendix B and B.1 for details)

1.

$$q^*(\tau_\eta^k) = \mathsf{Gam}(\tau_\eta^k | a_\eta^*, b_\eta^*) \tag{4.11}$$

with

$$\begin{aligned} a_\eta^* &= a_\eta + \frac{N}{2}, \\ b_\eta^* &= b_\eta + \frac{1}{2}\mathbb{E}_{\mathbf{s}^k}\Big\{ \sum_{n=1}^{N} (r_n^k - s_n^k)^2 \Big\}. \end{aligned}$$

2.

$$q^*(\tau_\epsilon^k) = \mathsf{Gam}(\tau_\epsilon^k | a_\epsilon^*, b_\epsilon^*) \tag{4.12}$$

with

$$\begin{aligned} a_\epsilon^* &= a_\epsilon + \frac{N}{2}, \\ b_\epsilon^* &= b_\epsilon + \frac{1}{2}\sum_{n=1}^{N} \Big\{ \mathbb{E}\{(s_n^k)^2\} - 2\boldsymbol{\mu}_{\mathbf{a}}^{*T}\mathbb{E}\{s_n^k \mathbf{s}_{n-1}^k\} \\ &+ \boldsymbol{\mu}_{\mathbf{a}}^{*T}\mathbb{E}\{\mathbf{s}_{n-1}^k \mathbf{s}_{n-1}^{kT}\}\boldsymbol{\mu}_{\mathbf{a}}^* + \mathsf{Tr}(\mathbb{E}\{\mathbf{s}_{n-1}^k \mathbf{s}_{n-1}^{kT}\}\boldsymbol{\Sigma}_{\mathbf{a}}^*) \Big\}. \end{aligned}$$

$\mathsf{Tr}(.)$ is the trace of the matrix argument.

3.

$$q^*(\mathbf{z}_a^k) = \prod_{i=1}^{M_a|\mathcal{L}|} (\gamma_i^k)^{z_{a,i}^k} \tag{4.13}$$

where

$$\gamma_i^k = \frac{\rho_i^k}{\sum_{i=1}^{M_a|\mathcal{L}|} \rho_i^k}$$

and

$$\begin{aligned}\log \rho_i^k &= -\frac{1}{2}\log|\boldsymbol{\Sigma}_i^a| - \frac{1}{2}(\boldsymbol{\mu}_{\mathbf{a}}^* - \boldsymbol{\mu}_i^a)^T \boldsymbol{\Sigma}_i^{a-1}(\boldsymbol{\mu}_{\mathbf{a}}^* - \boldsymbol{\mu}_i^a) \\ &- \frac{1}{2}\mathsf{Tr}(\boldsymbol{\Sigma}_i^{a-1}\boldsymbol{\Sigma}_{\mathbf{a}}^*) + \sum_{j=1}^{M_a|\mathcal{L}|} \gamma_j^{k-1}\log t_{ij} \\ &+ \sum_{n=1}^{M_a|\mathcal{L}|} \gamma_n^{k+1}\log t_{ni}.\end{aligned}$$

Recall that t_{ij} are the elements of the matrix $\mathbf{T}$ introduced in section 4.1.

4.

$$q^*(\mathbf{a}^k) = \mathcal{N}(\mathbf{a}^k; \boldsymbol{\mu}_{\mathbf{a}}^*, \boldsymbol{\Sigma}_{\mathbf{a}}^*) \tag{4.14}$$

with

$$\begin{aligned}\boldsymbol{\Sigma}_{\mathbf{a}}^* &= \Big[\sum_{n=1}^{N} \frac{a_\epsilon^*}{b_\epsilon^*}\mathbb{E}_{\mathbf{s}^k}\{\mathbf{s}_{n-1}^k \mathbf{s}_{n-1}^{kT}\} + \sum_{m=1}^{M_a|\mathcal{L}|} \gamma_i^k \boldsymbol{\Sigma}_i^{a-1}\Big]^{-1} \\ \boldsymbol{\mu}_{\mathbf{a}}^* &= \boldsymbol{\Sigma}_{\mathbf{a}}^*\Big[\sum_{n=1}^{N} \frac{a_\epsilon^*}{b_\epsilon^*}\mathbb{E}_{\mathbf{s}^k}\{s_n^k \mathbf{s}_{n-1}^k\} + \sum_{m=1}^{M_a|\mathcal{L}|} \gamma_i^k \boldsymbol{\Sigma}_i^{a-1}\boldsymbol{\mu}_i^a\Big]\end{aligned}$$

5. Turning to $q(\mathbf{s}^k)$ we have

$$\begin{aligned}\log q^*(\mathbf{s}^k) &= -\frac{1}{2}\sum_{n=1}^{N} \frac{a_\eta^*}{b_\eta^*}(r_n^k - s_n^k)^2 \\ &- \frac{1}{2}\sum_{n=1}^{N} \frac{a_\epsilon^*}{b_\epsilon^*}\Big((s_n^k)^2 - 2\boldsymbol{\mu}_{\mathbf{a}}^{*T} s_n^k \mathbf{s}_{n-1}^k \\ &+ \mathbf{s}_{n-1}^{kT}\boldsymbol{\mu}_{\mathbf{a}}^*\boldsymbol{\mu}_{\mathbf{a}}^{*T}\mathbf{s}_{n-1}^k + \mathbf{s}_{n-1}^{kT}\boldsymbol{\Sigma}_{\mathbf{a}}^*\mathbf{s}_{n-1}^k\Big) \\ &+ const.\end{aligned} \tag{4.15}$$

As discussed in appendix B, $\mathbb{E}\{\mathbf{s}_n^k\}$, $\mathbb{E}\{\mathbf{s}_n^k \mathbf{s}_n^{kT}\}$ and $\mathbb{E}\{\mathbf{s}_n^k \mathbf{s}_{n-1}^{kT}\}$ can be computed using a Kalman smoother [7].

The forms of the expressions (4.11)-(4.14) are typical in Bayesian computations. They include a contribution from the prior and one from the data. The nature of the prior determines the relative contribution of the data component to the posterior. When the prior is uninformative, the posterior largely depends on the data.

4.3 The VB Algorithm

Armed with closed form expressions for the approximate forms of the posteriors for the parameters $\mathbf{a}^k, \mathbf{z}_a^k, \tau_\epsilon^k$, and τ_η^k and a means to compute the source statistics, we can now present the VB algorithm. The VB algorithm is similar to the expectation maximization (EM) algorithm. It consists of a step similar to the E-step where the current source estimates are determined using a Kalman smoother using the current estimates of the posterior parameters. In the VB-M step, the current source statistic estimates are used to update the parameters of the posterior distributions.

To run the algorithm, the noisy utterance is divided into K segments of N samples each. The posterior parameters for each block are initialized and updated at each iteration. See algorithm 3.

```
Initialize the posterior distribution parameters {a*_η, b*_η, a*_ε, b*_ε, μ*_a, Σ*_a, γ^k_i} for all
blocks;
for n = 1 to Number of Iterations do
    for k = 1, ..., K do
        VB E-step: Run the Kalman smoother to estimate the source statistics
        for block k;
        VB M-Step: Update the posterior parameters for block k using
        (4.11)-(4.14);
    end
end
```

Algorithm 3: VB algorithm

4.4 Experimental Results

In this section we present experimental results that verify the performance of the algorithm. For the simulations we use the TIMIT database which contains recordings of 630 speakers drawn from 8 dialect regions across the USA with each speaker recording 10 sentences [57]. The sampling frequency of the utterances is 16kHz with 16 bit resolution. For our initial experiment a randomly generated library of four speakers was used. In order to train the speaker models we used 8 sentences and used the other 2 for testing. We assume an AR order of 8 with 10 mixture coefficients. To obtain training data for the AR models we divide the speech into 32ms frames and compute the AR coefficients corresponding to these frames using the Levinson-Durbin algorithm. We then use the EM algorithm to determine the GMM parameters. The EM algorithm is run until the relative change in model likelihood is less than 10^{-4}. 100 EM iterations are found to be sufficient. We also train speaker models using Mel Frequency Cepstral Coefficients (MFCCs) to allow us to compare the performance of our algorithm with that obtained using MFCCs. Here we use 13 coefficients obtained from 32ms frames with 50% overlap. Speaker GMMs are trained using the EM algorithm with the number of mixtures set at 32.

We found it necessary to augment the speaker library with a silence model to avoid erroneous classification of silent speech blocks. In our formulation, we treat 'silence' as an additional speaker therefore increasing the library size by one. The silence model consists of a single Gaussian with zero mean and small covariance. An added benefit of this is that we can now use the algorithm to perform voice activity detection (VAD)[54; 55]. We present experimental results comparing the VB algorithm's performance to that obtained using the ITU-G.729 standard [56]. We also need to define the speaker transition matrix $\mathbf{A}$. We assume $\mathbf{A}$ is defined so that the speaker states have a large self transition probability. Also we assume that speaker

changes can occur only after a silent state. That is (silence is considered the fifth speaker)

$$\mathbf{A} = \begin{bmatrix} p & 0 & 0 & 0 & \frac{1-q}{|\mathcal{L}|} \\ 0 & p & 0 & 0 & \frac{1-q}{|\mathcal{L}|} \\ 0 & 0 & p & 0 & \frac{1-q}{|\mathcal{L}|} \\ 0 & 0 & 0 & p & \frac{1-q}{|\mathcal{L}|} \\ 1-p & 1-p & 1-p & 1-p & q \end{bmatrix}. \tag{4.16}$$

The experiments were performed using additive white Gaussian noise as the source of contamination. To run the algorithm, the noisy utterance was divided into 32ms segments ($N = 512$). The hyperparameters of the gamma distributions were $a = b = 10^{-6}$. Thus the prior over the noise variance is uninformative and the noise variance for a particular segment is inferred from the observation. This makes the algorithm robust to changes in noise level from segment to segment. As with any iterative algorithm, initialization is very important and it affects the quality of the final solution. In our experiments, the following initialization scheme was found to work well: We initialize the posterior mean of the AR coefficients to the AR coefficients obtained from the noisy speech blocks. The posterior covariance of the AR coeficients was initialized as the identity matrix. a_η^* and b_η^* are initialized to one for all blocks. b_ϵ^* is initialized to the variance of the AR predection error determined using the noisy speech block and a_ϵ^* is initialized at one. Finally we initialize the parameters of $q(\mathbf{z}_\mathbf{a}^k)$ as $\gamma_i^k = \frac{1}{M_a|\mathcal{L}|}$. The parameters of the transition matrix were set to $p = q = 0.8$. These values were determined by computing the transition probabilities between silence and speech states for several files from the TIMIT data set. The silence and speech states were determined using an energy detector.

Since we update the posterior parameters one at a time, we need to specify a parameter update schedule. The parameter update schedule is as follows:

1. Update the parameters of $q^*(\mathbf{a}^k)$.

2. Update the parameters of $q^*(\tau_\eta^k)$.

3. Update the parameters of $q^*(\tau_\epsilon^k)$.

4. Update the parameters of $q^*(\mathbf{z}_a^k)$.

This schedule was observed in simulation to be numerically stable.

To quantify the algorithm's enhancement performance we measure the input and output SNR. If $\mathbf{s}$, $\mathbf{r}$ and $\hat{\mathbf{s}}$ denote the clean, noisy and enhanced signals respectively, then the input and output SNRs are defined as

$$\begin{aligned} \text{SNR}_{in} &= 20 \log \frac{\|\mathbf{s}\|}{\|\mathbf{s} - \mathbf{r}\|}, \\ \text{SNR}_{out} &= 20 \log \frac{\|\mathbf{s}\|}{\|\mathbf{s} - \hat{\mathbf{s}}\|}. \end{aligned}$$

In order to determine the appropriate number of iterations, we compute the average SNR improvement ($\text{SNR}_{out} - \text{SNR}_{in}$) after the final iteration of the algorithm for all the test utterances in the library for various values of number of iterations. Figure 4.2 shows a plot of SNR improvement versus number of iterations for two values of input SNR: 5 and 10dB. We see that there is minimal SNR improvement after 10 iterations. However, we set the number of iterations at 30 since this is observed to improve speaker identification performance. Figure 4.3 shows the spectrograms and speech waveforms corresponding to the utterance "The shot reverberated in diminishing whiplashes of sound" when corrupted by additive white Gaussian noise at 10dB and enhanced using the algorithm. Using a C implementation of the algorithm we can process a 3 second utterance in approximately 10 seconds when the algorithm is run for 10 iterations. A C implementation of the Ephraim-Malah enhancement algorithm processes the same utterance in less than one second.

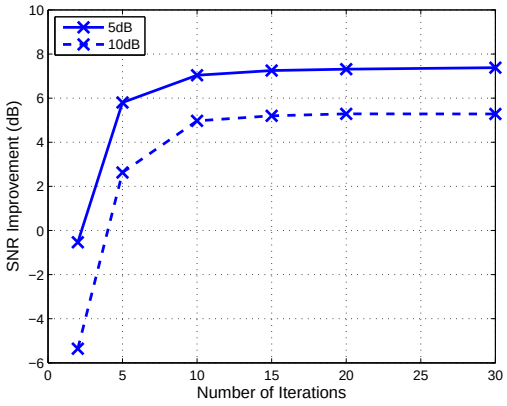


Figure 4.2: SNR improvement ($\text{SNR}_{out} - \text{SNR}_{in}$) after the final iteration of the algorithm versus number of iterations.

To measure the identification performance of the algorithm the posterior speaker probabilities are computed from the approximate posterior $q(\mathbf{z}_{\mathbf{a}}^k)$. The posterior probability that a given block was generated by a given speaker is

$$q(\ell^k = i) = \sum_{j=(i-1)M_a+1}^{iM_a} \gamma_j^k$$

for $i \in \mathcal{L}$. For each block, the most likely speaker is determined via the maximum *a posteriori* (MAP) criterion using the posterior distribution $q(\ell^k)$. That is

$$\hat{\ell}^k = \arg\max_{i \in \mathcal{L}} q(\ell^k = i).$$

In order to assign a speaker to the entire utterance we compute

$$q(\ell = i) \propto \exp\Big(\sum_{k=1}^{K} \log q(\ell^k = i)\Big).$$

Figure 4.4(a) shows a segment of the enhanced signal and the blockwise speaker assignment of the sentence "The shot reverberated in diminishing whiplashes of sound"

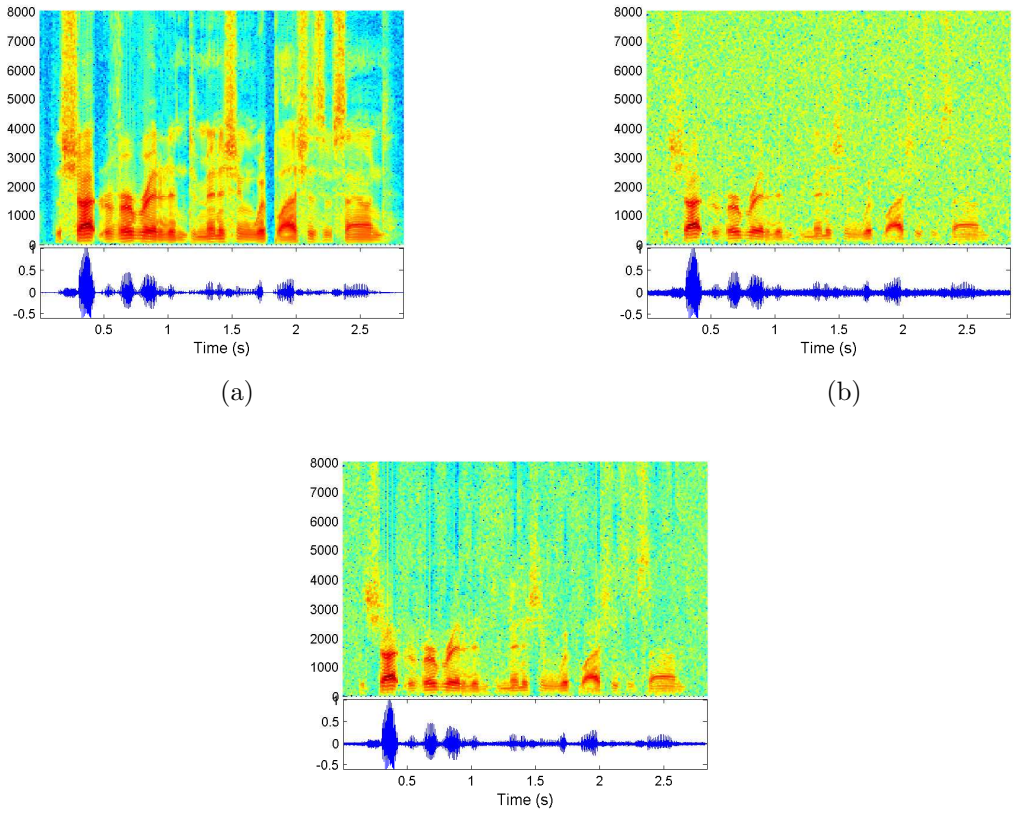
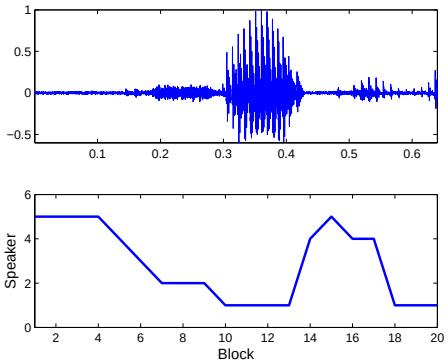


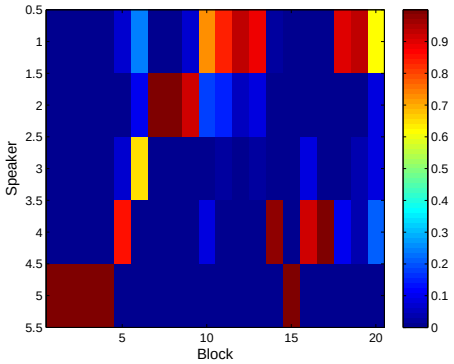
Figure 4.3: Spectrograms and speech waveforms corresponding to the utterance "The shot reverberated in diminishing whiplashes of sound". (a) clean (b) noisy at 10dB (c) enhanced to 14.3dB.

spoken by the first speaker in the library. As before the input SNR is 15dB and the algorithm is ran for 30 iterations. We see that a significant number of blocks are correctly assigned to speaker 1. Also, the initial silence is correctly identified. Figure 4.4(b) shows a plot of the blockwise probabilities $q(\ell^k = i)$ for the segment. This plot allows us to observe the level of certainty of the speaker assignments. Figure 4.5 shows a plot of the speaker posterior for the entire utterance. It is seen that a MAP estimate of the speaker would be correct.

We now present enhancement and recognition results for all the test utterances in



(a) The enhanced signal(top) and the blockwise speaker assignment(bottom).



(b) Blockwise speaker posterior probabilities.

Figure 4.4

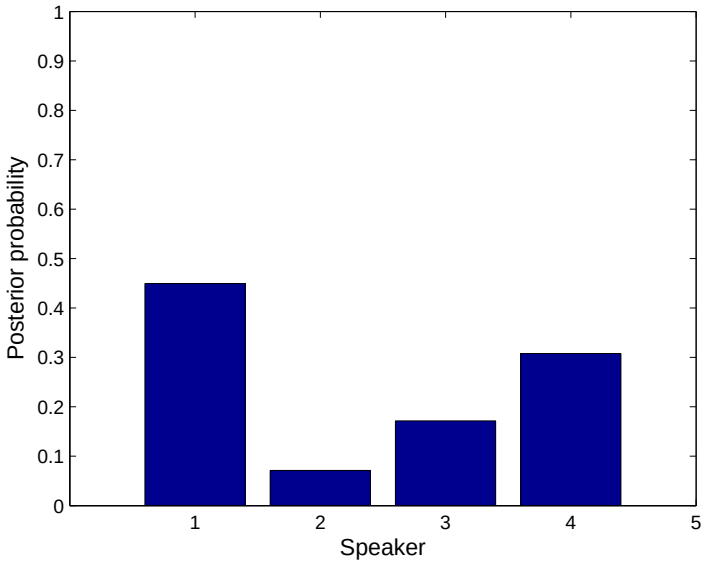
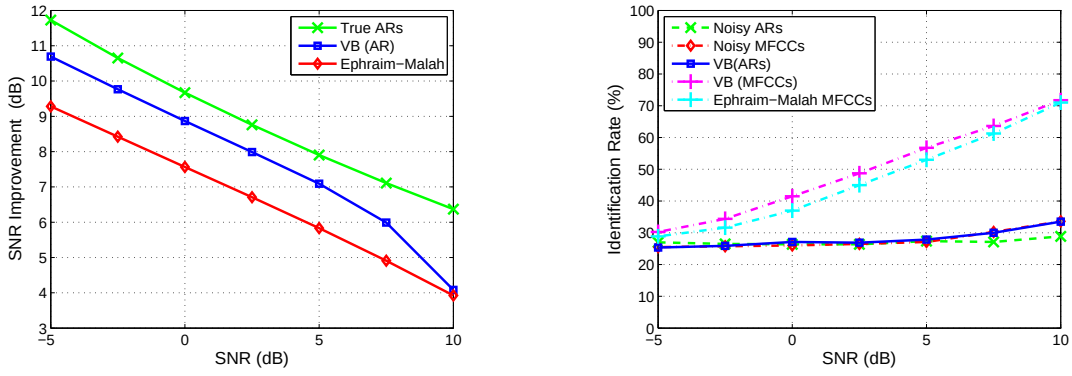


Figure 4.5: Speaker posterior probability.

a library averaged over 100 random libraries of four speakers drawn from the TIMIT database. We performed experiments to investigate the average SNR improvement and speaker recognition rates as a function of input SNR. The algorithm was run

for 30 iterations. Figure 4.6(a) shows a plot of the SNR improvement versus input SNR while figure 4.6(b) shows the recognition rates averaged over 100 random sets of four speakers each. We compare the SNR improvement of the algorithm to the SNR improvement obtained using the Ephraim-Malah enhancement algorithm [27] and using a Kalman smoother when the true AR coefficients are assumed known. That is, we obtain the AR coefficients from the clean speech and use these ARs to enhance the noisy speech using a Kalman smoother. The latter provides an upper bound to the performance of the algorithm since we employ a Kalman smoother in the VB E-step to enhance the noisy speech using the current estimate of the AR coefficients. Since we are working with an estimate of the AR coefficients obtained from noisy observations, we can not outperform the SNR improvement obtained by a Kalman smoother using the true AR coefficients. We also compare the recognition rates of the algorithm to those obtained when 1) AR coeffcients are obtained directly from the noisy signals 2) MFCCs are obtained from the noisy signal 3) MFCCs are obtained from the VB enhanced signal and 4) MFCCs are obtained from the Ephraim-Malah enhanced signal.



(a) (b)

Figure 4.6: SNR improvement versus input SNR (a) and recognition performance (b) for 4 speaker library.

From these results we see that significant SNR improvement is obtained by the algorithm with a maximum SNR improvement of approximately 10dB obtained when the input SNR is -5dB. The VB algorithm outperforms Ephraim-Malah when the input SNR is between -5 and 7.5 dB. When the input SNR is between -5dB and 5dB, the SNR impovement obtained by the VB algorithm is within 1 dB of the performance obtained when the true AR coefficients are known (the upper bound since we have to estimate the AR coefficients and can not outperform a method in which these coefficients are known). Turning to speaker identification results, we see that the VB algorithm which relies on AR coefficients achieves performance comparable to MFCCs obtained directly from the noisy speech. We see that the best identification rates are obtained when MFCCs obtained using the enhanced speech are used. The MFCCs obtained from speech enhanced using the VB algorithm outperform MFCCs from speech enhanced using the Ephraim-Malah algorithm by up to approximately 5%. This shows that the improved performance of the VB algorithm in speech enhancement allows for improved speaker identification.

We are also interested in the perceptual quality of the speech enhanced using our algorithm. To this end we evaluate the Perceptual Evaluation of Speech Quality (PESQ) score of the enhanced utterances. The PESQ score is highly correlated to the mean opinion score (MOS) which is a subjective measure of speech quality [69]. To evaluate the MOS, listeners are asked to rate speech quality on a scale ranging from 1 to 5 with 1 being the worst and 5 the best [23]. In our experiments 80 files corrupted at input SNRs ranging from 0-10 dB were enhanced using both our algorithm and Ephraim-Malah. For each file we compute both the input and output PESQ score. Figure 4.7 shows the PESQ scores for both the VB algorithm and Ephraim-Malah and the best-fit lines. We see that the VB algorithm outperforms the Ephraim-Malah algorithm in terms of perceptual quality.

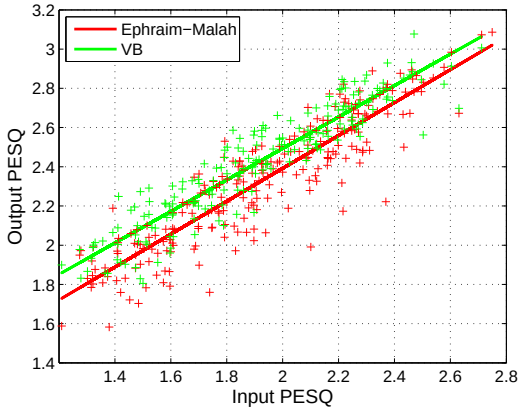


Figure 4.7: Comparison of perceptual quality performance between the VB algorithm and Ephraim-Malah

In order to evaluate the performance of the VB algorithm in more realistic noisy conditions, experiments were performed using the NOIZEUS data set [23]. This data set contains 30 IEEE sentences corrupted by real world noises at varios SNRs. The SNR improvement obtained by the VB algorithm is compared to that obtained using the Ephraim-Malah algorithm. Table 4.1 presents the average SNR improvement for all 30 sentences in the data set at input SNRs ranging from 0dB to 15dB. From the experimental results we see that the VB algorithm outperforms the Ephraim-Malah algorithm in the input SNR range 5dB to 15dB. However at 15dB, both algorithms introduce distortion leading to degradation of the signal.

We now present experimental results that demonstrate the algorithm's performance in voice activity detection (VAD). All blocks assigned to the 'silence' speaker are classified as silence while blocks assigned to other speakers in the library are collectively classified as 'speech'. Figures 4.8-4.9 show the VAD decisions obtained by the VB algorithm and the ITU-G.729 algorithm [56] when the speech is corrupted by additive white Gaussian noise at 10dB and -5dB. We compare the VAD decisions to the ground truth. To obtain the ground truth we perform energy thresholding on the

Table 4.1: SNR improvement for the NOIZEUS data set

Noise Type	Algorithm	Input SNR (dB)			
		0	5	10	15
Train	VB	2.41	2.64	1.86	-0.48
	Ephraim-Malah	3.07	1.00	-1.99	-5.98
Airport	VB	1.10	1.50	1.09	-0.74
	Ephraim-Malah	1.94	0.17	-2.49	-6.11
Car	VB	1.82	2.18	1.64	-0.57
	Ephraim-Malah	5.14	2.07	-1.45	-5.72

clean speech. Any blocks with energy 20dB lower than the maximum energy are classified as silence. To quantify VAD performance, we compare the percentage of speech samples correctly identified as either silence or speech by the VB algorithm and the ITU-G.729 algorithm. Table 4.2 presents the experimental results when 80 speech files were processed at SNRs ranging from -5dB to 10dB by the two algorithms. We see that the VB algorithm outperforms the ITU-G.729 algorithm at all input SNRs considered.

Table 4.2: % of speech samples correctly identified as either speech or silence

Algorithm	Input SNR (dB)			
	-5	0	5	10
VB	59.9	66.7	75.4	83.0
ITU-G.729	51.1	60.4	71.7	79.4

4.5 Conclusions

Experimental results reported in the previous section verify that the proposed VB algorithm does indeed perform joint speech enhancement and speaker identification.

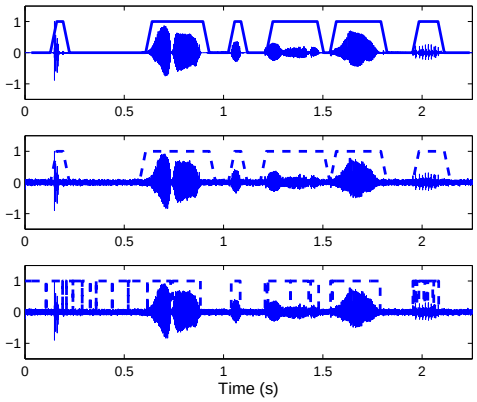


Figure 4.8: Voice activity detection results at 10 dB. Ground truth (top), VB decision with 93% of samples correctly identified (middle) and ITU-G.729 algorithm decision with 70.5% of samples correctly identified (bottom).

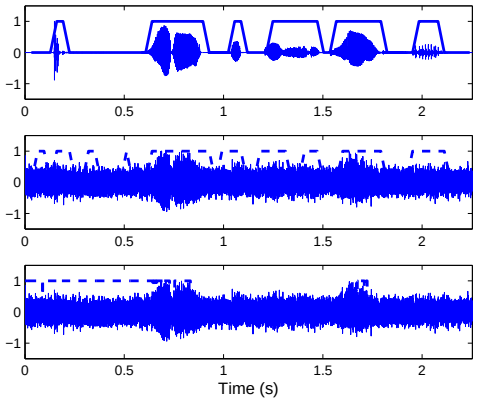


Figure 4.9: Voice activity detection results at -5 dB. Ground truth (top), VB decision with 77% of samples correctly identified (middle) and ITU-G.729 algorithm decision with 42% of samples correctly identified (bottom).

The significant SNR improvement of up to 10dB obtained by the VB algorithm over a wide range of input SNRs shows that speech enhancement is achieved. Furthermore, when the input SNR is between -5dB and 5dB, the SNR impovement obtained by the VB algorithm is within 1 dB of the upper bound obtained when the true AR coefficients are known. The enhancement performance is also confirmed by observing

the time domain speech plots and spectrograms in figure 4.3 and by informal listening tests. Also, the VB algorithm outperforms the Ephraim-Malah algorithm, a standard baseline which has been found to outperform several speech enhancement algorithms in the literature [23, chapter 11], in terms of SNR improvement and perceptual quality as measured using the PESQ score. This result suggests that the full Bayesian treatment employed in the VB algorithm improves speech enhancement performance when compared to an algorithm in which some parameters are assumed known as is the case with the Ephraim-Malah algorithm. In the identification experiments, MFCCs from speech enhanced using the VB algorithm outperform MFCCs from speech enhanced using the Ephraim-Malah algorithm in the input SNR range of -5dB to 10dB. As an added benefit, the VB algorithm allows us to perform VAD. From the experimental results, we see that the VB algorithm outperforms the ITU-G.729 algorithm [56].

In this chapter we have presented a variational Bayesian algorithm that performs speech enhacement and speaker identification jointly. We demonstrate the power of approximate Bayesian methods when applied to complex inference problems. The importance of considering speech enhancement and speaker identification jointly within a Bayesian framework is that we can use rich speaker dependent speech priors to mitigate the effects of noise and therefore improve speaker identification in noisy environments. The experimental results provided verify the performance of the algorithm.

5. Log Spectra Enhancement using Speaker Dependent Priors for Speaker Verification

The experimental results presented in the previous chapter showed the performance gains we can obtain in speech enhancement and speaker identification systems by making use of speaker dependent priors over the speech parameters. The speaker dependent priors over the linear prediction coefficients lead to significant performance improvement in speech enhancement but only moderate improvement in speaker identification. The main cause for this is that the enhancement is not in the ideal domain for speaker recognition. To improve speaker recognition, we should enhance features which capture the spectral properties of the speech signal in a robust manner since this spectrum is speaker dependent. This motivates the work in this chapter where we derive a variational Bayesian algorithm that enhances the log spectra of noisy speech using speaker dependent priors. This algorithm extends prior work by Frey *et al.* where the Algonquin algorithm was introduced to enhance speech log spectra in order to improve speech recognition in noisy environments. Our work is built on the intuition that speaker dependent priors would provide better enhancement and subsequent speaker verification performance than priors that attempt to capture global speech properties.

Working in the log spectral domain offers an advantage over the acoustic domain in the speaker verification setting because we can easily derive Mel frequency cepstral coefficients (MFCCs) from the enhanced log spectra. MFCCs, which were discussed in chapter 2, are features which have been successfully used in speaker recognition. Experimental results using the TIMIT data set and the MIT Mobile Device Speaker Verification Corpus (MDSVC) are presented that demonstrate the algorithm's performance to mitigate both additive noise and mismatch between training and testing

conditions. In both additive Gaussian white noise and realistic noise such as factory noise, we are able to reduce the equal error rate by up to 50% when we compare our system to a standard baseline.

5.1 Problem Formulation

We consider the enhancement of log-spectra of observed speech in order to improve the performance of speaker verification systems by using speaker specific speech priors in the log spectrum domain. In [70] an approximate relationship between the log spectra of observed speech and clean speech is derived. We assume that the clean speech is corrupted by a channel and additive noise. We have

$$y[t] = h[t] * s[t] + n[t], \tag{5.1}$$

where $y[t]$ is the observed speech, $h[t]$ is the impulse response of the channel, $s[t]$ is the clean speech $n[t]$ is the additive noise and $*$ denotes convolution.

Taking the DFT and assuming that the frame size is of sufficient length compared to the length of the channel impulse response we get

$$Y[k] = H[k]S[k] + N[k],$$

where k is the frequency bin index. Taking the logarithm of the power spectrum $\mathbf{y} = \log|Y[:]|^2$ it can be shown that [70]

$$\mathbf{y} \approx \mathbf{s} + \mathbf{h} + \log(\mathbf{1} + \exp(\mathbf{n} - \mathbf{h} - \mathbf{s})) \tag{5.2}$$

where $\mathbf{s} = \log|S[:]|^2$, $\mathbf{h} = \log|H[:]|^2$ and $\mathbf{n} = \log|N[:]|^2$. The approximate observation

likelihood is given by

$$p(\mathbf{y}|\mathbf{s},\mathbf{h},\mathbf{n}) = \mathcal{N}(\mathbf{y}|\mathbf{s}+\mathbf{h}+\log(\mathbf{1}+\exp(\mathbf{n}-\mathbf{h}-\mathbf{s})),\boldsymbol{\psi}) \tag{5.3}$$

where $\boldsymbol{\psi}$ is the covariance matrix of the modelling errors which are assumed to be Gaussian with zero mean.

In this work we assume that we can mitigate channel effects using methods such as mean subtraction and concentrate on mitigating the effects of additive distortion. In this case the observation likelihood becomes

$$p(\mathbf{y}|\mathbf{s},\mathbf{n}) = \mathcal{N}(\mathbf{y}|\mathbf{s}+\log(\mathbf{1}+\exp(\mathbf{n}-\mathbf{s})),\boldsymbol{\psi}).$$

To complete the probabilistic formulation we introduce priors over $\mathbf{s}$ and $\mathbf{n}$. In the speaker verification context, we assume two 'speakers': The target speaker and the 'universal' speaker represented by the universal background model. Thus the prior over $\mathbf{s}$ is given by

$$p(\mathbf{s}|\ell) = \sum_{m=1}^{M_s} \pi_{\ell m}^{s} \mathcal{N}(\mathbf{s};\boldsymbol{\mu}_{\ell m}^{s},\boldsymbol{\Sigma}_{\ell m}^{s}) \tag{5.4}$$

where $\ell \in \mathcal{L} = \{\text{TargetSpeaker}, \text{UBM}\}$

In chapter 4 where we dealt with speaker identification, $\mathcal{L}$ was a library of known speakers. In speaker verification, all speakers are not known before hand and only target speakers are known. Thus we have a library which varies with every test utterance depending on who the target speaker is.

We find it analytically convenient to introduce an indicator variable $\mathbf{z}_s$ that is a $M_s|\mathcal{L}| \times 1$ random binary vector which indicates whether the speech is produced by the target or 'universal' speaker and the mixture coefficient 'active' over a given frame. Thus $\mathbf{z}_s$ takes values from the columns of the $M_s|\mathcal{L}|$-by-$M_s|\mathcal{L}|$ identity matrix.

We have

$$p(\mathbf{s}|\mathbf{z}_s) = \prod_{i=1}^{M_s|\mathcal{L}|} \left[\mathcal{N}(\mathbf{s}; \boldsymbol{\mu}_i^s, \boldsymbol{\Sigma}_i^s)\right]^{z_{s,i}}, \tag{5.5}$$

and

$$p(\mathbf{z}_s) = \prod_{i=1}^{M_s|\mathcal{L}|} (\pi_i^s)^{z_{s,i}}. \tag{5.6}$$

The values of $\boldsymbol{\pi}^s = [\pi_1^s, \ldots, \pi_{M_s|\mathcal{L}|}^s]^T$ are computed from the mixture coefficients of the prior speech models as follows

$$\boldsymbol{\pi}^s = \begin{bmatrix} p\boldsymbol{\pi}_{Tar} \\ (1-p)\boldsymbol{\pi}_{UBM} \end{bmatrix}.$$

where $\boldsymbol{\pi}_{Tar}$ and $\boldsymbol{\pi}_{UBM}$ are the mixture coefficients of the target and UBM GMMs respectively and p is the prior probability that an utterance is from the target speaker. We select p as an uninformative prior for the experiments by setting $p = 0.5$.

We assume that the noise is well modelled by a single Gaussian. That is

$$p(\mathbf{n}) = \mathcal{N}(\mathbf{n}; \boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n). \tag{5.7}$$

This simplifies the derivation of the posterior and is sufficient for the noise types considered here. Extension to the Gaussian mixture model case is straightforward.

We can now write the joint distribution of this model as

$$p(\mathbf{y}, \mathbf{s}, \mathbf{z}_s, \mathbf{n}) = p(\mathbf{y}|\mathbf{s}, \mathbf{n})p(\mathbf{s}|\mathbf{z}_s)p(\mathbf{z}_s)p(\mathbf{n}). \tag{5.8}$$

Inference in this model is complicated due to the nonlinear likelihood term. To allow us to derive a tractable variational inference algorithm we linearize the likelihood as in [43; 44].

Let $g([\mathbf{s}, \mathbf{n}]) = \log(\mathbf{1} + \exp(\mathbf{n} - \mathbf{s}))$. We linearize $g(.)$ using a first order Taylor

series expansion about the point $[\mathbf{s}_0, \mathbf{n}_0]$. We have

$$g([\mathbf{s}, \mathbf{n}]) \approx g([\mathbf{s}_0, \mathbf{n}_0]) + \nabla g([\mathbf{s}_0, \mathbf{n}_0])([\mathbf{s}, \mathbf{n}] - [\mathbf{s}_0, \mathbf{n}_0]) \tag{5.9}$$

And the linearized likelihood is

$$\hat{p}(\mathbf{y}|\mathbf{s}, \mathbf{n}) = \mathcal{N}(\mathbf{y}|\mathbf{s} + g([\mathbf{s}_0, \mathbf{n}_0]) + \mathbf{G}([\mathbf{s}, \mathbf{n}] - [\mathbf{s}_0, \mathbf{n}_0]), \boldsymbol{\psi}) \tag{5.10}$$

Where $\mathbf{G} = [\mathbf{G}_s, \mathbf{G}_n] \stackrel{\text{def}}{=} \nabla g([\mathbf{s}_0, \mathbf{n}_0])$ with

$$\begin{aligned} \mathbf{G}_s &= \mathsf{diag}\Big[\frac{-\exp(n_0^1 - s_0^1)}{1 + \exp(n_0^1 - s_0^1)}, \ldots, \frac{-\exp(n_0^N - s_0^N)}{1 + \exp(n_0^N - s_0^N)}\Big] \\ \mathbf{G}_n &= \mathsf{diag}\Big[\frac{\exp(n_0^1 - s_0^1)}{1 + \exp(n_0^1 - s_0^1)}, \ldots, \frac{\exp(n_0^N - s_0^N)}{1 + \exp(n_0^N - s_0^N)}\Big] \end{aligned}$$

where N is the dimension of the log spectrum feature vector.

We can now derive a variational Bayesian inference algorithm to enhance the observed log spectrum.

5.2 Approximate Posterior

Returning to the context of our model, we assume an approximate posterior $q(\Theta)$ that factorizes as follows

$$q(\Theta) = q(\mathbf{s})q(\mathbf{z}_s)q(\mathbf{n}). \tag{5.11}$$

The factorization used in this work differs from that in Frey *et al.* [43] by enforcing independence between the mixture coefficient indicator variable and the clean log spectra. Thus instead of a mixture of Gaussians posterior over the clean log spectra we have a single Gaussian. Additionally, the algorithm has been designed to jointly verify the speaker and enhance the speech using this information. In [43] the factorization

is

$$q(\Theta) = q(\mathbf{n}) \sum_{m=1}^{M} \rho_m q(\mathbf{s}|m) \tag{5.12}$$

where ρ_m is the posterior probability of the mth mixture component. The optimal forms of the approximate posterior when the factorization (5.12) from [43] is assumed are as follows

$$q(\Theta) = \mathcal{N}(\mathbf{n}; \boldsymbol{\mu}_{\mathbf{n}}^*, \boldsymbol{\Sigma}_{\mathbf{n}}^*) \sum_{m=1}^{M} \rho_m \mathcal{N}(\mathbf{s}; \boldsymbol{\mu}_{\mathbf{s}}^{m,*}, \boldsymbol{\Sigma}_{\mathbf{s}}^{m,*}).$$

The update equations resulting from this factorization are presented in [44].

Using (2.4) we obtain expressions for the optimal form of the factors for the factorization used in this work given by (5.11). We obtain

1.

$$q^*(\mathbf{s}) = \mathcal{N}(\mathbf{s}; \boldsymbol{\mu}_{\mathbf{s}}^*, \boldsymbol{\Sigma}_{\mathbf{s}}^*) \tag{5.13}$$

with

$$\begin{aligned}
\boldsymbol{\Sigma}_{\mathbf{s}}^* &= \Big[\boldsymbol{\psi}^{-1} + \mathbf{G}_s^T \boldsymbol{\psi}^{-1} \mathbf{G}_s + \boldsymbol{\psi}^{-1} \mathbf{G}_s \\
&+ \mathbf{G}_s \boldsymbol{\psi}^{-1} + \sum_{i=1}^{M_s|\mathcal{L}|} \gamma_i \boldsymbol{\Sigma}_i^{s-1}\Big]^{-1} \\
\boldsymbol{\mu}_{\mathbf{s}}^* &= \boldsymbol{\Sigma}_{\mathbf{s}}^* \Big[(\mathbf{I} + \mathbf{G}_s^T) \boldsymbol{\psi}^{-1} (\mathbf{y} - g([\mathbf{s}_0, \mathbf{n}_0]) \\
&- \mathbf{G}_n \boldsymbol{\mu}_{\mathbf{n}}^* + \mathbf{G}_s \mathbf{s}_0 + \mathbf{G}_n \mathbf{n}_0) \\
&+ \sum_{i=1}^{M_s|\mathcal{L}|} \gamma_i \boldsymbol{\Sigma}_i^{s-1} \boldsymbol{\mu}_i^s \Big]
\end{aligned}$$

2.

$$q^*(\mathbf{n}) = \mathcal{N}(\mathbf{n}; \boldsymbol{\mu}_{\mathbf{n}}^*, \boldsymbol{\Sigma}_{\mathbf{n}}^*) \tag{5.14}$$

with

$$\begin{aligned}\boldsymbol{\Sigma}^*_{\mathbf{n}} &= \left[\mathbf{G}_n^T\boldsymbol{\psi}^{-1}\mathbf{G}_n + \boldsymbol{\Sigma}_n^{-1}\right]^{-1} \\ \boldsymbol{\mu}^*_{\mathbf{n}} &= \boldsymbol{\Sigma}^*_{\mathbf{n}}\Big[\mathbf{G}_n^T\boldsymbol{\psi}^{-1}(\mathbf{y} - \boldsymbol{\mu}^*_{\mathbf{s}} - g([\mathbf{s}_0, \mathbf{n}_0]) - \mathbf{G}_s\boldsymbol{\mu}^*_{\mathbf{s}} \\ &+ \mathbf{G}_s\mathbf{s}_0 + \mathbf{G}_n\mathbf{n}_0) + \boldsymbol{\Sigma}_n^{-1}\boldsymbol{\mu}_n\Big]\end{aligned}$$

3.

$$q^*(\mathbf{z}_s) = \prod_{i=1}^{M_s|\mathcal{L}|}(\gamma_i)^{z_{s,i}} \tag{5.15}$$

where

$$\gamma_i = \frac{\rho_i}{\sum_{i=1}^{M_s|\mathcal{L}|}\rho_i}$$

and

$$\begin{aligned}\log\rho_i &= -\frac{1}{2}(\boldsymbol{\mu}^*_{\mathbf{s}} - \boldsymbol{\mu}^s_i)^T\boldsymbol{\Sigma}^{s-1}_i(\boldsymbol{\mu}^*_{\mathbf{s}} - \boldsymbol{\mu}^s_i) \\ &- \frac{1}{2}\log|\boldsymbol{\Sigma}^s_i| - \frac{1}{2}\mathsf{Tr}(\boldsymbol{\Sigma}^{s-1}_i\boldsymbol{\Sigma}^*_{\mathbf{s}}) + \log\pi^s_i.\end{aligned}$$

5.3 The VB Algorithm

To run the algorithm, the observed utterance is divided into K frames and each frame is enhanced. The linearization point is critical to the performance of the algorithm. As in [43; 44] we linearize the likelihood at the current estimate of the posterior mean $[\boldsymbol{\mu}^*_{\mathbf{s}}, \boldsymbol{\mu}^*_{\mathbf{n}}]$. The overall algorithm is summarized in algorithm 4. The posterior mean of the speech log spectrum at the final iteration is used as the enhanced log spectrum of that frame. We then derive MFCCs from the enhanced log spectra and use these to compute scores for each verification trial.

for $k = 1, \ldots, K$ **do**
  Initialize the posterior distribution parameters $\{\boldsymbol{\mu}_{\mathbf{s}}^*, \boldsymbol{\Sigma}_{\mathbf{s}}^*, \boldsymbol{\mu}_{\mathbf{n}}^*, \boldsymbol{\Sigma}_{\mathbf{n}}^*, \gamma_i\}$;
  for $n = 1$ **to** *Number of Iterations* **do**
    Set $[\mathbf{s}_0, \mathbf{n}_0] = [\boldsymbol{\mu}_{\mathbf{s}}^*, \boldsymbol{\mu}_{\mathbf{n}}^*]$;
    Compute $\mathbf{G} = [\mathbf{G}_s, \mathbf{G}_n]$ and $g([\mathbf{s}_0, \mathbf{n}_0])$;
    Update $\{\boldsymbol{\mu}_{\mathbf{s}}^*, \boldsymbol{\Sigma}_{\mathbf{s}}^*, \boldsymbol{\mu}_{\mathbf{n}}^*, \boldsymbol{\Sigma}_{\mathbf{n}}^*\}$ using (5.13)-(5.14);
    Update γ_i using (5.15);
  end
end

Algorithm 4: VB algorithm

5.4 Computational Complexity

The computational complexity of the algorithm is dominated by the cost to update the posterior distribution of the clean speech log spectra. From equation (3.10) we see that the computation of $\boldsymbol{\mu}_{\mathbf{s}}^*$ is dominated by the term $\sum_{i=1}^{M_s|\mathcal{L}|} \gamma_i \boldsymbol{\Sigma}_i^{s-1} \boldsymbol{\mu}_i^s$. Since the model covariance matrices are diagonal, evaluation of each term has a computational complexity of $O(N)$ where N is the dimension of the log spectral features. Thus each update of the mean parameters has a computational cost of $O(M_s|\mathcal{L}|N)$ which is linear in the number of mixture coefficients.

5.5 Experimental Results

In this section we present experimental results that verify the performance of the algorithm presented in section 5.3. For the simulations we use the TIMIT database and the MIT Mobile Device Speaker Verification Corpus (MDSVC)[58]. The experiments investigate the equal error rate (EER) and detection error tradeoff (DET) curve improvement obtained when the VB log spectral enhancement algorithm is used in speaker verification systems in noisy environments. To obtain noisy speech from TIMIT data, we add additive white Gaussian noise and realistic noise from the NOISEX 92 data set.

The TIMIT data set contains recordings of 630 speakers drawn from 8 dialect regions across the USA with each speaker recording 10 sentences [57]. The sampling frequency of the utterances is 16kHz with 16 bit resolution. In order to train the speaker models we used 8 sentences and used the other 2 for testing. The MIT Mobile Device Speaker Verification Corpus is a data set that is designed to test speaker verification systems with limited enrollment data in noisy acoustic conditions. The speech data consists of recordings of speakers saying ice cream flavor phrases and names. The recordings are done in an office, hallway and street intersection in order to provide realistic noisy speech.

5.5.1 System Descriptions

In this section we present the various verification systems whose performance we measured.

Baseline System

In speaker verification the basic task is to determine whether a given target speaker is speaking in a particular speech segment. Thus given a speech segment X we test the following hypotheses

- H0: X is from speaker S
- H1: X is not from speaker S

Here the target speakers are modelled using speaker specific GMMs and a universal background model (UBM) is used to test the alternate hypothesis H1. The likelihood ratio is compared to a threshold in order to determine which hypothesis is correct. For each trial we compute the score

$$\mathsf{Score} = \log p(\mathbf{X}|\text{TargetModel}) - \log p(\mathbf{X}|\text{UBM}). \tag{5.16}$$

where $\mathbf{X}$ are the features computed from the test utterance. For the baseline system we use 13 dimensional MFCCs generated every 10ms using a 25ms window as features. Using the feature vectors extracted from training speech, we train speaker GMMs with 32 mixture coefficients.

Log Spectrum System

This system uses the log spectrum of the speech frames as features. Log spectra are generated every 10ms using a 25ms window which corresponds to 400 samples at 16kHz. The FFT length is 512 resulting in a feature vector of length 257. Using the feature vectors extracted from training speech, we train speaker GMMs with 8 mixture coefficients.

Variational Bayesian System

For this system, we form a library consisting of the target speaker and the UBM and run algorithm 4 to enhance the noisy log spectra. As with any iterative algorithm, initialization is very important and it affects the quality of the final solution. In our experiments, the following initialization scheme was found to work well: We initialize the posterior mean of the speech log spectrum, $\boldsymbol{\mu}_{\mathbf{s}}^*$, to the log spectrum of the noisy speech frame. The posterior covariance of the speech log spectrum, $\boldsymbol{\Sigma}_{\mathbf{s}}^*$, was initialized as the identity matrix. We initialize the posterior mean of the noise log spectrum, $\boldsymbol{\mu}_{\mathbf{n}}^*$, to the all zero vector. The posterior covariance of the noise log spectrum, $\boldsymbol{\Sigma}_{\mathbf{s}}^*$, was initialized as the identity matrix. Finally we initialize the parameters of $q(\mathbf{z_s})$ as $\gamma_i = \frac{1}{M_s|\mathcal{L}|}$.

Since we update the posterior parameters one at a time, we need to specify a parameter update schedule. The parameter update schedule is as follows:

1. Update the parameters of $q^*(\mathbf{n})$.

2. Update the parameters of $q^*(\mathbf{s})$.

3. Update the parameters of $q^*(\mathbf{z}_s)$.

This schedule was observed in simulation to be numerically stable.

For our experiments, the algorithm was run for 5 iterations and the posterior mean of the speech log spectrum at the final iteration was used as the enhanced log spectrum of that frame. Using the enhanced log spectra for a given utterance, scores for each verification trial are computed using (5.16).

We also derive MFCCs from the enhanced log spectra and use these to compute scores for each verification trial. Thus for the VB system we have two results: one using the enhanced log spectra and the other using the MFCCs derived from these log spectra.

Feature Domain Intersession Compesation (FDIC) System

This system is implemented as described in section 2.3. In order to train the intersession subspace for the TIMIT data experiments, training utterances from the target speakers were corrupted at various SNRs using additive white noise. These training utterances were then used to obtain speaker models via MAP adaptation of a UBM model with 32 mixture coefficients. Using the projection matrix obtained, feature compensation was performed during training and testing. For the MDSVC data set, speaker models from the three recording conditions: an office, hallway and street intersection were used to obtain the projection matrix.

5.5.2 TIMIT Speaker Verification Results

We now turn to experiments aimed at determining the speaker verification performance of the systems in noisy conditions. We assume that the TIMIT data is clean and the SNR only accounts for the additive distortion we introduce. In this work the

input SNR is defined as

$$\text{SNR}_{in} \quad = \quad 10 \log \frac{\sum_t s^2[t]}{\sum_t (s[t] - y[t])^2}.$$

where $s[t]$ is the clean speech and $y[t]$ is the observed speech.

The UBMs were trained using the training data for a random 300 speaker subset of the 630 speaker TIMIT data set. The MFCC UBMs and speaker models had 32 mixtures while the log spectra UBMs and speaker models had 8 mixtures.

The verification experiments were performed with the test utterances corrupted by additive white Gaussian noise at various input SNRs. For each of the 630 speakers we have two test utterances yielding 1260 true trials. To generate impostor trials, a random set of ten speakers was selected from the remaining speakers and the corresponding test utterances used to generate 20 impostor trials per speaker. Thus there are a total of 12600 impostor trials.

For the FDIC experiments, the projection matrix was trained using speaker models derived from the UBM with the training speech degraded by additive white Gaussian noise at SNRs ranging from 0dB to 30dB. For each speaker, 14 models were trained using data degraded at 0, 5, 10 ,20, 21, ..., 30dB. The pairwise differences between the 14 models for all the speakers were used to determine the projection matrix. In order to determine an appropriate subspace dimension, verification experiments were performed using speech corrupted by additive white Gaussian noise at 10dB, 20dB and 22dB and the subspace dimension varied from 2 to 10. Table 5.1 shows the EERs obtained. From these results a 2 dimensional subspace was used for the experiments.

Table 5.2 shows the equal error rates (EER) obtained in our verification experiments at various input SNRs. Figures 5.1-5.3 show the corresponding DET curves. We see that the VB algorithm improves the performance of both the MFCC and log spectral systems. We see that in the range 20-30dB the VB algorithm reduces

Table 5.1: Speaker verification EER (%) as a function of subspace dimension for the TIMIT data set

System	Dimension	SNR (dB)		
		10	20	22
MFCCs (Baseline)	-	43.49	23.25	18.97
FDIC	2	33.17	20.08	17.94
FDIC	5	35.40	25.40	25.32
FDIC	10	36.51	28.81	27.38

the EER by approximately 50% in all cases. For example at 30dB the the MFCC EER is reduced from 6.83% to 3.65%. Also the VB algorithm outperforms the FDIC algorithm.

Table 5.2: Speaker verification EER (%) for the entire TIMIT data set

System	SNR (dB)						
	10	20	22	24	26	28	30
MFCCs (Baseline)	43.49	23.25	18.97	15.24	11.98	9.21	6.83
VB (MFCC)	**26.51**	**11.83**	**9.44**	**7.46**	**6.27**	**4.84**	**3.65**
FDIC	33.25	20.56	17.94	15.63	14.84	12.62	10.56
Log Spectra	49.68	45.16	43.89	43.17	42.06	40.79	40.48
VB (Log Spectra)	44.68	43.57	42.78	42.22	41.51	40.40	40.71

TIMIT Speaker Verification Results in Realistic Noise

We now turn to experiments aimed at demonstrating the performance of the algorithm in realisitic noisy conditions. To this end we add noise from the NOISEX 92 data set [22] to the clean TIMIT data at various SNRs. This data set consists of recordings of various types of noise including factory noise and speech babble. The recordings are sampled at 19.98kHz and it is necessary to resample the recordings

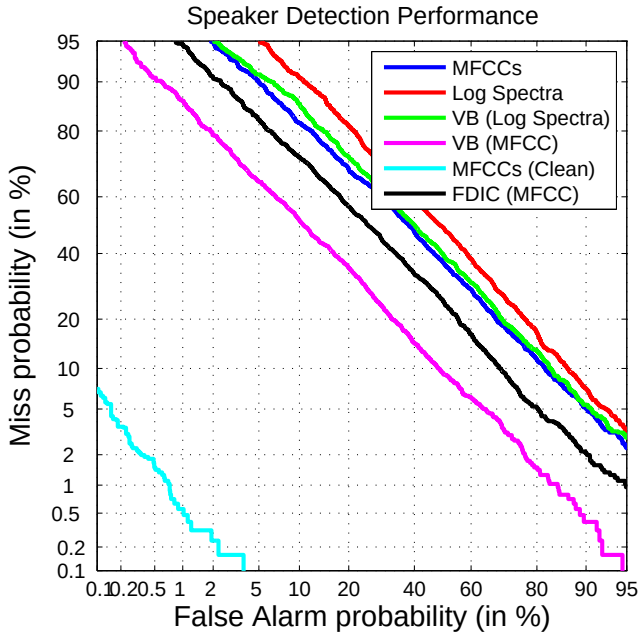


Figure 5.1: Speaker verification performance for the entire TIMIT data set at 10dB. We see that MFCCs obtained from enhanced log spectra yield the best performance.

since TIMIT recordings are sampled at 16kHz.

The experiments using the entire TIMIT data set were repeated using factory noise and speech babble. Table 5.3 shows the equal error rates (EER) obtained in our verification experiments at various input SNRs using factory noise. Table 5.4 shows the equal error rates (EER) obtained in our verification experiments at various input SNRs using speech babble. As in the white noise case, the MFCCs obtained from enhanced log spectra give the best performance. However these results are better than those obtained using white noise. For example in factory noise at 20dB the EER is reduced from 7.54% to 3.17% using the VB algorithm. Similarly in speech babble, the EER is reduced from 9.52% to 4.84% at 10dB.

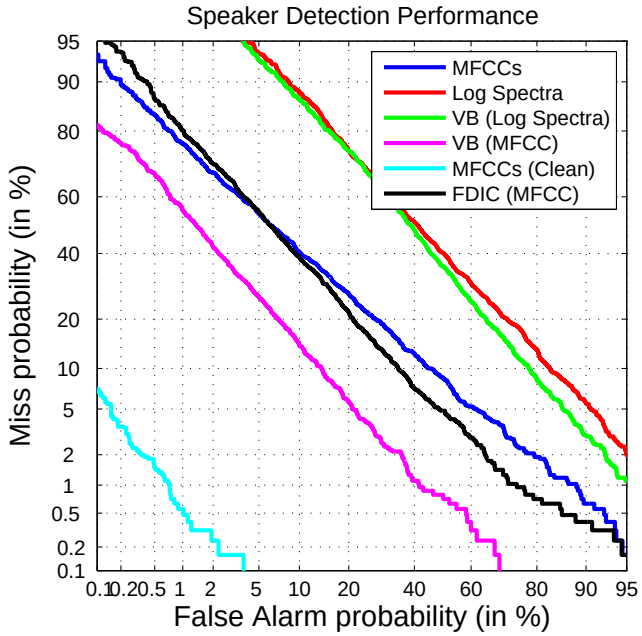


Figure 5.2: Speaker verification performance for the entire TIMIT data set at 20dB. The VB algorithm outperforms the FDIC algorithm.

5.5.3 MDSVC Speaker Verification Results

In the MDSVC data set, each speaker records 54 utterances in two sessions, one for training and the other for testing. The 54 utterances are recorded in three conditions: in an office, a hallway and a noisy street intersection. 18 utterances are recorded in each environment. The speaker models are trained using the 18 utterances recorded in an office since these are the closest to clean. Each utterance is approximately two seconds long. There are 48 target speakers in the data set with 22 female speakers and 26 male speakers. There are 40 impostors with 23 male and 17 female. In our experiments, all trials are same sex trials and all 18 utterances recorded in a given environment are used. This yields a total of 864 true trials and 17496 impostor trials. For the FDIC system, the projection matrix is trained using models derived from the three recording environments and experiments were performed to determine the

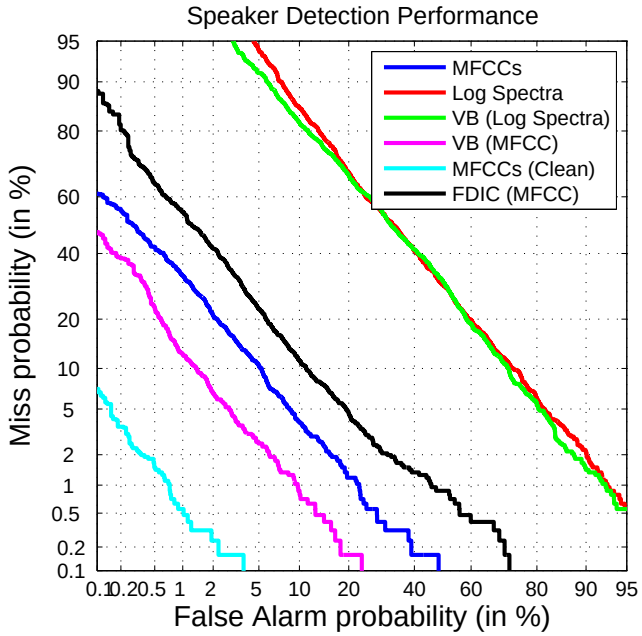


Figure 5.3: Speaker verification performance for the entire TIMIT data set at 30dB. Here the FDIC algorithm degrades the system performance.

appropriate subspace dimension. A subspace dimension of 2 was found to work well.

In our initial experiment we examine the performance of a baseline GMM-UBM speaker verification system. We investigate the EER performance of the system when the test utterances are recorded in the three different environments. Table 5.5 shows the EERs for the test data from different locations. Figure 5.4 shows the corresponding DET curves. We see that mismatch between training and testing data leads to performance degradation. The EER increases from 14.24% to 28.82% when the training data is recorded in an office but the test data is obtained in a noisy street intersection. These EERs are comparable to those obtained in [3, Fig. 7].

In order to investigate the performance of the VB log spectral algorithm on this data set, experiments were performed to determine the EER improvement obtained when the test speech was recorded in various locations with both the MFCC and log spectral models trained using office speech. Table 5.6 shows the EERs obtained

Table 5.3: Speaker verification EER (%) for the entire TIMIT data set in factory noise

System	SNR (dB)						
	0	5	10	15	20	25	30
MFCCs (Baseline)	46.79	39.13	27.78	15.95	7.54	2.94	1.67
VB (MFCC)	**35.48**	**23.49**	**11.90**	**6.11**	**3.17**	**2.06**	**1.51**
Log Spectra	47.22	46.35	44.05	40.85	37.54	35.40	34.84
VB (Log Spectra)	44.84	42.06	39.92	37.78	35.87	35.08	35.48

Table 5.4: Speaker verification EER (%) for the entire TIMIT data set in speech babble

System	SNR (dB)				
	0	5	10	20	30
MFCCs	33.25	20.69	9.52	2.22	1.27
VB (MFCC)	**22.62**	**11.11**	**4.84**	**2.14**	**1.27**
Log Spectra	45.40	42.78	39.68	35.87	35.71
VB (Log Spectra)	41.98	38.89	36.98	34.84	35.56

by the systems described in section 5.5.1. Figure 5.5 shows the corresponding DET curves when the test data is recorded at a noisy street intersection. We see that the VB algorithm significantly improves the EER from 28.82% to 24.54%. Also, the VB algorithm outperfoms the FDIC technique which improves the EER to 27.89%.

5.5.4 SRE Speaker Verification Results

For the SRE data, we report results on the core test of the 2004 evaluation where one conversation side is used for both training and testing (1side-1side). The speaker models are GMMs with 512 mixtures and the features are 18 dimensional MFCCs with delta features. We also make use of gender dependent UBMs. The VB algorithm is run in the same manner as for the TIMIT data. However since all SRE data is corrupted by additive noise and the telephone channel, the speaker models we

Table 5.5: Speaker verification results for MDSVC test data in the three different environments

Location	EER (%)
Office	14.24
Hallway	22.92
Intersection	28.82

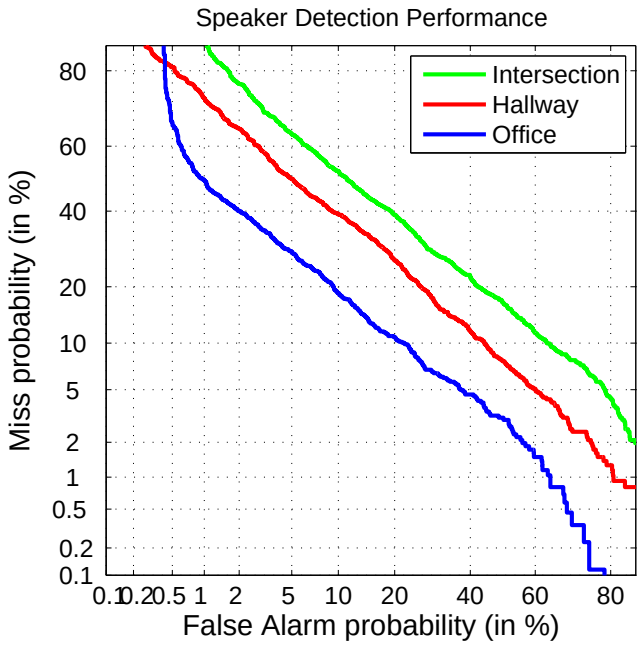


Figure 5.4: Baseline GMM-UBM speaker verification system performance for test data drawn from different environments when training data was recorded in an office. These EERs are comparable to the baseline performance obtained in [3, Fig. 7].

obtain are not as good as those obtained with TIMIT data. Also, we estimate the noise distribution by computing the mean and variance of the frames discarded by the energy detector. To determine the improvement in performance in trials with telephone type mismatch between training data and testing data, the trials were divided into two sets: those in which training and testing data were obtained from the same telephone type (matched) and those where they differ (mismatched). Figure

Table 5.6: Speaker verification EER (%) for the MDSVC data set

System	Intersection EER
MFCCs (Baseline)	28.82
VB (MFCC)	24.54
FDIC	27.89
Log Spectra	42.71
VB (Log Spectra)	40.63

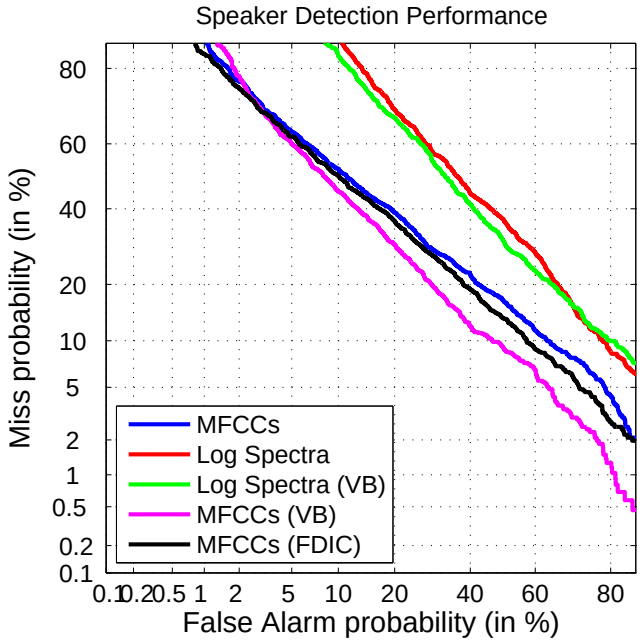


Figure 5.5: Speaker verification system performance for test data drawn from a noisy street intersection for the VB log spectral enhancement algorithm.

5.6 shows the DET curves corresponding to the 1side-1side trials. Overall we see that a slight improvement is obtained in EER with our baseline system yielding an EER of 13.89% and the VB system yielding an EER of 13.43%.This performance is comparable to that obtained by other authors on SRE 2004 data [65]. Furthermore a greater relative improvement of 5% is obtained when mismatched trials are considered separately with the EER reducing from 16.53% to 15.70% as compared to matched

trials where the relative improvement is 3% with the EER reducing from 11.58% to 11.23%.

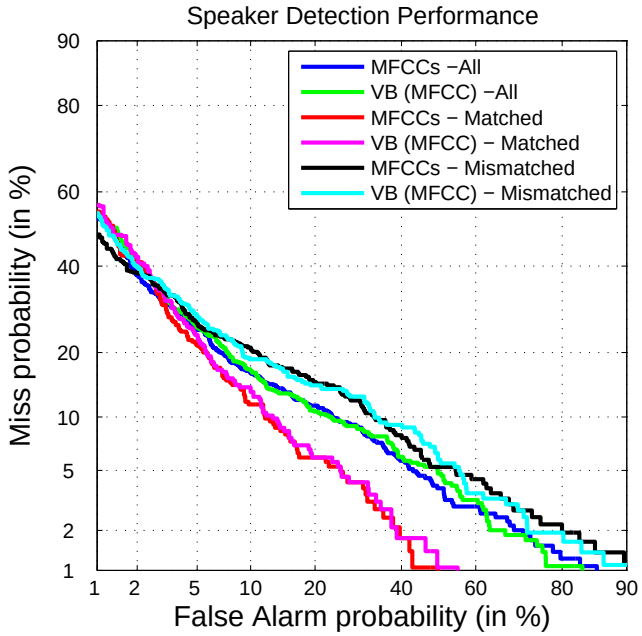


Figure 5.6: Speaker verification performance on SRE 2004 data for the 1side-1side condition.

5.6 Conclusions

The experimental results reported in the previous section verify that the proposed log spectrum enhancement algorithm does indeed improve speaker verification in noisy environments and compensates for mismatch between training and testing conditions. For the TIMIT data set, significant improvements in EER performance are obtained in both white noise and realistic noisy conditions. In white noise, the EER is reduced from 6.83% to 3.65 at 30dB, in factory noise at 20dB the EER is reduced from 7.54% to 3.17% using the VB algorithm. Similarly in speech babble, the EER is reduced

from 9.52% to 4.84% at 10dB. In each of these cases, the algorithm presented in this paper has reduced the verification system EER by approximately 50%. Also, we see that the VB algorithm outperforms FDIC which is a state of the art feature domain technique. At 20 dB, the VB algorithm reduces the EER from 23.25% to 11.83% while FDIC reduces the EER to 20.56%.

The experimental results using the MIT Mobile Device Speaker Verification Corpus demonstrate the compensation of mismatch in realistic environments. Using the VB algorithm, we are able to improve the EER from 28.82% to 24.54% when training data is recorded in an office and test data is recorded at a noisy street intersection. Once again the VB algorithm outperforms FDIC which reduces the EER to 27.89%.

The improvement in performance on SRE data is less than that obtained on TIMIT data. This could be due to the lack of clean training data in this data set. Thus the extension of the model to handle channel and handset mismatch and a means to train clean speaker models could yield improvement in SRE performance similar to that currently obtained on TIMIT. The fact that greater relative improvement in performance is obtained when mismatched trials are considered shows that this algorithm does indeed compensate mismatch between training and testing conditions in speaker verification systems even on the SRE dataset where no clean speech is available to train models.

In summary this chapter has demonstrated the performance of a log spectra enhancement algorithm to improve speaker verification performance in noisy acoustic environments. The encouraging experimental results indicate the potential of using speaker dependent priors in the log spectrum domain to improve the performance of speaker verification systems in noisy environments.

6. Conclusions

The work presented in this thesis is aimed at improving the performance of speech processing systems in noisy acoustic environments. In particular, we present algorithms that perform speech enhancement, voice activity detection and speaker recognition. The algorithms developed in this thesis are all Bayesian algorithms which offer a means of robust estimation of speech parameters from speech signals corrupted by noise. Due to the computational complexity of Bayesian inference, our algorithms employ approximations to make inference possible. In this thesis we develop variational Bayesian algorithms to improve the performance of several speech processing problems.

In chapter 4 we derive a joint speech enhancement and speaker identification algorithm that takes advantage of the fact that speech enhancement and speaker identification are inextricably linked. With enhanced speech, speaker identification decisions are more accurate and on the other hand with accurate speaker identification we can use speaker dependent priors over the speech parameters to improve speech enhancement. This relationship is captured in an iterative VB algorithm that exchanges information between the speech enhancement and speaker identification tasks. The experimental results presented in this chapter show that significant SNR improvement is obtained by the VB algorithm with a maximum SNR improvement of approximately 10dB. Also, we achieve SNR improvements within 1 dB of the performance obtained by the theoretical upper limit. Furthermore, the VB algorithm outperforms the Ephraim-Malah algorithm which is a standard baseline in both SNR improvement and perceptual quality as measured using the PESQ score.

In addition to performing joint speech enhancement and speaker identification, the algorithm presented in chapter 4 is capable of performing robust voice activity

detection (VAD). The algorithm makes use of priors over linear prediction coefficients in silence dominated regions to accurately classify speech segments as either speech or non-speech. The experimental results show that the VB algorithm outperforms the ITU-G.729 algorithm which is the international telecommunications union standard.

In chapter 5 we present a VB algorithm for the enhancement of log spectral features and show how this algorithm can be applied to speaker verification to improve equal error rate performance. Working in the log spectral domain offers an advantage over the acoustic domain in the speaker verification setting because we can easily derive Mel frequency cepstral coefficients (MFCCs) from the enhanced log spectra. We make use of speaker dependent priors over the log spectral features and we demonstrate improved system performance in various noise conditions such as additive Gaussian noise, factory noise and speech babble. We are able to reduce the EER by up to 50% when we compare our system to a standard baseline.

Appendix A. Approximate Posterior Derivations for Chapter 3

We now present the details for each of the factors starting with $q(\tau_\eta)$. We have

$$\begin{aligned}
\log q^*(\tau_\eta) &= \mathbb{E}_{\Theta\setminus\tau_\eta}\{\log p(\mathbf{X},\Theta)\} + const. \\
&= \mathbb{E}_{\mathbf{S},\mathbf{h}}\{\log p(\mathbf{X}|\mathbf{S},\mathbf{h},\tau_\eta)\} + \log p(\tau_\eta) + const. \\
&= \mathbb{E}_{\mathbf{S},\mathbf{h}}\Big\{\sum_{n=0}^{N-1}\log\mathcal{N}(x_n;\mathbf{h}^T\mathbf{s}_n,\tau_\eta^{-1})\Big\} + \log p(\tau_\eta) + const. \\
&= \frac{1}{2}\mathbb{E}_{\mathbf{S},\mathbf{h}}\Big\{\sum_{n=0}^{N-1}\log(\tau_\eta) - \tau_\eta(x_n-\mathbf{h}^T\mathbf{s}_n)^2\Big\} + (a_\eta-1)\log(\tau_\eta) - b_\eta\tau_\eta + const. \\
&= (a_\eta+\frac{N}{2}-1)\log(\tau_\eta) - \tau_\eta\Big[b_\eta + \frac{1}{2}\mathbb{E}_{\mathbf{S},\mathbf{h}}\Big\{\sum_{n=0}^{N-1}(x_n-\mathbf{h}^T\mathbf{s}_n)^2\Big\}\Big] + const. \quad \text{(A.1)}
\end{aligned}$$

From (A.1) we can write

$$q^*(\tau_\eta) = \mathsf{Gam}(\tau_\eta|a_\eta^*,b_\eta^*)$$

with

$$\begin{aligned}
a_\eta^* &= a_\eta + \frac{N}{2}, \quad \text{(A.2)} \\
b_\eta^* &= b_\eta + \frac{1}{2}\mathbb{E}_{\mathbf{S},\mathbf{h}}\Big\{\sum_{n=0}^{N-1}(x_n-\mathbf{h}^T\mathbf{s}_n)^2\Big\}. \quad \text{(A.3)}
\end{aligned}$$

Similarly

$$\begin{aligned}
\log q^*(\beta) &= \mathbb{E}_{\Theta\setminus\beta}\{\log p(\mathbf{X},\Theta)\} + const. \\
&= \mathbb{E}_{\mathbf{a}}\{\log p(\mathbf{a}|\beta)\} + \log p(\beta) + const. \\
&= \mathbb{E}_{\mathbf{a}}\Big\{\frac{P}{2}\log(\beta) - \frac{\beta}{2}\mathbf{a}^T\mathbf{a}\Big\} + (a_\beta-1)\log(\beta) - b_\beta\beta + const. \\
&= (a_\beta+\frac{P}{2}-1)\log(\beta) - \beta\Big[b_\beta + \frac{1}{2}\mathbb{E}_{\mathbf{a}}\{\mathbf{a}^T\mathbf{a}\}\Big] \quad \text{(A.4)}
\end{aligned}$$

From (A.4) we can write $q^*(\beta) = \mathsf{Gam}(\beta|a^*_\beta, b^*_\beta)$ with

$$a^*_\beta = a_\beta + \frac{P}{2}, \tag{A.5}$$

$$b^*_\beta = b_\beta + \frac{1}{2}\mathbb{E}_{\mathbf{a}}\{\mathbf{a}^T\mathbf{a}\}. \tag{A.6}$$

The optimal factor $q^*(\boldsymbol{\pi})$ is now derived. We have

$$\begin{aligned}
\log q^*(\boldsymbol{\pi}) &= \mathbb{E}_{\Theta\setminus\boldsymbol{\pi}}\{\log p(\mathbf{X}, \Theta)\} + const. \\
&= \mathbb{E}_{\mathbf{Z}}\{\log p(\mathbf{Z}|\boldsymbol{\pi})\} + \log p(\boldsymbol{\pi}) + const. \\
&= \mathbb{E}_{\mathbf{Z}}\Big\{ \sum_{n=0}^{N-1}\sum_{m=1}^{M} z_{nm}\log(\pi_m)\Big\} + (\alpha_0 - 1)\sum_{m=1}^{M}\log(\pi_m) + const. \\
&= \sum_{m=1}^{M}\Big\{\alpha_0 + \sum_{n=0}^{N-1}\mathbb{E}_{\mathbf{Z}}\{z_{nm}\} - 1\Big\}\log(\pi_m) + const.
\end{aligned} \tag{A.7}$$

Thus $q^*(\boldsymbol{\pi}) = \mathsf{Dir}(\boldsymbol{\pi}|\alpha^*_0)$ with

$$\alpha^*_0 = \alpha_0 + \sum_{n=0}^{N-1}\mathbb{E}_{\mathbf{Z}}\{z_{nm}\}. \tag{A.8}$$

Turning to $q(\mathbf{Z})$ and following [10, p. 476] we have

$$\begin{aligned}
\log q^*(\mathbf{Z}) &= \mathbb{E}_{\Theta\setminus\mathbf{Z}}\{\log p(\mathbf{X},\Theta)\} + const. \\
&= \mathbb{E}_{\mathbf{S},\mathbf{a},\boldsymbol{\tau}}\{\log p(\mathbf{S}|\mathbf{Z},\mathbf{a},\boldsymbol{\tau})\} + \mathbb{E}_{\boldsymbol{\pi}}\{\log p(\mathbf{Z}|\boldsymbol{\pi})\} + const. \\
&= \mathbb{E}_{\mathbf{S},\mathbf{a},\boldsymbol{\tau}}\Big\{ \sum_{n=0}^{N-1}\sum_{m=1}^{M} z_{nm} \log \mathcal{N}(s_n; \mathbf{a}^T\mathbf{s}^*_{n-1}, \tau_m^{-1})\Big\} + \mathbb{E}_{\boldsymbol{\pi}}\Big\{ \sum_{n=0}^{N-1}\sum_{m=1}^{M} z_{nm}\log(\pi_m)\Big\} + const. \\
&= \mathbb{E}_{\mathbf{S},\mathbf{a},\boldsymbol{\tau}}\Big\{ \sum_{n=0}^{N-1}\sum_{m=1}^{M} z_{nm}[\frac{1}{2}\log(\tau_m) - \frac{\tau_m}{2}(s_n - \mathbf{a}^T\mathbf{s}^*_{n-1})^2]\Big\} \\
&+ \mathbb{E}_{\boldsymbol{\pi}}\Big\{ \sum_{n=0}^{N-1}\sum_{m=1}^{M} z_{nm}\log(\pi_m)\Big\} + const. \\
&= \sum_{n=0}^{N-1}\sum_{m=1}^{M} z_{nm} \underbrace{\Big[\frac{1}{2}\mathbb{E}_{\boldsymbol{\tau}}\{\log(\tau_m)\} - \frac{\mathbb{E}_{\boldsymbol{\tau}}\{\tau_m\}}{2}\mathbb{E}_{\mathbf{S},\mathbf{a}}\{(s_n - \mathbf{a}^T\mathbf{s}^*_{n-1})^2\} + \mathbb{E}_{\boldsymbol{\pi}}\{\log(\pi_m)\}\Big]}_{\log(\rho_{nm})} \\
&+ const. \qquad (A.9)
\end{aligned}$$

From (A.9) we see that

$$q^*(\mathbf{Z}) \propto \prod_{n=0}^{N-1}\prod_{m=1}^{M} \rho_{nm}^{z_{nm}}.$$

If

$$\gamma_{nm} = \frac{\rho_{nm}}{\sum_{m=1}^{M}\rho_{nm}} \qquad (A.10)$$

then

$$q^*(\mathbf{Z}) = \prod_{n=0}^{N-1}\prod_{m=1}^{M} \gamma_{nm}^{z_{nm}}. \qquad (A.11)$$

Considering $q^*(\boldsymbol{\tau})$ we have

$$\begin{aligned}
\log q^*(\boldsymbol{\tau}) &= \mathbb{E}_{\Theta\setminus\boldsymbol{\tau}}\{\log p(\mathbf{X},\Theta)\} + const. \\
&= \mathbb{E}_{\mathbf{S},\mathbf{Z},\mathbf{a}}\{\log p(\mathbf{S}|\mathbf{Z},\mathbf{a},\boldsymbol{\tau})\} + \log p(\boldsymbol{\tau}) + const. \\
&= \mathbb{E}_{\mathbf{S},\mathbf{Z},\mathbf{a}}\Big\{\sum_{n=0}^{N-1}\sum_{m=1}^{M} z_{nm}\log\mathcal{N}(s_n;\mathbf{a}^T\mathbf{s}^*_{n-1},\tau_m^{-1})\Big\} \\
&+ \sum_{m=1}^{M}\Big\{(a_0-1)\log(\tau_m) - b_0\tau_m\Big\} + const. \\
&= \mathbb{E}_{\mathbf{S},\mathbf{Z},\mathbf{a}}\Big\{\sum_{n=0}^{N-1}\sum_{m=1}^{M} z_{nm}[\frac{1}{2}\log(\tau_m) - \frac{\tau_m}{2}(s_n-\mathbf{a}^T\mathbf{s}^*_{n-1})^2]\Big\} \\
&+ \sum_{m=1}^{M}\Big\{(a_0-1)\log(\tau_m) - b_0\tau_m\Big\} + const. \\
&= \sum_{m=1}^{M}\Big\{\frac{1}{2}\Big(\sum_{n=0}^{N-1}\mathbb{E}_{\mathbf{Z}}\{z_{nm}\}\Big) + a_0 - 1\Big\}\log(\tau_m) \\
&- \sum_{m=1}^{M}\tau_m\Big[b_0 + \frac{1}{2}\sum_{n=0}^{N-1}\Big(\mathbb{E}_{\mathbf{Z}}\{z_{nm}\}\mathbb{E}_{\mathbf{S},\mathbf{a}}\{(s_n-\mathbf{a}^T\mathbf{s}^*_{n-1})^2\}\Big)\Big] + const. \qquad \text{(A.12)}
\end{aligned}$$

Which implies that

$$q^*(\boldsymbol{\tau}) = \prod_{m=1}^{M}\mathsf{Gam}(\tau_m|a^*_m,b^*_m)$$

with

$$\begin{aligned}
a^*_m &= a_0 + \frac{1}{2}\Big(\sum_{n=0}^{N-1}\mathbb{E}_{\mathbf{Z}}\{z_{nm}\}\Big), \qquad \text{(A.13)} \\
b^*_m &= b_0 + \frac{1}{2}\sum_{n=0}^{N-1}\Big(\mathbb{E}_{\mathbf{Z}}\{z_{nm}\}\mathbb{E}_{\mathbf{S},\mathbf{a}}\{(s_n-\mathbf{a}^T\mathbf{s}^*_{n-1})^2\}\Big). \qquad \text{(A.14)}
\end{aligned}$$

Similarly

$$\begin{aligned}
\log q^*(\boldsymbol{\lambda}) &= \mathbb{E}_{\Theta\setminus\boldsymbol{\lambda}}\{\log p(\mathbf{X},\Theta)\} + const. \\
&= \mathbb{E}_{\mathbf{h}}\{\log p(\mathbf{h}|\boldsymbol{\lambda})\} + \log p(\boldsymbol{\lambda}) + const. \\
&= \mathbb{E}_{\mathbf{h}}\Big[\frac{1}{2}\sum_{i=0}^{L_h-1}\log\lambda_i - \frac{1}{2}\lambda_i h_i^2\Big] + \sum_{i=0}^{L_h-1}\{(a_\lambda-1)\log\lambda_i - b_\lambda\lambda_i\} + const. \\
&= \sum_{i=0}^{L_h-1}(a_\lambda+\frac{1}{2}-1)\log\lambda_i - \sum_{i=0}^{L_h-1}\lambda_i(b_\lambda+\frac{1}{2}\mathbb{E}_{\mathbf{h}}\{h_i^2\}) + const. \qquad \text{(A.15)}
\end{aligned}$$

Which implies that

$$q^*(\boldsymbol{\lambda}) = \prod_{i=0}^{L_h-1}\mathsf{Gam}(\lambda_i|a^*_{\lambda i}, b^*_{\lambda i})$$

with

$$\begin{aligned}
a^*_{\lambda i} &= a_\lambda + \frac{1}{2}, \qquad \text{(A.16)} \\
b^*_{\lambda i} &= b_\lambda + \frac{1}{2}\mathbb{E}_{\mathbf{h}}\{h_i^2\}. \qquad \text{(A.17)}
\end{aligned}$$

Turning to the AR coefficients we have

$$\begin{aligned}
\log q^*(\mathbf{a}) &= \mathbb{E}_{\Theta\setminus\mathbf{a}}\{\log p(\mathbf{X},\Theta)\} + const. \\
&= \mathbb{E}_{\mathbf{S},\mathbf{Z},\boldsymbol{\tau}}\{\log p(\mathbf{S}|\mathbf{Z},\mathbf{a},\boldsymbol{\tau})\} + \mathbb{E}_\beta\{\log p(\mathbf{a}|\beta)\} + const. \\
&= \mathbb{E}_{\mathbf{S},\mathbf{Z},\boldsymbol{\tau}}\Big\{\sum_{n=0}^{N-1}\sum_{m=1}^{M} z_{nm}[\frac{1}{2}\log\tau_m - \frac{\tau_m}{2}(s_n - \mathbf{a}^T\mathbf{s}^*_{n-1})^2]\Big\} \\
&+ \mathbb{E}_\beta\Big\{\frac{P}{2}\log\beta - \frac{\beta}{2}\mathbf{a}^T\mathbf{a}\Big\} + const. \qquad \text{(A.18)}
\end{aligned}$$

(A.18) is quadratic in $\mathbf{a}$ and we can write

$$\begin{aligned}\log q^*(\mathbf{a}) &= -\frac{1}{2}\mathbf{a}^T\Big[\sum_{m=1}^{M}\mathbb{E}_{\boldsymbol{\tau}}\{\tau_m\}\Big(\sum_{n=0}^{N-1}\mathbb{E}_{\mathbf{Z}}\{z_{nm}\}\mathbb{E}_{\mathbf{S}}\{\mathbf{s}_{n-1}^*\mathbf{s}_{n-1}^{*T}\}\Big)+\mathbb{E}_{\beta}\{\beta\}\mathbf{I}\Big]\mathbf{a}\\ &+ \mathbf{a}^T\Big[\sum_{m=1}^{M}\mathbb{E}_{\boldsymbol{\tau}}\{\tau_m\}\Big(\mathbb{E}_{\mathbf{Z}}\{z_{nm}\}\mathbb{E}_{\mathbf{S}}\{s_n\mathbf{s}_{n-1}^*\}\Big)\Big]+const. \qquad (A.19)\end{aligned}$$

From (A.19) we see that $q^*(\mathbf{a})=\mathcal{N}(\mathbf{a};\boldsymbol{\mu}_{\mathbf{a}}^*,\boldsymbol{\Sigma}_{\mathbf{a}}^*)$ with

$$\boldsymbol{\Sigma}_{\mathbf{a}}^* = \Big[\sum_{m=1}^{M}\mathbb{E}_{\boldsymbol{\tau}}\{\tau_m\}\Big(\sum_{n=0}^{N-1}\mathbb{E}_{\mathbf{Z}}\{z_{nm}\}\mathbb{E}_{\mathbf{S}}\{\mathbf{s}_{n-1}^*\mathbf{s}_{n-1}^{*T}\}\Big)+\mathbb{E}_{\beta}\{\beta\}\mathbf{I}\Big]^{-1}, \qquad (A.20)$$

$$\boldsymbol{\mu}_{\mathbf{a}}^* = \boldsymbol{\Sigma}_{\mathbf{a}}^*\Big[\sum_{m=1}^{M}\mathbb{E}_{\boldsymbol{\tau}}\{\tau_m\}\Big(\sum_{n=0}^{N-1}\mathbb{E}_{\mathbf{Z}}\{z_{nm}\}\mathbb{E}_{\mathbf{S}}\{s_n\mathbf{s}_{n-1}^*\}\Big)\Big]. \qquad (A.21)$$

Considering $q^*(\mathbf{h})$ we have

$$\begin{aligned}\log q^*(\mathbf{h}) &= \mathbb{E}_{\Theta\setminus\mathbf{h}}\{\log p(\mathbf{X},\Theta)\}+const.\\ &= \mathbb{E}_{\mathbf{S},\tau_\eta}\{\log p(\mathbf{X}|\mathbf{S},\mathbf{h},\tau_\eta)\}+\mathbb{E}_{\boldsymbol{\lambda}}\{\log p(\mathbf{h}|\boldsymbol{\lambda})\}+const.\\ &= \mathbb{E}_{\mathbf{S},\tau_\eta}\Big\{\sum_{n=0}^{N-1}\Big(\frac{1}{2}\log\tau_\eta-\frac{\tau_\eta}{2}(x_n-\mathbf{h}^T\mathbf{s}_n)^2\Big)\Big\}-\frac{1}{2}\mathbf{h}^T\mathbb{E}_{\boldsymbol{\lambda}}\{\boldsymbol{\Lambda}\}\mathbf{h}+const. \qquad (A.22)\end{aligned}$$

where $\boldsymbol{\Lambda}=\mathsf{diag}(\boldsymbol{\lambda})$. (A.22) is quadratic in $\mathbf{h}$ and we can write

$$\begin{aligned}\log q^*(\mathbf{h}) &= -\frac{1}{2}\mathbf{h}^T\Big[\mathbb{E}_{\tau_\eta}\{\tau_\eta\}\Big(\sum_{n=0}^{N-1}\mathbb{E}_{\mathbf{S}}\{\mathbf{s}_n\mathbf{s}_n^T\}\Big)+\mathbb{E}_{\boldsymbol{\lambda}}\{\boldsymbol{\Lambda}\}\Big]\mathbf{h}\\ &+ \mathbf{h}^T\Big(\mathbb{E}_{\tau_\eta}\{\tau_\eta\}\sum_{n=0}^{N-1}\mathbb{E}_{\mathbf{S}}\{x_n\mathbf{s}_n\}\Big)+const. \qquad (A.23)\end{aligned}$$

From (A.23) we see that $q^*(\mathbf{h}) = \mathcal{N}(\mathbf{h}; \boldsymbol{\mu}^*_{\mathbf{h}}, \boldsymbol{\Sigma}^*_{\mathbf{h}})$ with

$$\boldsymbol{\Sigma}^*_{\mathbf{h}} = \Big[\mathbb{E}_{\tau_\eta}\{\tau_\eta\}\Big(\sum_{n=0}^{N-1}\mathbb{E}_{\mathbf{S}}\{\mathbf{s}_n\mathbf{s}_n^T\}\Big) + \mathbb{E}_{\boldsymbol{\lambda}}\{\boldsymbol{\Lambda}\}\Big]^{-1}, \tag{A.24}$$

$$\boldsymbol{\mu}^*_{\mathbf{h}} = \boldsymbol{\Sigma}^*_{\mathbf{h}}\Big(\mathbb{E}_{\tau_\eta}\{\tau_\eta\}\sum_{n=0}^{N-1}\mathbb{E}_{\mathbf{S}}\{x_n\mathbf{s}_n\}\Big). \tag{A.25}$$

Finally we derive $q^*(\mathbf{S})$. We have

$$\begin{aligned}
\log q^*(\mathbf{S}) &= \mathbb{E}_{\Theta\setminus\mathbf{S}}\{\log p(\mathbf{X},\Theta)\} + const.\\
&= \mathbb{E}_{\mathbf{h},\tau_\eta}\{\log p(\mathbf{X}|\mathbf{S},\mathbf{h},\tau_\eta)\} + \mathbb{E}_{\mathbf{Z},\mathbf{a},\boldsymbol{\tau}}\{\log p(\mathbf{S}|\mathbf{Z},\mathbf{a},\boldsymbol{\tau})\} + const.\\
&= \mathbb{E}_{\mathbf{h},\tau_\eta}\sum_{n=0}^{N-1}\Big\{\frac{1}{2}\log\tau_\eta - \frac{\tau_\eta}{2}(x_n - \mathbf{h}^T\mathbf{s}_n)^2\Big\}\\
&+ \mathbb{E}_{\mathbf{Z},\mathbf{a},\boldsymbol{\tau}}\sum_{n=0}^{N-1}\sum_{m=1}^{M}\Big\{z_{nm}[\frac{1}{2}\log\tau_m - \frac{\tau_m}{2}(s_n - \mathbf{a}^T\mathbf{s}^*_{n-1})^2]\Big\} + const.\\
&= -\frac{1}{2}\mathbb{E}_{\mathbf{h},\tau_\eta}\Big\{\sum_{n=0}^{N-1}\tau_\eta(x_n - \mathbf{h}^T\mathbf{s}_n)^2\Big\}\\
&- \frac{1}{2}\sum_{n=0}^{N-1}\underbrace{\Big(\sum_{m=1}^{M}\mathbb{E}_{\mathbf{Z}}\{z_{nm}\}\mathbb{E}_{\boldsymbol{\tau}}\{\tau_m\}\Big)}_{\zeta^*_n}\mathbb{E}_{\mathbf{a}}\{(s_n - \mathbf{a}^T\mathbf{s}^*_{n-1})^2\} + const. \qquad \text{(A.26)}\\
&= -\frac{1}{2}\mathbb{E}_{\mathbf{h},\tau_\eta}\Big\{\sum_{n=0}^{N-1}\tau_\eta(x_n - h_0 s_n - \tilde{\mathbf{h}}^T\tilde{\mathbf{s}}_n)^2\Big\}\\
&- \frac{1}{2}\sum_{n=0}^{N-1}\zeta^*_n\mathbb{E}_{\mathbf{a}}\{(s_n - \mathbf{a}^T\mathbf{s}^*_{n-1})^2\} + const.\\
&= -\frac{1}{2}\mathbb{E}_{\mathbf{h},\tau_\eta}\Big\{\sum_{n=0}^{N-1}\tau_\eta h_0^2\Big(s_n - \frac{1}{h_0}(x_n - \tilde{\mathbf{h}}^T\tilde{\mathbf{s}}_n)\Big)^2\Big\}\\
&- \frac{1}{2}\sum_{n=0}^{N-1}\zeta^*_n\mathbb{E}_{\mathbf{a}}\{(s_n - \mathbf{a}^T\mathbf{s}^*_{n-1})^2\} + const. \qquad \text{(A.27)}
\end{aligned}$$

where $\tilde{\mathbf{h}} = [h_1, \ldots, h_{L_h-1}]^T$ and $\tilde{\mathbf{s}}_n = [s_{n-1}, \ldots, s_{n-L_h+1}]^T$.

Recognizing that $q^*(\mathbf{S}) = \prod_{n=0}^{N-1} q(s_n|s_{n-1},\ldots,s_0)$ and

$$\log q^*(\mathbf{S}) = \sum_{n=0}^{N-1} \log q(s_n|s_{n-1},\ldots,s_0).$$

From (A.26) we have

$$\begin{aligned}\log q^*(\mathbf{S}) &= -\frac{1}{2}\sum_{n=0}^{N-1}(\zeta_n^* + \mathbb{E}_{\mathbf{h},\tau_\eta}\{\tau_\eta h_0^2\})s_n^2 \\ &+ \sum_{n=0}^{N-1} s_n\Big(\mathbb{E}_{\mathbf{h},\tau_\eta}\{\tau_\eta h_o(x_n - \tilde{\mathbf{h}}^T\tilde{\mathbf{s}}_n)\} + \zeta_n^*\mathbb{E}_{\mathbf{a}}\{\mathbf{a}^T\mathbf{s}_{n-1}^*\}\Big) + const. \quad \text{(A.28)}\end{aligned}$$

We see that (A.28) is the sum of terms that are quadratic in s_n and we conclude that

$$q^*(\mathbf{S}) = \prod_{n=0}^{N-1} \mathcal{N}(s_n; \mu^*_{n|n-d:n-1}, \tau^{*-1}_{n|n-d:n-1}) \quad \text{(A.29)}$$

with

$$\begin{aligned}\tau^*_{n|n-d:n-1} &= \zeta_n^* + \mathbb{E}_{\mathbf{h},\tau_\eta}\{\tau_\eta h_0^2\}, & \text{(A.30)}\\ \mu^*_{n|n-d:n-1} &= \tau_n^{*-1}\Big(\mathbb{E}_{\mathbf{h},\tau_\eta}\{\tau_\eta h_o(x_n - \tilde{\mathbf{h}}^T\tilde{\mathbf{s}}_n)\} + \zeta_n^*\mathbb{E}_{\mathbf{a}}\{\mathbf{a}^T\mathbf{s}_n^*\}\Big). & \text{(A.31)}\end{aligned}$$

where $d = \max\{L_h, P\}$.

From the form of the posterior, we observe that it can be derived from a Gaussian linear state space model (GLSSM) [7]. To see this consider a GLSSM described by

$$\begin{aligned}\mathbf{s}_n &= \mathbf{A}\mathbf{s}_{n-1} + \mathbf{e}_1\epsilon_n \quad \epsilon_n \sim \mathcal{N}(\epsilon_n; 0, \tau_{\epsilon,n}^{-1}) & \text{(A.32)}\\ x_n &= \mathbf{h}^T\mathbf{s}_n + \eta_n \quad \eta_n \sim \mathcal{N}(\eta_n; 0, \tau_{\eta,n}^{-1}). & \text{(A.33)}\end{aligned}$$

Where $\mathbf{A}$ is the $d \times d$ state transition matrix, $\mathbf{h}$ is the $d \times 1$ observation vector and

$\mathbf{e}_1$ is the first column of the $d \times d$ identity matrix. From (A.32) and (A.33) we have

$$p(s_n|\mathbf{s}_{n-1}) = \mathcal{N}(s_n; \mathbf{a}^T\mathbf{s}_{n-1}, \tau_{\epsilon,n}^{-1}), \tag{A.34}$$

$$p(x_n|\mathbf{s}_n) = \mathcal{N}(x_n; \mathbf{h}^T\mathbf{s}_n, \tau_{\eta,n}^{-1}). \tag{A.35}$$

where $\mathbf{a}$ is the first row of $\mathbf{A}$.

We now consider the posterior $q(s_n|\mathbf{s}_{n-1}, x_{0:n})$ where $x_{0:n} = \{x_0, \ldots, x_n\}$. We have

$$\begin{aligned} q(s_n|\mathbf{s}_{n-1}, x_{0:n}) &= \frac{p(s_n, \mathbf{s}_{n-1}, x_{0:n})}{p(\mathbf{s}_{n-1}, x_{0:n})} \\ &\propto p(x_{0:n}|s_n, \mathbf{s}_{n-1})p(s_n|\mathbf{s}_{n-1}) \\ &= p(s_n|\mathbf{s}_{n-1}) \prod_{i=0}^{n} p(x_i|s_i, \mathbf{s}_{i-1}) \\ &\propto p(s_n|\mathbf{s}_{n-1})p(x_n|s_n, \mathbf{s}_{n-1}) \\ &= p(s_n|\mathbf{s}_{n-1})p(x_n|\mathbf{s}_n) \end{aligned}$$

where all terms independent of s_n have been lumped into a constant. Using (A.34) and (A.35) we have

$$\begin{aligned} q(s_n|\mathbf{s}_{n-1}, x_{0:n}) &\propto \mathcal{N}(s_n; \mathbf{a}^T\mathbf{s}_{n-1}, \tau_{\epsilon,n}^{-1}) \times \mathcal{N}(x_n; \mathbf{h}^T\mathbf{s}_n, \tau_{\eta,n}^{-1}) \\ &\propto \exp\Big[-\frac{\tau_{\epsilon,n}}{2}(s_n - \mathbf{a}^T\mathbf{s}_{n-1})^2 - \frac{\tau_{\eta,n}}{2}(x_n - \mathbf{h}^T\mathbf{s}_n)^2 \Big] \end{aligned} \tag{A.36}$$

Comparing (A.26) and (A.36) we see that we can determine moments with respect to $q^*(\mathbf{S})$ using a Kalman filter applied to the GLSSM characterized by (A.32) and

(A.33) with

$$\mathbf{A} = \begin{bmatrix} \mu_{a1}^* & \mu_{a2}^* & \dots & \mu_{aP}^* & 0 & \dots & 0 \\ 1 & 0 & \dots & 0 & 0 & \dots & 0 \\ 0 & 1 & \dots & & & \dots & 0 \\ \vdots & & \ddots & & & & \vdots \\ & & & & & & \\ 0 & \dots & & & & 1 & 0 \end{bmatrix} \tag{A.37}$$

where $\boldsymbol{\mu}_{\mathbf{a}}^* = [\mu_{a1}^*, \mu_{a2}^*, \dots, \mu_{aP}^*]^T$.

$$\begin{aligned} \mathbf{h} &= [\underbrace{\mu_{h0}^*, \mu_{h2}^*, \dots, \mu_{h(L_h-1)}^*, 0, \dots, 0}_{d \quad \text{terms}}]^T, \\ \tau_{\epsilon,n} &= \zeta_n^*, \\ \tau_{\eta,n} &= \mathbb{E}_{\tau_\eta}\{\tau_\eta\} = \tau_\eta^*, \end{aligned}$$

and $d = \max\{L_h, P\}$.

Appendix B. Approximate Posterior Derivations for Chapter 4

In this appendix we derive the optimal factors of the approximate posterior presented in section 5.2. Starting with the optimal form of $q(\tau_\eta^k)$ we have

$$\begin{aligned}
\log q^*(\tau_\eta^k) &= \mathbb{E}_{\Theta\setminus\tau_\eta^k}\{\log p(\mathbf{r}^{1:K},\Theta)\} + const. \\
&= \mathbb{E}_{\mathbf{s}^k}\{\log p(\mathbf{r}^k|\mathbf{s}^k,\tau_\eta^k)\} + \log p(\tau_\eta^k) + const. \\
&= \mathbb{E}_{\mathbf{s}^k}\{\sum_{n=1}^{N}\log \mathcal{N}(r_n^k; s_n^k, \tau_\eta^k)\} + \log p(\tau_\eta^k) + const. \\
&= \mathbb{E}_{\mathbf{s}^k}\{\sum_{n=1}^{N}\frac{1}{2}\log\tau_\eta^k - \frac{\tau_\eta^k}{2}(r_n^k - s_n^k)^2\} \\
&+ (a_\eta - 1)\log\tau_\eta^k - b_\eta\tau_\eta^k + const. \\
&= (a_\eta + \frac{N}{2} - 1)\log\tau_\eta^k \\
&- \tau_\eta^k[b_\eta + \frac{1}{2}\mathbb{E}_{\mathbf{s}^k}\{\sum_{n=1}^{N}(r_n^k - s_n^k)^2\}] + const. \qquad \text{(B.1)}
\end{aligned}$$

From (B.1) we obtain (4.11)

$$q^*(\tau_\eta^k) = \mathsf{Gam}(\tau_\eta^k|a_\eta^*, b_\eta^*)$$

with

$$\begin{aligned}
a_\eta^* &= a_\eta + \frac{N}{2}, \\
b_\eta^* &= b_\eta + \frac{1}{2}\mathbb{E}_{\mathbf{s}^k}\Big\{\sum_{n=1}^{N}(r_n^k - s_n^k)^2\Big\}.
\end{aligned}$$

For $q(\tau_\epsilon^k)$ we have

$$
\begin{aligned}
\log q^*(\tau_\epsilon^k) &= \mathbb{E}_{\Theta\setminus\tau_\epsilon^k}\{\log p(\mathbf{r}^{1:K},\Theta)\} + const. \\
&= \mathbb{E}_{\mathbf{s}^k,\mathbf{a}^k}\{\log p(\mathbf{s}^k|\mathbf{a}^k,\tau_\epsilon^k)\} + \log p(\tau_\epsilon^k) + const. \\
&= \mathbb{E}_{\mathbf{s}^k,\mathbf{a}^k}\{\sum_{n=1}^{N} \log \mathcal{N}(s_n^k; \mathbf{a}^{kT}\mathbf{s}_{n-1}^k, (\tau_\epsilon^k)^{-1})\} \\
&+ \log p(\tau_\epsilon^k) + const. \\
&= \mathbb{E}_{\mathbf{s}^k,\mathbf{a}^k}\Big\{ \sum_{n=1}^{N} \Big(\frac{1}{2}\log\tau_\epsilon^k - \frac{\tau_\epsilon^k}{2}(s_n^k - \mathbf{a}^{kT}\mathbf{s}_{n-1}^k)^2\Big)\Big\} \\
&+ (a_\epsilon - 1)\log\tau_\epsilon^k - b_\epsilon\tau_\epsilon^k + const. \qquad \text{(B.2)}
\end{aligned}
$$

From (B.2) we obtain (4.12)

$$q^*(\tau_\epsilon^k) = \mathsf{Gam}(\tau_\epsilon^k|a_\epsilon^*, b_\epsilon^*)$$

with

$$
\begin{aligned}
a_\epsilon^* &= a_\epsilon + \frac{N}{2}, \\
b_\epsilon^* &= b_\epsilon + \frac{1}{2}\mathbb{E}_{\mathbf{s}^k,\mathbf{a}^k}\Big\{\sum_{n=1}^{N}(s_n^k - \mathbf{a}^{kT}\mathbf{s}_{n-1}^k)^2\Big\}.
\end{aligned}
$$

Turning to $q(\mathbf{z}_a^k)$ we have

$$
\begin{aligned}
\log q^*(\mathbf{z}_a^k) &= \mathbb{E}_{\Theta\setminus\mathbf{z}_a^k}\{\log p(\mathbf{r}^{1:K},\Theta)\} + const. \\
&= \mathbb{E}_{\mathbf{a}^k}\{\log p(\mathbf{a}^k|\mathbf{z}_a^k)\} + \mathbb{E}_{\mathbf{z}_a^{k-1}}\{\log p(\mathbf{z}_a^k|\mathbf{z}_a^{k-1})\} \\
&+ \mathbb{E}_{\mathbf{z}_a^{k+1}}\{\log p(\mathbf{z}_a^{k+1}|\mathbf{z}_a^k)\} + const. \\
&= \mathbb{E}_{\mathbf{a}^k}\Big\{\sum_{i=1}^{M_a|\mathcal{L}|} z_{a,i}^k \log \mathcal{N}(\mathbf{a}^k;\boldsymbol{\mu}_i^a,\boldsymbol{\Sigma}_i^a)\Big\} \\
&+ \sum_{i=1}^{M_a|\mathcal{L}|} z_{a,i}^k \Big\{\mathbb{E}_{\mathbf{z}_a^{k-1}}\Big(\sum_{j=1}^{M_a|\mathcal{L}|} z_{a,j}^{k-1}\log t_{ij}\Big) \\
&+ \mathbb{E}_{\mathbf{z}_a^{k+1}}\Big(\sum_{n=1}^{M_a|\mathcal{L}|} z_{a,n}^{k+1}\log t_{ni}\Big)\Big\} + const. \\
&= \sum_{i=1}^{M_a|\mathcal{L}|} z_{a,i}^k \Big\{-\frac{1}{2}\log|\boldsymbol{\Sigma}_i^a| \\
&- \frac{1}{2}\mathbb{E}_{\mathbf{a}^k}\{(\mathbf{a}^k-\boldsymbol{\mu}_i^a)^T\boldsymbol{\Sigma}_i^{a-1}(\mathbf{a}^k-\boldsymbol{\mu}_i^a)\} \\
&+ \sum_{j=1}^{M_a|\mathcal{L}|} \mathbb{E}_{\mathbf{z}_a^{k-1}}\{z_{a,j}^{k-1}\}\log t_{ij} \\
&+ \sum_{n=1}^{M_a|\mathcal{L}|} \mathbb{E}_{\mathbf{z}_a^{k+1}}\{z_{a,n}^{k+1}\}\log t_{ni}\Big\} \\
&+ const. \qquad \text{(B.3)}
\end{aligned}
$$

From (B.3) we obtain (4.13)

$$q^*(\mathbf{z}_a^k) = \prod_{i=1}^{M_a|\mathcal{L}|} (\gamma_i^k)^{z_{a,i}^k}$$

where

$$\gamma_i^k = \frac{\rho_i^k}{\sum_{i=1}^{M_a|\mathcal{L}|}\rho_i^k}$$

and

$$\begin{aligned}
\log \rho_i^k &= -\frac{1}{2}\log|\boldsymbol{\Sigma}_i^a| \\
&- \frac{1}{2}\mathbb{E}_{\mathbf{a}^k}\{(\mathbf{a}^k - \boldsymbol{\mu}_i^a)^T \boldsymbol{\Sigma}_i^{a-1}(\mathbf{a}^k - \boldsymbol{\mu}_i^a)\} \\
&+ \sum_{j=1}^{M_a|\mathcal{L}|} \mathbb{E}_{\mathbf{z}_a^{k-1}}\{z_{a,j}^{k-1}\}\log t_{ij} \\
&+ \sum_{n=1}^{M_a|\mathcal{L}|} \mathbb{E}_{\mathbf{z}_a^{k+1}}\{z_{a,n}^{k+1}\}\log t_{ni}
\end{aligned}$$

Considering $q(\mathbf{a}^k)$ we have

$$\begin{aligned}
\log q^*(\mathbf{a}^k) &= \mathbb{E}_{\Theta\setminus\mathbf{a}^k}\{\log p(\mathbf{r}^{1:K}, \Theta)\} + const. \\
&= \mathbb{E}_{\mathbf{s}^k,\tau_\epsilon^k}\{\log p(\mathbf{s}^k|\mathbf{a}^k, \tau_\epsilon^k)\} \\
&+ \mathbb{E}_{\mathbf{z}_a^k}\{\log p(\mathbf{a}^k|\mathbf{z}_a^k)\} + const. \\
&= \mathbb{E}_{\mathbf{s}^k,\tau_\epsilon^k}\Big\{\sum_{n=1}^{N}\log\mathcal{N}(s_n^k; \mathbf{a}^{kT}\mathbf{s}_{n-1}^k, (\tau_\epsilon^k)^{-1})\Big\} \\
&+ \mathbb{E}_{\mathbf{z}_a^k}\Big\{\sum_{i=1}^{M_a|\mathcal{L}|} z_{a,i}^k \log\mathcal{N}(\mathbf{a}^k; \boldsymbol{\mu}_i^a, \boldsymbol{\Sigma}_i^a)\Big\} + const. \\
&= -\frac{\mathbb{E}_{\tau_\epsilon^k}\{\tau_\epsilon^k\}}{2}\mathbb{E}_{\mathbf{s}^k}\Big\{\sum_{n=1}^{N}(s_n^k - \mathbf{a}^{kT}\mathbf{s}_{n-1}^k)^2\Big\} \\
&- \frac{1}{2}\sum_{i=1}^{M_a|\mathcal{L}|}\mathbb{E}_{\mathbf{z}_a^k}\{z_{a,i}^k\}\Big\{(\mathbf{a}^k - \boldsymbol{\mu}_i^a)^T\boldsymbol{\Sigma}_i^{a-1}(\mathbf{a}^k - \boldsymbol{\mu}_i^a)\Big\} \\
&+ const. \qquad \text{(B.4)}
\end{aligned}$$

(B.4) is quadratic in $\mathbf{a}^k$ and we can write

$$\begin{aligned}
\log q^*(\mathbf{a}^k) &= -\frac{1}{2}\mathbf{a}^{kT}\Big[\sum_{n=1}^{N}\mathbb{E}_{\tau_\epsilon^k}\{\tau_\epsilon^k\}\mathbb{E}_{\mathbf{s}^k}\{\mathbf{s}_{n-1}^k\mathbf{s}_{n-1}^{kT}\} \\
&+ \sum_{i=1}^{M_a|\mathcal{L}|}\mathbb{E}_{\mathbf{z}_a^k}\{z_{a,i}^k\}\boldsymbol{\Sigma}_i^{a-1}\Big]\mathbf{a}^k \\
&+ \mathbf{a}^{kT}\Big[\sum_{n=1}^{N}\mathbb{E}_{\tau_\epsilon^k}\{\tau_\epsilon^k\}\mathbb{E}_{\mathbf{s}^k}\{s_n^k\mathbf{s}_{n-1}^k\} \\
&+ \sum_{i=1}^{M_a|\mathcal{L}|}\mathbb{E}_{\mathbf{z}_a^k}\{z_{a,i}^k\}\boldsymbol{\Sigma}_i^{a-1}\boldsymbol{\mu}_i^a\Big] + const. \qquad \text{(B.5)}
\end{aligned}$$

From (B.5) we obtain (4.14)

$$q^*(\mathbf{a}^k) = \mathcal{N}(\mathbf{a}^k; \boldsymbol{\mu}_\mathbf{a}^*, \boldsymbol{\Sigma}_\mathbf{a}^*)$$

with

$$\begin{aligned}
\boldsymbol{\Sigma}_\mathbf{a}^* &= \Big[\sum_{n=1}^{N}\mathbb{E}_{\tau_\epsilon^k}\{\tau_\epsilon^k\}\mathbb{E}_{\mathbf{s}^k}\{\mathbf{s}_{n-1}^k\mathbf{s}_{n-1}^{kT}\} \\
&+ \sum_{i=1}^{M_a|\mathcal{L}|}\mathbb{E}_{\mathbf{z}_a^k}\{z_{a,i}^k\}\boldsymbol{\Sigma}_i^{a-1}\Big]^{-1} \\
\boldsymbol{\mu}_\mathbf{a}^* &= \boldsymbol{\Sigma}_\mathbf{a}^*\Big[\sum_{n=1}^{N}\mathbb{E}_{\tau_\epsilon^k}\{\tau_\epsilon^k\}\mathbb{E}_{\mathbf{s}^k}\{s_n^k\mathbf{s}_{n-1}^k\} \\
&+ \sum_{i=1}^{M_a|\mathcal{L}|}\mathbb{E}_{\mathbf{z}_a^k}\{z_{a,i}^k\}\boldsymbol{\Sigma}_i^{a-1}\boldsymbol{\mu}_i^a\Big]
\end{aligned}$$

Turning to $q^*(\mathbf{s}^k)$ we have

$$\begin{aligned}
\log q^*(\mathbf{s}^k) &= \mathbb{E}_{\Theta\setminus\mathbf{s}^k}\{\log p(\mathbf{r}^{1:K},\Theta)\} + const. \\
&= \mathbb{E}_{\tau_\eta^k}\{\log p(\mathbf{r}^k|\mathbf{s}^k,\tau_\eta^k)\} \\
&+ \mathbb{E}_{\mathbf{a}^k,\tau_\epsilon^k}\{\log p(\mathbf{s}^k|\mathbf{a}^k,\tau_\epsilon^k)\} + const. \\
&= \mathbb{E}_{\tau_\eta^k}\Big\{\sum_{n=1}^{N}\log\mathcal{N}(r_n^k;s_n^k,\tau_\eta^k)\Big\} \\
&+ \mathbb{E}_{\mathbf{a}^k,\tau_\epsilon^k}\Big\{\sum_{n=1}^{N}\log\mathcal{N}(s_n^k;\mathbf{a}^{kT}\mathbf{s}_{n-1}^k,(\tau_\epsilon^k)^{-1})\Big\} \\
&+ const. \\
&= \mathbb{E}_{\tau_\eta^k}\{\sum_{n=1}^{N}-\frac{\tau_\eta^k}{2}(r_n^k-s_n^k)^2\} \\
&+ \mathbb{E}_{\mathbf{a}^k,\tau_\epsilon^k}\Big\{-\frac{\tau_\epsilon^k}{2}\sum_{n=1}^{N}(s_n^k-\mathbf{a}^{kT}\mathbf{s}_{n-1}^k)^2\Big\} \\
&+ const. \qquad \text{(B.6)}
\end{aligned}$$

Expanding the terms in (B.6) and evaluating the expectations yields (4.15).

$$\begin{aligned}
\log q^*(\mathbf{s}^k) &= -\frac{1}{2}\sum_{n=1}^{N}\frac{a_\eta^*}{b_\eta^*}(r_n^k-s_n^k)^2 \\
&- \frac{1}{2}\sum_{n=1}^{N}\frac{a_\epsilon^*}{b_\epsilon^*}\Big((s_n^k)^2 - 2\boldsymbol{\mu}_{\mathbf{a}}^{*T}s_n^k\mathbf{s}_{n-1}^k \\
&+ \mathbf{s}_{n-1}^{kT}\boldsymbol{\mu}_{\mathbf{a}}^*\boldsymbol{\mu}_{\mathbf{a}}^{*T}\mathbf{s}_{n-1}^k + \mathbf{s}_{n-1}^{kT}\boldsymbol{\Sigma}_{\mathbf{a}}^*\mathbf{s}_{n-1}^k\Big) \\
&+ const.
\end{aligned}$$

To arrive at the conclusion that $\mathbb{E}\{\mathbf{s}_n^k\}$, $\mathbb{E}\{\mathbf{s}_n^k\mathbf{s}_n^{kT}\}$ and $\mathbb{E}\{\mathbf{s}_n^k\mathbf{s}_{n-1}^{kT}\}$ can be computed using a Kalman smoother consider the following state space model where

$\mathbf{y}_n^k = [r_n^k, 0, \ldots, 0]^T$

$$\mathbf{s}_n^k = \mathbf{A}\mathbf{s}_{n-1}^k + \mathbf{G}u_n^k \tag{B.7}$$

$$\mathbf{y}_n^k = \mathbf{H}\mathbf{s}_n^k + \mathbf{v}_n^k \tag{B.8}$$

with

$$u^k \sim \mathcal{N}(u^k; 0, (\bar{\tau}_\epsilon^k)^{-1}) \tag{B.9}$$

$$\mathbf{v}^k \sim \mathcal{N}(\mathbf{v}^k; \mathbf{0}, \mathbf{\Sigma}_v^k) \tag{B.10}$$

where

$$\mathbf{A} = \begin{bmatrix} \mu_{1,a}^* & \mu_{2,a}^* & \ldots & \ldots & \mu_{P,a}^* \\ 1 & 0 & \ldots & \ldots & 0 \\ 0 & 1 & \ldots & \ldots & 0 \\ \vdots & & \ddots & & \vdots \\ & & & & \\ 0 & \ldots & & 1 & 0 \end{bmatrix}, \tag{B.11}$$

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & \ldots & 0 \end{bmatrix}^T \tag{B.12}$$

and

$$\mathbf{H} = \begin{bmatrix} 1, 0, \ldots, 0 \\ \mathbf{I}_{P\times P} \end{bmatrix} \tag{B.13}$$

Also

$$\mathbf{\Sigma}_v^k = \begin{bmatrix} (\bar{\tau}_\eta^k)^{-1} & \\ & (\bar{\tau}_\epsilon^k)^{-1}\mathbf{\Sigma}_{\mathbf{a}}^{*-1} \end{bmatrix} \tag{B.14}$$

Consider the sequence of observations $\{\mathbf{y}_1^k, \ldots, \mathbf{y}_N^k\}$ and the corresponding states

$\{\mathbf{s}_1^k, \ldots, \mathbf{s}_N^k\}$. The joint distribution for the state space model is

$$\begin{aligned} p(\mathbf{y}_1^k, \ldots, \mathbf{y}_N^k, \mathbf{s}_1^k, \ldots, \mathbf{s}_N^k) &= \prod_{n=1}^{N} p(\mathbf{y}_n^k|\mathbf{s}_n^k) p(\mathbf{s}_n^k|\mathbf{s}_{n-1}^k) \\ &= \prod_{n=1}^{N} p(\mathbf{y}_n^k|\mathbf{s}_n^k) p(s_n^k|\mathbf{s}_{n-1}^k). \end{aligned}$$

The posterior

$$p(\mathbf{s}_1^k, \ldots, \mathbf{s}_N^k|\mathbf{y}_1^k, \ldots, \mathbf{y}_N^k) \propto p(\mathbf{y}_1^k, \ldots, \mathbf{y}_N^k, \mathbf{s}_1^k, \ldots, \mathbf{s}_N^k)$$

and

$$\begin{aligned} \log p(\mathbf{s}_1^k, \ldots, \mathbf{s}_N^k|\mathbf{y}_1^k, \ldots, \mathbf{y}_N^k) = &\sum_{n=1}^{N} \log p(\mathbf{y}_n^k|\mathbf{s}_n^k) \\ &+ \sum_{n=1}^{N} \log p(s_n^k|\mathbf{s}_{n-1}^k) + const. \end{aligned} \tag{B.15}$$

From (B.7) to (B.10) we can write

$$\begin{aligned} p(\mathbf{y}_n^k|\mathbf{s}_n^k) &= \mathcal{N}(\mathbf{y}_n^k; \mathbf{H}\mathbf{s}_n^k, \boldsymbol{\Sigma}_v^k) \\ p(s_n^k|\mathbf{s}_{n-1}^k) &= \mathcal{N}(s_n^k; \boldsymbol{\mu}_{\mathbf{a}}^{*T}\mathbf{s}_{n-1}^k, (\bar{\tau}_\epsilon^k)^{-1}) \end{aligned}$$

And evaluating (B.15) we obtain

$$\begin{aligned}\log p(\mathbf{s}_1^k,\ldots,\mathbf{s}_N^k|\mathbf{y}_1^k,\ldots,\mathbf{y}_N^k) &= -\frac{\bar{\tau}_\epsilon^k}{2}\sum_{n=1}^{N}(s_n^k-\boldsymbol{\mu}_{\mathbf{a}}^{*T}\mathbf{s}_{n-1}^k)^2 \\ &-\frac{1}{2}\sum_{n=1}^{N}(\mathbf{y}_n^k-\mathbf{H}\mathbf{s}_n^k)^T\boldsymbol{\Sigma}_v^{k-1}(\mathbf{y}_n^k-\mathbf{H}\mathbf{s}_n^k)+const. \\ &= -\frac{1}{2}\sum_{n=1}^{N}\left\{\bar{\tau}_\eta^k(r_n^k-s_n^k)^2+\bar{\tau}_\epsilon^k\mathbf{s}_n^{kT}\boldsymbol{\Sigma}_{\mathbf{a}}^*\mathbf{s}_n^{kT}\right\} \\ &-\frac{\bar{\tau}_\epsilon^k}{2}\sum_{n=1}^{N}(s_n^k-\boldsymbol{\mu}_{\mathbf{a}}^{*T}\mathbf{s}_{n-1}^k)^2+const. \end{aligned} \tag{B.16}$$

Comparing (4.15) and (B.16) we see that the two expressions are equivalent and we conclude that we can compute $\mathbb{E}\{\mathbf{s}_n^k\}$, $\mathbb{E}\{\mathbf{s}_n^k\mathbf{s}_n^{kT}\}$ and $\mathbb{E}\{\mathbf{s}_n^k\mathbf{s}_{n-1}^{kT}\}$ using a Kalman smoother if we assume that the observations are generated by the state space model described by (B.7) to (B.10). We have $\mathbb{E}\{\mathbf{s}_n^k\}=\mathbb{E}\{[s_n^k,\ldots,s_{n-P+1}^k]^T\}$ and the quantity $\mathbb{E}\{s_n^k\}$ is obtained from the posterior means computed by the Kalman smoother. Also $\mathbb{E}\{\mathbf{s}_n^k\mathbf{s}_n^{kT}\}=\mathrm{Cov}\{\mathbf{s}_n^k\}+\mathbb{E}\{\mathbf{s}_n^k\}\mathbb{E}\{\mathbf{s}_n^k\}^T$. $\mathrm{Cov}\{\mathbf{s}_n^k\}$ is obtained from the Kalman smoother and the second order moments $\mathbb{E}\{(s_n^k)^2\}$ are obtained as follows

$$\mathbb{E}\{(s_n^k)^2\} \;=\; [\mathbb{E}\{\mathbf{s}_n^k\mathbf{s}_n^{kT}\}]_{1,1}.$$

Similarly $\mathbb{E}\{\mathbf{s}_n^k\mathbf{s}_{n-1}^{kT}\}=\mathrm{Cov}\{\mathbf{s}_n^k,\mathbf{s}_{n-1}^{kT}\}+\mathbb{E}\{\mathbf{s}_n^k\}\mathbb{E}\{\mathbf{s}_{n-1}^k\}^T$. $\mathrm{Cov}\{\mathbf{s}_n^k,\mathbf{s}_{n-1}^{kT}\}$ is obtained from the Kalman smoother and $\mathbb{E}\{s_n^k\mathbf{s}_{n-1}^k\}$ is obtained from the first row of $\mathbb{E}\{\mathbf{s}_n^k\mathbf{s}_{n-1}^{kT}\}$.

B.1 Required Expectations

To characterize the parameters of the posterior distributions derived in appendix B we need to compute the following expectations:

1.

$$\mathbb{E}_{\mathbf{s}^k}\Big\{\sum_{n=1}^{N}(r_n^k - s_n^k)^2\Big\} = \mathbb{E}_{\mathbf{s}^k}\Big\{\sum_{n=1}^{N}(r_n^k)^2 - 2r_n^k s_n^k + (s_n^k)^2\Big\}$$

The first and second order moments $\mathbb{E}\{s_n^k\}$, and $\mathbb{E}\{(s_n^k)^2\}$ are computed using a Kalman smoother as discussed in appendix B.

2.

$$\mathbb{E}_{\mathbf{s}^k,\mathbf{a}^k}\Big\{\sum_{n=1}^{N}(s_n^k - \mathbf{a}^{kT}\mathbf{s}_{n-1}^k)^2\Big\} =$$

$$\mathbb{E}_{\mathbf{s}^k,\mathbf{a}^k}\Big\{\sum_{n=1}^{N}\Big((s_n^k)^2 - 2\mathbf{a}^{kT}s_n^k\mathbf{s}_{n-1}^k + \mathbf{a}^{kT}\mathbf{s}_{n-1}^k\mathbf{s}_{n-1}^{kT}\mathbf{a}^k\Big)\Big\}$$

$$= \sum_{n=1}^{N}\Big\{\mathbb{E}\{(s_n^k)^2\} - 2\boldsymbol{\mu}_{\mathbf{a}}^{*T}\mathbb{E}\{s_n^k\mathbf{s}_{n-1}^k\}$$

$$+\boldsymbol{\mu}_{\mathbf{a}}^{*T}\mathbb{E}\{\mathbf{s}_{n-1}^k\mathbf{s}_{n-1}^{kT}\}\boldsymbol{\mu}_{\mathbf{a}}^* + \mathsf{Tr}(\mathbb{E}\{\mathbf{s}_{n-1}^k\mathbf{s}_{n-1}^{kT}\}\boldsymbol{\Sigma}_{\mathbf{a}}^*)\Big\}$$

3.

$$\mathbb{E}_{\mathbf{a}^k}\{(\mathbf{a}^k - \boldsymbol{\mu}_i^a)^T\boldsymbol{\Sigma}_i^{a-1}(\mathbf{a}^k - \boldsymbol{\mu}_i^a)\} =$$

$$(\boldsymbol{\mu}_{\mathbf{a}}^* - \boldsymbol{\mu}_i^a)^T\boldsymbol{\Sigma}_i^{a-1}(\boldsymbol{\mu}_{\mathbf{a}}^* - \boldsymbol{\mu}_i^a) + \mathsf{Tr}(\boldsymbol{\Sigma}_i^{a-1}\boldsymbol{\Sigma}_{\mathbf{a}}^*)$$

4.

$$\bar{\tau}_\eta^k \stackrel{\text{def}}{=} \mathbb{E}_{\tau_\eta^k}\{\tau_\eta^k\} = \frac{a_\eta^*}{b_\eta^*}$$

$$\bar{\tau}_\epsilon^k \stackrel{\text{def}}{=} \mathbb{E}_{\tau_\epsilon^k}\{\tau_\epsilon^k\} = \frac{a_\epsilon^*}{b_\epsilon^*}$$

5.

$$\mathbb{E}_{\mathbf{z}_a^k}\{z_{a,i}^k\} = \gamma_i^k$$

Bibliography

[1] F. Bimbot, J.-F. Bonastre, C. Fredouille, G. Gravier, I. Magrin-Chagnolleau, S. Meignier, T. Merlin, J. Ortega-García, D. Petrovska-Delacétaz, D. A. Reynolds. A Tutorial on Text-Independent Speaker Verification. *EURASIP J. Applied Signal Processing*, (4):430–451, 2004.

[2] Toon van Waterschoot, Geert Rombouts, and Marc Moonen. Optimally regularized adaptive filtering algorithms for room acoustic signal enhancement. *Signal Process.*, 88(3):594–611, 2008.

[3] Ji Ming, T.J. Hazen, J.R. Glass, and D.A. Reynolds. Robust speaker recognition in noisy conditions. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(5):1711–1723, July 2007.

[4] J. M. Bernardo and A. F. M. Smith. *Bayesian Theory*. Wiley, New York, 1994.

[5] Steven M. Kay. *Fundamentals of Statistical Signal Processing, Volume I: Estimation Theory*. Prentice Hall PTR, March 1993.

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

[7] Olivier Cappé, Eric Moulines, and Tobias Rydén. *Inference in Hidden Markov Models*. Springer Science and Business Media, 2005.

[8] T. Cover and J. Thomas. *Elements of Information Theory*. John Wiley and Sons, 2006.

[9] Hagai Attias. A Variational Bayesian Framework for Graphical Models. In *Advances in Neural Information Processing Systems 12*, pages 209–215. MIT Press, 2000.

[10] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[11] John Winn and Christopher M. Bishop. Variational Message Passing. *J. Mach. Learn. Res.*, 6:661–694, 2005.

[12] Steffen L. Lauritzen and David J. Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society, Series B*, 50:157–224, 1988.

[13] M. J. Wainwright and M. I. Jordan. A Variational Principle for Graphical Models. In S. Haykin, J. Príncipe, T. J. Sejnowski, and J. McWhirter, editor, *New Directions in Statistical Signal Processing From Systems to Brains*, pages 155–202. MIT press, 2005.

[14] Andrew Gelman, John B. Carlin, Hal S. Stern, and Donald B. Rubin. *Bayesian Data Analysis, Second Edition*. Chapman & Hall/CRC, July 2003.

[15] Lawrence Rabiner and Biing-Hwang Juang. *Fundamentals of speech recognition*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1993.

[16] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems : Networks of Plausible Inference*. Morgan Kaufmann, September 1988.

[17] Brendan J. Frey and David J. C. Mackay. A revolution: Belief propagation in graphs with cycles. In *Neural Information Processing Systems*, pages 479–485. MIT Press, 1998.

[18] Kevin P. Murphy, Yair Weiss, and Michael I. Jordan. Loopy belief propagation for approximate inference: An empirical study. In *Proceedings of Uncertainty in AI*, pages 467–475, 1999.

[19] T. Heskes. Stable fixed points of loopy belief propagation are minima of the Bethe free energy. In *Neural Information Processing Systems*, volume 15, pages 343–350. MIT Press, 2002.

[20] W.R. Gilks, S. Richardson, and D.J. Spiegelhalter. *Markov Chain Monte Carlo in Practice*. Chapman and Hall, London, 1996.

[21] C. Févotte and S. Godsill. A Bayesian Approach to Blind Separation of Sparse Sources. *IEEE Transactions on Audio, Speech and Language Processing*, 14(6):2174–2188, Nov 2006.

[22] Andrew Varga and Herman J.M. Steeneken. Assessment for automatic speech recognition: II. NOISEX-92: A database and an experiment to study the effect of additive noise on speech recognition systems. *Speech Communication*, 12(3):247 – 251, 1993.

[23] P. Loizou. *Speech Enhancement: Theory and Practice*. CRC Press, 2007.

[24] S. Boll. Suppression of acoustic noise in speech using spectral subtraction. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 27(2):113 – 120, Apr. 1979.

[25] Yang Lu and Philipos C. Loizou. A geometric approach to spectral subtraction. *Speech Communication*, 50(6):453–466, 2008.

[26] Y. Ephraim and H.L. Van Trees. A signal subspace approach for speech enhancement. *IEEE Transactions on Speech and Audio Processing*, 3(4):251 –266, Jul. 1995.

[27] Y. Ephraim and D. Malah. Speech enhancement using a minimum-mean square error short-time spectral amplitude estimator. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 32(6):1109 – 1121, Dec. 1984.

[28] P.J. Wolfe and S.J. Godsill. Simple alternatives to the Ephraim and Malah suppression rule for speech enhancement. In *Proceedings of the 11th IEEE Signal Processing Workshop on Statistical Signal Processing*, pages 496 –499, 2001.

[29] I. Cohen. Noise spectrum estimation in adverse environments: improved minima controlled recursive averaging. *IEEE Transactions on Speech and Audio Processing*, 11(5):466 – 475, Sept. 2003.

[30] O. Cappé. Elimination of the musical noise phenomenon with the Ephraim and Malah noise suppressor. *IEEE Transactions on Speech and Audio Processing*, 2(2):345 –349, Apr 1994.

[31] R. Martin. Speech Enhancement Based on Minimum Mean-Square Error Estimation and Supergaussian Priors. *IEEE Transactions on Speech and Audio Processing*, 13(5):845 – 856, Sept. 2005.

[32] P.C. Loizou. Speech enhancement based on perceptually motivated bayesian estimators of the magnitude spectrum. *IEEE Transactions on Speech and Audio Processing*, 13(5):857 – 869, sept. 2005.

[33] J. P. Campbell. Speaker recognition: a tutorial. *Proc. IEEE*, 85(9):1437–1462, Sep. 1997.

[34] D. E. Sturim, W. M. Campbell, and D. A. Reynolds. Classification methods for speaker recognition. pages 278–297, 2007.

[35] D. Reynolds and R. Rose. Robust text-independent speaker identification using gaussian mixture speaker models. *IEEE Trans. Speech Audio Processing*, 3(1):72–83, 1995.

[36] S. Young, et al. *The HTK Book for version 3.4*. Microsoft Corporation, CUED, 2006.

[37] D. A. Reynolds, T. F. Quatieri, R. B. Dunn. Speaker Verification Using Adapted Gaussian Mixture Models. *Digital Signal Processing*, 10:19–41, 2000.

[38] Li Deng, J. Droppo, and A. Acero. Estimating cepstrum of speech under the presence of noise using a joint prior of static and dynamic features. *IEEE Transactions on Speech and Audio Processing*, 12(3):218–233, May 2004.

[39] R. Vogt and S. Sridharan. Experiments in session variability modelling for speaker verification. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 1, page I, May 2006.

[40] H. Hermansky and N. Morgan. RASTA processing of speech. *IEEE Transactions on Speech and Audio Processing*, 2(4):578–589, 1994.

[41] Hagai Attias, John C. Platt, Alex Acero, and Li Deng. Speech denoising and dereverberation using probabilistic models. In *Advances in Neural Information Processing Systems 13*. MIT Press, 2001.

[42] Jiucang Hao, H. Attias, S. Nagarajan, Te-Won Lee, and T.J. Sejnowski. Speech Enhancement, Gain, and Noise Spectrum Adaptation Using Approximate Bayesian Estimation. *IEEE Transactions on Audio, Speech, and Language Processing*, 17(1):24–37, Jan. 2009.

[43] B. J. Frey, T. T. Kristjansson, L. Deng, and A. Acero. ALGONQUIN Learning dynamic noise models from noisy speech for robust speech recognition. In *Advances in Neural Information Processing Systems 14*, pages 1165–1172, January 2002.

[44] Kristjansson, T. *Speech Recognition in Adverse Environments: a Probabilistic Approach*. PhD thesis, 2002.

[45] Li Deng, J. Droppo, and A. Acero. Recursive estimation of nonstationary noise using iterative stochastic approximation for robust speech recognition. *IEEE Transactions on Speech and Audio Processing*, 11(6):568–580, Nov. 2003.

[46] J. Droppo, L. Deng, A. Acero. A Comparison of Three Non-Linear Observation Models for Noisy Speech Features. In *Eurospeech*, pages 681–684, 2003.

[47] Li Deng, J. Droppo, and A. Acero. Enhancement of log mel power spectra of speech using a phase-sensitive model of the acoustic environment and sequential estimation of the corrupting noise. *IEEE Transactions on Speech and Audio Processing*, 12(2):133–143, March 2004.

[48] A. Solomonoff, W. M. Campbell, and I. Boardman. Advances in channel compensation for SVM speaker recognition. In *Proc. IEEE Int. Conf. Acoust. Speech, Signal Process. (ICASSP)*, volume 1, page 629632, Philadelphia, PA, March 2005.

[49] A. Solomonoff, C. Quillen, W. M. Campbell. Channel Compensation for SVM Speaker Recognition. In *In Proc. Odyssey: The Speaker and Language Recognition Workshop*, pages 41–44, Toledo, Spain, June 2004.

[50] F. Castaldo, D. Colibro, E. Dalmasso, P. Laface, C. Vair. Compensation of Nuisance Factors for Speaker and Language Recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(7):1969 – 1978, September 2007.

[51] P. Kenny, G. Boulianne, P. Dumouchel. Eigenvoice Modeling with Sparse Training Data. *IEEE Transactions on Speech and Audio Processing*, 13:345–359, May 2005.

[52] P. Kenny, G. Boulianne, P. Ouellet, P. Dumouchel. Speaker and session variability in GMM-based speaker verification. *IEEE Transactions on Audio, Speech and Language Processing*, 15(4):1448–1460, May 2007.

[53] S. Lucey, T. Chen. Improved speaker verification through probabilistic subspace adaptation. In *Proceedings of the 8th European Conference on Speech Communication and Technology (Eurospeech 2003)*, pages 2021–2024, Geneva, Switzerland, September 2003.

[54] Jongseo Sohn, Nam Soo Kim, and Wonyong Sung. A statistical model-based voice activity detection. *IEEE Signal Processing Letters*, 6(1):1 –3, Jan. 1999.

[55] J.-M. Gorriz, J. Ramirez, E.W. Lang, and C.G. Puntonet. Jointly Gaussian PDF-Based Likelihood Ratio Test for Voice Activity Detection. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(8):1565 –1578, Nov. 2008.

[56] A. Benyassine, E. Shlomot, H.-Y. Su, D. Massaloux, C. Lamblin, and J.-P. Petit. ITU-T Recommendation G.729 Annex B: a silence compression scheme for use with G.729 optimized for V.70 digital simultaneous voice and data applications. *IEEE Communications Magazine*, 35(9):64 –73, Sep. 1997.

[57] J.S. Garofolo, L.F. Lamel, W.M. Fisher, J.G. Fiscus, D.S. Pallett and N.L. Dahlgren. The DARPA TIMIT acoustic-phonetic continuous speech corpus CDROM , 1993. http://www.ldc.upenn.edu/Catalog.

[58] R. Woo, A. Park, T. J. Hazen. The MIT Mobile Device Speaker Verification Corpus: Data collection and preliminary experiments. In *Proc. Odyssey: The Speaker and Language Recognition Workshop*, pages 1–6, San Juan, Puerto Rico, 2006.

[59] M. Cooke and Te-Won Lee. Speech separation challenge, 2006. http://www.dcs.shef.ac.uk/martin/SpeechSeparationChallenge.htm.

[60] NIST 2004 Speaker Recognition Evaluation plan", 2004. http://www.nist.gov/speech/tests/spk/2004/SRE-04_evalplanv1a.pdf.

[61] D.A. Reynolds, W. Campbell, T.T. Gleason, C. Quillen, D. Sturim, P. Torres-Carrasquillo, and A. Adami. The 2004 MIT Lincoln Laboratory Speaker Recognition System. In *IEEE International Conference on Acoustics, Speech, and Signal Processing,*, volume 1, pages 177–180, 18-23, 2005.

[62] S. Kajarekar, L. Ferrer, E. Shriberg, K. Sonmez, A. Stolcke, A. Venkataraman, and Jing Zheng. SRI's 2004 NIST Speaker Recognition Evaluation System. In *IEEE International Conference on Acoustics, Speech, and Signal Processing,*, volume 1, pages 173–176, 18-23, 2005.

[63] J.-F. Bonastre, Nicolas Scheffer, Corinne Fredouille, Driss Matrouf. NIST04 speaker recognition evaluation campaign: New LIA speaker detection platform based on ALIZE toolkit,. In *NIST SRE04 Workshop: Speaker Detection Evaluation Campaign*, Toledo, Spain, 2004.

[64] J.-F. Bonastre, F. Wils, and S. Meignier. ALIZE, a free toolkit for speaker recognition. In *IEEE International Conference on Acoustics, Speech, and Signal Processing,*, volume 1, pages 737–740, 18-23, 2005.

[65] D. A. van Leeuwen. Speaker adaptation in the NIST speaker recognition evaluation 2004. In *Interspeech*, pages 1981–1984, 2005.

[66] J.R. Hershey and P.A. Olsen. Approximating the Kullback Leibler Divergence Between Gaussian Mixture Models. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 4, pages 317 –320, April 2007.

[67] Qinghua Huang, Jie Yang, and Shoushui Wei. Temporally correlated source separation using variational Bayesian learning approach. *Digital Signal Processing*, 17(5):873 – 890, 2007. Special Issue on Bayesian Source Separation.

[68] S.J. Roberts and W.D. Penny. Variational Bayes for generalized autoregressive models. *IEEE Transactions on Signal Processing*, 50(9):2245–2257, Sep 2002.

[69] Yi Hu and P.C. Loizou. Evaluation of objective quality measures for speech enhancement. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(1):229 –238, Jan. 2008.

[70] B. Frey, L. Deng, A. Acero, and T. Kristjansson. Algonquin: iterating Laplace's method to remove multiple types of acoustic distortion for robust speech recognition. In *Eurospeech*, pages 901–904, January 2001.