UNIVERSITY OF CALIFORNIA, SAN DIEGO

Speech Enhancement and Source Separation using Probabilistic Models

A dissertation submitted in partial satisfaction of the requirements for the degree
Doctor of Philosophy

in

Physics

by

Jiucang Hao

Committee in charge:

Professor Terrence Sejnowski, Co-Chair
Professor Herbert Levine, Co-Chair
Professor Gert Cauwenberghs
Professor Terrence Hwa
Professor David Kleinfeld

2008

The dissertation of Jiucang Hao is approved, and it is acceptable in quality and form for publication on microfilm.

Co-Chair

Co-Chair

University of California, San Diego

2008

To my family

Table of Contents

List of Figures

List of Tables

Acknowledgments

I would like to take this opportunity to acknowledge many people without whom this work would not have been possible.

First, I thank my advisor Dr. Terrence Sejnowski. As a physics student, I would never work on machine learning and signal processing without Dr. Sejnowski's supervision. I benefit a lot from his support and deep knowledge in many fields. He introduced Dr. Te-Won Lee who became my research advisor. When Dr. Lee was on leave of absence in my last year of study, Dr. Sejnowski generously offered me a position in his Computational Neurobiology Laboratory (CNL) where I could concentrate on and finish my thesis. Dr. Sejnowski also provided valuable suggestions on my research and this thesis. Second, I thank my advisor Dr. Te-Won Lee. As a engineer and researcher, Dr. Lee introduced a new dimension into my thinking and research. Apart from theoretical rigorousness, he focused on the very challenging practical problems and created the cutting edge techniques to solve them. The inspiration, intuition and caring of Dr. Lee carried me along the whole graduate path. Third, I thank Dr. Hagai Attias. As a physicist, Dr. Attias shares a lot common sense with me. His successful transition from physicist to machine learning expert encouraged me whenever I felt depressed. Dr. Attias has been the great help whenever I need on almost anything, for example, mathematical derivation, code debugging, manuscript review. Significant part of this thesis was based on the work done together with Dr. Attias.

I am extremely fortunate to have such a super mentor.

I would like to thank members of the Medical Advice from Glaucomatous Informatics (MAGI) project: Dr. Michael Goldbaum, Dr. Pamela Sample, Dr. Linda Zangwill, Dr. Christopher Bowd, Dr. Catherine Boden, Dr. Lyne Racette. I thank the support for my study and I am proud to be a member of MAGI.

Special thanks go to the laboratories I stayed. The CNL hosts tea everyday. My knowledge was broadened in talking to people with different background. Members of Dr. Te-Won Lee's Lab of Machine Learning for Signal Processing has been my friends, we discussed almost everything. I spent a wonderful summer in Dr. Srikantan Nagarajan's Biomagnetic Imaging Laboratory in UCSF. I am proud to be member of these laboratories.

Last, but not the least, I owe so much to my family. Thanks for their love, support and understanding – the time I worked on this thesis could have been the time I spent with my family.

Chapter 2, Chapter 3 and Chapter 5 contain materials in J. Hao, H. Attias, S. Nagarajan, T.-W. Lee and T. Sejnowski, "Speech Enhancement, Gain, and Noise Spectrum Adaptation Using Approximate Bayesian Inference", *IEEE Transactions on Audio, Speech, and Language Processing*, accepted for publication. Chapter 2, Chapter 4 and Chapter 5 contain materials in J. Hao, T-W. Lee and T. Sejnowski, "Speech Enhancement Using Gaussian Scale Mixture Models", *IEEE Transactions on Audio, Speech, and Language Processing*, submitted for publication. Chapter 6, Chapter 7, Chapter 8 and Chapter 9 contain materials in J. Hao, I. Lee, T.-W. Lee and T. Sejnowski, "Source Separation with Independent Vector Analysis", to be submitted.

Curriculum Vitæ

Education

2008
Doctor of Philosophy in Physics
University of California at San Diego, La Jolla, CA

2004
Master of Science in Physics
University of California at San Diego, La Jolla, CA

2001
Bachelor of Science in Physics
University of Science and Technology of China, Hefei, Anhui, P. R. China.

Research Experience

2003–2008
Research Assistant, Institute for Neural Computation
University of California at San Diego, California, USA
Co-advisor: Terrence Sejnowski, Computational Neurobiology Laboratory, The Salk Institute, and Department of Biology, University of California at San Diego
Co-advisor: Te-Won Lee, Institute for Neural Computation, University of California at San Diego
Developed algorithms for speech enhancement and source separation using probabilistic models. Applied machine learning algorithms to analyze the medical data and evaluated the performance.

1999-2001
Research Assistant, Key Laboratory of Quantum Information
University of Science and Technology of China, P.R. China
Advisor: Guang-Can Guo, Key Laboratory of Quantum Information and Department of Physics, University of Science and Technology of China
Proposed schemes for quantum teleportation and dense coding using entanglement.

Teaching Experience

2001-2003
Teaching Assistant, Department of Physics
University of California at San Diego, California, USA

Instructed physics experiments, led discussion session and problem session, graded homework and exams.

Selected Publication

J. Hao, I. Lee, T.-W. Lee, T. Sejnowski. Source Separation Using Independent Vector Analysis. To be submitted.

J. Hao, T-W. Lee, T. Sejnowski. Speech Enhancement Using Gaussian Scale Mixture Models. Submitted to *IEEE Transactions on Audio, Speech, and Language Processing*.

J. Hao, H. Attias, S. Nagarajan, T.-W. Lee, T. Sejnowski. Speech Enhancement, Gain, and Noise Spectrum Adaptation Using Approximate Bayesian Estimation. *IEEE Transactions on Audio, Speech, and Language Processing*. Accepted for publication.

C. Bowd, J. Hao, I Tavares, F. Medeiros, L. Zangwill, T.-W. Lee, P. Sample, R. Weinreb, M. Goldbaum. Bayesian Machine Learning Classifiers for Combining Structural and Functional Measurements to Classify Healthy and Glaucomatous Eyes. *Investigative Ophthalmology & Visual Science, Vol.* 49*, pp.*945 − 953*,* 2008.

C. Boden, K. Chan, P. Sample, J. Hao, T.-W. Lee, L. Zangwill, R. Weinreb, M. Goldbaum. Assessing Visual Field Clustering Schemes Using Machine Learning Classifiers in Standard Perimetry. *Investigative Ophthalmology & Visual Science, Vol.* 48*, pp.*5582 − 5590*,* 2007.

I. Kozak, P. Sample, J. Hao, W. Freeman, R. Weinreb, T.-W. Lee, M. Goldbaum. Machine Learning Classifiers Detect Subtle Field Defects in Eyes of HIV Individuals. *Transactions of the American Ophthalmological Society, Vol.* 105*, pp.*111 − 120*,* 2007.

K. Petersen, J. Hao, T.-W. Lee. Generative and Filtering Approaches for Overcomplete Representations. *Neural Information Processing - Letters and Reviews, Vol.* 8*, No.* 1*,* 2005.

M. Goldbaum, P. Sample, Z. Zhang, K. Chan, J. Hao, T.-W. Lee, C. Boden, C. Bowd, R. Bourne, L. Zangwill, T. Sejnowski, D. Spinak, R. Weinreb. Using unsupervised learning with independent component analysis to identify patterns of glaucomatous visual field defects. *Investigative Ophthalmology & Visual Science, Vol.* 46*, pp.*3676 − 3683*,* 2005.

C. Bowd, F. Medeiros, Z. Zhang, L. Zangwill, J. Hao, T.-W. Lee, T. Sejnowski, R. Weinreb, M. Goldbaum. Relevance vector machine and support vector machine classifier analysis of scanning laser polarimetry retinal nerve fiber layer measurements. *Investigative Ophthalmology & Visual Science, Vol.* 46*, pp.*$1322 - 1329$*,* 2005.

L. Zangwill, K. Chan, C. Bowd, J. Hao, T.-W. Lee, R. Weinreb, T. Sejnowski, M. Goldbaum. Heidelberg retina tomograph measurements of the optic disc and parapapillary retina for detecting glaucoma analyzed by machine learning classifiers. *Investigative Ophthalmology & Visual Science, Vol.* 45*, pp.*$3144 - 3151$*,* 2004.

C. Bowd, L. Zangwill, F. Medeiros, J. Hao, K. Chan, T.-W. Lee, T. Sejnowski, M. Goldbaum, P. Sample, J. Crowston, R. Weinreb. Confocal scanning laser ophthalmoscopy classifiers and stereophotograph evaluation for prediction of visual field abnormalities in glaucoma-suspect eyes. *Investigative Ophthalmology & Visual Science, Vol* 45*, pp.*$2255 - 2262$*,* 2004.

P. Sample, K. Chan, C. Boden, T.W. Lee, E. Blumenthal, R. Weinreb, A. Bernd, J. Pascual, J. Hao, T. Sejnowski, M. Goldbaum. Using unsupervised learning with variational bayesian mixture of factor analysis to identify patterns of glaucomatous visual field defects. *Investigative Ophthalmology & Visual Science, Vol* 45*, pp.*$2596 - 2605$*,* 2004.

J. Hao, C. Li, G. Guo. Controlled Dense Coding using the Greenberger-Horne-Zeilinger State. *Physical Review A, Vol.* 63*,* 054301*,* 2001.

J. Hao, C. Li, G. Guo. Probabilistic Dense Coding and Teleportation. *Physics Letters A, Vol.* 278(3)*, pp.*$113 - 117$*,* 2000.

ABSTRACT OF THE DISSERTATION

Speech Enhancement and Source Separation using Probabilistic Models

by

Jiucang Hao
Doctor of Philosophy in Physics

University of California, San Diego, 2008

Professor Terrence Sejnowski, Co-Chair

Professor Herbert Levine, Co-Chair

Statistical signal processing has been very successful. We proposed novel probabilistic models and developed efficient algorithms for two important problems: speech enhancement and source separation.

Part I focused on the speech enhancement. We developed two models with efficient algorithms. The first one assumed a Gaussian Mixture Model (GMM) in the log-spectral domain for speech prior which was trained by expectation maximization (EM) algorithm. Three approximations were employed to enhance the computational efficiency. The Laplace method estimated the signal by computing the mode of the posterior distribution, either in the frequency domain or in the log-spectrum domain. The Gaussian approximation converted the GMM in the log-spectrum domain into a GMM in the frequency domain by minimizing the KL-divergency. It provided an efficient gain and noise spectrum estimation with the EM algorithm. The second one used a Gaussian scale mixture model (GSMM) as speech prior. This model specified a stochastic dependency between the log-spectra and the frequency components which can be estimated simultaneously with GSMM. The algorithms for training the model and signal estimation were developed. All these algorithms were evaluated by applying them to enhance the speeches corrupted by the speech shaped noise (SSN). The experimental results demonstrated that the proposed algorithms improved the signal-to-noise ratio and lowered the word recognition error rate.

In part II, a novel probabilistic framework based on Independent Vector Analysis (IVA) was proposed to separate the convolutive mixture of sources. IVA assumed a multidimensional GMM for the source priors. The joint modeling of all frequency bins originating from the same source prevented the permutation disorder that associated with independent component analysis (ICA). The GMM source priors could adapt to the statistics of the sources and enable IVA to separate different type of signals. We developed EM algorithms for both the noiseless case and noisy case. For noiseless case, an online algorithm was developed to handle non-stationary environments. For noisy case, noise reduction was achieved together with the separation processes. The algorithms were evaluated by applying them to separate the mixtures of speech and music. The experimental results showed improved performance over other algorithms.

1

Introduction to Probabilistic Models and Acoustic Signals

Probabilistic models have been very powerful and successful in many fields, for example, Bayesian data analysis, signal processing, communications, financial analysis, and bioinformatics. As the focus of the machine learning research, researchers have been actively developing more precise models and efficient algorithms. The main advantage of the probabilistic models comes from the allowance of uncertainty and they are appropriate for problems with unknown or nonexist exact processes. Consider the process of the speech generation and recording. The repeated recordings of the same content are different. So it is not possible to specify the exact relationship between the recorded signals and their content. However, human can easily understand the content of the recorded signals, because they are not completely random conditioned on that the noise is small. Thus we need to specify the stochastic relationship and the probabilistic models are a perfect fit.

Another advantage of the probabilistic models comes for the inference mechanism.

Due to the stochastic relationship, the clean signals cannot be extracted from the observations deterministically. But we can say something about the signal from a probabilistic perspective. The Bayesian inference is to compute the posterior distribution, conditioned on the observations. Reconsider the speech problem. The human perception can be thought as an inference. Listening to the recorded signal, human can infer the words and the content. This idea has been successfully applied to speech recognition engines.

In this chapter, we present the preliminaries for the probabilistic models and the applications to speech enhancement and source separation.

1.1 Preliminaries for Probabilistic Models

There are four problems for probabilistic models: model specification, parameter estimation, Bayesian inference and approximations to enhance computational efficiency.

1.1.1 Model Specification

A probabilistic model is determined by the prior distribution, the conditional distribution and the parameters. Let x and y be two random variables. We write their joint distribution as

$$p(x, y) = p(y|x)p(x|\theta) \tag{1.1}$$

The $p(x|\theta)$ is the prior distribution with parameter θ. In general, the prior contains the structures we are looking for. For example, a density in a lower dimensional space could extract the important components that explain the observations. A peaky shape for $p(x)$ enforces the sparseness for efficient coding. It can also model the dynamics and the Kalman filter is an example. The conditional probability $p(y|x)$ describes the dependency of y on x

which is often considered as noise and assumed to be a Gaussian. The y obeys some process specified by $p(x)$ in a stochastic manner.

1.1.2 Parameter Estimation

The parameters θ can be either learned from the training set or estimated from the observations, with maximum likelihood. Let $\{y_1, \cdots, y_T\}$ be the training set and θ be the parameters for the prior $p(x|\theta)$. Assuming the independent samples, the likelihood is given by

$$\begin{aligned}
\mathcal{L}(\theta) &= \sum_n \log p(y_n|\theta) = \sum_n \log \int dx_n p(y_n|x_n) p(x_n|\theta) && (1.2)\\
&\geq \sum_n \int dx_n q(x_n) \log \frac{p(y_n|x_n)p(x_n|theta)}{q(x_n)} && (1.3)\\
&= \mathcal{F}(q, \theta) && (1.4)
\end{aligned}$$

The inequality holds for any $q(x_n)$ and the $\mathcal{F}(q, \theta) = \mathcal{L}(\theta)$ when $q(x_n) = p(x_n|y_n, \theta)$, the posterior density.

The maximum likelihood estimator for θ is

$$\hat{\theta} = \arg\max_\theta \sum_n \log p(y_n|\theta) \quad (1.5)$$

which can be efficiently estimated by the Expectation Maximization (EM) algorithm [1] that iterates the E-step and M-step,

$$\begin{aligned}
&\text{E-step:} && q(x_n) = p(x_n|y_n, \theta) = \frac{p(y_n|x_n)p(x_n|\theta)}{p(y_n)} && (1.6)\\
&\text{M-step:} && \theta = \arg\max \mathcal{F}(q, \theta) && (1.7)
\end{aligned}$$

The E-step computes the posterior probability and the M-step updates the parameters. One appealing property of EM algorithm is that the cost function $\mathcal{F}(q, \theta)$ increases monotonically, which is very useful to monitor the convergence.

1.1.3 Bayesian Inference

Bayesian inference extracts the information about x from the observation y based on the model assumptions. The x is not determined by y when they are related stochastically. Fortunately, Bayesian theory offers a systematic inference mechanism for the hidden variable x by computing its posterior distribution, per Bayes' rule

$$p(x_n|y_n, \theta) = \frac{p(y_n|x_n)p(x_n, \theta)}{p(y_n)} \tag{1.8}$$

Sometimes it is necessary to use the point estimators, of which the minimum mean square error (MMSE) and the maximum a posterior (MAP) estimators are the most popular ones

$$\begin{aligned} \text{MMSE:} \qquad \hat{x}_n &= \int x_n p(x_n|y_n) dx_n && (1.9) \\ \text{MAP:} \qquad \hat{x}_n &= \arg\max_{x_n} p(x_n|y_n) && (1.10) \end{aligned}$$

1.1.4 Approximations

Exact parameter estimation and inference for the probabilistic models are rarely tractable. Approximations are required to enhance the computational efficiency. The bottleneck of the EM algorithm is the E-step which is often very hard. The Gaussian approximation, Laplace method and variational approximation are the popular ones that are used in this thesis.

The Gaussian approximation computes the mean and the covariance of the posterior probability,

$$\mu_n = \int x_n p(x_n|y_n) dx_n \tag{1.11}$$

$$\frac{1}{\nu_n} = \int (x_n - \mu_n)^2 p(x_n|y_n) dx_n \tag{1.12}$$

The resulted Gaussian, $p(x_n|y_n, \theta) \approx \mathcal{N}(x_n|\mu_n, \nu_n)$, has the correct first and second order statistics.

The Laplace method computes the Taylor expansion of the likelihood around its mode (the MAP) to the second order

$$\log p(x_n|y_n, \theta) \approx c + \frac{1}{2} H (x_n - \mu_n)^2 \tag{1.13}$$

where $H = \frac{\partial^2 \log p(x_n|y_n,\theta)}{\partial x_n^2}$ is the Hessian. This Laplace method, $p(x_n|y_n, \theta) \approx \mathcal{N}(x_n|\mu_n, H)$, focuses on the most likely region of the posterior distribution and has the correct MAP value.

The variational approximation [2] is very effective for models with more than one hidden variables. We consider two hidden variables here x and ξ for simplicity. When the posterior density is hard to compute, a factorized $p(x_n, \xi_n|y_n) \approx q(x_n)q(\xi_n)$ is used. The accuracy of the approximation is measured by the Kullback-Leibler (KL) divergence defined by

$$D(q||p) = \int dx_n d\xi_n q(x_n) q(\xi_n) \frac{p(x_n, \xi_n|y_n)}{q(x_n)q(\xi_n)} \tag{1.14}$$

which is positive and equals to zero when $q = p$. The optimal $q(x_n)$ and $q(\xi)$ minimize the

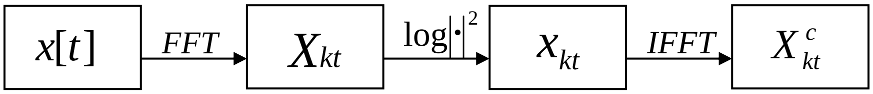

Figure 1.1: Diagram for the relationship among the time domain, the frequency domain, the log-spectral domain and the cepstral domain.

KL-divergency and they satisfy

$$
\begin{aligned}
q(x_n) &= \frac{1}{Z_1} e^{\int q(\xi_n)(\log p(y_n|x_n,\xi_n)+\log p(x_n,\xi_n))d\xi_n} && (1.15)\\
q(\xi_n) &= \frac{1}{Z_2} e^{\int q(x_n)(\log p(y_n|x_n,\xi_n)+\log p(x_n,\xi_n))dx_n} && (1.16)
\end{aligned}
$$

where Z_1 and Z_2 are the normalization factors. The $q(x_n)$ and $q(\xi_n)$ can be iteratively optimized.

1.2 Representations of Acoustic Signals

The signals we consider in this thesis were limited to digital signals that are recorded from microphones. The time domain signal is represented by $x[t]$ at time t. Because the time domain signal has less perceptual meaning, signals are transformed into other domains. Frequency coefficients X_k are obtained by applying the fast Fourier transformation (FFT) on the segmented and windowed signal $x[t]$,

$$X_k = \sum_{n=0}^{K-1} x[n] e^{-2\pi i k n/K} \tag{1.17}$$

The log-spectra are computed as the logarithm of the magnitude of the FFT coefficients,

$$x_k = \log(|X_k|^2) \tag{1.18}$$

The cepstral coefficients x_k^c are computed by applying the inverse FFT (IFFT) on the log-spectra x_k,

$$x_k^c = \frac{1}{K} \sum_{n=0}^{K-1} x_k e^{2\pi ikn/K} \tag{1.19}$$

Figure 1.1 shows the relationship among different domains. Due to the symmetry of the FFT coefficients, the k^{th} component X_k is the complex conjugate of X_{K-k}. Thus we only keep the first $K/2 + 1$ components, because the rest provides no additional information. And IFFT contains the same property. Due to this symmetry, the cepstral coefficients x_{kt}^c are real.

1.3 Acoustic Signal Processing using Probabilistic Models

Acoustic signals, for instance speech and music, have intrinsic statistical properties. The generation process, propagations channels, and the recording devices are subject to uncertainty. Human or musical instrument generate different acoustic waves of the same content. The microphones have its own noise and the recordings in time domain are subject to noise and distortion. The exact process for speech and noise does not exist. Thus probabilistic models have been popular and successful in acoustic signal processing.

Building novel probabilistic models and developing efficient algorithms are two focuses of this thesis. We approach the goals by controlling the model complexity and applying efficient approximations. Two problems are addressed: speech enhancement and source separation.

1.3.1 Part I: Speech Enhancement

Speech enhancement improves the quality of the signals by recovering them from the noisy recordings. The speech model is described by a prior probability $p(x)$ which can be trained by samples of clean signals or estimated from noisy observations y. Speech enhancement estimate the original signal x from y based on the model assumptions. As in Eq.(1.8), the posterior distribution contains all the information about x. To reconstruct the signal, a point estimator for x is necessary. Two approaches are commonly used: the MMSE estimator given by Eq.(1.10) and the MAP estimator given by Eq.(1.10).

In part I, speech enhancement algorithms based on Bayesian inference are developed. First, we derive several approximations to infer the signals using the GMM in the log-spectral domain as a speech prior. Under the Gaussian approximation, an EM algorithm for gain and noise spectrum estimation is developed. Second, a novel Gaussian scale mixture model with two hidden variables is proposed for speech signals. This model provides the inference for both the frequency component and the log-spectra, which are useful for different applications: the estimated FFT coefficients provide better signal reconstruction in the time domain, while the estimated log-spectra are more appropriate for recognizing the noisy speech.

1.3.2 Part II: Source Separation

Different from speech enhancement, the source separation uses two microphones and separate the mixed signals originating from two sources. The key assumption is that the two sources are statistically independent. Let $\mathbf{X} = (X_1, X_2)^T$ denote the original signals from the two sources, and $\mathbf{Y} = (Y_1, Y_2)^T$ be the recorded signals. The independency is described by the factorized source prior $p(X_1)p(X_2)$. The shape of the density reflects the statistics of the signal and the product form enforces the independency. Independent vector analysis (IVA) assumes a linear model, $\mathbf{y} = \mathbf{Ax}$. It searches the unmixing matrix W such that $\mathbf{WY}$ contains independent components and achieves separation. The maximum likelihood estimator for W is

$$\hat{W} = \arg\max_{W} \log \int p(\mathbf{Y}|\mathbf{X}, W)p(\mathbf{X}|\theta)d\mathbf{X} \tag{1.20}$$

In part II, we develop several algorithms for IVA by maximizing the likelihood. Signals are estimated using the MMSE estimator. Efficient EM algorithms to separate speech from music are developed. An online algorithm is proposed to handle the nonstationary environment or sources. The IVA is extended to the noisy case, where noise reduction and source separation are achieved simultaneously.

Part I

Speech Enhancement

2

Introduction to Speech Enhancement

In the real environment, speech signals are usually corrupted by adverse noise, such as competing speakers, background noise, or car noise, and also they are subject to distortion caused by communication channels, examples are room reverberation, low quality microphones, etc. Other than specialized studios or laboratories when audio signal is recorded, noise is recorded as well. In some circumstances such as cars in traffic, noise level could exceed speech signal. Speech enhancement improves the signal quality by suppression of noise and reduction of distortion. Speech enhancement has many applications, for example, mobile communications, robust speech recognition, low quality audio devices and hearing aids.

Because of its broad application range, speech enhancement has attracted intensive research for many years. The difficulty arises from the fact that precise models for both speech signal and noise are unknown [3], thus speech enhancement problem remains unsolved

[4]. A vast variety of models and speech enhancement algorithms are developed which can be broadly classified into two categories: single microphone class and multi-microphone class. While the second class can be potentially better because of having multiple inputs from microphones, it also involves complicated joint modeling of microphones such as beamforming [4, 5, 6]. Algorithms based on a single microphone have been a major research focus and a popular subclass is spectral domain algorithms.

It is believed that when measuring the speech quality, the spectral magnitude is more important than its phase. Boll proposed the spectral subtraction method [7] where the signal spectra are estimated by subtracting the noise from a noisy signal spectra. When the noisy signal spectra fall below the noise level, the method produces negative values which need to be suppressed to zero or replaced by a small value. Alternatively, signal subspace methods [8] aim to find a desired signal subspace, which is disjoint with the noise subspace. Thus the components that lie in the complementary noise subspace can be removed. A more general task is source separation. Ideally, if there exists a domain where the subspaces of different signal sources are disjoint, then perfect signal separation can be achieved by projecting the source signal onto its subspace [9]. This method can also be applied to the single channel source separation problem where the target speaker is considered as signal and the competing speaker is considered as noise. Other approaches include algorithms based on audio coding algorithms[10], independent component analysis (ICA) [11] and perceptual models [12].

Performance of speech enhancement is commonly evaluated using some distortion measures. Therefore enhanced signals can be estimated by minimizing its distortion, where the expectation value is utilized, because of the stochastic property of speech signal. Thus statistical-model-based speech enhancement systems [13] have been particularly successful. Statistical approaches require pre-specified parametric models for both the signal and the noise. The model parameters are obtained by maximizing the likelihood of the training samples of the clean signals using expectation maximization (EM) algorithm. Because the true model for speech remains unknown [3], a variety of statistical models have been proposed. Short-time spectral amplitude (STSA) estimator [14] and log-spectral amplitude estimator (LSAE) [15] assume that the spectral coefficients of both signal and noise obey Gaussian distribution. Their difference is that STSA minimizes the mean square error (MMSE) of the spectral amplitude while the LSAE uses the MMSE estimator of the log-spectra. LSAE is more appropriate because log-spectrum is believed more suitable for speech processing. Hidden Markov model (HMM) is also developed for clean speech. The developed HMM with gain adaptation has been applied to the speech enhancement [16] and to the recognition of clean and noisy speech [17]. In contrast to the frequency domain models [14, 15, 16, 17], the density of log-spectral amplitudes is modeled by a Gaussian mixture model (GMM) with parameters trained on the clean signals [18, 19, 20]. Spectrally similar signals are clustered and represented by their mixture components. Though the quality of fitting the signal distribution using the GMM depends on the number of mixture components [1], the density of the speech log-spectral amplitudes can be accurately represented with very small number of mixtures. However, this approach leads to a complex model in the frequency domain and exact signal estimation becomes intractable, therefore approximation methods have been proposed. The MIXMAX algorithm [18] simplifies the mixing process such that the noisy signal takes the maximum of either the signal or the noise, which offers a closed-form signal estimation. Linear approximation [19, 20] expands the logarithm function locally using Taylor expansion. This leads to a linear Gaussian model where the estimation is easy, although finding the point of Taylor expansion needs iterative optimization. The spectral domain algorithms offer high quality speech enhancement while remain low in computational complexity.

In Chapter 3, differ from the frequency domain models [14, 15, 16, 17], we start with a GMM in the log-spectral domain as proposed in [18, 19, 20]. Converting the GMM in the log-spectral domain into the frequency domain directly produces a mixture of log-normal distributions which causes the signal estimation difficult to compute. Approximating the logarithm function [18, 19, 20] is accurate only locally for a limited interval, thus may not be optimal. We propose three methods based on Bayesian estimation. The first is to substitute the log-normal distribution by an optimal Gaussian distribution in the Kullback-Leibler (KL) divergence [2] sense. This way in the frequency domain, we obtain a GMM with a closed-form signal estimation. The second approach uses Laplace method [21], where the spectral amplitude is estimated by computing the maximum *a posteriori* (MAP). The Laplace method approximates the posterior distribution by a Gaussian derived from the second order Taylor expansion of the log likelihood. The third approach is also based on Laplace method, but the log-spectra of signals are estimated using the MAP. The spectral amplitudes are obtained by exponentiating their log-spectra.

The statistical approaches discussed above rely on parameters estimated from the training samples that reflect the statistical properties of the signal. However, the statistics of the test signals may not match those of the training signals perfectly. For example, movement of the speakers and changes of the recording conditions are causes of mismatches. Such difficulty can be overcome by introducing parameters that adapt to the environmental changes. Gain and noise adaptation partially solves this problem [16, 17]. Different from the aspect of audio gain estimation in [14, 22] the gain here means the energy of signals corresponding to the volume of the audio. In [19], noise estimation is proposed, but the gain is fixed to 1. We propose an EM algorithm with efficient gain and noise estimation under the Gaussian approximation.

In Chapter 4, we use Gaussian Scale Mixture model (GSMM) for speech prior. The GSMM enables us estimate both the frequency coefficients and the log-spectra, which are not possible for previous models. The estimated frequency coefficients usually produces better signal quality measured by the signal to noise ratio (SNR), but the estimated log-spectra usually provides lower recognition error rate, because higher SNR may not necessarily give a lower error rate. The propose GSMM estimates both features at the same time. Instead of forcing a deterministic relation between the log-spectra and frequency coefficients, we model them stochastically. We model the log-spectra using a GMM following [18, 19, 20]. The frequency coefficients obey a Gaussian density whose covariances are the exponentials of the log-spectra. In a probabilistic setting, both features can be estimated. An approximate EM algorithm is developed to train the model and two approaches, the Laplace method [21] and the variational approximation [2], are used for signal estimation. The enhanced signals can be constructed from either the estimated frequency coefficients or the estimated log-spectra, depending on the applications.

Acknowledgement: This chapter contains materials in J. Hao, H. Attias, S. Nagarajan, T.-W. Lee and T. Sejnowski, “Speech Enhancement, Gain, and Noise Spectrum Adaptation Using Approximate Bayesian Inference”, *IEEE Transactions on Audio, Speech, and Language Processing*, accepted for publication, and J. Hao, T-W. Lee and T. Sejnowski,

“Speech Enhancement Using Gaussian Scale Mixture Models”, *IEEE Transactions on Audio, Speech, and Language Processing*, submitted for publication.

3

Speech Enhancement, Gain and Noise Spectrum Adaption Using Approximate Bayesian Estimation

The log-spectra of speech are accurately modeled by a Gaussian mixture model. However, signal estimation based on log-spectral domain model is hard. We derive three methods: Gaussian approximation, Laplace method in frequency domain and Laplace method in log-spectral domain. These methods can effectively recover the signal from the noisy recordings. Further, the Gaussian approximation provides an efficient EM algorithm for gain and noise spectrum adaption.

3.1 Prior Speech Model and Signal Estimation

3.1.1 Speech and Noise Models

We consider the clean signal $x[n]$ is contaminated by statistically independent and zero mean noise $n[t]$ in time domain. Under the assumption of additive noise, the observed signal can be described by

$$y[t] = h[t] * x[t] + n[t] = \sum_{m} h_m x[t-m] + n[t]$$

where $h[t]$ is the impulse response of the filter and $*$ denotes convolution. Such signal is often processed in frequency domain by applying FFT

$$Y_k = H_k X_k + N_k \tag{3.1}$$

where k denotes the frequency bin and H_k is the gain. In this chapter we will focus on stationary channel where H_k is time-independent.

Statistical models characterize the signals by its probability density function (PDF). The GMM, provided sufficient number of mixtures, can approximate any given density function to arbitrary accuracy, when the parameters (weights, means, and covariances) are correctly chosen [1, page 214]. The number of parameters for GMM is usually small and can be reliably estimated using the EM algorithm [1]. Here, we assume the log-spectral amplitudes $\{x_0, \cdots, x_{K-1}\}$ obey a GMM,

$$p(x) = \sum_{s} p(x|s)p(s) = \sum_{s} \prod_{k} \mathcal{N}(x_k|\mu_{ks}, B_{ks})p(s) \tag{3.2}$$

where s is the state of the mixture component. For state s, $\mathcal{N}(x_k|\mu_{ks}, B_{ks})$ denotes a Gaussian with mean μ_{ks} and precision B_{ks} defined as the inverse of the covariance,

$$\mathcal{N}(x_k|\mu_{ks}, B_{ks}) = \sqrt{|\frac{B_{ks}}{2\pi}|}e^{-\frac{B_{ks}}{2}(x_k-\mu_{ks})^2} \tag{3.3}$$

Though each frequency bin is statistically independent for state s, they are dependent overall because the marginal density $p(x)$ doesn't factorize.

Use the definition of log-spectrum $|x_k| = \log(|X_k|^2)$, X_k can be written as $X_k = X'_k + iX''_k$, where $X'_k = e^{x_k/2}\cos\theta_k$ and $X''_k = e^{x_k/2}\sin\theta_k$ are its real part and imaginary part, θ_k is its phase. Assume that the phase is uniformly distributed $p(\theta_k) = \frac{1}{2\pi}$ and the PDF for x_k is given in Eq(3.3), we compute the PDF for the FFT coefficients as,

$$\begin{aligned} p(X_k|s) &= p(X'_k, X''_k|s) = |\frac{\partial(X'_k, X''_k)}{\partial(x_k, \theta_k)}|^{-1}p(x_k|s)p(\theta_k) \\ &= \frac{1}{\pi|X_k|^2}\mathcal{N}(\log(|X_k|^2)|\mu_{ks}, B_{ks}) \\ &= \frac{1}{\pi|X_k|^2}\sqrt{\frac{B_{ks}}{2\pi}}e^{-\frac{B_{ks}}{2}(\log(|X_k|^2)-\mu_{ks})^2} \end{aligned} \tag{3.4}$$

where the Jacobian $|\frac{\partial(X'_k,X''_k)}{\partial(x_k,\theta_k)}| = e^{x_k}/2 = |X_k|^2/2$. We call this density log-normal, because the logarithm of a random variable obeys a normal distribution. The frequency domain model is preferred compared to the log-spectral domain, because of simple corruption dynamics in Eq(3.1).

We consider a noise process independent on the signal and assume the FFT coefficients obey a Gaussian distribution with zero mean and precision matrix $\Gamma = \text{diag}(\gamma_1, \cdots, \gamma_K)$

$$\begin{aligned} p(N) &= p(Y|X) = \prod_k \mathcal{N}(Y_k - H_k X_k | 0, \gamma_k) \\ &= \prod_k \frac{\gamma_k}{\pi} e^{-\gamma_k |Y_k - H_k X_k|^2} \end{aligned} \tag{3.5}$$

Note that this Gaussian density is for the complex variables. The precisions γ_k satisfy $\gamma_k = 1/E\{|Y_k - H_k X_k|^2\}$. In contrast, Eq(3.3) is Gaussian density for the log-spectrum x_k which is a real random variable.

The parameters μ_{ks}, B_{ks} and $p(s)$ of speech model given in Eq(3.2) are estimated from the training samples using an EM algorithm. The details for EM algorithm can be found in [1]. The precision matrix $\Gamma = \text{diag}(\gamma_1, \cdots, \gamma_K)$ of the noise model can be estimated from either pure noise or the noisy signals.

3.1.2 Signal Estimation

Under the assumption that the noise is independent on the signal, the full probabilistic model is

$$p(Y, X, s) = p(Y|X)p(X|s)p(s) \tag{3.6}$$

Signal estimation is done as a summation of the posterior distributions of a signal

$$p(X|Y) = \sum_s p(X|Y, s)p(s|Y) \tag{3.7}$$

For example, the MMSE estimator of a signal is given by

$$\hat{X} = \sum_s \int X p(X|Y, s) dX p(s|Y) = \sum_s \hat{X}_s p(s|Y) \tag{3.8}$$

where $\hat{X}_s$ is the signal estimator for state s. This signal estimator makes intuitive sense. Each mixture component enhances the noisy signal separately. Because the hidden state is unknown, the MMSE estimator is consisted of the average of the individual estimators $\hat{X}_s$, weighted by the posterior probability $p(s|Y)$. The block diagram is shown in Figure 3.1.

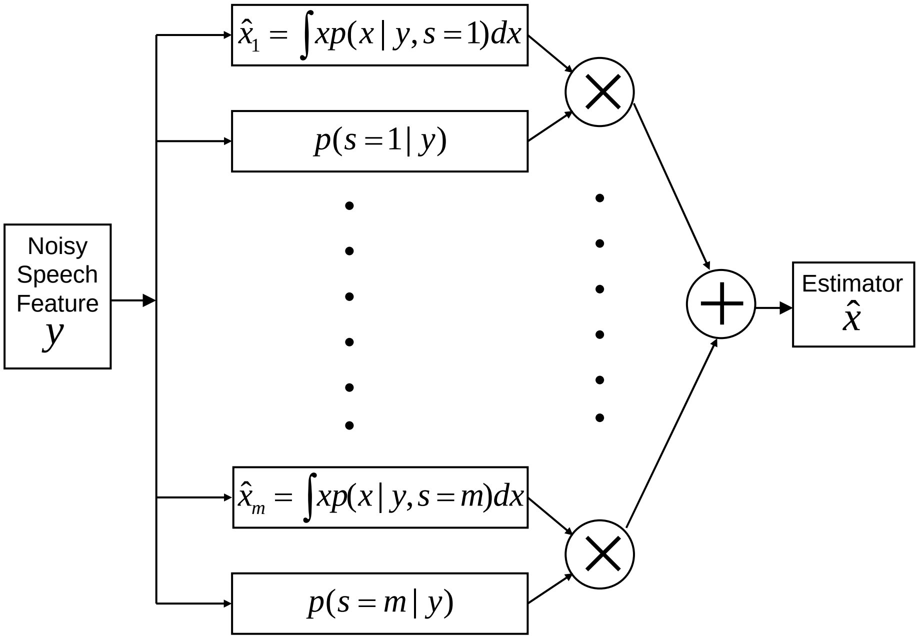
$\hat{x}_1 = \int xp(x \mid y, s=1)dx$
$p(s=1 \mid y)$
Noisy Speech Feature y
$\hat{x}_m = \int xp(x \mid y, s=m)dx$
$p(s=m \mid y)$
Estimator $\hat{x}$

Figure 3.1: Block diagram for speech enhancement based on mixture models. Each mixture component enhances the signal separately. The signal estimator $\hat{x}$ is computed by the summation of individual estimator weighted by its posterior probability $p(s|y)$.

The MMSE estimator suggests a general signal estimation method for the mixture models. First, an estimator based on each mixture state $\hat{X}_s$ is computed. Then the posterior state probability $p(s|Y)$ is calculated to reflect the contribution from state s. Finally, the system output is the summation of the estimators for the states, weighted by the posterior state probability. However, such straightforward scheme can not be carried out directly for the model considered. Neither the individual estimator $\hat{X}_s$ nor the posterior state probability $p(s|Y)$ is easy to compute. The difficulty originates from the log-normal distributions for speech in the frequency domain. We propose approximations to compute both terms. Because we assume a diagonal precision matrix for B_s in the GMM, $\hat{X}_s$ can be estimated separately for each frequency bin k.

3.2 Signal Estimation based on Approximate Bayesian Estimation

Intractability often limits the application of sophisticated models. A great amount of research has been devoted to develop accurate and efficient approximations [2, 21]. Although there are popular methods that have been applied successfully, the effectiveness of such approximations is often model dependent. As indicated in Eq(3.8), two terms, $\hat{X}_s$ and $p(s|Y)$ are required. Three algorithms are derived to estimate both terms. One is based on Gaussian approximation. The other two methods are based on Laplace methods in the time-frequency domain and the log-spectral domain.

3.2.1 Gaussian Approximation (Gaussian)

As shown in Section 3.1.1, the mixture of log-normal distributions for FFT coefficients makes the signal estimation difficult. If we substitute the log-normal distribution $p(X|s)$ in Eq(3.4) by a Gaussian for each state s, the frequency domain model becomes a GMM, which is analytically tractable.

For each state s, we choose the optimal Gaussian that minimizes the KL divergence D_{KL} [23],

$$q = \arg\min_q D_{KL}(p\|q) = \arg\min_q \int p(X) \log \frac{p(X)}{q(X)} dX \tag{3.9}$$

D_{KL} is non-negative and equals to zero if and only if p equals to q almost surely. Note that D_{KL} is asymmetric about its arguments p and q, and $D_{KL}(p\|q)$ is chosen because a closed-form solution for q exists.

It can be shown that the optimal Gaussian q that minimizes the KL-divergence having mean and covariance corresponding to those of the conditional probability in state s, $p(X_k|s)$. The mean of $p(X_k|s)$ is zero due the assumption of a uniform phase distribution. The second order moments are

$$\lambda_{ks} = \int |X_k|^2 p(X_k|s) dX_k = \exp[\mu_{ks} + 1/(2B_{ks})] \tag{3.10}$$

The Gaussian $q(X_k|s) = \mathcal{N}(X_k|0, 1/\lambda_{ks})$ minimizes D_{KL}.

Under the Gaussian approximation, we have converted the GMM in log-spectral domain into a GMM in frequency domain. We denote this converted GMM by $q(X)$

$$q(X) = \sum_s \prod_k q(X_k|s)p(s) = \sum_s \prod_k \mathcal{N}(X_k|0, 1/\lambda_{ks})p(s) \tag{3.11}$$

This approach avoids the complication from the log-normal distribution and offers efficient signal enhancement.

Under the assumption of a Gaussian noise model in Eq(3.5), the posterior distribution over X for state s is computed as

$$p(X_k|Y_k, s) = \frac{p(Y_k|X_k)q(X_k|s)}{p(Y_k|s)} = \mathcal{N}(X_k|\hat{X}_{ks}, \phi_{ks}) \tag{3.12}$$

It is a Gaussian with precision ϕ_{ks} and mean $\hat{X}_{ks}$ given by

$$\begin{aligned} \phi_{ks} &= \lambda_{ks}^{-1} + \gamma_k && (3.13) \\ \hat{X}_{ks} &= \frac{\gamma_k}{\phi_{ks}} Y_k && (3.14) \end{aligned}$$

where λ_{ks} is the covariance of the speech prior and γ_k is the precision of noise PDF. Note that we have used the approximated speech prior $q(X_k|s)$ in Eq(3.12). The individual signal estimator for each state s is given by Eq(3.14).

The posterior state probability $p(s|Y)$ is computed,

$$p(s|Y) = \frac{p(Y|s)p(s)}{p(Y)} \tag{3.15}$$

using the Bayes' rule. Under the speech prior $q(X|s)$ in Eq(3.11), $p(Y|s)$ is computed as,

$$p(Y|s) = \prod_k \int p(Y_k|X_k)q(X_k|s)dX_k = \prod_k \mathcal{N}(Y_k|0, \psi_{ks}) \tag{3.16}$$

where the precision ψ_{ks} is given by

$$\psi_{ks} = \frac{1}{\lambda_{ks} + 1/\gamma_k} \tag{3.17}$$

Using Eq(3.8) and substituting $\hat{X}_{ks}$ in Eq(3.14), $p(s|Y)$ in Eq(3.15), the signal estimation function can be written as

$$\hat{X}_k = \sum_s \hat{X}_{ks} p(s|Y) = \left(\sum_s \frac{\gamma_k}{\phi_{ks}} p(s|Y) \right) Y_k \tag{3.18}$$

Each individual estimator has resembled the power response of a Wiener filter and is a linear function of Y . Note that the state probability depends on Y, therefore the signal estimator in Eq(3.18) is a nonlinear function of Y. This is analogous to a time varying Wiener filter where the signal and noise power is known or can be estimated from a short period of the signal such as using a decision directed estimation approach [14, 22]. Here, the temporal variation is integrated through the changes of the posterior state probability $p(s|Y)$ over time.

3.2.2 Laplace Method in Frequency Domain (LaplaceFFT)

Laplace method approximates a complicated distribution using a Gaussian around its MAP. This method suggests the MAP estimator for the original distribution which is equivalent to the more popular MMSE estimator of the resulted Gaussian. Computing the MAP can be considered as an optimization problem and many optimization tools can be applied. We use the Newton's method to find the MAP. The Laplace method is also applied to compute the posterior state probability which requires an integration over a hidden variable X. It expands the logarithm of the integrand around its mode using Taylor series expansion, and transforms the process into a Gaussian integration which has a closed-form solution. However, such method for computing the posterior state probability is not accurate for our problem and we use an alternative approach. The final signal estimator is constructed using Eq(3.8).

We derive the MAP estimator $\hat{X}_{ks}$ for each state s. The logarithm of signal the posterior probability, conditioned on state s, is given by

$$\begin{aligned}\log p(X_k|Y_k, s) &= \log p(Y_k|X_k, s) + \log p(X_k|s) + c \\ &= -\gamma_k|Y_k - X_k|^2 + \log \frac{1}{\pi|X_k|^2} \\ &\quad -\frac{B_{ks}}{2}(\log|X_k|^2 - \mu_{ks})^2 + c \qquad (3.19)\end{aligned}$$

where c is a constant independent on X_k. It is more convenient to represent X_k using its magnitude r_k and phase θ_k, $X_k = r_k e^{i\theta_k}$. And we compute the MAP estimator for the magnitude r_k and phase θ_k for each state s,

$$\begin{aligned}(\hat{r}_{ks}, \hat{\theta}_{ks}) &= \arg\max_{r_k,\theta_k}\{\log p(r_k, \theta_k|Y_k, s)\}, \\ &= \arg\max_{r_k,\theta_k}\{\log r_k p(X_k|Y_k, s)\}. \qquad (3.20)\end{aligned}$$

Use Eq(3.19) and neglect the constant c, maximizing Eq(3.20) is equivalent to minimizing the function h_1 defined by

$$h_1(r_k, \theta_k) = \gamma_k|Y_k - r_k e^{i\theta_k}|^2 + \frac{B_{ks}}{2}(\log(r_k^2) - \beta_{ks})^2 \qquad (3.21)$$

where $\beta_{ks} = \mu_{ks} - 1/(2B_{ks})$. It is obvious from the above equation that the MAP estimator for θ_k is $\hat{\theta}_k = \angle Y_k$, which is independent on state s. And the magnitude estimator $\hat{r}_{ks}$ minimizes

$$h_1(r_k) = \gamma_k|r_{yk} - r_k|^2 + \frac{B_{ks}}{2}(\log(r_k^2) - \beta_{ks})^2 \qquad (3.22)$$

where $r_{yk} = |Y_k|$. The minimization over r_k doesn't have an analytical solution, but it can be solved with the Newton's method. For this we need the first order and second order derivatives of $h_1(r_k)$ with respect to r_k:

$$h_1'(r_k) = 2\gamma_k(r_k - r_{yk}) + B_{ks}(\log(r_k^2) - \beta_{ks})\frac{2}{r_k} \tag{3.23}$$

$$h_1''(r_k) = 2\gamma_k + B_{ks}\frac{4}{r_k^2} - B_{ks}(\log(r_k^2) - \beta_{ks})\frac{2}{r_k^2} \tag{3.24}$$

Then the Newton's method iterates

$$\hat{r}_{ks} \leftarrow \hat{r}_{ks} - \eta\frac{h_1'(\hat{r}_{ks})}{|h_1''(\hat{r}_{ks})|}. \tag{3.25}$$

The absolute value of h_1'' indicates the search of the minima of h_1. The $\eta = 1$ denotes the learning rate.

The Newton's method is sensitive to the initialization and may give local minima. The two squared terms in Eq(3.22) indicate that the optimal estimator $\hat{r}_{ks}$ is bounded between $e^{\beta_{ks}/2}$ and r_{yk}. We use both values to initialize $\hat{r}_{ks}$ and select the one that produces a smaller $h_1(r_k)$. Empirically, we observe that this scheme always finds a global minimum. The first term in Eq(3.22) is quadratic, thus Newton's method converges to the optimal solution faster, less than 5 iterations for our case, than other methods such as gradient decent.

Computing the posterior state probability $p(s|Y)$ requires the knowledge of $p(Y_k|s)$. Marginalization over X_k gives

$$p(Y_k|s) = \int p(Y_k|X_k)p(X_k|s)dX_k \tag{3.26}$$

However, because of the log-normal distribution $p(X_k|s)$ provided in Eq(3.4), the integration can not be solved with a closed-form answer. Either numerical methods or approximations are needed. Numerical integration is computationally expensive, leaving approximation more efficient. We propose the following two approaches based on Laplace method and Gaussian approximation.

Evaluate $p(s|Y)$ using Laplace Method

Laplace method is widely used to approximate integrals with continuous variables in statistical models to facilitate probabilistic inference [21] such as computing the high order statistics. It expands the logarithm of the integrand up to its second order, leading to a Gaussian integral which has a closed-form solution. We rewrite Eq(3.26) as

$$p(Y_k|s) = \int \frac{\gamma_k}{\pi} \sqrt{\frac{B_{ks}}{2\pi}} \exp(-f(X_k) - \beta_{ks}) dX_k \tag{3.27}$$

where we define

$$f(X_k) = \gamma_k |Y_k - X_k|^2 + \frac{B_{ks}}{2} (\log(|X_k|^2) - \alpha_{ks})^2 \tag{3.28}$$

and $\alpha_{ks} = \mu_{ks} - 1/B_{ks}$, $\beta_{ks} = \mu_{ks} - 1/(2B_{ks})$. Laplace method expands the logarithm of the integrand $f(X_k)$ around its minimum $\hat{X}_{ks}$ up to the second order and carries out a Gaussian integration,

$$\int e^{-f(X_k)} dX_k \approx e^{-f(\hat{X}_{ks})} \sqrt{|\frac{2\pi}{J}|} \tag{3.29}$$

where J is the Hessian of $f(X_k)$ evaluated at $\hat{X}_{ks}$. Denote $\hat{X}_{ks} = \hat{X}'_{ks} + i\hat{X}''_{ks}$ by its real part $\hat{X}'_{ks}$ and imaginary part $\hat{X}''_{ks}$, its magnitude by $\hat{r}_{ks} = |\hat{X}_{ks}|$. J is computed as

$$\begin{aligned} J &= \begin{pmatrix} \frac{\partial^2 f}{\partial X' \partial X'} & \frac{\partial^2 f}{\partial X' \partial X''} \\ \\ \frac{\partial^2 f}{\partial X' \partial X''} & \frac{\partial^2 f}{\partial X'' \partial X''} \end{pmatrix} && (3.30) \\ &= \begin{pmatrix} a_k + \frac{4\hat{X}_k'^2}{\hat{r}_{ks}^2} b_k & \frac{4\hat{X}'_k \hat{X}''_k}{\hat{r}_{ks}^2} b_k \\ \frac{4\hat{X}'_k \hat{X}''_k}{\hat{r}_{ks}^2} b_k & a_k + \frac{4\hat{X}_k''^2}{\hat{r}_{ks}^2} b_k \end{pmatrix} && (3.31) \end{aligned}$$

The a_k and b_k here are defined as

$$\begin{aligned} a_k &= 2\gamma_k + B_{ks}(\log(\hat{r}_{ks}^2) - \alpha_{ks})\frac{2}{\hat{r}_{ks}^2} && (3.32) \\ b_k &= \frac{B_{ks} - B_{ks}(\log(\hat{r}_{ks}^2) - \alpha_{ks})}{\hat{r}_{ks}^2} && (3.33) \end{aligned}$$

The determinant of Hessian J is

$$\det(J) = a_k^2 + 4a_k b_k \tag{3.34}$$

Thus the marginal probability is

$$p(Y_k|s) \propto \sqrt{|B_{ks}|} e^{-\beta_{ks}} e^{-f(\hat{X}_{ks})} \sqrt{|\frac{1}{\det(J)}|} \tag{3.35}$$

This gives $p(s|Y)$

$$p(s|Y) = \frac{p(Y|s)p(s)}{p(Y)} \propto \prod_k p(Y_k|s)p(s) \tag{3.36}$$

Laplace method in essence approximates the posterior $p(X_k|Y_k, s)$ using a Gaussian density. This is very effective in Bayesian networks where the training set includes a large number of samples. The posterior distribution of the (hyper-) parameters has a peaky shape that closely resembles a Gaussian. The Laplace method has an error that scales as $O(T^{-1})$, where T is the number of samples [21]. However, the estimation here is based on a single sample Y. Further, the normalization factor of $p(Y_k|s)$ in Eq(3.35) depends on the state s but it is ignored. Thus this approach does not yield good experimental results and we derive another method.

Evaluate $p(s|Y)$ using Gaussian Approximation

As discussed in section 3.2.1, the log-normal distribution $p(X_k|s)$ has a Gaussian approximation $q(X_k|s) = \mathcal{N}(X_k|0, 1/\lambda_{ks})$ given in Eq(3.11). Thus we can compute the marginal distribution $p(Y_k|s)$ for state s as

$$\begin{aligned} p(Y_k|s) &= \int p(Y_k|X_k)p(X_k|s)dX_k \\ &\approx \int p(Y_k|X_k)q(X_k|s)dX_k \\ &= \mathcal{N}(0, \psi_{ks}) \end{aligned} \tag{3.37}$$

where the precision ψ_{ks} is given in Eq(3.17). The posterior state probability $p(s|Y)$ is obtained using the Bayes' rule. It is

$$p(s|Y) = \frac{\prod_k p(Y_k|s)p(s)}{p(Y)} \tag{3.38}$$

This approach uses the same procedure shown in section 3.2.1.

The signal estimator is the summation of the MAP estimator $\hat{r}_{ks}e^{i\angle Y_k}$ for each state s weighted by the posterior state probability $p(s|Y)$ in Eq(3.38),

$$X_k = \sum_s \hat{r}_{ks} e^{i\angle Y_k} p(s|Y) \tag{3.39}$$

The MAP estimator for phase, $\angle Y_k$, is utilized.

3.2.3 Laplace Method in Log-Spectral Domain (LaplaceLS)

It is suggested that the human auditory system perceives signal on the logarithmic scale, therefore log-spectral analysis such as LSAE [15] is more suitable for speech processing. Thus we can expect better performance if the log-spectra can be directly estimated. The idea is to find the log-amplitude $\hat{v}_k = \log(|X_k|^2)$ that maximizes the log posterior probability $\log(p(X|Y,s))$ given in Eq(3.19). Note that $\hat{v}_k$ is not the MAP of $p(\log(|X|^2)|Y,s)$. A similar case is LSAE [15] where the expectation of the log-spectral error is taken over $p(X)$ rather than $p(\log|X|)$. Optimization over v_k also has the advantage of avoiding negative amplitude due to local minima.

Substitute $v_k = \log(|X_k|^2)$ into Eq(3.19), we compute the MAP estimator for the phase and log-amplitude v_k. Note that the optimal phase is that of the noisy signal, $\hat{\theta}_k = \angle Y_k$. The MAP estimator for the log-amplitude maximizes Eq(3.19), which is equivalent to minimizing

$$h_2(v_k) = \gamma_k (r_{yk} - e^{v_k/2})^2 + v_k + \frac{B_{ks}}{2}(v_k - \mu_{ks})^2 \tag{3.40}$$

where $r_{yk} = |Y_k|$. And h_2 can be minimized using Newton's method. The first and second order derivatives are given by

$$h_2'(v_k) = -\gamma_k (r_{yk} - e^{v_k/2}) e^{v_k/2} + 1 + B_{ks}(v_k - \mu_{ks}) \tag{3.41}$$

$$h_2''(v_k) = -\frac{1}{2}\gamma_k(r_{yk} - e^{v_k/2})e^{v_k/2} + \frac{1}{2}\gamma_k e^{v_k} + B_{ks} \tag{3.42}$$

The Newton's method updates the log-amplitude v_{ks} as

$$\hat{v}_{ks} \leftarrow \hat{v}_{ks} - \eta \frac{h_2'(\hat{v}_{ks})}{|h_2''(\hat{v}_{ks})| + \tau} \tag{3.43}$$

where η is the learning rate and τ is the regularization to avoid divergence when h_2'' is close to zero. This avoids the numerical instability caused by the exponential term in Eq(3.40).

In the experiment, we use the noisy signal log-spectra for initialization, $\hat{v}_{ks} = \log(|Y_k|^2)$. We set $\eta = 0.5$, $\tau = 3$ and run 10 Newton's iterations.

We use the same strategy as described in section 3.2.2 to compute $p(s|Y)$ using Eq(3.38). The signal estimator follows

$$\begin{aligned} \bar{v}_k &= \sum_s \hat{v}_{ks} p(s|Y) && (3.44) \\ X_k &= \exp(\bar{v}_k/2) e^{i\angle Y_k} && (3.45) \end{aligned}$$

The MAP estimator of phase from the noisy signal is used.

In contrast to Eq(3.39) where the amplitude estimators are averaged, Eq(3.44) provides the log-amplitude estimator. The magnitude is obtained by taking the exponential. The exponential function is convex, thus Eq(3.44) provides a smaller magnitude estimation than Eq(3.39) when $e^{\hat{v}_{ks}/2} = \hat{r}_{ks}$. Furthermore, this log-spectral estimator fits a speech recognizer, which extracts the Mel Frequency Cepstral Coefficients (MFCC).

3.3 Learning Gain and Noise with Gaussian Approximation

One drawback of the system comes from the assumption that the statistical properties of the training set match those of the testing set, which means a lack of adaptability. However, the energy of the test signals may not be reliably estimated from a training set because of uncontrolled factors such as variations of the speech loudness or the distance between the speaker and microphone. This mismatch results in poor enhancement because the pre-trained model may not capture the statistics of samples under the testing conditions. One strategy to compensate for these variations is to estimate the gain H instead of a fixed value of 1 used in the previous sections. Two conditions will be considered: frequency independent gain, which is a scalar gain and frequency dependent gain. Gain-adaptation needs to carry out efficiently. For the signal prior given in Eq(3.2), it is difficult to estimate the gain because of the involvement of log-normal distributions. See section 3.1.1. But under Gaussian approximation, the gain can be estimated using the EM algorithm.

Recall that the acoustic model is $Y_k = H_k X_k + N_k$ as given in Eq(3.1). If $p(X_k)$ has the form of GMM and $p(N_k)$ is Gaussian, the model becomes exactly a mixture of factor analysis (MFA) model. The gain H can be estimated in the same way as estimating a loading matrix for MFA. For this purpose, we take the approach in section 3.2.1 and approximate the log-normal PDF $p(X_k|s)$ by a normal distribution $q(X_k|s) = \mathcal{N}(X_k|0, 1/\lambda_{ks})$, where the signal covariance λ_{ks} is given in Eq(3.10). In addition, we assume additive Gaussian noise as provided in Eq(3.5). Treating X_k as a hidden variable, we derive an EM algorithm, which contains an expectation step (E-step) and a maximization step (M-step), to estimate the gain H_k and the noise spectrum $\Gamma = \text{diag}(\gamma_1, \cdots, \gamma_K)$.

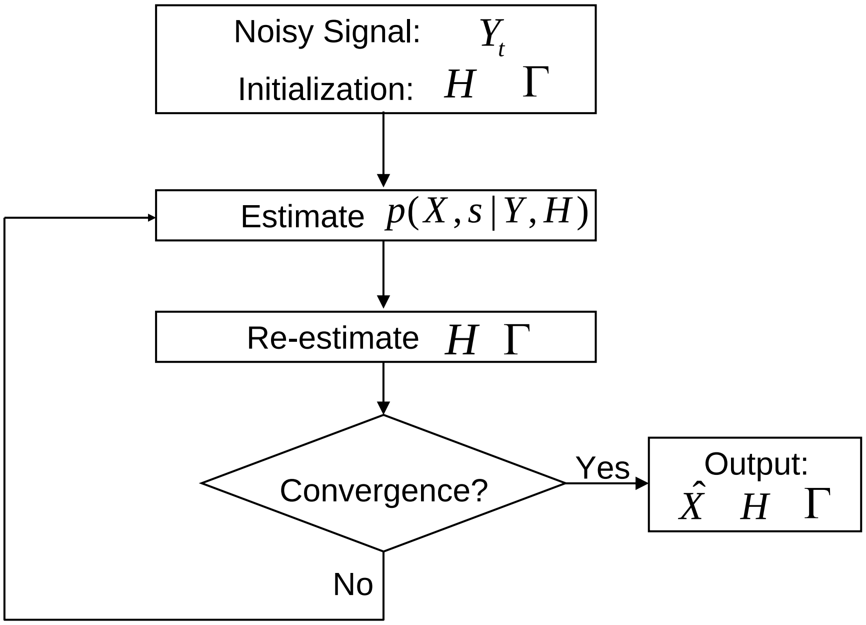

Figure 3.2: Block diagram of EM algorithm for the gain and noise spectrum estimation. The E-step, computing $p(X, s|Y, H)$, and M-step, updating H and Γ, iterate until convergence.

3.3.1 EM Algorithm for Gain and Noise Spectrum Estimation

The data log-likelihood denoted by $\mathcal{L}$ is

$$\begin{aligned}\mathcal{L} &= \sum_t \log p(Y_t) = \sum_t \log\left(\sum_{s_t}\int p(Y_t, X_t, s_t)dX_t\right) \\ &\geq \sum_{ts_t}\int \tilde{q}(X_t, s_t)[\log p(Y_t, X_t, s_t) - \log \tilde{q}(X_t, s_t)]dX_t\end{aligned}$$

where t is the frame index. The above inequality is true for all choices of the distribution $\tilde{q}(X_t, s_t)$. When $\tilde{q}(X_t, s_t)$ equals the posterior probability $p(X_t, s_t|Y_t)$, the inequality becomes an equality. The EM algorithm is a typical technique to maximize the likelihood. It iterates between updating the auxiliary distribution $\tilde{q}(X_t, s_t)$ (E-step) and optimizing the model parameters $\{H, \Gamma\}$ (M-step), until some convergence criterion is satisfied.

The E-step computes the posterior distribution over X_t, $\tilde{q}(X_t|s_t) = p(X_t|Y_t, s_t) = \prod_k p(X_{kt}|Y_{kt}, s_t)$ with gain H fixed. And $p(X_{kt}|Y_{kt}, s_t)$ is computed as

$$p(X_{kt}|Y_{kt}, s_t) = \frac{p(Y_{kt}|X_{kt})q(X_{kt}|s_t)}{p(Y_{kt}|s_t)} \tag{3.46}$$

Note we use the approximated signal prior $q(X_{kt}|s_t)$ given in Eq(3.11). Thus, the computation is a standard Bayesian inference in a Gaussian system, and one can show that $p(X_{kt}|Y_{kt}, s_t) = \mathcal{N}(X_{kt}|\tilde{X}_{kst}, \Sigma_{ks})$, whose mean $\tilde{X}_{kst}$ and precision Σ_{ks} are given by

$$\Sigma_{ks} = H_k^2 \gamma_k + 1/\lambda_{ks} \tag{3.47}$$

$$\tilde{X}_{kst} = \frac{\gamma_k H_k^* Y_{kt}}{\Sigma_{ks}} \tag{3.48}$$

Here H^* denotes the complex conjugate of H. We point out that the precisions are time-independent while the means are time dependent.

The posterior state probability $\tilde{q}(s_t) = p(s_t|Y_t)$ is computed as

$$\begin{aligned} \tilde{q}(s_t) &= p(s_t|Y_t) = \frac{p(Y_t|s_t)p(s_t)}{p(Y_t)}, \\ &\propto \prod_k \mathcal{N}(Y_{kt}|0, \frac{1}{H_k^2 \lambda_{ks} + 1/\gamma_k})p(s_t) \end{aligned} \tag{3.49}$$

The M-step updates the gain H and noise spectrum $\Gamma = \text{diag}(\gamma_1, \cdots, \gamma_K)$ with $\tilde{q}$ fixed. Now we consider two conditions: frequency dependent gain and frequency independent gain.

Frequency Independent Gain: H is scalar, its update rule is

$$H = \frac{\sum_{t,s_t,k} \tilde{q}(s_t)\gamma_k Y_{kt} \tilde{X}^*_{ks_t t}}{\sum_{t,s_t,k} \tilde{q}(s_t)\gamma_k (\tilde{X}_{ks_t t}\tilde{X}^*_{ks_t t} + \Sigma^{-1}_{ks})} \tag{3.50}$$

Frequency Dependent Gain: $H = \{H_1, \cdots, H_K\}$ is a vector. The update rule is, for $k = \{1, \cdots, K\}$,

$$H_k = \frac{\sum_{t,s_t} \tilde{q}(s_t) Y_{kt} \tilde{X}^*_{ks_t t}}{\sum_{t,s_t} \tilde{q}(s_t)(\tilde{X}_{ks_t t}\tilde{X}^*_{ks_t t} + \Sigma^{-1}_{ks})} \tag{3.51}$$

The update rule for the precision of noise γ_k is

$$1/\gamma_k = \frac{1}{T}\sum_{t,s_t} \int \tilde{q}(X_{kt}, s_t|Y_t)|Y_{kt} - H_k X_{kt}|^2 dX_{kt}. \tag{3.52}$$

The goal of the EM algorithm is to provide an estimation for the gain and the noise spectrum. Note that it is not necessary to compute the intermediate results $\tilde{X}_{ks_t t}$ in every iteration. Thus substantial computation can be saved if we substitute Eq(3.48) into the learning rules. This significantly improves the computational efficiency and saves memory. After some mathematical manipulation, the EM algorithm for the frequency dependent gain is as follows:

1. Initialize H_k and γ_k;
2. Compute $\tilde{q}(s_t)$ using Eq(3.49);
3. Update the precisions Σ_{ks} using Eq(3.47);

4. Update the gain

$$H_k \leftarrow \frac{\sum_{ts_t} \tilde{q}(s_t)\Sigma_{ks}^{-1} H_k \gamma_k |Y_{kt}|^2}{\sum_{ts_t} \tilde{q}(s_t)((\Sigma_{ks}^{-1}\gamma_k)^2 H_k^2 |Y_{kt}|^2 + \Sigma_{ks}^{-1})} \tag{3.53}$$

5. Update the noise precision

$$\frac{1}{\gamma_k} \leftarrow \frac{1}{T}\sum_{ts_t} \tilde{q}(s_t)((1 - \Sigma_{ks}^{-1}\gamma_k H_k^2)|Y_{kt}|^2 + \Sigma_{ks}^{-1} H_k^2) \tag{3.54}$$

6. Iterate step 2), 3), 4), 5) until convergence.

For frequency independent gain, the gain is updated as follows

$$H \leftarrow \frac{\sum_{ts_tk} \tilde{q}(s_t)\Sigma_{ks}^{-1} H_k \gamma_k^2 |Y_{kt}|^2}{\sum_{ts_tk} \tilde{q}(s_t)\gamma_k((\Sigma_{ks}^{-1}\gamma_k)^2 H_k^2 |Y_{kt}|^2 + \Sigma_{ks}^{-1})} \tag{3.55}$$

The block diagram is shown in Figure 3.2. In the above EM algorithm, Σ_{ks} is time independent, thus it is computed only once for all the frames. And $|Y_{kt}|^2$ is computed in advance.

In our experiment, because the test files are 1-2 seconds long segments, the parameters can not be reliably learned using a single segment. Thus we concatenate 4 segments as a testing file. The gain is initialized to be 1. The noise covariance is initialized to be 30% of the signal covariance for all signal to noise ratio (SNR) conditions, which does not include any prior SNR knowledge. Because the EM algorithm for estimating the gain and noise is efficient, we set strict convergence criteria: a minimum of 100 EM iterations, the change of likelihood less than 1 and the change of gain less than 10^{-4} per iteration.

3.3.2 Identifiability of Model Parameters

The MFA is not identifiable because it is invariant under the proper re-scaling of the parameters. However, in our case, the parameters H and Γ are identifiable, because the model for speech, a GMM trained by clean speech signals, remains fixed during the learning of parameters. The fixed speech prior removes the scaling uncertainty of the gain H. Second, the speech model is a GMM while the noise is modeled by a single Gaussian. The structure of speech, captured by the GMM through its higher order statistics, doesn't resemble a single Gaussian. This makes the noise spectrum Γ identifiable. As shown in our experiments, the gain H and noise spectrum Γ are reliably estimated using the EM algorithm.

Acknowledgement: This chapter contains materials in J. Hao, H. Attias, S. Nagarajan, T.-W. Lee and T. Sejnowski, "Speech Enhancement, Gain, and Noise Spectrum Adaptation Using Approximate Bayesian Inference", *IEEE Transactions on Audio, Speech, and Language Processing*, accepted for publication.

4

Speech Enhancement using Gaussian Scale Mixture Models

The GMM in log-spectral domain is a accurate model for speech log-spectra. However, when we convert the GMM into the frequency domain, it results in mixture of log-normals for FFT coefficients. The mixture of log-normal density is inappropriate for speech because it has low density around 0, in contrast the FFT coefficients of speech has a sharp peak around 0. The reason is that, the argument of the logarithm function can not be zero, and magnitudes of the FFT coefficients can not be zero. In this chapter, we propose an stochastic relationship between the log-spectra and FFT coefficients. The speech model is a Gaussian scale mixture model (GSMM), which matches the statistical properties of speech signal in frequency domain. We develop and EM algorithm to train the GSMM, and two methods for signal estimation: Laplace method and variational approximation.

4.1 Gaussian Scale Mixture Model

4.1.1 Acoustic Model

Assuming additive noise, the time domain acoustic model is

$$y[t] = h[t] * x[t] + n[t] = \sum_m h_m x[t-m] + n[t] \tag{4.1}$$

where h is the impulse response of the filter and $*$ denotes the convolution. Applying a fast Fourier transform (FFT), this becomes

$$Y_k = H_k X_k + N_k, \tag{4.2}$$

where k is the frequency bin and H_k is the gain. We assume $H_k = 1$ in this chapter.

The noise is modeled by a Gaussian

$$p(Y_k|X_k) = \mathcal{N}(Y_k|X_k, \gamma_k) = \frac{\gamma_k}{\pi} e^{-\gamma_k |Y_k - X_k|^2} \tag{4.3}$$

with zero mean and precision $1/\gamma_k = E\{|Y_k - X_k|^2\}$. Note this Gaussian is of a complex variable, because the FFT coefficients are complex.

4.1.2 Improperness of the Log-Normal Distribution for X_k

If the log-spectra $x_k = \log(|X_k|^2)$ are modeled by a GMM, for each mixture s,

$$p(x_k|s) = \sqrt{\frac{\nu_{ks}}{2\pi}} e^{-\frac{\nu_{ks}}{2}(x_k - \mu_{ks})^2} \tag{4.4}$$

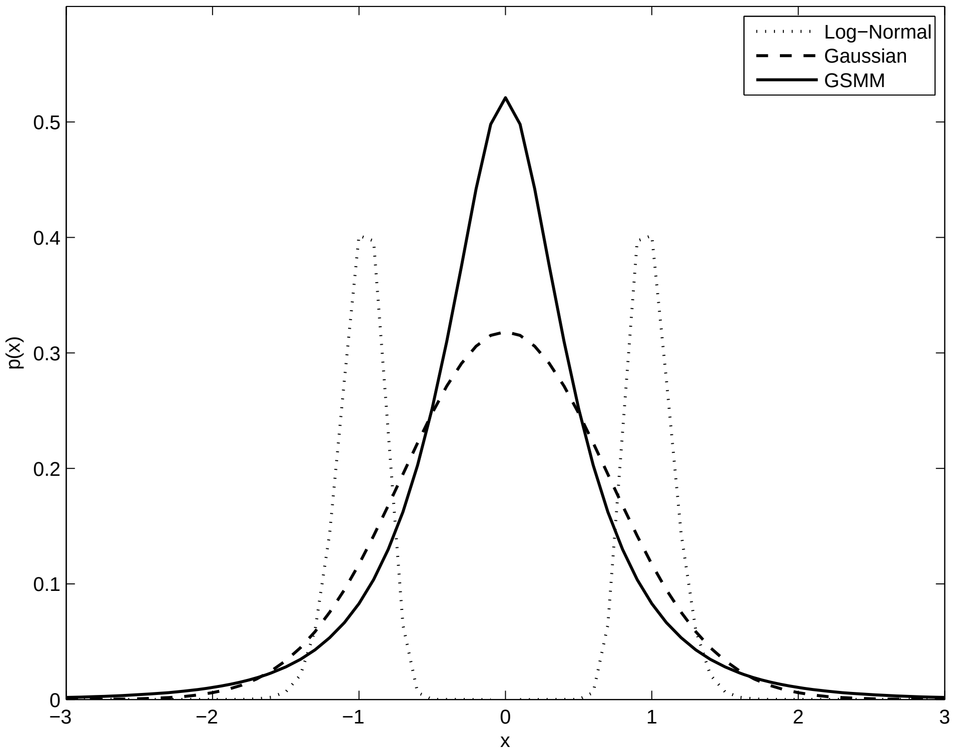

Figure 4.1: Distributions for the real part of X_k, with its imaginary part fixed at 0. The log-normal (dotted) has two modes. The GSMM (solid) is more peaky than Gaussian (dashed).

is a Gaussian with mean μ_{ks} and precision ν_{ks}. Express $X_k = X'_k + iX''_k$ by its real and imaginary parts. Then $X'_k = e^{x_k/2}\cos\theta_k$ and $X''_k = e^{x_k/2}\sin\theta_k$, where θ_k is the phase. If the phase is uniformly distributed, $p(\theta_k) = \frac{1}{2\pi}$, the PDF for X_k is $p(X_k|s) = p(X'_k, X''_k|s) = \frac{1}{J_k}p(x_k|s)p(\theta_k)$, where J_k is the Jacobian $J_k = \frac{\partial(X'_k, X''_k)}{\partial(x_k, \theta_k)} = |X_k|^2/2$. We have

$$p(X_k|s) = \frac{1}{\pi|X_k|^2}\sqrt{\frac{\nu_{ks}}{2\pi}}e^{-\frac{\nu_{ks}}{2}(\log(|X_k|^2)-\mu_{ks})^2} \tag{4.5}$$

as plotted in Figure 4.1. This is a log-normal PDF because $\log(|X_k|^2)$ is normally distributed. Note that it has a saddle shape around zero. In contrast to this log-normal density, the FFT coefficients X_k of speech is super-Gaussian and has a peak at zero, which can not be modeled by the log-normal density.

4.1.3 Gaussian Scale Mixture Model for Speech Prior

Instead of assuming $x_k = \log(|X_k|^2)$, we model this relation stochastically. To avoid confusion, we denote the random variable for the log-spectra as ξ_k. The conditional probability is

$$p(X_k|\xi_k) = \frac{e^{-\xi_k}}{\pi} e^{-e^{-\xi_k}|X_k|^2} \tag{4.6}$$

This is a Gaussian PDF with mean zero and precision $e^{-\xi_k}$. Note that ξ_k controls the scaling of X_k. Consider $\log p(X_k|\xi_k) = -\xi_k - e^{-\xi_k}|X|^2 - \log \pi$, and its maximum given by

$$\hat{\xi}_k = \arg\max_{\xi_k} p(X_k|\xi_k) = \log |X_k|^2 \tag{4.7}$$

Thus we term ξ_k the log-spectrum.

It has been proposed in [19, 20, 18] to model the log-spectra with a GMM

$$\begin{aligned} p(\xi_k|s) &= \mathcal{N}(\xi_k|\mu_{ks}, \nu_{ks}) = \sqrt{\frac{\nu_{ks}}{2\pi}} e^{-\frac{\nu_{ks}}{2}(\xi_k - \mu_{ks})^2} && (4.8) \\ p(\xi_1, \cdots, \xi_K) &= \sum_s p(s) \prod_k p(\xi_k|s) && (4.9) \end{aligned}$$

where s is the state indexing the mixtures. Though the precision is diagonal for each s, $p(\xi_1, \cdots, \xi_K)$ doesn't factorize over k, i.e. the frequency bins are dependent. The PDF for X_k is

$$p(X_1, \cdots, X_k) = \sum_s p(s) \prod_k \int d\xi_k p(X_k|\xi_k) p(\xi_k|s) \tag{4.10}$$

This is called Gaussian scale mixture model (GSMM) because ξ_k controls the scaling of X_k and the scaling parameter ξ_k obeys a GMM given in Eq(4.9). Note that $\{X_1, \cdots, X_K\}$ are statistically dependent because of the dependency among $\{\xi_1, \cdots, \xi_K\}$.

The GSMM has a peak at zero, in contrast to the log-normal PDF which has low probability near zero. To prove it, observe $p(X_k = 0|\xi_k) > p(X_k = \bar{X}_k|\xi_k)$, for any $\bar{X}_k \neq 0$, then

$$\begin{aligned} p(X_k = 0) &= \sum_s \int d\xi_k p(X_k = 0|\xi_k) p(\xi_k|s) p(s) \\ &> \sum_s \int d\xi_k p(X_k = \bar{X}_k|\xi_k) p(\xi_k|s) p(s) \\ &= p(X_k = \bar{X}_k) \end{aligned} \tag{4.11}$$

The GSMM is super-Gaussian, defined to have positive kurtosis, for each mixture. To show this property, we express $X_k = X_k' + iX_k''$ by the real and imaginary parts. The Kurtosis of X_k' conditioned on state s is defined by

$$Kurt(X_k'|s) = E\{(X_k')^4|s\} - 3[E\{(X_k')^2|s\}]^2 \tag{4.12}$$

where E stands for the expectation. We need

$$\begin{aligned} E\{(X_k')^4|s\} &= \int (X_k')^4 \frac{e^{-\xi_k}}{\pi} e^{-e^{-\xi_k}((X_k')^2 + (X_k'')^2)} \\ &\quad \times \sqrt{\frac{\nu_{ks}}{2\pi}} e^{-\frac{\nu_{ks}}{2}(\xi_k - \mu_{ks})^2} dX_k' dX_k'' d\xi_k \\ &= \int \frac{3}{4} e^{2\xi_k} \sqrt{\frac{\nu_{ks}}{2\pi}} e^{-\frac{\nu_{ks}}{2}(\xi_k - \mu_{ks})^2} d\xi_k \\ &= \frac{3}{4} e^{2\mu_{ks} + 2/\nu_{ks}} \end{aligned} \tag{4.13}$$

$$\begin{aligned}
E\{(X'_k)^2|s\} &= \int (X'_k)^2 \frac{e^{-\xi_k}}{\pi} e^{-e^{-\xi_k}((X'_k)^2+(X''_k)^2)} \\
&\quad \times \sqrt{\frac{\nu_{ks}}{2\pi}} e^{-\frac{\nu_{ks}}{2}(\xi_k-\mu_{ks})^2} dX'_k dX''_k d\xi_k \\
&= \int \frac{1}{2} e^{\xi_k} \sqrt{\frac{\nu_{ks}}{2\pi}} e^{-\frac{\nu_{ks}}{2}(\xi_k-\mu_{ks})^2} d\xi_k \\
&= \frac{1}{2} e^{\mu_{ks}+1/(2\nu_{ks})} \qquad (4.14)
\end{aligned}$$

Because the precision $\nu_{ks} > 0$,

$$Kurt(X'_k|s) = \frac{3}{4} e^{2\mu_{ks}+1/\nu_{ks}} (e^{1/\nu_{ks}} - 1) > 0 \qquad (4.15)$$

The real and imaginary parts are symmetric due to uniform phase, $Kurt(X''_k|s) = Kurt(X'_k|s)$. A Gaussian PDF has zero kurtosis. A super-Gaussian PDF is more peaky and has heavier tails than Gaussian, as shown in Figure 4.1. The GSMM, which is unimodal and super Gaussian, is a proper model for the FFT coefficients of speech [24]. The mixture model offers extra flexibility.

4.2 An EM Algorithm for Training the GSMM

The parameters of the GSMM are $\theta = \{\mu_{ks}, \nu_{ks}, p(s)\}$ which can be estimated from the training samples. We estimate the parameters by the maximum likelihood (ML) using EM algorithm. The log-likelihood is

$$
\begin{aligned}
\mathcal{L}(\theta) &= \sum_t \log p(X_{1t}, \cdots, X_{kt}) \\
&= \sum_t \log \left(\sum_{s_t} p(s_t) \prod_k \int p(X_{kt}|\xi_{kt}) p(\xi_{kt}|s_t) d\xi_{kt} \right) \\
&\geq \sum_{ts_t} \int q(s_t) \prod_k q(\xi_{kt}|s_t) \\
&\quad \times \log \frac{p(s_t) \prod_k p(X_{kt}|\xi_{kt}) p(\xi_{kt}|s_t)}{q(s_t) \prod_k q(\xi_{kt}|s_t)} d\xi_{1t} \cdots d\xi_{Kt} \\
&= \mathcal{F}(q, \theta). \qquad (4.16)
\end{aligned}
$$

The inequality holds for any choice of distribution q due to Jensen's inequality [23]. The EM algorithm iteratively optimizes $\mathcal{F}(q, \theta)$ over q and θ, while optimizing $\mathcal{L}$ with respect to θ directly is more difficult. When q equals the posterior distribution $q(\xi_{1t}, \cdots, \xi_{Kt}, s_t) = p(\xi_{1t}, \cdots, \xi_{Kt}, s_t | X_{1t}, \cdots, X_{kt})$, the lower bound is tight, $\mathcal{F}(q, \theta) = \mathcal{L}(\theta)$.

4.2.1 The Expectation Step

The E-step updates the distribution q. The optimal $q(\xi_{kt}|s_t)$ that maximizes $\mathcal{F}$ satisfies

$$
\begin{aligned}
\log q(\xi_{kt}|s_t) &= \log p(X_{kt}|\xi_{kt}) + \log p(\xi_{kt}|s_t) + c \\
&= -\xi_{kt} - e^{-\xi_{kt}} |X_{kt}|^2 - \frac{\nu_{ks}}{2} (\xi_{kt} - \mu_{ks})^2 + c \qquad (4.17)
\end{aligned}
$$

where c is a constant. There is no closed-form density, so we compute a Gaussian approximation by expanding $\log q$ to the second order around the mode $\hat{\xi}_{kts_t}$. We have

$$\begin{aligned}
q(\xi_{kt}|s_t) &= \mathcal{N}(\xi_{kt}|\bar{\xi}_{kts_t}, \phi_{kts_t}) && (4.18)\\
\bar{\xi}_{kts_t} &= \hat{\xi}_{kts_t} + \frac{1}{\phi_{kts_t}}(e^{-\hat{\xi}_{kts_t}}|X_{kt}|^2 \\
&\quad -\nu_{ks_t}\hat{\xi}_{kts_t} + \nu_{ks_t}\mu_{ks_t} - 1) && (4.19)\\
\phi_{kts_t} &= e^{-\hat{\xi}_{kts_t}}|X_{kt}|^2 + \nu_{ks_t}. && (4.20)
\end{aligned}$$

The accuracy of the Gaussian approximation depends on the point $\hat{\xi}_{kts_t}$ where we expand $\log q(\xi_{kt}|s_t)$. We choose the mode of the posterior. Substituting $\hat{\xi}_{kts_t} = \bar{\xi}_{kts_t}$ into Eq(4.19) yields an iterative update,

$$\bar{\xi}_{kts_t} \leftarrow \bar{\xi}_{kts_t} + \frac{1}{\phi_{kts_t}}(e^{-\bar{\xi}_{kts_t}}|X_{kt}|^2 - \nu_{ks_t}\bar{\xi}_{kts_t} + \nu_{ks_t}\mu_{ks_t} - 1) \qquad (4.21)$$

This update rule is equivalent to maximizing $\log q(\xi_{kt}|s_t)$ in Eq(4.17) using the the Newton's method,

$$\bar{\xi}_{kts_t} \leftarrow \bar{\xi}_{kts_t} - \frac{[\log q(\xi_{kt}|s_t)]'_{\xi_{kt}=\bar{\xi}_{kts_t}}}{[\log q(\xi_{kt}|s_t)]''_{\xi_{kt}=\bar{\xi}_{kts_t}}} \qquad (4.22)$$

The second-order approximation to $\log q(\xi_{kt}|s_t)$ from Newton's method results in a Gaussian PDF.

The posterior state probability $q(s_t)$ is computed by maximizing $\mathcal{F}(q, \theta)$ with respect to $q(s_t)$. We define

$$\begin{aligned} f_{kts_t} &= \int q(\xi_{kt}|s_t)(\log p(X_{kt}, \xi_{kt}|s_t) - \log q(\xi_{kt}|s_t)) \\ &= \log \frac{\sqrt{\nu_{ks_t}}}{\pi\sqrt{\phi_{kts_t}}} - e^{-\bar{\xi}_{kts_t}+1/(2\phi_{kts_t})}|X_{kt}|^2 - \bar{\xi}_{kts_t} \\ &\quad -\frac{\nu_{ks_t}}{2}(\frac{1}{\phi_{kts_t}} + (\bar{\xi}_{kts_t} - \mu_{ks_t})^2) + \frac{1}{2} \end{aligned} \tag{4.23}$$

And $q(s_t)$ can be obtained as

$$q(s_t) = \frac{\exp(\sum_k f_{kts_t})p(s_t)}{Z_t} \tag{4.24}$$

$$Z_t = \sum_{s_t} \exp(\sum_k f_{kts_t})p(s_t) \tag{4.25}$$

4.2.2 The Maximization Step

The M-step optimizes $\mathcal{F}(q, \theta)$ over model parameters θ.

$$\mu_{ks} = \frac{\sum_t q(s_t = s)\xi_{kts_t}}{\sum_t q(s_t = s)} \tag{4.26}$$

$$\frac{1}{\nu_{ks}} = \frac{\sum_t q(s_t = s)[(\bar{\xi}_{kts_t} - \mu_{ks})^2 + 1/(\phi_{kts_t})]}{\sum_t q(s_t = s)} \tag{4.27}$$

$$p(s) = \frac{\sum_t q(s_t = s)}{\sum_{ts} q(s_t = s)} \tag{4.28}$$

The cost $\mathcal{F}$ is computed as $\mathcal{F} = \sum_t \log(Z_t)$ which can be used empirically to monitor the convergence, because the $\mathcal{F}$ is not guaranteed to increase due to the approximation in the E-step.

The parameters of a GMM trained in the log-spectral domain are used to initialize the EM algorithm. The E-step and M-step are iterated until convergence, which is very quick because ξ_k simulates the log-spectra.

4.3 Two Signal Estimation Approaches

The task of speech enhancement is to recover the signal from noisy recordings based on the model assumption. For the probabilistic models, signal is usually estimated from the its posterior PDF given the observed samples. For example, the MMSE estimator computes the posterior mean. However, for sophisticated models, the closed-form solutions for the posterior PDF and MMSE estimator are difficult to obtain. To enhance the tractability, we use the Laplace method [21] and a variational approximation [2].

Each frame is independent and processed sequentially. The frame index t is omitted for simplicity. We rewrite the full model as

$$\prod_k p(Y_k|X_k)p(X_k|\xi_k)p(\xi_k|s)p(s) \tag{4.29}$$

where $p(Y_k|X_k)$ is given by Eq.(4.3), $p(X_k|\xi_k)$ is given by Eq.(4.6), $p(\xi_k|s)$ is a GMM given in Eq.(4.9) and $p(s)$ is the state probability.

4.3.1 Laplace Method for Signal Estimation

The Laplace method computes the maximum a posterior (MAP) estimator for each state s. Conditioned on the state s, the logarithm of the posterior distribution over X_k and ξ_k is

$$\begin{aligned}
\log p(X_k, \xi_k|Y_k, s) &= \log p(Y_k|X_k) + \log p(X_k|\xi_k) + \log p(\xi_k|s) + c \\
&= -\gamma_k|Y_k - X_k|^2 - \xi_k - e^{-\xi_k}|X_k|^2 - \frac{\nu_{ks}}{2}(\xi_k - \mu_{ks})^2 + c \\
&= h_s(X_k, \xi_k) \qquad (4.30)
\end{aligned}$$

For fixed ξ_k, the MAP estimator for X_k is

$$X_k = \frac{\gamma_k Y_k}{\gamma_k + e^{-\xi_k}} \tag{4.31}$$

For fixed X_k, the optimization over ξ_k can be performed using Newton's method. We need

$$\frac{\partial h_s(X_k, \xi_k)}{\partial \xi_k} = -1 + e^{-\xi_k}|X_k|^2 - \nu_{ks}(\xi_k - \mu_{ks}) \tag{4.32}$$

$$\frac{\partial^2 h_s(X_k, \xi_k)}{\partial \xi_k^2} = -e^{-\xi_k}|X_k|^2 - \nu_{ks} \tag{4.33}$$

Substituting X_k by Eq(4.31), we obtain the update rule for ξ_{ks}

$$\xi_{ks} \leftarrow \xi_{ks} - \frac{\partial h_s(X_k, \xi_k)/\partial \xi_k|_{\xi_k=\xi_{ks}}}{\partial^2 h_s(X_k, \xi_k)/\partial \xi_k^2|_{\xi_k=\xi_{ks}}} \tag{4.34}$$

This update rule is initialized by both $\xi_{ks} = \mu_{ks}$, the means of GSMM and $\xi_{ks} = \log|Y_k|^2$, the noisy log-spectra. After iterating to convergence, the ξ_{ks} that gives higher value of $h_s(X_k, \xi_k)$ is selected. Note that because $h_s(X_k, \xi_k)$ is a concave function in ξ_k, $\frac{\partial^2 h_s(X_k,\xi_k)}{\partial \xi_k^2} < 0$, Newton's method works efficiently.

Denote the convergent value for ξ_{ks} from Eq(4.34) as $\bar{\xi}_{ks}$ and compute $\bar{X}_{ks} = \frac{\gamma_k Y_k}{\gamma_k + e^{-\tilde{\xi}_{ks}}}$ using Eq(4.31). We obtain the MAP estimators

$$(\bar{X}_{ks}, \bar{\xi}_{ks}) = \arg\max_{X_k, \xi_k} \log p(X_k, \xi_k | Y_k, s) \tag{4.35}$$

Because the true state s is unknown, the estimators are averaged over all states.

The posterior state probability is

$$p(s|Y_1, \cdots, Y_K) \propto p(s)\prod_k \int p(Y_k|X_k)p(X_k|s)dX_k \tag{4.36}$$

$$p(X_k|s) = \int p(X_k|\xi_k)p(\xi_k|s)d\xi_k \tag{4.37}$$

The above integral is intractable. Approximate the $p(X_k|s)$ by a Gaussian with the same first and second order momenta. The mean is zero and the variance is

$$\beta_{ks} = \int |X_k|^2 p(X_k|s)dX_k = e^{\mu_{ks}+1/(2\nu_{ks})} \tag{4.38}$$

Using the approximate PDF $p(X_k|s) \approx \mathcal{N}(X_k|0, 1/\beta_{ks})$, the Gaussian integral in Eq(4.36) becomes $p(Y_k|s) \approx \mathcal{N}(Y_k|0, \frac{1}{1/\gamma_k + e^{\mu_{ks}+1/(2\nu_{ks})}})$ and

$$p(s|Y_1, \cdots, Y_K) \propto p(s)\prod_k \mathcal{N}(Y_k|0, \frac{1}{1/\gamma_k + e^{\mu_{ks}+1/(2\nu_{ks})}}) \tag{4.39}$$

The estimated signal can be constructed from the average of either $\bar{X}_{ks}$ or $\bar{\xi}_{ks}$, weighted by the posterior state probability,

$$\hat{X}_k = \sum_s \bar{X}_{ks} p(s|Y_1, \cdots, Y_K) \tag{4.40}$$

$$\hat{\xi}_k = \sum_s \bar{\xi}_{ks} p(s|Y_1, \cdots, Y_K) \tag{4.41}$$

$$\hat{X}_k^{ls} = e^{\hat{\xi}_k/2} e^{i\angle Y_k} \tag{4.42}$$

where the phase of the noisy signal $\angle Y_k$ is used. The time domain signal is synthesized by applying IFFT.

4.3.2 Variational Approximation for Signal Estimation

Variational approximation employs a factorized posterior PDF. Here, we assume the posterior PDF over X_k and ξ_k conditioned on state s factorizes

$$p(X_k, \xi_k, |Y_k, s) \approx q(X_k|s)q(\xi_k|s) \tag{4.43}$$

The difference between q and the true posterior is measured by the KL-divergence [23], D, defined as

$$D(q||p) = -E^q \left\{ \log \frac{p(s|Y_1, \cdots, Y_K) \prod_k p(X_k, \xi_k|Y_k, s)}{q(s) \prod_k q(X_k|s)q(\xi_k|s)} \right\} \tag{4.44}$$

where E^q is the expectation over q. Choose the optimal q that is closest to the true posterior in the sense of the KL-divergence, $q = \arg\min_q D(q||p)$.

The optimal $q(X_k|s)$ that minimizes $D(q||p)$ is

$$\begin{aligned} \log q(X_k|s) &\propto \log p(Y_k|X_k) + \int d\xi_k q(\xi_k|s) \log p(X_k|\xi_k) \\ &\propto -\gamma_k |Y_k - X_k|^2 - \int e^{-\xi_k} q(\xi_k|s) d\xi_k |X_k|^2 \end{aligned} \tag{4.45}$$

As shown later in Eq(4.50), we can use $q(\xi_k|s) = \mathcal{N}(\xi_k|\bar{\xi}_{ks}, \psi_{ks})$. Because the above equation is quadratic in X_k, $q(X_k|s)$ is Gaussian

$$q(X_k|s) = \mathcal{N}(X_k|\bar{X}_{ks}, \varphi_{ks}) \tag{4.46}$$

$$\bar{X}_{ks} = \frac{\gamma_k}{\varphi_{ks}} Y_k \tag{4.47}$$

$$\varphi_{ks} = \gamma_k + e^{-\bar{\xi}_{ks} + 1/(2\psi_{ks})} \tag{4.48}$$

The optimal $q(\xi_k|s)$ that minimizes $D(q||p)$ is

$$\begin{aligned}\log q(\xi_k|s) &\propto \int dX_k q(X_k|s) \log p(X_k|\xi_k) + \log p(\xi_k|s) \\ &\propto -\xi_k - e^{-\xi_k} \int |X_k|^2 q(X_k|s) dX_k - \frac{\nu_{ks}}{2}(\xi_k - \mu_{ks})^2 \end{aligned} \tag{4.49}$$

Because this PDF is hard to work with, a Gaussian approximation is made by expanding $\log q(\xi_k|s)$ around its mode ρ_{ks} up to the second order. We have

$$q(\xi_k|s) = \mathcal{N}(\xi_k|\bar{\xi}_{ks}, \psi_{ks}) \tag{4.50}$$

$$\begin{aligned}\bar{\xi}_{ks} &= \rho_{ks} + \frac{1}{\psi_{ks}}\Big(e^{-\rho_{ks}}(|\bar{X}_{ks}|^2 + \frac{1}{\varphi_{ks}}) \\ &\quad -\nu_{ks}(\rho_{ks} - \mu_{ks}) - 1\Big)\end{aligned} \tag{4.51}$$

$$\psi_{ks} = e^{-\rho_{ks}}(|\bar{X}_{ks}|^2 + \frac{1}{\varphi_{ks}}) + \nu_{ks} \tag{4.52}$$

Because we chose ρ_{ks} to be the posterior mode, $\rho_{ks} = \bar{\xi}_{ks}$. Substituting this into Eq(4.51) and Eq(4.52), we obtain the update equations

$$\begin{aligned}\bar{\xi}_{ks} &\leftarrow \bar{\xi}_{ks} + \frac{1}{\psi_{ks}}\Big(e^{-\bar{\xi}_{ks}}(|\bar{X}_{ks}|^2 + \frac{1}{\varphi_{ks}}) \\ &\quad -\nu_{ks}(\bar{\xi}_{ks} - \mu_{ks}) - 1\Big)\end{aligned} \tag{4.53}$$

$$\psi_{ks} \leftarrow e^{-\bar{\xi}_{ks}}(|\bar{X}_{ks}|^2 + \frac{1}{\varphi_{ks}}) + \nu_{ks} \tag{4.54}$$

This is equivalent to maximizing the $\log q(\xi_k|s)$ of Eq(4.49) using Newton's method. The $\psi_{ks} > 0$ indicates $\log q(\xi_k|s)$ is a concave function in ξ_k, thus Newton's method is efficient. Expanding $\log q(\xi_k|s)$ to the second order results in the Gaussian PDF.

The variational algorithm is initialized with $\xi_{ks} = \log(|Y_k|^2)$ and $\varphi_{ks} = \gamma_k +$

$\exp(-\xi_{ks})$. Note that $\bar{X}_{ks}$ in Eq(4.47) can be substituted into Eq(4.53) and Eq(4.54) to avoid redundant computation. Then the updates over ψ_{ks}, ξ_{ks} and φ_{ks} iterate until convergence.

To compute $q(s)$ that minimizes $D(q||p)$, define

$$\begin{aligned} g_{ks} &= \int q(X_k|s)q(\xi_k|s)\log\frac{p(Y_k|X_k)p(X_k|\xi_k)p(\xi_k|s)}{q(X_k|s)q(\xi_k|s)} \\ &= \log\frac{\gamma_k\sqrt{\nu_{ks}}}{\pi\varphi_{ks}\sqrt{\psi_{ks}}} - \gamma_k|Y_k|^2 + \varphi_{ks}|\bar{X}_{ks}|^2 - \bar{\xi}_{ks} \\ &\quad -\frac{\nu_{ks}}{2}\left[(\bar{\xi}_{ks}-\mu_{ks})^2 + \frac{1}{\psi_{ks}}\right] + \frac{1}{2} \end{aligned} \tag{4.55}$$

The posterior state probability is

$$q(s) = \frac{\exp(\sum_k g_{ks})p(s)}{Z} \tag{4.56}$$

$$Z = \sum_s \exp(\sum_k g_{ks})p(s) \tag{4.57}$$

The function $\log(Z) = \log p(Y_1,\cdots,Y_K) - D(q||p)$ increases when $D(q||p)$ decreases. Because we use a Gaussian for $q(\xi_k|s)$, $\log(Z)$ is not theoretically guaranteed to increase, but it is used empirically to monitor the convergence.

Because the hidden state s is unknown, the estimator for each state is averaged with weights $q(s)$. Similar to the Laplace method, we can construct the signal in two ways

$$\hat{X}_k = \sum_s \bar{X}_{ks}q(s) \tag{4.58}$$

$$\hat{\xi}_k = \sum_s \bar{\xi}_{ks}q(s) \tag{4.59}$$

$$\hat{X}_k^{ls} = e^{\hat{\xi}_k/2}e^{i\angle Y_k} \tag{4.60}$$

where $\angle Y_k$ is the phase of the noisy signal. Time domain signal is synthesized by IFFT.

Acknowledgement: This chapter contains materials in J. Hao, T-W. Lee and T. Sejnowski, “Speech Enhancement Using Gaussian Scale Mixture Models”, *IEEE Transactions on Audio, Speech, and Language Processing*, submitted for publication.

5

Experimental Results for Speech Enhancement

We evaluate the performances of the proposed algorithms by applying them to enhance the speeches corrupted by various levels of SSN. The signal to noise ratio (SNR) and word recognition error rate serve as the criteria to compare them with the other benchmark algorithms quantitatively.

5.1 Task and Dataset Description

For all the experiments, we use the materials provided by the speech separation challenge [25]. This data set contains six-word sentences from 34 speakers. The speech follows the sentence grammar, $\langle$\$command$\rangle$ $\langle$\$color$\rangle$ $\langle$\$preposition$\rangle$ $\langle$\$letter$\rangle$ $\langle$\$number$\rangle$ $\langle$\$adverb$\rangle$. There are 25 choices for the letter (*a-z except w*), 10 choices for the number (*0-9*), 4 choices for the command (*bin, lay, place, set*), 4 choices for the color (*blue, green, red, white*), 4 choices for the preposition (*at, by, in, with*) and 4 choices for the ad-

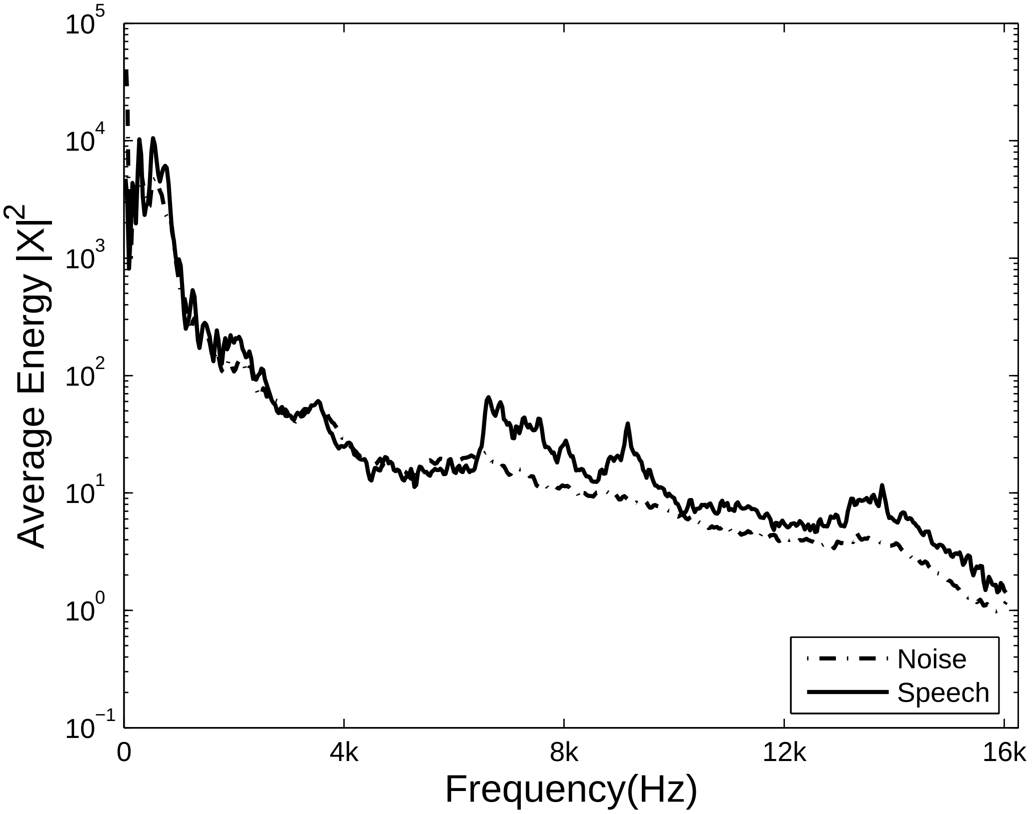

Figure 5.1: Plot of SSN spectrum (dotted line) and speech spectrum (solid line) averaged over one segment under 0dB SNR. Note the similar shapes.

verb (*again, now, please, soon*). The time domain signals are sampled at 25kHz. Provided with the training samples, the task is to recover speech signals and recognize the key words (*color, letter, digit*) in the presence of different levels of SSN. Figure 5.1 shows the speech and the SSN spectrum averaged over a segment under 0dB SNR. The average spectra of the speech and the noise have the similar shape, hence the name speech shaped noise. The testing set includes the noisy signals under 4 SNR conditions, −12dB, −6dB, 0dB, and 6dB, each consisting of 600 utterances from 34 speakers.

5.2 Training the Log-Spectral Domain GMM

The training set consists of clean signal segments that are $1-2$ seconds long. They are used to train our prior speech model. To obtain a reliable speech model, we randomly concatenate 2 minutes of signals from the training set and analyze them using Hanning windows, each of size 800 samples and overlapping by half of the window. Frequency coefficients are obtained by performing a 1024 points FFT to the time domain signals. Coefficients in the log-spectral domain are obtained by taking the logarithm of the magnitude of the FFT coefficients. Due to FFT/IFFT symmetry, only the first 513 frequency components are kept. Cepstral coefficients are obtained by applying IFFT on the log-spectral amplitudes.

The speech model for each speaker is a GMM with 30 states in the log-spectral domain. First, we take the first 40 cepstral coefficients and apply a k-mean algorithm to obtain $k = 30$ clusters. Next, the outputs of the k-mean clustering are used to initialize the GMM on those 40 cepstral coefficients. Then, we convert the GMM from the cepstral domain into the log-spectral domain using FFT. Finally, the EM algorithm initialized by the converted GMM is used to train the GMM in the log-spectral domain. After training, this log-spectral domain GMM with 30 states for speech is fixed when processing the noisy signals.

5.3 Training the Gaussian Scale Mixture Model

The GSMM with 30 states for each speaker was trained using 2 minutes of signal concatenated from the training set. We first applied a Hanning window of size 800 samples with half overlapping, then a 1024-point FFT to extract the frequency components. The log-spectral coefficients were obtained by taking the log magnitude of the FFT coefficients.

Due to the symmetry of FFT, only first 513 components were kept.

We applied the k-mean algorithm to partition the log-spectra into $k = 30$ clusters. They were used to initialize the GMM which was further trained by standard EM algorithm. Initialized by the GMM, we ran the derived EM algorithm in section 4.2 to train the GSMM. After training, the speech model was fixed and served as signal prior. It was not updated when processing the noisy signals.

5.4 Benchmark Algorithms for Comparison

In this section, we present the benchmark algorithms with which we compare the proposed algorithms: the Wiener filter, the perceptual model [12], the linear approximation [19, 20], and the model based on super Gaussian prior [24]. We assume that parameters of the model for noise are available, and they are estimated by concatenating 50 segments in the experiment.

5.4.1 Wiener Filter (Wiener)

Time varying Wiener filter assumes that both of the signal and noise power are known, and they are stationary for a short period of time. In the experiment, we first divide the signals into frames of 800 samples long with half overlapping. Both speech and noise are assumed to be stationary within each frame. To estimate speech and noise power, for each frame, the 200-sample long sub-frames are chosen with half overlapping. On the sub-frames, Hanning windows are applied. Then, 256 points FFT are performed on those sub-frames to obtain the frequency coefficients. The power of signal within each frame t for frequency bin k, denoted by P^x_{tk}, is computed by averaging the power of FFT coefficients over all the sub-frames that belong to the frame t. The same method is used to compute the noise power denoted by P^n_{tk}. The signal estimation is computed as

$$X_{tjk} = \frac{P^x_{tk}}{P^x_{tk} + P^n_{tk}} Y_{tjk} \tag{5.1}$$

where j is the sub-frame index and k denotes the frequency bins. After IFFT, in the time domain, each frame can be synthesized by overlap-adding the sub-frames, and the estimated speech signal is obtained by overlap-adding the frames.

Because the signal and noise powers are derived locally for each frame from the speech and noise, the Wiener filter contains strong speech prior in detail. Its performance can be regarded as a sort of experimental upper bound for the proposed methods.

5.4.2 Perceptual Model (Wolfe)

Perceptually motivated noise reduction technique can be seen as a masking process. The original signal is estimated by applying some suppression rules. For comparison, we use the method described in [12]. The algorithm estimates the spectral amplitude by minimizing the following cost function

$$C(\hat{a}_k, a_k) = \begin{cases} (\hat{a}_k - a_k - \frac{m_k}{2})^2 - (\frac{m_k}{2})^2 & \text{if } |\hat{a}_k - a_k - \frac{m_k}{2}| > \frac{m_k}{2}; \\ 0 & \text{otherwise.} \end{cases} \tag{5.2}$$

where $\hat{a}_k$ is the estimated spectral amplitude and a_k is the true spectral amplitude. This cost function penalizes the positive and negative errors differently, because positive estimation errors are perceived as additive noise and negative errors are perceived as signal attenuation [12]. The stochastic property of speech is that real spectral amplitude is unavailable, therefore $\hat{a}_k$ is computed by minimizing the expected cost function

$$\hat{a}_k = \arg\min_{\hat{a}_k} \int \int C(\hat{a}_k, a_k) p(\alpha_k, a_k | Y_k) d\alpha_k da_k \tag{5.3}$$

where α_k is the phase and $p(\alpha_k, a_k|Y_k)$ is the posterior signal distribution. Details of the algorithm can be found in [12]. The MATLAB code is available online [26]. The original code adds synthetic white noise to the clean signal, we modified it to add SSN to corrupt a speech at different SNR levels.

The reason we chose this method is because we hypothesize that this spectral analysis based approach fails to enhance the SSN corrupted speech, due to the spectral similarity between the speech and noise as shown in Figure 5.1. This method, motivated from a different aspect by human perception, also serves as a benchmark with which we can compare our methods.

5.4.3 Linear Approximation (Linear)

It can be shown that the relationship among the log-spectra of the signal x, the noisy signal y and the noise n is given by [19, 20]

$$y_k = x_k + \log(1 + \exp(n_k - x_k)) + \epsilon_k, \tag{5.4}$$

where ϵ_k is an error term.

The speech model remains the same which is GMM given by Eq(3.2). But the noise log-spectrum n has a Gaussian density with the mean ρ and precision D, while the error term ϵ obeys a Gaussian with zero-mean and precision R,

$$\begin{aligned} p(n) &= \mathcal{N}(n|\rho, D) = \prod_k \mathcal{N}(n_k|\rho_k, D_k) && (5.5) \\ p(\epsilon) &= \mathcal{N}(\epsilon|0, R) = \prod_k \mathcal{N}(\epsilon_k|0, R_k) && (5.6) \end{aligned}$$

This essentially assumes a log-normal PDF for the noise FFT coefficients, in contrast to the noise model in Eq(3.5).

Linear approximation to Eq(5.4) has been proposed in [19, 20] to enhance the tractability. Note that there are two hidden variables x and n due to the error term ϵ. Let $z_k = (x_k, n_k)^T$. Define $g(z_k) = x_k + \log(1 + \exp(n_k - x_k))$ and its derivatives $g'_x(z_k) = \frac{\partial g}{\partial x_k} = \frac{1}{1+\exp(n_k - x_k)}$, $g'_n(z_k) = \frac{\partial g}{\partial n_k} = \frac{1}{1+\exp(x_k - n_k)}$, $g'(z_k) = (g'_x(z_k), g'_n(z_k))^T$. Use Eq(5.4) and expand $g(z_k)$ around $\tilde{z}_{ks} = (\tilde{x}_{ks}, \tilde{n}_{ks})^T$ linearly, y_k becomes a linear function of z_k,

$$y_k \approx l(z_k) + \epsilon_k \tag{5.7}$$

where

$$l(z_k) = g(\tilde{z}_{ks}) + g'(\tilde{z}_{ks})^T (z_k - \tilde{z}_{ks}) \tag{5.8}$$

The choice for $\tilde{z}_{ks}$ will be discussed later. Now we have a linear Gaussian system and the posterior distribution over z_k is Gaussian, $\mathcal{N}(z_k|\hat{z}_{ks}, \Lambda)$. The mean $\hat{z}_{ks}$ and the precision Λ satisfy

$$\Lambda(z_k - \hat{z}_{ks}) = -R_k(y_k - l(z_k))g'(\tilde{z}_{ks}) - G_{ks}(\zeta_{ks} - z_k) \tag{5.9}$$

$$\Lambda = g'(\tilde{z}_{ks}) R_k g'(\tilde{z}_{ks})^T + G_{ks} \tag{5.10}$$

where $\zeta_{ks} = (\mu_{ks}, \rho_k)^T$, the means of GMM for the speech and noise log-spectrum, and $G_{ks} = \text{diag}(B_{ks}, D_k)$, the precisions.

The accuracy of linear approximation strongly depends on the point $\tilde{z}_{ks}$ which is the point of expansion for $g(z_k)$. A reasonable choice is the MAP. Substitute $z_k = \tilde{z}_{ks}$ in Eq(5.9) and use $\tilde{z}_{ks} = \hat{z}_{ks}$, we can obtain an iterative update for $\tilde{z}_{ks}$:

$$\tilde{z}_{ks} \leftarrow \tilde{z}_{ks} + \eta\Lambda^{-1}\{R_k(y_k - g(\tilde{z}_{ks}))g'(\tilde{z}_{ks}) + G_{ks}(\zeta_{ks} - \tilde{z}_{ks})\} \tag{5.11}$$

The η is the learning rate, and is introduced to avoid oscillation. This iterative update gives the signal log-spectral estimator, $\tilde{x}_{ks}$, which is the first element of the $\tilde{z}_{ks}$.

The state probability $p(s|y)$ is computed as, per Bayes' rule, $p(s|y) \propto p(y|s)p(s)$. The state dependent probability is

$$p(y|s) = \prod_k \sqrt{|\frac{\Gamma_{ks}}{2\pi}|} \exp(-\frac{\Gamma_{ks}}{2}(y_k - l(\zeta_{ks}))^2), \tag{5.12}$$

where the mean $l(\zeta_{ks})$ is given in Eq(5.8) and the precision $\Gamma_{ks} = \frac{1}{g'^T G^{-1} g' + 1/R_k}$.

The log-spectral estimator is $\bar{x}_k = \sum_s \tilde{x}_{ks} p(s|y)$. Using the phase of the noisy signal $\angle Y_k$, the signal estimation in frequency domain is given by $X_k = \exp(\bar{x}/2)e^{i\angle Y_k}$.

It is observed that Newton's method with learning rate 1 oscillates, therefore we set $\eta = 0.5$ in our experiments. We initialize the iteration of Eq(5.11) with two conditions, $(y_k, \rho_k)^T$ and $(\mu_{ks}, \rho_k)^T$, and choose the one that offers higher likelihood value. The number of iterations is 7 which is enough for convergence. Note that the optimization of the two variables x and n increases computational cost.

5.4.4 Super Gaussian Prior (SuperGauss)

This method is developed in [24]. Let $X_R = Re\{X\}$ and $X_I = Im\{X\}$ denote the real and the imaginary part of the signal FFT coefficients. The super Gaussian priors for X_R and X_I obey double-sided exponential distribution, given by

$$p(X_R) = \frac{1}{\sigma_x} e^{-\frac{2|X_R|}{\sigma_x}} \tag{5.13}$$

$$p(X_I) = \frac{1}{\sigma_x} e^{-\frac{2|X_I|}{\sigma_x}} \tag{5.14}$$

Assume the Gaussian density for the noise N, $p(N) = \mathcal{N}(0, 1/\sigma_n^2)$. Here, σ_x^2 and σ_n^2 are the means of $|X|^2$ and $|N|^2$, respectively. Let $\xi = \sigma_x^2/\sigma_n^2$ be the *a priori* SNR, $Y_R = Re\{Y\}$ be the real part of the noisy signal FFT coefficient. Define $L_{R+} = 1/\sqrt{\xi} + Y_R/\sigma_n$, and $L_{R-} = 1/\sqrt{\xi} - Y_R/\sigma_n$. It was shown in [24, Eq(11)] that the optimal estimator for the real part is

$$\hat{X}_R = Y_R + \frac{\sigma_n}{\sqrt{\xi}} \frac{e^{\frac{2Y_R}{\sigma_x}} \operatorname{erfc}(L_{R+}) - e^{-\frac{2Y_R}{\sigma_x}} \operatorname{erfc}(L_{R-})}{e^{\frac{2Y_R}{\sigma_x}} \operatorname{erfc}(L_{R+}) + e^{-\frac{2Y_R}{\sigma_x}} \operatorname{erfc}(L_{R-})} \tag{5.15}$$

where erfc(x) denotes the complementary error function. The optimal estimator for the imaginary part $\hat{X}_I$ is derived analogously in the same manner. The FFT coefficient estimator is given by $\hat{X} = \hat{X}_R + i\hat{X}_I$.

5.5 Comparison Criteria

The performance of the algorithms are subject to some quality measures. We employ three criteria to evaluate the performances of all algorithms: SNR and word recognition error rate. For all experiments, the estimated signal $\hat{x}[t]$ are normalized such that it has the same covariance as the clean signal $x[t]$ before computing the signal quality measures.

5.5.1 Signal to Noise Ratio (SNR)

In time domain, SNR is defined by

$$SNR = 10\log_{10}\frac{\sum_t (x[t])^2}{\sum_t (\hat{x}[t] - x[t])^2} \tag{5.16}$$

where $x[t]$ is original clean signal and $\hat{x}[t]$ is estimated signal.

5.5.2 Word Recognition Error Rate

We use the speech recognition engine provided on the ICSLP website [25]. The recognizer is based on the HTK package. The inputs of the recognizer include MFCC, its velocity (Δ MFCC) and its acceleration ($\Delta\Delta$ MFCC) that are extracted from speech waveforms. The words are modeled by the HMM with no skipover states and 2 states for each phoneme. The emission probability for each state is a GMM of 32 mixtures, of which the covariance matrices are diagonal. The grammar used in the recognizer is the same as the sentence grammar shown in section 5.1. More details about the recognition engine can be found at [25].

For each input SNR condition, the estimated signals are fed into the recognizer. A score of $\{0, 1, 2, 3\}$ is assigned to each utterance depending on how many key words (*color, letter, digit*) that are incorrectly recognized. The word recognition error rate in percentage is the average of the scores of all 600 testing utterances divided by 3.

Table 5.1: Signal to Noise Ratio (dB) of the speech enhanced by the algorithms listed in the leftmost column. The speech is corrupted by SSN at 4 input SNR values. The gain and the noise spectrum are assumed to be known. Wiener: Wiener filter, Wolfe99: perceptual model, Linear: linear approximation,SuperGauss: super Gaussian prior, LaplaceFFT: Laplace method in frequency domain, LaplaceLS: Laplace method in log-spectral domain, Gaussian: Gaussian approximation, GSMM Lap FFT: FFT coefficients estimation by GSMM using Laplace method, GSMM Lap LS: log-spectra estimation by GSMM using Laplace method, GSMM Var FFT: FFT coefficients estimation by GSMM using variational approximation, GSMM Var LA: log-spectra estimation by GSMM using variational approximation.

Input SNR	-12dB	-6dB	0dB	6dB
Wiener	1.05	3.49	6.73	10.70
Wolfe99	-3.29	-1.58	1.70	6.17
Linear	-1.44	1.53	5.63	9.85
SuperGauss	-2.94	1.61	5.25	9.37
LaplaceFFT	-0.37	2.72	6.63	10.92
LaplaceLS	-0.48	2.82	6.81	11.16
Gaussian	-0.45	2.39	5.99	10.02
GSMM Lap FFT	-1.14	1.80	5.97	10.44
GSMM Lap LS	-1.31	1.49	5.45	9.29
GSMM Var FFT	-1.14	2.28	6.56	11.09
GSMM Var LS	-1.72	1.73	5.81	9.59

5.6 Performance Comparison with Fixed Gain and Known Noise Spectrum

All the algorithms are applied to enhance the speech corrupted by SSN at various SNR levels. They are compared by SNR and word recognition error rate. The Wiener filer, which contains the strong and detailed signal prior from a clean speech, can be regarded as an experimental upper bound.

Figure 5.2 and Figure 5.3 show the spectrograms of a female speech and a male speech, respectively. The SNR for the noisy speech is 6dB. The Wiener filter can recover the spectrogram of the speech. The methods based on the models in log-spectral domain

Table 5.2: Word recognition error rate of the speech enhanced by the algorithms listed in the leftmost column. The speech is corrupted by SSN at 4 input SNR values. The gain and the noise spectrum are assumed to be known. See Table 5.1 for the description of algorithms. No Denoising stands for noisy signal input without processing.

Input SNR	**-12dB**	**-6dB**	**0dB**	**6dB**
Wiener	18.51%	11.38%	7.43%	5.17%
Wolfe99	87.33%	84.78%	78.06%	62.61%
Linear	85.06%	68.67%	31.33%	9.78%
SuperGauss	87.56%	83.61%	62.17%	27.17%
LaplaceFFT	78.83%	59.06%	29.22%	11.83%
LaplaceLS	77.83%	50.22%	20.78%	6.22%
Gaussian	80.94%	65.78%	37.33%	16.56%
GSMM Lap FFT	83.91%	73.94%	57.66%	31.63%
GSMM Lap LS	82.21%	67.77%	38.14%	16.04%
GSMM Var FFT	84.24%	76.57%	45.62%	13.72%
GSMM Var LS	85.80%	76.33%	39.02%	14.08%
No Denoising	88.33%	88.22%	81.06%	43.33%

(Linear, LaplaceFFT, LaplaceLS, and Gaussian) can effectively suppress the SSN and recover the spectrogram. Because the SuperGauss estimates the real and imaginary part separately, the spectral amplitude is not optimally estimated which leads to a blurred spectrogram. The perceptual model (Wofle99) fails to suppress SSN because of its spectral similarity to speech. For the algorithms based on the GSMM, the spectrograms of the signals are recovered. The Laplace methods give clearer spectrogram than the variational approximation.

Table 5.1 presents the output SNR's for all the algorithms. They are graphically shown in Figure 5.4. Wiener filter performs the best. Laplace methods (LaplaceFFT and LaplaceLS) are very effective, and the LaplaceLS is better. This coincides with the belief that the log-spectral amplitude estimator is more suitable for speech processing. The Gaussian approximation works comparably well to the Laplace methods with the advantage of greater computational efficiency where no iteration is necessary. The linear approximation provides inferior SNR. The reason is that this approach involves two hidden variables, which may increase the uncertainty for signal estimation. The SuperGauss works better than perceptual model (Wolfe99) which fails to suppress SSN. For GSMM, the frequency domain algorithms (GSMM Lap FFT and GSMM Var FFT) give higher SNR than log-spectral domain algorithms (GSMM Lap LS and GSMM Var LS), because the frequency coefficients are more reliably estimated which produce better signals in time domain. Further, the algorithms using GSMM is close to the top performer (LaplaceLS). Because GSMM estimates both the FFT coefficients and log-spectra simultaneously which are correlated, and it is in general harder to estimate two random variables than one, the Laplace method with GMM in log-spectral domain perform better than GSMM.

The word recognition error rate of speeches enhanced by all the algorithms are shown in Table 5.2 and Figure 5.5. The outstanding performance of Wiener filter may be considered as an upper bound. The Linear and LaplaceLS give very low word recognition error rate in the high SNR range, because they estimate the log-spectral amplitude, which is a strong fit to the recognizer input (MFCC). LaplaceLS is better than Linear in the low SNR range, because Linear involves two hidden variables to estimate. The LaplaceFFT and Gaussian also improve the recognition remarkably. Because SuperGauss offers less accurate spectral amplitude estimation, it gives lower word recognition rate. The Wolfe99 is not able to suppress SSN and the decrease in performance may be caused by the spectral distortion. It is interesting to observe that for GSMM, the log-spectral domain algorithms provide lower word recognition rate than the frequency domain algorithms, because the more reliably estimated log-spectra fits the speech recognition engine and reduce the error rate. Note that the LaplaceLS only estimates the log-spectra, while the GSMM estimated both FFT coefficients and log-spectra which is in general harder. LaplaceLS gives the lowest word recognition error rate.

5.7 Performance Comparison with Estimated Gain and Noise Spectrum

Table 5.3: Signal to Noise Ratio (dB) of speech enhanced by algorithms based on Gaussian approximation. The speech is corrupted by SSN. Known Noise: known gain and noise spectrum. Scalar Gain: estimated frequency-independent gain and noise spectrum. Vector Gain: estimated frequency dependent gain and noise spectrum.

Input SNR	-12dB	-6dB	0dB	6dB	clean
Known Noise	-0.45	2.39	5.99	10.02	
Scalar Gain	-0.62	2.21	5.90	9.84	32.71
Vector Gain	-0.93	1.77	5.12	8.40	15.32

Table 5.4: word recognition error rate of speech enhanced by algorithms based on Gaussian approximation. See Table 5.3 for the explanation of the algorithms.

Input SNR	-12dB	-6dB	0dB	6dB	clean
Known Noise	80.94%	65.78%	37.33%	16.56%	1.44%
Scalar Gain	82.56%	65.61%	37.89%	17.33%	1.56%
Vector Gain	82.61%	69.11%	39.61%	17.00%	1.67%
No Denoising	88.33%	88.22%	81.06%	43.33%	1.44

The performances of the Gaussian approximation with the fixed gain versus the estimated gain and noise spectrum are compared. The SNR and word recognition error rate of the enhanced speech are shown in Table 5.3 and Table 5.4, respectively. They are graphically represented in Figure 5.6 and Figure 5.7. The performances are almost identical, which demonstrate that, under Gaussian approximation, the learning of gain and noise spectrum is very effective. Estimation of gain and noise degrades the performance compared to the scenario of fixed gain and known noise spectrum very slightly. Furthermore, with clean signal input, the estimated signal still has 32.71dB SNR for scalar gain and 15.32dB SNR for vector gain. The recognition error rate is also close to the results of the clean signal input. The slight degradation in the vector gain case is because we have more parameters to estimate.

Acknowledgement: This chapter contains materials in J. Hao, H. Attias, S. Nagarajan, T.-W. Lee and T. Sejnowski, “Speech Enhancement, Gain, and Noise Spectrum Adaptation Using Approximate Bayesian Inference”, *IEEE Transactions on Audio, Speech, and Language Processing*, accepted for publication and J. Hao, T-W. Lee and T. Sejnowski, “Speech Enhancement Using Gaussian Scale Mixture Models”, *IEEE Transactions on Audio, Speech, and Language Processing*, submitted for publication.

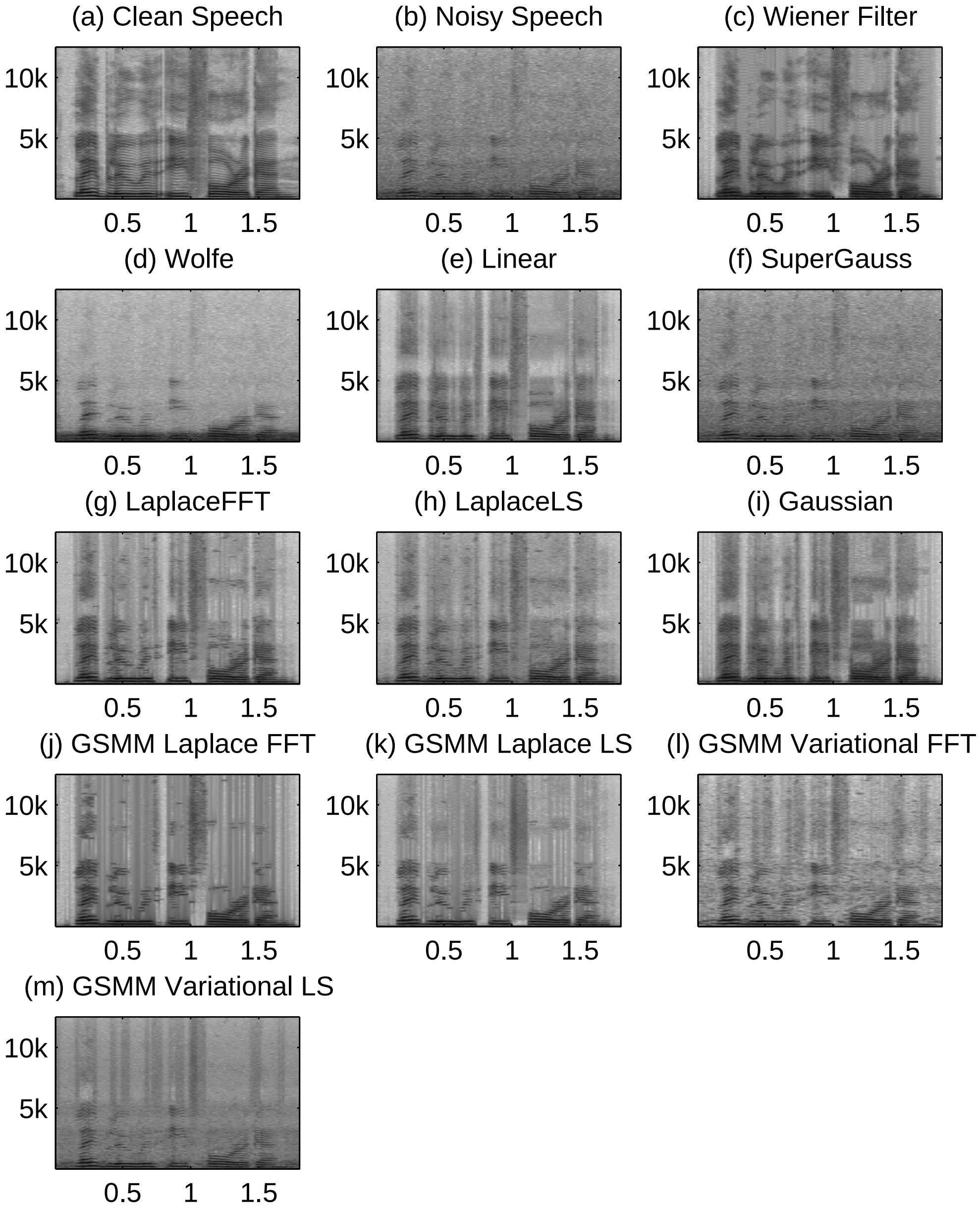

Figure 5.2: Spectrogram of a female speech "lay blue with e four again". (a) clean speech; (b) noisy speech of 6dB SNR; (c-i) enhanced signals by (c) Wiener filter, (d) perceptual model (Wolfe), (e) linear approximation (Linear), (f) super Gaussian prior (SuperGauss), (g) Laplace method in frequency domain (LaplaceFFT) and (h)in log-spectral domain (LaplaceLS), (i) Gaussian approximation (Gaussian), (j) FFT coefficients estimation by GSMM using Laplace method (GSMM Lap FFT), see Eq(4.40), (k) log-spectra estimation by GSMM using Laplace method (GSMM Lap LS), see Eq(4.42), (l) FFT coefficients estimation by GSMM using variational approximation (GSMM Var FFT), see Eq(4.58), (m) log-spectra estimation by GSMM using variational approximation (GSMM Var LS), see Eq(4.60).

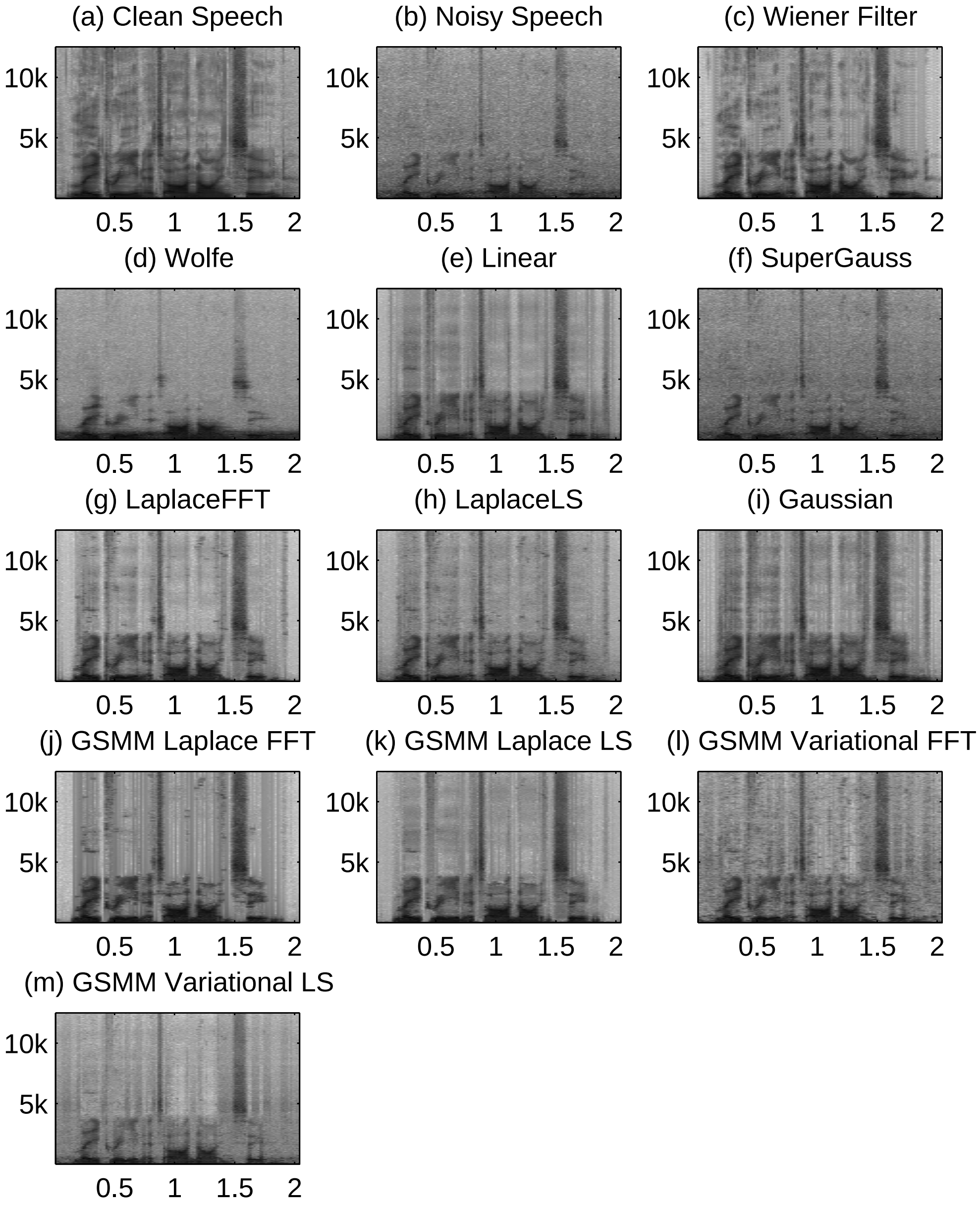

Figure 5.3: Spectrogram of a male speech "lay green at r nine soon". (a) clean speech; (b) noisy speech of 6dB SNR; (c-m) enhanced signal by various algorithms. See Figure 5.2.

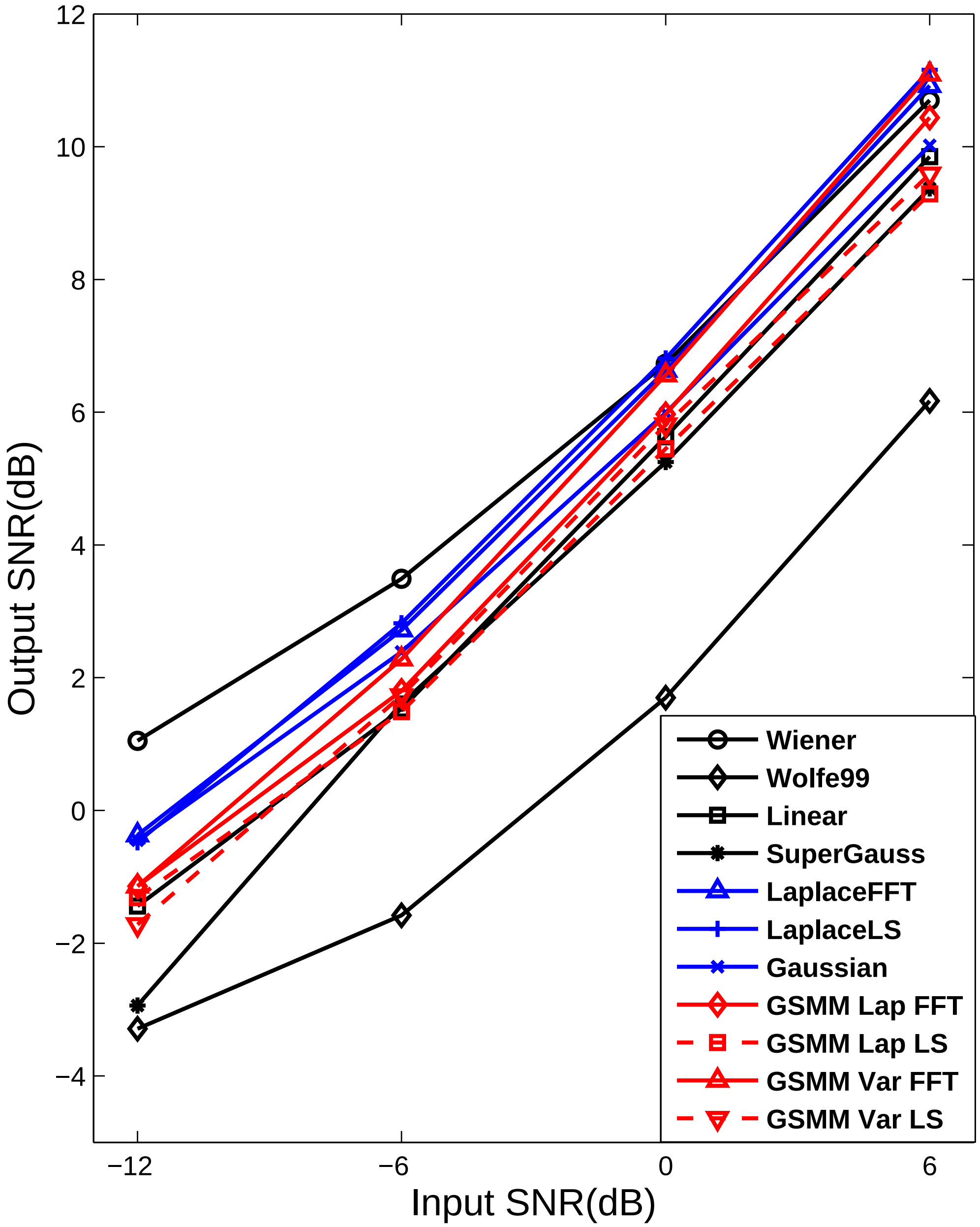

Figure 5.4: Plot of the signal to noise ratios of speeches enhanced by various algorithms. The speech is corrupted at 4 input SNR values. The gain and the noise spectrum are assumed to be known. See Table 5.1 for description of the algorithms.

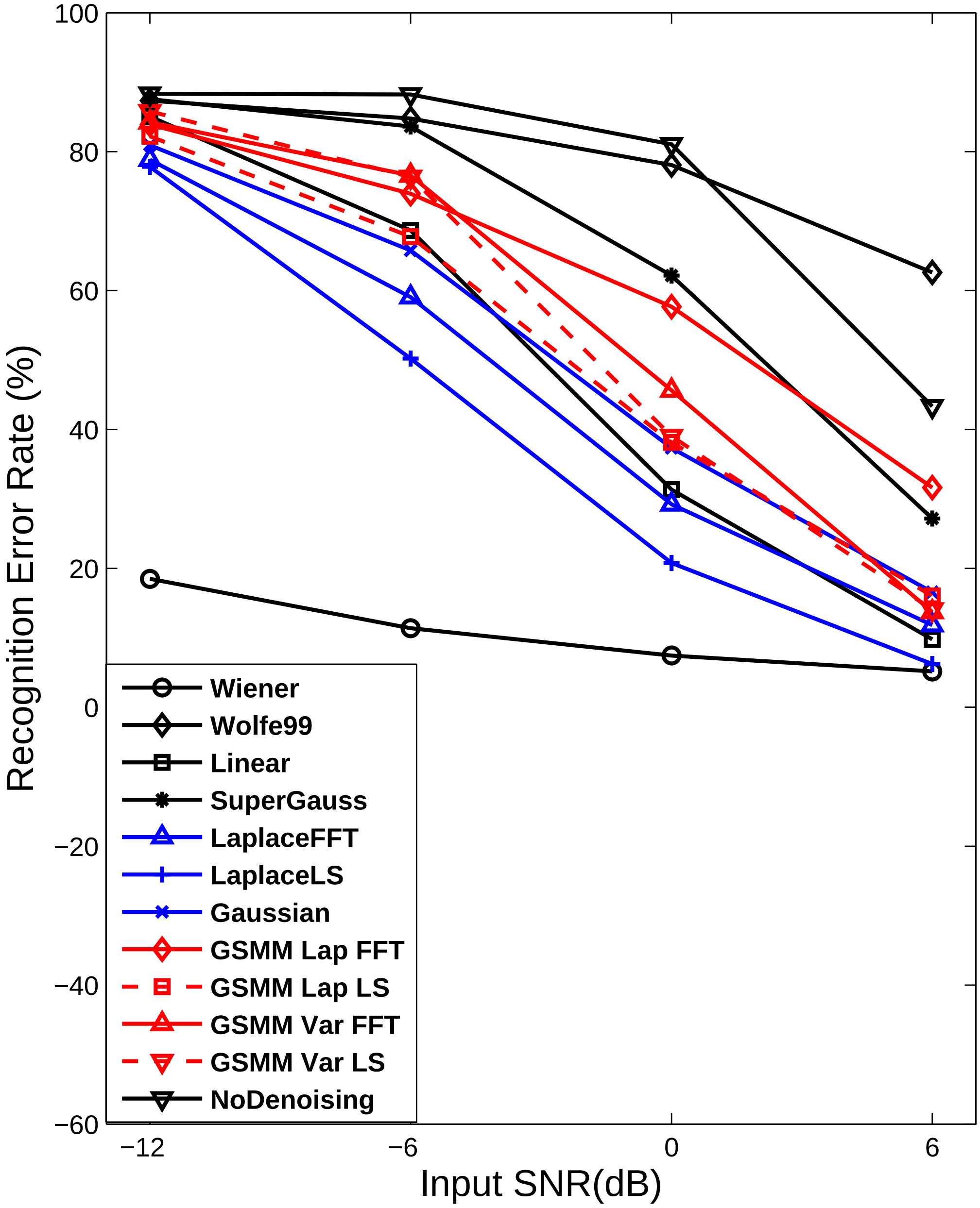

Figure 5.5: Plot of the word recognition error rate of speeches enhanced by the algorithms. The speech is corrupted at 4 input SNR values. The gain and the noise spectrum are assumed to be known. See Table 5.2 for description of the algorithms.

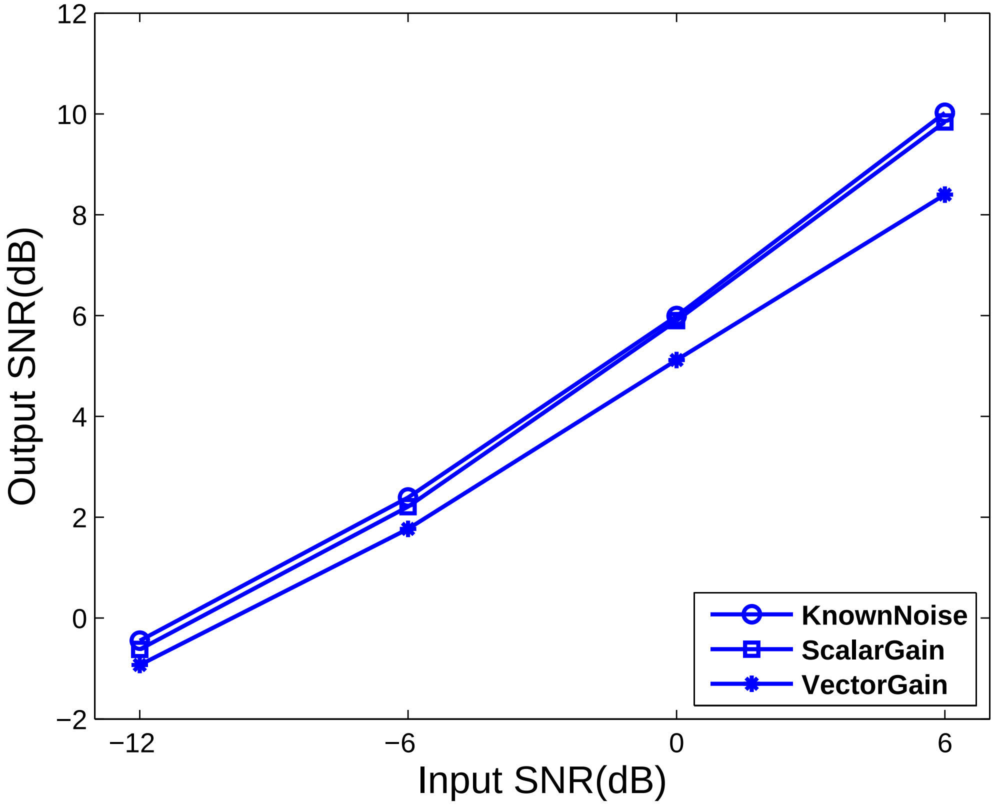

Figure 5.6: Plot of the signal to noise ratio of speeches enhanced by the algorithms. The speech is corrupted at 4 input SNR values. KnownNoise: known gain and noise spectrum; ScalarGain: estimated frequency-independent gain and noise spectrum; VectorGain: estimated frequency dependent gain and noise spectrum.

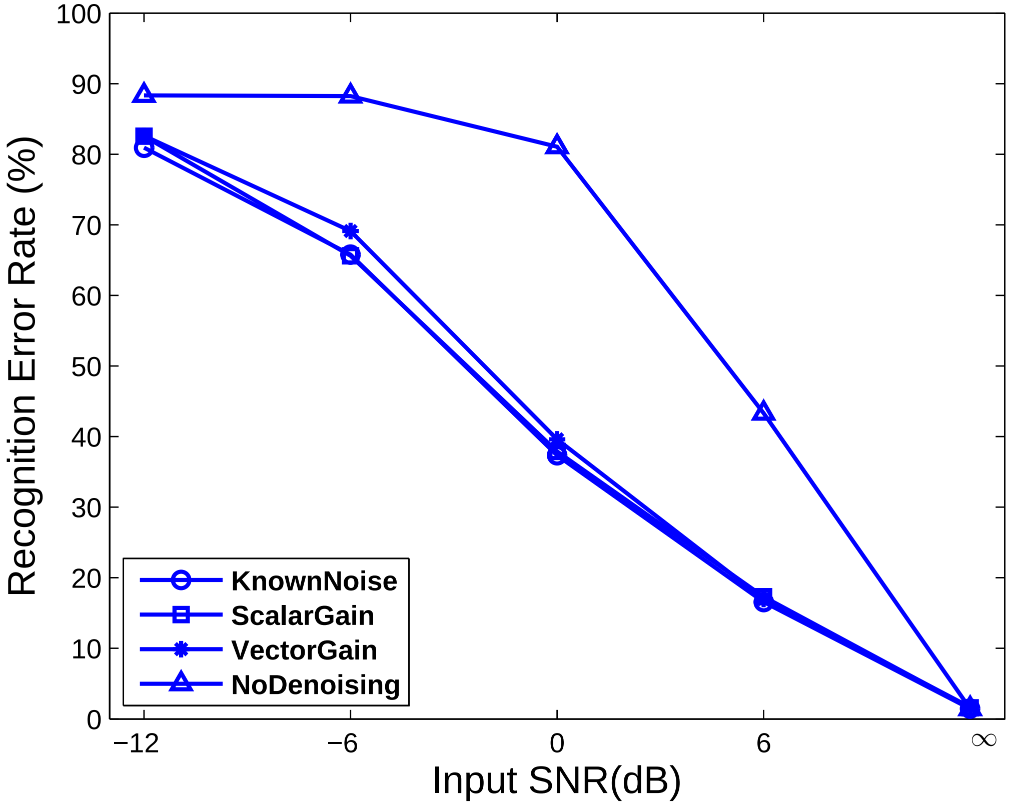

Figure 5.7: Plot of the recognition error rate of speeches enhanced by algorithms based on Gaussian approximation. The speech is corrupted by SSN. KnownNoise: known gain and noise spectrum; ScalarGain: estimated frequency-independent gain and noise spectrum; VectorGain: estimated frequency dependent gain and noise spectrum; NoDenoising: noisy speech input.

Part II

Source Separation

6

Introduction to Independent Vector Analysis Models

Independent component analysis (ICA) [27] is a well-known algorithmic method that is very successful in solving the blind source separation (BSS) problem. The underlying assumption of ICA is that the observations are linear mixtures of hidden sources that are statistically independent and thus the sources can be separated by maximizing the independence of the outputs. Various ICA algorithms have been proposed based on the source models and the characterization of independence (See [28]).

Separation of convolutive mixtures have been tackled in the frequency (or time-frequency) domain where the bin-wise mixture is approximately linear and thus simple ICA algorithms can be applied. However, because ICA is blind to permutation and the permutation disorder results in erratic signal reconstruction, permutation has to be aligned after bin-wise separation. Attempts to solve the permutation problem have depended on computing the cross-correlation [29] or direction of arrival [30] of the frequency components, or by smoothing the filter [31].

A more desirable approach is to exploit the dependency among the frequency components during the separation process. Independent vector analysis (IVA) is a recent framework whose mixture model consists of multiple layers of linear ICA mixtures as in frequency-domain BSS, where the source components have dependency across the layers to form a multivariate source, or vector, and is independent of other vectors (Figure 6.1) [32, 33]. IVA captures the intra-source dependency while it enforces the inter-source independence, and thus avoids the permutation problem [32].

ICA and IVA algorithms have been derived in maximum likelihood (ML) framework where independence is achieved via factorized source priors. Typically, the source priors are pre-specified and only the mixing matrix (or its inverse) is estimated by maximizing the likelihood. In this case, reliable statistical properties of the sources have to be known in advance. The true source models, however, are usually unknown or unavailable. Also, in IVA, it becomes even more difficult to model the sources since they are multidimensional and often, high-dimensional.

Independent factor analysis (IFA) assumes that the sources are statistically independent and each source can be modeled with a Gaussian mixture model (GMM). Contrary to those ICA algorithms where the source priors are fixed, IFA also estimates the parameters of the source priors from the data. Given the fact that GMM can model any distribution accurately with sufficiently large number of mixtures, in principle, IFA can separate sources of arbitrary distribution.

We present an adaptive IVA approach to separate convolutively mixed signals. Motivated by IFA and IVA, we model the joint probability density function (PDF) of the frequency components originating from the same source by GMM. The proposed approach employs the flexibility of source prior similar to IFA while avoiding the permutation problem. In addition, our model allows sensor noise, which has not been considered in the previous ML approaches of IVA. An efficient EM algorithm is derived to estimate the mixing matrices and the parameters of GMM. Signal estimation is achieved through Bayesian inference by computing the minimum mean square error (MMSE) of the signal posterior distribution.

6.1 Independent Vector Analysis Model

6.1.1 Acoustic Model for Convolutive Mixing

We will focus on 2 × 2 problem, i.e. two sources and two microphones. Some of our algorithms can be generalized to multiple sources/microphones. Let $x_j[t]$ be the sources j and $y_l[t]$ be the channel l, at time t. The mixing process can be accurately described by the convolution. We consider both noisy case and noiseless case here,

$$\text{Noiseless IVA:} \qquad y_l[t] = \sum_{j=1}^{2} \sum_{\tau=0}^{T-1} h_{lj}[t](\tau) x_j[t-\tau] \tag{6.1}$$

$$\text{Noisy IVA:} \qquad y_l[t] = \sum_{j=1}^{2} \sum_{\tau=0}^{T-1} h_{lj}[t](\tau) x_j[t-\tau] + n_l[t] \tag{6.2}$$

where $h_{lj}[t]$ is time domain transfer function from j^{th} source to l^{th} channel, and $n_i[t]$ is the noise. Although the Noiseless IVA is a special case of noisy IVA by setting $n_i[t] = 0$, the algorithms are quite different and treated separately.

Let $\mathbf{Y}_{kt} = (Y_{1kt}, Y_{2kt})^T$, $\mathbf{X}_{kt} = (X_{1kt}, X_{2kt})^T$, $\mathbf{N}_{kt} = (N_{1kt}, N_{2kt})^T$, be the vectors of the FFT coefficients of the mixed signals, the sources, and the sensor noise, respectively.

Applying the fast Fourier transform (FFT), the convolution becomes multiplicative

$$\text{Noiseless IVA:} \qquad \mathbf{Y}_{kt} = \mathbf{A}_k(t)\mathbf{X}_{kt} \tag{6.3}$$

$$\text{Noisy IVA:} \qquad \mathbf{Y}_{kt} = \mathbf{A}_k(t)\mathbf{X}_{kt} + \mathbf{N}_{kt} \tag{6.4}$$

where $\mathbf{A}_k(t)$ is frequency domain response function corresponding to $h_{ij}[t]$. The $\mathbf{A}_k(t)$ is called the mixing matrix because it mixes the sources. Its inverse, $\mathbf{W}_k(t) = \mathbf{A}_k^{-1}(t)$, is called unmixing matrix, which separates the mixed signals. Figure 6.1 shows the mixture model of IVA.

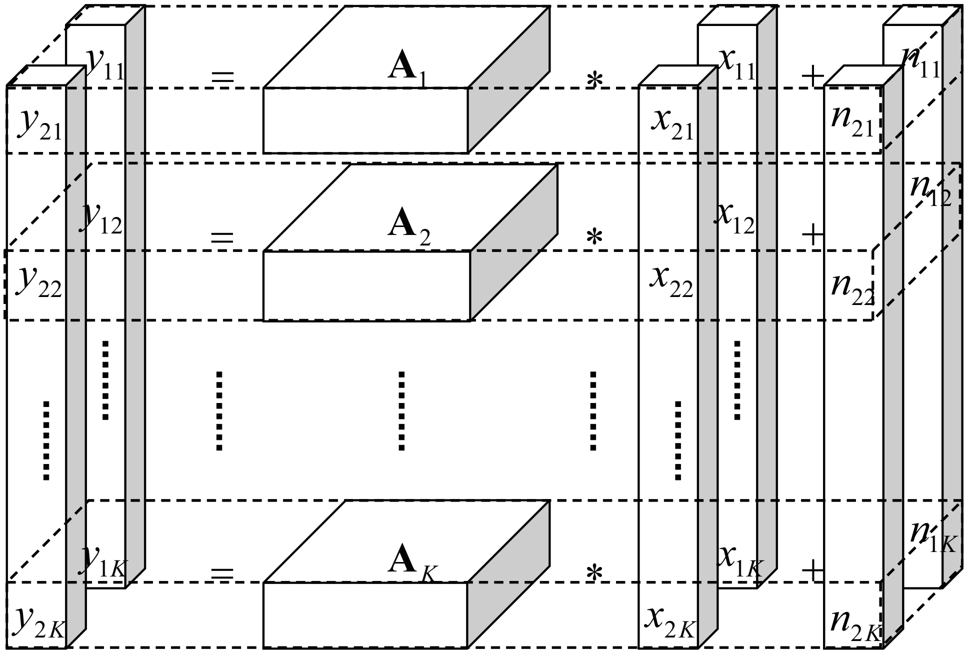

Figure 6.1: The mixture model of independent vector analysis (IVA). Dependent source components across the layers of linear mixtures are grouped into a multidimensional source, or vector.

6.1.2 Probabilistic Models for Source Priors

Because there are no true models for speech [3], a flexible model that can capture the statistical properties of each source is often desired. The probability density function (PDF) for each source is assumed to be a Gaussian mixture model (GMM) which can approximate any continuous distributions given the parameters are properly chosen [1]. Assuming the sources are statistically independent

$$\begin{aligned} p(\mathbf{X}_1, \cdots, \mathbf{X}_K) &= \prod_{j=1}^{2} p(X_{j1}, \cdots, X_{jK}) \\ p(X_{j1}, \cdots, X_{jK}) &= \sum_{s_j} p(s_j) \prod_k \mathcal{N}(X_{jk}|0, \nu_{ks_j}) \end{aligned} \tag{6.5}$$

The s_j is the state indexing the mixture components. The Gaussian PDF

$$\mathcal{N}(X_{jk}|0, \nu_{ks_j}) = \frac{\nu_{ks_j}}{\pi} e^{-\nu_{ks_j}|X_{jk}|^2} \tag{6.6}$$

is of the complex variables X_{jk}. The precision, defined as the inverse of the covariance, satisfies $1/\nu_{ks_j} = E\{|X_{jk}|^2|s_j\}$.

Consider the vector of frequency components from the same source j, $\{X_{j1}, \cdots, X_{jK}\}$. Note that although the GMM has a diagonal precision matrix for each state, the joint PDF $p(X_{j1}, \cdots, X_{jK})$ doesn't factorize, i.e. the inter-dependency among the components of a vector of the same source is captured. However, the vectors originating from different sources are independent. This model is called Independent Vector Analysis (IVA). The advantage of IVA over ICA is that the inter-frequency dependency prevents the permutation. All the frequency bins are separated in a correlated manner, rather than separately as in ICA.

For noisy IVA, we assume a Gaussian noise with precision γ,

$$p(\mathbf{Y}_k|\mathbf{X}_k) = \frac{\gamma_k^L}{\pi^L} e^{-\gamma_k|\mathbf{Y}_k - \mathbf{A}_k\mathbf{X}_k|^2} \tag{6.7}$$

where we assume the L channels have the same noise level.

The full probability is given by

$$p(\mathbf{Y}_1, \cdots, \mathbf{Y}_K, \mathbf{X}_1, \cdots, \mathbf{X}_K, \mathbf{s}) = \prod_{k=1}^{K} p(\mathbf{Y}_k|\mathbf{X}_k) \prod_{j=1}^{2} \left(\prod_k p(X_{jk}|s_j)p(s_j) \right) \tag{6.8}$$

where $\mathbf{s} = (s_1, s_2)$ is the collective state index.

The source priors can be trained in advance or estimated directly from the mixed observations. Due to the complexity of our model and the limited data, estimating the GMM for all source priors is difficult. Depending on the application, we can pre-train the priors for some sources and leave the others adaptive. The mixing matrices $\mathbf{A}_k(t)$ and the noise spectrum γ_k are estimated from the mixed observations using an EM algorithm described later. Separated signals are constructed using minimum mean square error (MMSE) estimator.

6.1.3 Related Work

The independent factor analysis (IFA) [34] also uses a GMM for each independent component whose parameters are learned from the mixed signal together with the mixing matrices. The GMM's for each source could be different, thus it is able to separate various types of sources with different statistical properties. However, IFA is derived in the time domain and can't solve sources with convolutions. IVA generalizes IFA to the vector version, in which frequency bins originating from the same source are jointly modeled by a GMM. This dependency can group the frequency components from each source together, thus preventing the permutation. The GMM has a small number of mixtures and efficiently represents the joint PDF.

Motivated by the super-Gaussian nature of speech signals, a multivariate Laplace is often used for the source priors in [32],

$$p(X^{j1}, \cdots, X_{jK}) \propto e^{-\sqrt{\|X_{j1}\|^2 + \cdots + \|X_{jK}\|^2}} \tag{6.9}$$

Note that this model captures the inter-frequency dependency. However, it assumes the identical priors for all sources and all the frequency bins have the same marginal PDF. In contrast, for acoustic signal, the PDF's of different frequency bins are likely to differ.

Our IVA model includes the advantage of both previous models. By using a GMM for the joint PDF, the inter-dependency is preserved and permutation is prevented. IVA, like IFA, uses GMM source prior and can handle noisy mixing and noiseless mixing. In contrast to multivariate Laplacian, the GMM source prior, can adapt to each source and separate various types of signals, e.g. speech and music. Further, for noisy mixing, the IVA can suppress the noise and enhance the sources quality.

Acknowledgement: This chapter contains materials in J. Hao, I. Lee, T.-W. Lee and T. Sejnowski, "Source Separation with Independent Vector Analysis", to be submitted.

7

Independent Vector Analysis for Noiseless Case

When the sensor noise is absent, the mixing process is given by Eq(6.3),

$$\mathbf{Y}_{kt} = \mathbf{A}_k(t)\mathbf{X}_{kt} \tag{7.1}$$

The parameters $\theta = \{\mathbf{A}_{kt}(t), \nu_{ks_j}, p(s_j)\}$ are estimated by maximum likelihood using the EM algorithm.

7.1 Pre-whitening and Unitary Mixing/Unmixing Matrices

The scaling of $\mathbf{X}_{kt}$ and $\mathbf{A}_k(t)$ in Eq(7.1) can not be uniquely determined by observations $\mathbf{Y}_k(t)$. Thus we can pre-whitened the observations

$$\begin{aligned} \mathbf{Q}_{k\bar{t}} &= \sum_{t=0}^{\bar{t}} \lambda^{\bar{t}-t}\mathbf{Y}_{kt}\mathbf{Y}_{kt}^{\dagger} = \lambda\mathbf{Q}_{k\bar{t}-1} + \mathbf{Y}_{k\bar{t}}\mathbf{Y}_{k\bar{t}}^{\dagger} && (7.2) \\ \mathbf{Y}_{kt} &\leftarrow \mathbf{Q}_{k\bar{t}}^{-\frac{1}{2}}\mathbf{Y}_{kt} && (7.3) \end{aligned}$$

where λ is a parameter for the online learning that will be explained later. The whitening process removes the second order correlation and $\mathbf{Y}_k$ has an identity covariance matrix, which facilitates the separation.

To be consistent with this whitening processes, we assume the priors are also white, $E\{|X_k|^2\} = 1$. The speech priors capture the high-order statistics of the sources, which enables IVA to achieve source separation.

It is more convenient to work with the demixing matrix defined as $\mathbf{W}_{kt}(t) = \mathbf{A}_{kt}^{-1}(t)$. Because of the pre-whitening process, both mixing matrix $\mathbf{A}_k(t)$ and demixing matrix $\mathbf{W}_k(t)$ are unitary, $\mathbf{I} = E\mathbf{Y}_{kt}\mathbf{Y}_{kt}^{\dagger} = E\mathbf{A}_k(t)\mathbf{X}_{kt}\mathbf{X}_{kt}^{\dagger}\mathbf{A}_k(t)^{\dagger} = \mathbf{A}_k(t)\mathbf{A}_k(t)^{\dagger}$. The inverse of unitary matrix is also unitary.

We consider the case of two sources and two sensors, and express the 2 × 2 unitary matrices $\mathbf{W}_k(t)$ using the Cayley-Klein parametrization

$$\mathbf{W}_{kT} = \begin{pmatrix} a_{kT} & b_{kT} \\ -b_{kT}^{*} & a_{kT}^{*} \end{pmatrix} \qquad \text{s.t.} \;\; a_{kT}a_{kT}^{*} + b_{kT}b_{kT}^{*} = 1 \qquad (7.4)$$

7.2 The Weighted Likelihood Function

We propose a general likelihood function

$$\begin{aligned}\mathcal{L}(\theta) &= \sum_{t=1}^{T} \lambda^{T-t} \log p(\mathbf{Y}_{1t}, \cdots, \mathbf{Y}_{Kt}) \\ &= \sum_{t=1}^{T} \lambda^{T-t} \log \left(\sum_{\mathbf{s}_t} \prod_{k=1}^{K} p(\mathbf{Y}_{kt}|\mathbf{s}_t) p(\mathbf{s}_t) \right)\end{aligned} \tag{7.5}$$

For $0 \leq \lambda \leq 1$ the past samples are weighted less and the recent samples are weighted more. The regular likelihood is obtained when $\lambda = 1$. The lower bound of $\mathcal{L}(\theta)$ is

$$\begin{aligned}\mathcal{L}(\theta) &\geq \sum_{t\mathbf{s}_t} \lambda^{T-t} q(\mathbf{s}_t) \log \frac{\prod_{k=1}^{K} p(\mathbf{Y}_{kt}|\mathbf{s}_t) p(\mathbf{s}_t)}{q(\mathbf{s}_t)} \\ &= \mathcal{F}(q, \theta)\end{aligned} \tag{7.6}$$

for distribution $q(\mathbf{s}_t)$ due to Jensen's inequality. Note that because of the absence of noise, $\mathbf{X}_{kt}$ is determined by $\mathbf{Y}_{kt}$ and is not hidden variable. We maximized $\mathcal{L}(\theta)$ using the EM algorithm, which iteratively maximized $\mathcal{F}(q, \theta)$ over $q(\mathbf{s}_t)$ (E-step) and over θ (M-step) until convergence.

7.3 The Expectation Maximization Algorithm

7.3.1 Expectation Step

For fixed θ, the $q(\mathbf{s}_t)$ that maximize $\mathcal{F}(q, \theta)$ satisfies

$$q(\mathbf{s}_t) = \frac{\prod_{k=1}^{K} p(\mathbf{Y}_{kt}|\mathbf{s}_t)p(\mathbf{s}_t)}{p(\mathbf{Y}_{1t}, \cdots, \mathbf{Y}_{Kt})} \tag{7.7}$$

Using $\mathbf{Y}_{kt} = \mathbf{W}_k(t)\mathbf{X}_{kt}$, we obtain

$$p(\mathbf{Y}_{kt}|\mathbf{s}_t) \quad = \quad p(\mathbf{X}_{kt} = \mathbf{W}_k(t)\mathbf{Y}_{kt}|\mathbf{s}_t) = \mathcal{N}(\mathbf{Y}_{kt}|0, \mathbf{\Sigma}_{k\mathbf{s}_t}) \tag{7.8}$$

The precision matrix $\mathbf{\Sigma}_{k\mathbf{s}_t}$ is given by

$$\mathbf{\Sigma}_{k\mathbf{s}_t} = \mathbf{W}_k^{\dagger}(t)\mathbf{\Phi}_{k\mathbf{s}_t}\mathbf{W}_k(t); \qquad \mathbf{\Phi}_{k\mathbf{s}_t} = \begin{pmatrix} \nu_{ks_1} & 0 \\ 0 & \nu_{ks_2} \end{pmatrix} \tag{7.9}$$

Its determinant is $\det(\mathbf{\Sigma}_{k\mathbf{s}_t}) = \nu_{ks_1}\nu_{ks_2}$, because $\mathbf{W}_k(t)$ is unitary.

We define the function $f(\mathbf{s}_t)$ as following

$$f(\mathbf{s}_t) \quad = \quad \sum_k \log p(\mathbf{Y}_{kt}|\mathbf{s}_t) + \log p(\mathbf{s}_t) \tag{7.10}$$

Use Eq(7.7), $q(\mathbf{s}_t) \propto e^{f(\mathbf{s}_t)}$

$$Z_t \quad = \quad \sum_{\mathbf{s}_t} e^{f(\mathbf{s}_t)} \tag{7.11}$$

$$q(\mathbf{s}_t) \quad = \quad \frac{1}{Z_t} e^{f(\mathbf{s}_t)} \tag{7.12}$$

7.3.2 Maximization Step

The parameters θ was estimated by maximizing the cost function $\mathcal{F}$. We consider two cases: batch algorithm and online algorithm.

Batch M-step: $\lambda = 1$

First, we consider the maximization of $\mathcal{F}$ over $\mathbf{W}_k$ under a unitary constraint. To preserve the unitarity of $\mathbf{W}_k$, using the Cayley-Klein parametrization in Eq(7.4), rewrite the precision as $\mathbf{\Phi}_{k\mathbf{s}_t} = \begin{pmatrix} \nu_{ks_1} - \nu_{ks_2} & 0 \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} \nu_{ks_2} & 0 \\ 0 & \nu_{ks_2} \end{pmatrix}$ and introduce the Lagrangian multiplier β_k. After some manipulation and ignoring the constant terms in Eq(7.6), the $\mathbf{W}_k$ maximize

$$\begin{aligned} & -\sum_{tk\mathbf{s}_t} \lambda^{T-t} q(\mathbf{s}_t) \left\{ (\nu_{ks_1} - \nu_{ks_2}) \mathbf{Y}_{kt}^\dagger \mathbf{W}_k^\dagger \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \mathbf{W}_k \mathbf{Y}_{kt} \right\} + \beta_k (a_k a_k^* + b_k b_k^* - 1) \\ = \quad & -\sum_{tk\mathbf{s}_t} \lambda^{T-t} q(\mathbf{s}_t)(\nu_{ks_1} - \nu_{ks_2}) |a_k Y_{1kt} + b_k Y_{2kt}|^2 + \beta_k (a_k a_k^* + b_k b_k^* - 1) \qquad (7.13) \end{aligned}$$

Because this is quadratic in a_k and b_k, an analytical solution exists. Setting the derivatives with respect to a_k and b_k to zero, we have

$$\mathbf{M}_{kT} \begin{pmatrix} a_k^* \\ b_k^* \end{pmatrix} = \beta_k \begin{pmatrix} a_k^* \\ b_k^* \end{pmatrix} \qquad (7.14)$$

where $\mathbf{M}_{kT}$ is defined as

$$\mathbf{M}_{kT} = \sum_{t\mathbf{s}_t} q(\mathbf{s}_t)(\nu_{ks_1} - \nu_{ks_2}) \mathbf{Y}_{kt} \mathbf{Y}_{kt}^\dagger \qquad (7.15)$$

The vector $(a_k, b_k)^\dagger$ is the eigenvector of $\mathbf{M}_{kT}$ with smaller eigenvalue. This can be shown as follows.

Use Eq(7.15), we compute the value of the objective function Eq(7.13)

$$
\begin{aligned}
& -\mathrm{Tr}\left\{ \sum_{tk\mathbf{s}_t} q(\mathbf{s}_t)(\nu_{ks_1} - \nu_{ks_2})\mathbf{Y}_{kt}\mathbf{Y}_{kt}^\dagger \mathbf{W}_k^\dagger \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \mathbf{W}_k \right\} && (7.16) \\
= & -\mathrm{Tr}\left\{ \mathbf{M}_{kT} \begin{pmatrix} a_k^* \\ b_k^* \end{pmatrix} (a_k \quad b_k) \right\} = -\beta_k && (7.17)
\end{aligned}
$$

Thus the eigenvector associated with the smaller eigenvalue gives the higher value of the cost function. Thus $(a_k, b_k)^\dagger$ is the eigenvector of $\mathbf{M}_{kT}$ with the smaller eigenvalue.

The eigenvalue problem in Eq(7.14) can be solved analytically for the 2 × 2 case. Write $\mathbf{M}_{kT}$

$$
\mathbf{M}_{kT} = \begin{pmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{pmatrix} \tag{7.18}
$$

where M_{11}, M_{22} are real and $M_{21} = M_{12}^*$, because $\mathbf{M}_{kT}$ is Hermitian. Ignored the subscript k for simplicity. Its eigenvalues are $\frac{M_{11}+M_{22}}{2} \pm \sqrt{\frac{(M_{11}-M_{22})^2}{4} + |M_{12}|^2}$ which are real and the smaller one is

$$
\beta_k = \frac{M_{11} + M_{22}}{2} - \sqrt{\frac{(M_{11} - M_{22})^2}{4} + |M_{12}|^2} \tag{7.19}
$$

and the corresponding eigenvector is

$$
\begin{pmatrix} a_k^* \\ b_k^* \end{pmatrix} = \frac{1}{\sqrt{1 + (\frac{\beta_k - M_{11}}{M_{12}})^2}} \begin{pmatrix} 1 \\ \frac{\beta_k - M_{11}}{M_{12}} \end{pmatrix} \tag{7.20}
$$

This analytical solution avoids complicated matrix calculations and greatly improves the efficiency.

Maximizing $\mathcal{F}(q, \theta)$ over $\{\nu_{ks_j}, p(s_j)\}$ is straightforward. For the precision ν_{ks_j}, we have

$$\frac{1}{\nu_{ks_j=r}} = \frac{\left[\sum_{t,\mathbf{s}_t} q(s_{jt} = r)\mathbf{W}_k \mathbf{Y}_{kt} \mathbf{Y}_{kt}^\dagger \mathbf{W}_k^\dagger\right]_{jj}}{\sum_{t,\mathbf{s}_t} q(s_{jt} = r)} \tag{7.21}$$

where $[\cdot]_{jj}$ denotes the (j, j) element of the matrix. The state probability is

$$p(s_j = r) = \frac{\sum_{t=1}^{T} q(s_{jt} = r)}{T} \tag{7.22}$$

The cost function $\mathcal{F}$ is easily accessible as a by-product of the E-step. Use Eq(7.10) and Eq(7.11), we have $Z_t = p(\mathbf{Y}_{1t}, \cdots, \mathbf{Y}_{Kt})$ and

$$\mathcal{F}(q, \theta) = \sum_t \log p(\mathbf{Y}_{1t}, \cdots, \mathbf{Y}_{Kt}) = \sum_t \log(Z_t), \tag{7.23}$$

One appealing property of the EM algorithm is that the cost function $\mathcal{F}$ always increases. This property can be used to monitor convergence. The above E-step and M-step iterate until some convergent criterion is satisfied.

Online M-step: general λ

Note the in the E-step, the posterior PDF was computed for each observation. For online EM algorithm, the E-step used the updated parameters at $\bar{t}$ when computing the posterior PDF of the sample $\mathbf{Y}_{k\bar{t}}$. Each sample was used only once and the E-step was sequential.

We now derive an M-step that updates the parameters sequentially. As in the batch algorithm, the $(a_k(t), b_k(t))^\dagger$ is the eigenvector of $\mathbf{M}_k(\bar{t})$ with the smaller eigenvalue,

$$\mathbf{M}_k(\bar{t}) \begin{pmatrix} a_k^*(\bar{t}) \\ \\ b_k^*(\bar{t}) \end{pmatrix} = \beta_k \begin{pmatrix} a_k^*(\bar{t}) \\ \\ b_k^*(\bar{t}) \end{pmatrix} \tag{7.24}$$

where $\mathbf{M}_k(\bar{t})$ is defined as

$$\begin{aligned} \mathbf{M}_k(\bar{t}) &= \sum_{t\mathbf{s}_t}^{\bar{t}} \lambda^{\bar{t}-t} q(\mathbf{s}_t)(\nu_{ks_1} - \nu_{ks_2})\mathbf{Y}_{kt}\mathbf{Y}_{kt}^\dagger & (7.25) \\ &= \lambda \mathbf{M}_k(\bar{t}-1) + \sum_{\mathbf{s}_{\bar{t}}} q(\mathbf{s}_{\bar{t}})(\nu_{ks_1} - \nu_{ks_2})\mathbf{Y}_{k\bar{t}}\mathbf{Y}_{k\bar{t}}^\dagger & (7.26) \end{aligned}$$

This matrix was computed online and recursively.

To derive the update rules for $\{\nu_{ks_j}, p(s_j)\}$, define the effective number of samples belong to state r, up to time $\bar{t}$, for source j

$$m_{jr}(\bar{t}) = \sum_{t=1}^{\bar{t}} \lambda^{\bar{t}-t} q(s_{jt} = r) = \lambda m_{jr}(\bar{t}-1) + q(s_{j\bar{t}} = r) \tag{7.27}$$

Then

$$\begin{aligned} \frac{1}{\nu_{ks_j=r}(\bar{t})} &= \frac{\left[\sum_{t,\mathbf{s}_t} \lambda^{\bar{t}-t} q(s_{jt} = r)\mathbf{W}_k(\bar{t})\mathbf{Y}_{kt}\mathbf{Y}_{kt}^\dagger \mathbf{W}_k(\bar{t})^\dagger\right]_{jj}}{\sum_{t,\mathbf{s}_t} \lambda^{\bar{t}-t} q(s_{jt} = r)} & (7.28) \\ &= \frac{1}{\nu_{ks_j=r}(\bar{t}-1)} \frac{m_{jr}(\bar{t}-1)}{m_{jr}(\bar{t})} + \frac{q(s_{j\bar{t}} = r)}{m_{jr}(\bar{t})} \left[\mathbf{W}_k(\bar{t})\mathbf{Y}_{kt}\mathbf{Y}_{kt}^\dagger \mathbf{W}_k(\bar{t})^\dagger\right]_{jj} & (7.29) \\ p(s_j = r) &= \frac{\sum_{t=1}^{\bar{t}} \lambda^{\bar{t}-t} q(s_{jt} = r)}{T} = \frac{m_{jr}(\bar{t})}{\sum_r m_{jr}(\bar{t})} & (7.30) \end{aligned}$$

7.4 Post-processing for Spectral Compensation

Because the estimated signal $\hat{\mathbf{X}}_{kt} = \mathbf{W}_k(t)\hat{\mathbf{Y}}_{kt}$ has a flat spectrum inherited from the whitening processes, it is not appropriate for signal reconstruction and the signal spectrum needs scaling corrections.

Let $\mathbf{X}^o_{kt}$ denote the original sources without whitening and $\mathbf{A}^o_{kt}$ denote the real mixing matrix. The whitened mixed signal satisfies both $\mathbf{Y}_{kt} = \mathbf{Q}_{kt}^{-1/2}\mathbf{A}^o_t k\mathbf{X}^o_{kt}$ and $\mathbf{Y}_{kt} = \mathbf{A}_{kt}\hat{\mathbf{X}}_{kt}$. Thus $\hat{\mathbf{X}}_{kt} = \mathbf{D}_{kt}\mathbf{X}^o_{kt}$, where $\mathbf{D}_{kt} = \mathbf{A}_{kt}^{-1}\mathbf{Q}_{kt}^{-1/2}\mathbf{A}^o_{kt}$. Recall that the components of $\hat{\mathbf{X}}_{kt}$ and $\mathbf{X}^o_{kt}$ are independent, $\hat{\mathbf{X}}_{kt}$ must be the scaled version of $\mathbf{X}^o_{kt}$ because the IVA prevents the permutations, i.e. the matrix $\mathbf{D}_{kt}$ is diagonal. Thus,

$$\mathrm{diag}(\mathbf{A}^o_{kt})\mathbf{X}^o_{kt} = \mathrm{diag}(\mathbf{Q}_{kt}^{1/2}\mathbf{A}_{kt}\mathbf{D}_{kt})\mathbf{X}^o_{kt} = \mathrm{diag}(\mathbf{Q}_{kt}^{1/2}\mathbf{A}_{kt})\hat{\mathbf{X}}_{kt} \tag{7.31}$$

where “diag” takes the diagonal elements of a matrix. This commutes with the diagonal matrix $\mathbf{D}_{kt}$. We term the matrix $\mathrm{diag}(\mathbf{Q}_{kt}^{1/2}\mathbf{A}_{kt})$ the spectrum compensation operator, which compensates the estimated spectrum $\hat{\mathbf{X}}_{kt}$,

$$\tilde{\mathbf{X}}_{kt} = \mathrm{diag}\left(\mathbf{Q}_{kT}^{1/2}\mathbf{W}_{kt}^{-1}\right)\hat{\mathbf{X}}_{kt} \tag{7.32}$$

Note the separated signals are filtered by $\mathrm{diag}(\mathbf{A}^o_{kt})$ and could suffer from reverberations. The estimated signals can be considered as the recorded version of the original sources. After applying the inverse FFT to $\tilde{\mathbf{X}}_{kt}$, the time domain signals can be constructed by overlap-adding, if some window is applied.

Acknowledgement: This chapter contains materials in J. Hao, I. Lee, T.-W. Lee and T. Sejnowski, “Source Separation with Independent Vector Analysis”, to be submitted.

8

Independent Vector Analysis for the Noisy Case

When the sensor noise N_{kt} exists, the mixing process is given in Eq.(6.4),

$$\mathbf{Y}_{kt} = \mathbf{A}_k \mathbf{X}_{kt} + \mathbf{N}_{kt} \tag{8.1}$$

The parameters $\theta = \{A_k, \nu_{ks_j}, p(s_j), \gamma_k\}$ are estimated by maximum likelihood using the EM algorithm. If the prior for some sources are pre-trained, their corresponding parameters $\{\nu_{ks_j}, p(s_j)\}$ are fixed.

8.1 Mixing/Unmixing Matrices are Not Unitary

Due to the existence of noise, the mixing matrices $\mathbf{A}_k$ are not unitary. We assume the noise on the channels are uncorrelated. The whitening process causes the noise to be correlated, which is difficult to model and learn. For the noisy IVA, the mixed signal are not pre-whitened and the mixing/unmixing matrices are not assumed to be unitary. In experiments, initializing $\mathbf{A}_k$ to be the whitening matrix was not optimal, because a single valuation decomposition using MATLAB gave the eigenvalues in decreasing order and introduced an initial permutation bias. Thus we simply initialized $\mathbf{A}_k$ to be the identity matrix.

8.2 The Log-Likelihood Function

The log-likelihood function is

$$\begin{aligned}
\mathcal{L}(\theta) &= \sum_{t=1}^{T} \log p(\mathbf{Y}_{1t}, \cdots, \mathbf{Y}_{Kt}) \\
&= \sum_{t} \log \left(\sum_{\mathbf{s}_t=(s_{1t},s_{2t})} \prod_{k=1}^{K} \int p(\mathbf{Y}_{kt}, \mathbf{X}_{kt}|\mathbf{s}_t) p(\mathbf{s}_t) d\mathbf{X}_{kt} \right) \\
&\geq \sum_{t\mathbf{s}_t} \int \prod_{k=1}^{K} q(\mathbf{X}_{kt}|\mathbf{s}_t) q(\mathbf{s}_t) \times \log \frac{\prod_{k=1}^{K} p(\mathbf{Y}_{kt}, \mathbf{X}_{kt}|\mathbf{s}_t) p(\mathbf{s}_t)}{\prod_{k=1}^{K} q(\mathbf{X}_{kt}|\mathbf{s}_t) q(\mathbf{s}_t)} d\mathbf{X}_{kt} \\
&= \mathcal{F}(q, \theta)
\end{aligned} \tag{8.2}$$

The inequality is due to Jensen's inequality and is valid for any PDF $q(\mathbf{X}_{kt}, \mathbf{s}_t)$. Equality $\mathcal{F} = \mathcal{L}$ occurs q equals to the posterior PDF $q(\mathbf{X}_{kt}, \mathbf{s}_t) = p(\mathbf{X}_{kt}, \mathbf{s}_t|\mathbf{Y}_{1t}, \cdots, \mathbf{Y}_{Kt})$.

8.3 Expectation Maximization Algorithm

The EM algorithm for maximizing $\mathcal{L}$ consists of iteratively maximizing $\mathcal{F}(q, \theta)$ over q (Expectation step) and θ (Maximization step), until convergence. The EM algorithm is presented below.

8.3.1 Expectation Step

For fixed θ, the $q(\mathbf{X}_{kt}|\mathbf{s}_t)$ that maximizes $\mathcal{F}(q,\theta)$ satisfies

$$q(\mathbf{X}_{kt}|\mathbf{s}_t) = p(\mathbf{X}_{kt}|\mathbf{s}_t, \mathbf{Y}_{kt}) = \frac{p(\mathbf{Y}_{kt}|\mathbf{X}_{kt})p(\mathbf{X}_{kt}|\mathbf{s}_t)}{p(\mathbf{Y}_{kt}|\mathbf{s}_t)} \tag{8.3}$$

which is a Gaussian PDF given by

$$\begin{aligned} q(\mathbf{X}_{kt}|\mathbf{s}_t) &= \mathcal{N}(\mathbf{X}_{kt}|\boldsymbol{\mu}_{kt\mathbf{s}_t}, \boldsymbol{\Phi}_{k\mathbf{s}_t}) && (8.4)\\ \boldsymbol{\mu}_{kt\mathbf{s}_t} &= \gamma_k \boldsymbol{\Phi}_{k\mathbf{s}_t}^{-1} \mathbf{A}_k^{\dagger} \mathbf{Y}_{kt} && (8.5)\\ \boldsymbol{\Phi}_{k\mathbf{s}_t} &= \gamma_k \mathbf{A}_k^{\dagger}\mathbf{A}_k + \begin{pmatrix} \nu_{ks_{1t}} & 0 \\ 0 & \nu_{ks_{2t}} \end{pmatrix} && (8.6) \end{aligned}$$

where † denotes the Hermitian (complex conjugate transpose).

To compute the optimal $q(\mathbf{s}_t)$, define the function $f(\mathbf{s}_t)$

$$\begin{aligned} f(\mathbf{s}_t) &= \sum_k \log p(\mathbf{Y}_{kt}|\mathbf{s}_t) + \log p(\mathbf{s}_t) \\ &= \sum_k \left(\log |\frac{\boldsymbol{\Sigma}_{k\mathbf{s}_t}}{\pi}| - \mathbf{Y}_{kt}^{\dagger}\boldsymbol{\Sigma}_{k\mathbf{s}_t}\mathbf{Y}_{kt} \right) + \log p(\mathbf{s}_t) && (8.7) \end{aligned}$$

where $p(\mathbf{Y}_{kt}|\mathbf{s}_t) = \int p(\mathbf{Y}_{kt}|\mathbf{X}_{kt})p(\mathbf{X}_{kt}|\mathbf{s}_t)d\mathbf{X}_{kt} = \mathcal{N}(\mathbf{Y}_{kt}|0, \boldsymbol{\Sigma}_{k\mathbf{s}_t})$ and the precision $\boldsymbol{\Sigma}_{k\mathbf{s}_t}$ is

$$\boldsymbol{\Sigma}_{k\mathbf{s}_t}^{-1} = \mathbf{A}_k \begin{pmatrix} \frac{1}{\nu_{ks_1}} & 0 \\ 0 & \frac{1}{\nu_{ks_2}} \end{pmatrix} \mathbf{A}_k^{\dagger} + \frac{1}{\gamma_k} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \tag{8.8}$$

The optimal $q(\mathbf{s}_t) \propto e^{f(\mathbf{s}_t)}$ is

$$
\begin{aligned}
q(\mathbf{s}_t) &= \frac{1}{Z_t} e^{f(\mathbf{s}_t)} && (8.9)\\
Z_t &= \sum_{\mathbf{s}_t} e^{f(\mathbf{s}_t)} && (8.10)
\end{aligned}
$$

8.3.2 Maximization Step

The M-step maximize the cost $\mathcal{F}$ over θ, which is achieved by setting the derivatives of $\mathcal{F}$ to zero.

Setting the derivative of $\mathcal{F}(q, \theta)$ with respect to $\mathbf{A}_k$ to zero, we obtain

$$
\mathbf{A}_k \mathbf{U}_k = \mathbf{V}_k \tag{8.11}
$$

where

$$
\begin{aligned}
\mathbf{U}_k &= \sum_{t=0}^{T} \sum_{\mathbf{s}_t} E^q\{\mathbf{X}_{kt} \mathbf{X}_{kt}^\dagger\} && (8.12)\\
\mathbf{V}_k &= \sum_{t=0}^{T} \sum_{\mathbf{s}_t} E^q\{\mathbf{Y}_{kt} \mathbf{X}_{kt}^\dagger\} && (8.13)
\end{aligned}
$$

The expectations are given by

$$
\begin{aligned}
E^q\{\mathbf{Y}_{kt} \mathbf{X}_{kt}^\dagger\} &= \sum_{\mathbf{s}_t} q(\mathbf{s}_t) \mathbf{Y}_{kt} \boldsymbol{\mu}_{kt\mathbf{s}_t}^\dagger && (8.14)\\
&= \sum_{\mathbf{s}_t} q(\mathbf{s}_t) \gamma_k \mathbf{Y}_{kt} \mathbf{Y}_{kt}^\dagger \mathbf{A}_k \boldsymbol{\Phi}_{k\mathbf{s}_t}^{-1} && (8.15)\\
E^q\{\mathbf{X}_{kt} \mathbf{X}_{kt}^\dagger\} &= \sum_{\mathbf{s}t} q(\mathbf{s}_t) \left(\boldsymbol{\Phi}_{k\mathbf{s}_t}^{-1} + \boldsymbol{\mu}_{kt\mathbf{s}_t} \boldsymbol{\mu}_{kt\mathbf{s}_t}^\dagger \right) && (8.16)\\
&= \sum_{\mathbf{s}t} q(\mathbf{s}_t) \left(\boldsymbol{\Phi}_{k\mathbf{s}_t}^{-1} + \gamma_k^2 \boldsymbol{\Phi}_{k\mathbf{s}_t}^{-1} \mathbf{A}_k^\dagger \mathbf{Y}_{kt} \mathbf{Y}_{kt}^\dagger \mathbf{A}_k \boldsymbol{\Phi}_{k\mathbf{s}_t}^{-1} \right) && (8.17)
\end{aligned}
$$

The bottleneck of the EM algorithm lies in the computation of $\boldsymbol{\mu}_{kt\mathbf{s}_t}$ in the expectation step, which is avoid by using Eq(8.5). Fortunately, the common terms can be computed once and $\mathbf{Y}_{kt}\mathbf{Y}_{kt}^{\dagger}$ can be computed in advance.

Similarly, we can obtain the update rules for the parameters of the source j, $\{\nu_{ks_j}, p(s_j)\}$ and the noise precision γ_k.

$$\frac{1}{\nu_{ks_j=r}} = \frac{\left[\sum_{t,\mathbf{s}_t} \delta_{rs_{jt}} q(\mathbf{s}_t)(\boldsymbol{\Phi}_{k\mathbf{s}_t}^{-1} + \boldsymbol{\mu}_{kt\mathbf{s}_t}\boldsymbol{\mu}_{kt\mathbf{s}_t}^{\dagger})\right]_{jj}}{\sum_{t,\mathbf{s}_t} \delta_{rs_{jt}} q(\mathbf{s}_t)} \tag{8.18}$$

$$p(s_j = r) = \frac{\sum_{t,\mathbf{s}_t} \delta_{rs_{jt}} q(\mathbf{s}_t)}{\sum_{t,\mathbf{s}_t} q(\mathbf{s}_t)} = \frac{1}{T}\sum_t q(s_{jt} = r) \tag{8.19}$$

$$\begin{aligned} \frac{1}{\gamma_k} &= \frac{\sum_t E^q\{(\mathbf{Y}_{kt} - \mathbf{A}_k\mathbf{X}_{kt})^{\dagger}(\mathbf{Y}_{kt} - \mathbf{A}_k\mathbf{X}_{kt})\}}{2T} \\ &= \frac{1}{2T}\sum_t \mathrm{Tr}\left[\mathbf{Y}_{kt}\mathbf{Y}_{kt}^{\dagger} - \mathbf{A}_k E^q\{\mathbf{X}_{kt}\mathbf{Y}_{kt}^{\dagger}\}\right. \\ &\quad \left. -\mathbf{A}_k^{\dagger} E^q\{\mathbf{Y}_{kt}\mathbf{X}_{kt}^{\dagger}\} + \mathbf{A}_k E^q\{\mathbf{X}_{kt}\mathbf{X}_{kt}^{\dagger}\}\mathbf{A}_k^{\dagger}\right] \end{aligned} \tag{8.20}$$

where the $\delta_{rs_{jt}}$ is the the Kronecker delta function: $\delta_{rs_{jt}} = 1$ if $s_{jt} = r$ and $\delta_{rs_{jk}} = 0$ otherwise. Essentially, the state for the second source is fixed to be r. The $[\cdot]_{jj}$ denotes the (j, j) element of the matrix. The identity $\mathbf{Y}_k^{\dagger}\mathbf{Y}_k = \mathrm{Tr}[\mathbf{Y}_k\mathbf{Y}_k^{\dagger}]$ is used. The $E^q\{\mathbf{Y}_{kt}\mathbf{X}_{kt}^{\dagger}\}$ is given by Eq(8.14), $E^q\{\mathbf{X}_{kt}\mathbf{Y}_{kt}^{\dagger}\} = E^q\{\mathbf{Y}_{kt}\mathbf{X}_{kt}^{\dagger}\}^{\dagger}$ and $E^q\{\mathbf{X}_{kt}\mathbf{X}_{kt}^{\dagger}\}$ is given by Eq(8.16).

8.4 Signal Estimation and Spectral Compensation

Unlike the noiseless case, the signal estimation is nonlinear. The MMSE estimator is

$$\hat{\mathbf{X}}_{kt} = \sum_{\mathbf{s}_t} q(\mathbf{s}_t) \boldsymbol{\mu}_{kt\mathbf{s}_t} \tag{8.21}$$

which is the average of the means $\boldsymbol{\mu}_{kt\mathbf{s}_t}$ weighted by the posterior state probability.

Because the estimated signal $\hat{\mathbf{X}}_{kt}$ had a flat spectrum and was not appropriate for signal reconstruction, it needed scaling correction. Let $\mathbf{X}^o_{kt}$ denote the original sources without whitening and $\mathbf{A}^o_{kt}$ denote the real mixing matrix. Under the small noise assumption, the mixed signal satisfies both $\mathbf{Y}_{kt} = \mathbf{A}^o_{kt}\mathbf{X}^o_{kt}$ and $\mathbf{Y}_{kt} = \mathbf{A}_{kt}\hat{\mathbf{X}}_{kt}$. Thus $\hat{\mathbf{X}}_{kt} = \mathbf{D}_{kt}\mathbf{X}^o_{kt}$, where $\mathbf{D}_{kt} = \mathbf{A}^{-1}_{kt}\mathbf{A}^o_{kt}$. Recall that the components of $\hat{\mathbf{X}}_{kt}$ and $\mathbf{X}^o_{kt}$ were independent, so $\hat{\mathbf{X}}_{kt}$ must be the scaled version of $\mathbf{X}^o_{kt}$ because the IVA prevents permutations, i.e. the matrix $\mathbf{D}_{kt}$ has to be diagonal. Thus,

$$\operatorname{diag}(\mathbf{A}^o_{kt})\mathbf{X}^o_{kt} = \operatorname{diag}(\mathbf{A}_{kt}\mathbf{D}_{kt})\mathbf{X}^o_{kt} = \operatorname{diag}(\mathbf{A}_{kt})\hat{\mathbf{X}}_{kt} \tag{8.22}$$

where “diag” takes the diagonal elements of a matrix which commutes with the diagonal matrix $\mathbf{D}_{kt}$. We term the matrix $\operatorname{diag}(\mathbf{A}_{kt})$ the spectrum compensation operator, which compensates the estimated spectrum $\hat{\mathbf{X}}_{kt}$,

$$\tilde{\mathbf{X}}_{kt} = \operatorname{diag}\left(\mathbf{A}_{kt}\right)\hat{\mathbf{X}}_{kt} \tag{8.23}$$

Note the separated signals are filtered by $\operatorname{diag}(\mathbf{A}^o_{kt})$ and could suffer from reverberations. The estimated signals can be considered as the recorded version of the original sources.

After applying the inverse FFT on $\tilde{\mathbf{X}}_{kt}$, time domain signals can be constructed by overlap-adding, if some window is applied.

8.5 On the Convergence and the Online Algorithm

The mixing process reduces to noiseless case in the limit of zero noise. Contrary to the intuition, the EM algorithm for estimating the mixing matrices will not reduces to the noiseless case. The convergence is slow when the noise level is low because the update rule for $\mathbf{A}_k$ depends on the precision of noise. It has been shown in [35], the Taylor expansion of the learning rule is

$$\mathbf{A}_k \leftarrow \mathbf{A}_k + \frac{1}{\gamma_k}\tilde{\mathbf{A}}_k + \mathcal{O}(\frac{1}{\gamma_k^2}) \tag{8.24}$$

Thus the learning rate is zero when the noise goes to zero, $\gamma_k = \infty$, essentially, $\mathbf{A}_k$ won't be updated. For this reason, the EM algorithm for noiseless IVA is derived in Chapter 7.

In principle, we can derive an online algorithm for the noisy case in a similar manner to the noiseless case. All the variables needed for the EM algorithm can be computed recursively. Thus the parameters of the source priors and the mixing matrices can be updated online. However, online algorithm for the noisy case is difficult because the speed of convergence depends on the precision of noise and as well as the learning rate λ we used in Chapter 7.

Acknowledgement: This chapter contains materials in J. Hao, I. Lee, T.-W. Lee and T. Sejnowski, “Source Separation with Independent Vector Analysis”, to be submitted.

9

Experimental Results for Source Separation with IVA

We demonstrate the performance of the proposed algorithm by using it to separate the speech from music. Music and speech have different statistical properties, which pose difficulties for IVA using identical source priors.

9.1 Dataset Description

The music signal is a disco with a singer's voice. It is about 4.5 minutes long and sampled as $8k$ Hz. The speech signal is a male voice downloaded from the University of Florida audio news. It is about 7.5 minutes long and sampled at $8k$ Hz. These two sources were mixed together, and the task was to separate them. In the noisy IVA case, a Gaussian noise at 10 dB is added to the mixtures. The goal was to suppress the noise as well as separate the signals.

Due to the flexibility of our model, it cannot learn the separation matrices and source priors from random initialization. Thus we used the first 2 minutes of signals to train the GMM as an initialization, which was done using the standard EM algorithm [1]. First, a Hanning window of 1024 samples with a 50% overlap was applied to the time domain signals. Then FFT was performed on each frame. Due to the symmetry of the FFT, only the first 512 components are kept because the rest provides no additional information. The next 30 seconds of the recordings were used to evaluate the algorithms.

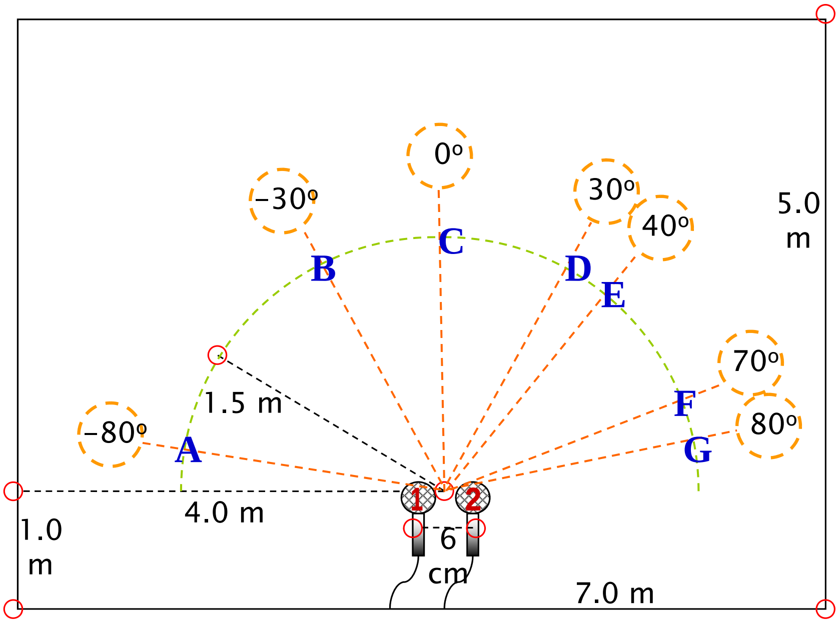

Figure 9.1: The size of the room is $7m \times 5m \times 2.75m$. The distance between two microphones is $6cm$. The sources are $1.5m$ away from the microphones. The heights of all sources and microphones are $1.5m$. The letters (A-G) indicates the position of sources.

The 30 seconds long mixed signals were obtained by simulating impulse responses of a rectangular room based on the image model technique [36, 37]. The geometry of the room is shown in 9.1. Similarly, a 1024-point Hanning window with 50% overlap was applied and the FFT was used on each frame to extract the frequency components. The mixed signals in the frequency domain were processed by the proposed algorithms, as well as the benchmark algorithms.

9.2 Benchmark: Independent Vector Analysis with Laplacian Prior

The independent vector analysis was originally proposed in [32] where the joint distribution of the frequency bins was assumed to be a multivariate Laplacian

$$p(X_{j1}, \cdots, X_{jK}) \propto e^{-\sqrt{|X_{j1}|^2 + \cdots + |X_{jK}|^2}} \tag{9.1}$$

This IVA models assumed no noise. As a results, the unmixing matrix $\mathbf{W}_k$ could be assumed to be unitary, because the mixed signals were pre-whitened and estimated by maximum likelihood, defined as

$$\begin{aligned} \mathcal{L} &= \sum_t \log p(X_{11t}, \cdots, X_{1Kt}) + \log p(X_{21t}, \cdots, X_{2Kt}) \\ &= -\sum_t \sqrt{\sum_k |X_{1kt}|^2} - \sum_t \sqrt{\sum_k |X_{2kt}|^2} + c \end{aligned} \tag{9.2}$$

where c is a constant and $\mathbf{X}_{kt} = (X_{1kt}; X_{2kt})$ is computed as $\mathbf{X}_{kt} = \mathbf{W}_k \mathbf{Y}_{kt}$.

Optimizing $\mathcal{L}$ over $\mathbf{W}_k$ was done using gradient ascent

$$\Delta \mathbf{W}_k = \eta \frac{\partial \mathcal{L}}{\mathbf{W}_k} \tag{9.3}$$

$$= \eta \sum_t \boldsymbol{\varphi}_{kt} \mathbf{Y}_{kt}^{\dagger} \tag{9.4}$$

where $\boldsymbol{\varphi}_{kt} = (\frac{X_{1kt}}{\sqrt{\sum_k |X_{1kt}|^2}}, \frac{X_{2kt}}{\sqrt{\sum_k |X_{2kt}|^2}})^\dagger$, is the derivative of the logarithm of the source prior. The natural gradient is obtained by multiplying the right hand side by $\mathbf{W}_k^\dagger \mathbf{W}_k$. The update rules become

$$\begin{aligned} \Delta \mathbf{W}_k &= \eta \sum_t \boldsymbol{\varphi}_{kt} \mathbf{X}_{kt}^\dagger \mathbf{W}_k & (9.5) \\ \mathbf{W}_k &\leftarrow \left(\mathbf{W}_k \mathbf{W}_k^\dagger \right)^{-\frac{1}{2}} \mathbf{W}_k & (9.6) \end{aligned}$$

where η is the learning rate and in all experiments we used $\eta = 5$. Eq(9.6) guarantees that $\mathbf{W}_k$ is unitary.

Because the mixed signals are pre-whiten and the scaling of the spectrum needs correction, as done in section 7.4 of Chapter 7.

9.3 Signal to Interference Ratio

The signal-to-interference ratio (SIR) for source j is defined as

$$\begin{aligned} SIR_j &= 10 \log \left(\frac{\sum_{tk} |[\hat{\mathcal{W}}_{kt} \mathbf{X}_{kt}^o]_{jj}|^2}{\sum_{tk} |[\hat{\mathcal{W}}_{kt} \mathbf{X}_{kt}^o]_{lj}|^2} \right) & (9.7) \\ \hat{\mathcal{W}}_{kt} &= \operatorname{diag}\left(\mathbf{Q}^{\frac{1}{2}} \hat{\mathbf{W}}_{kt} \right) \hat{\mathbf{W}}_{kt} \mathbf{Q}_{kt}^{-\frac{1}{2}} \mathbf{A}_{kt}^o & (9.8) \end{aligned}$$

where $\mathbf{X}_{kt}^o$ is the original source. The overall impulse response $\hat{\mathcal{W}}_{kt}$ consists of the real mixing matrix, $\mathbf{A}_{kt}^o$, obtained by performing FFT on the time domain impulse response $h_{ij}[t]$, the whitening matrix, $\mathbf{Q}_{kt}^{-\frac{1}{2}}$, the separation matrix, $\hat{\mathbf{W}}_{kt}$, estimated by the EM algorithm, and the spectrum compensation, $\operatorname{diag}\left(\mathbf{Q}^{\frac{1}{2}} \hat{\mathbf{W}}_{kt} \right)$. The numerator in Eq(9.7) takes the power of the estimated signal j, which is on the diagonal. The denominator in Eq(9.7) computes the power of the interference, which is on the off-diagonal, $l \neq j$. Note that the permutation is prevented by IVA and its correction is not needed.

9.4 Results for Noiseless IVA

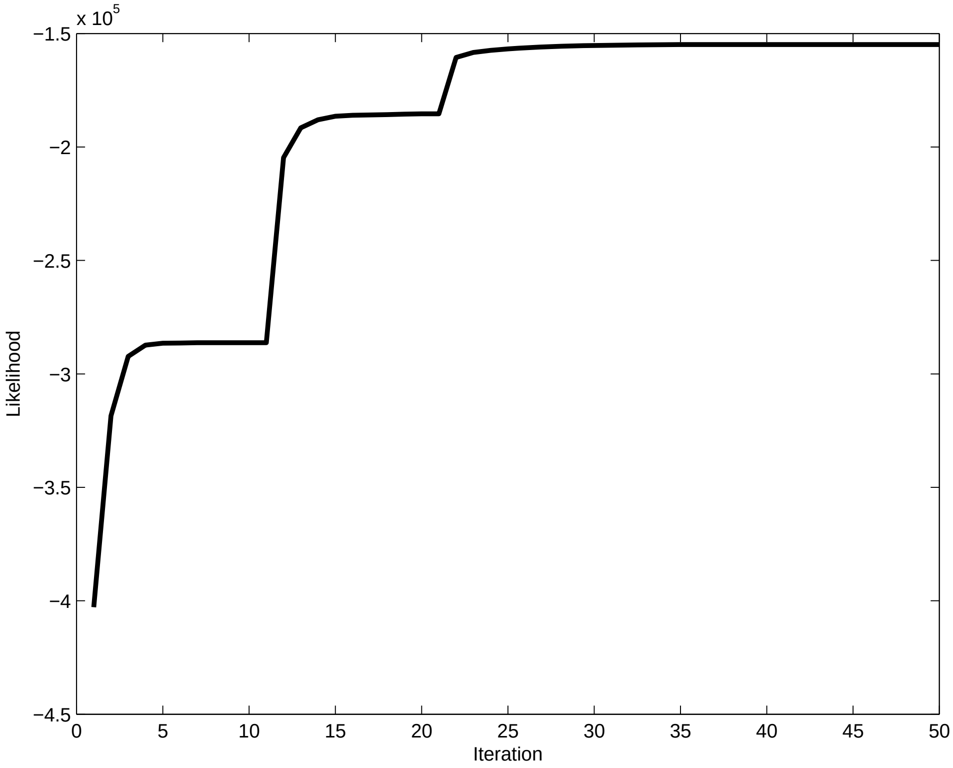

Figure 9.2: The plot of the likelihood value as a function of iterations. The EM algorithm guarantees the increase of the likelihood. When the source priors are updated, the likelihood has a jump.

The noiseless IVA optimizes the likelihood using the EM algorithm, which guarantees that the cost function increases. The mixed signal is whitened and the unmixing matrices are initialized to be identity. The number of states in the GMM was 15. The typical learning curve is shown in Figure 9.2. The two jumps correspond to the updating of the source priors. When the source priors are not updated, the curve has no jumps. The convergence was very fast, less than 50 iterations, with each iteration taking about 1

Table 9.1: Signal to interference ratio for noiseless IVA for various source locations. IVA-GMM stands for the proposed IVA using GMM as source prior, and IVA-Lap is benchmark with Laplacian source prior. IVA-GMM1 updates sources, while IVA-GMM2 with source prior fixed. The first number in each cell is the SIR of the speech, and the second number is the SIR of the music.

Source Location	**D,A**	**D,B**	**D,C**	**D,E**	**D,F**	**D,G**
IVA-Lap	11.5,18.9	11.3,13.7	11.1,12.7	10.7,15.0	11.7,18.9	12.4,19.3
IVA-GMM1	17.9,20.6	17.5,13.8	16.4,12.9	16.8,17.6	19.0,19.9	20.3,20.4
IVA-GMM2	19.7,20.7	15.1,15.7	14.0,14.0	16.8,18.6	19.6,20.2	21.4,20.8

second. In contrast, the IVA with a Laplacian prior took around 300 iterations to converge. The speech source was placed at 30°, and the music was placed at several positions. The proposed IVA-GMM improved the SIR of the speech, compared to the IVA with a Laplacian prior, IVA-Lap. Because the disco music is a mixture of many instruments and is a more Gaussian signal due to the center limit theorem, the Laplacian distribution cannot model the music accurately. As a result, the music signal leaks into the speech channel and degrades the SIR of speech. The proposed IVA use a GMM to model music, which is more accurate than Laplacian. Thus it prevented the leaking of music into speech and improved the separation by $5-8$ dB SIR. However, the improvement of the music is not significant because both models properly model the speech and prevent it from leaking into music.

9.5 Results for Online Noiseless IVA

We applied the online IVA algorithm to separate non-stationary mixtures. The speech was fixed at location $-50°$. The musical source was initially located at $-40°$ and moved to 50° at constant speed of 1° per second, then move backward at the same speed to 20°. Figure 9.3 shows the trajectory of the source: $B_1 \rightarrow B_2 \rightarrow B_3$.

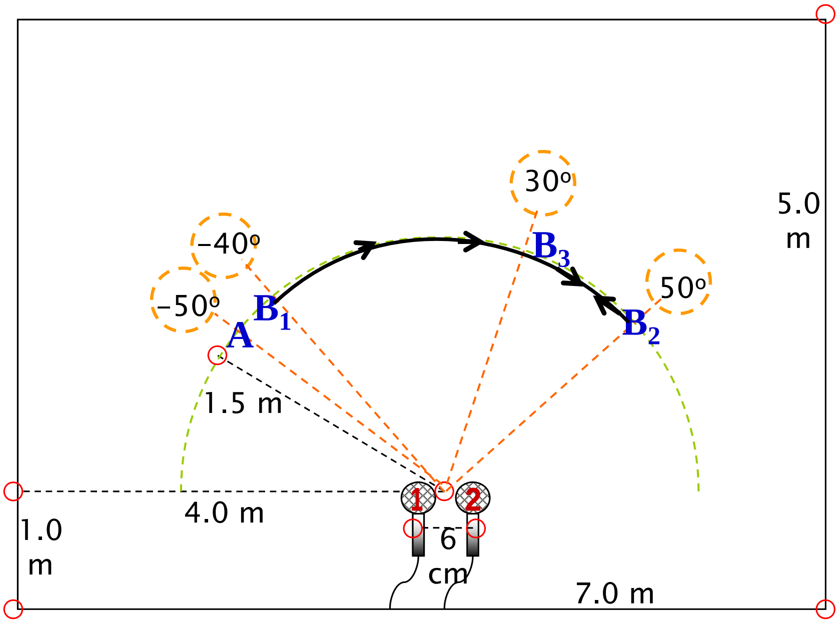

Figure 9.3: The speech is fixed at position A and the music move from B_1 to B_2 and back to B_3 at speed of 1° per second.

We set the weight $\lambda = 0.95$ in our experiment, which corresponds roughly to a 5% change in the statistics for each sample. Too small λ overfits the recent samples, while too large a value slows down the adaption. The choice of $\lambda = 0.95$ provided good adaption as well as reliable source separation. We trained a GMM with 15 states using the first 2 minutes of original signals, which was used to initialize the source priors of the online algorithm. The unmixing matrices were initialized to be identity. The number of EM iteration for the online algorithm is set to 1. Running more than one iteration was ineffective, because the state probability computed in the E-step change very little when the parameters are changed by one sample. The output SIR for speech and music are shown in Figure 9.4 and

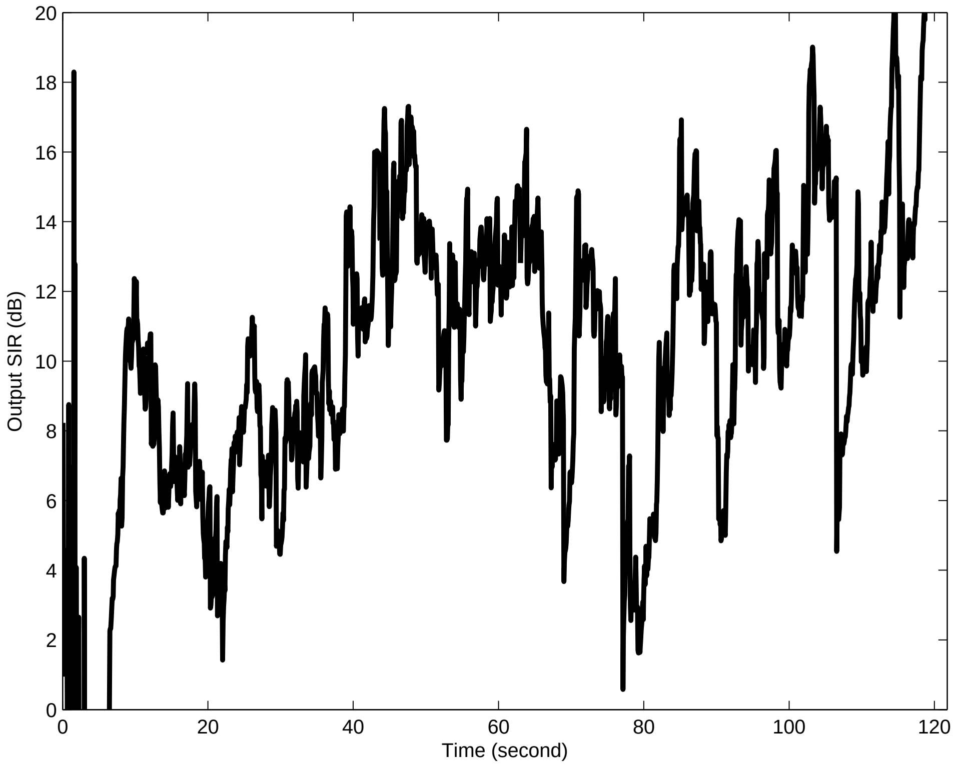

Figure 9.4: The output SIR for speech separated by online IVA algorithm. The speech is fixed at -50° and music moves from B_1 to B_2 and back to B_3 as indicated in Figure 9.3 at a speed of 1° per second.

Figure 9.5, respectively. The beginning period has low SIR values. The reason is due to the adaptation processes. The statistics for the beginning period were not estimated accurately and the separation performance was low for the first 10 seconds. The SIR improved as more samples were available and the sources were separated after 10 seconds. The SIR's for both speech and music were computed locally using the unmixing matrix for each frame and 5 seconds of original signals. The silent period of speech had very low energy, which decreased the SIR. The drops of the SIR in Figure 9.4 corresponded to the silences in the speech singles. The output SIR for the disco music was more consistent than that of speech. However, there was a drop of the SIR for both speech and music at around 80 second, when the singer's voice reached a climax in disco music which confused the IVA with the human

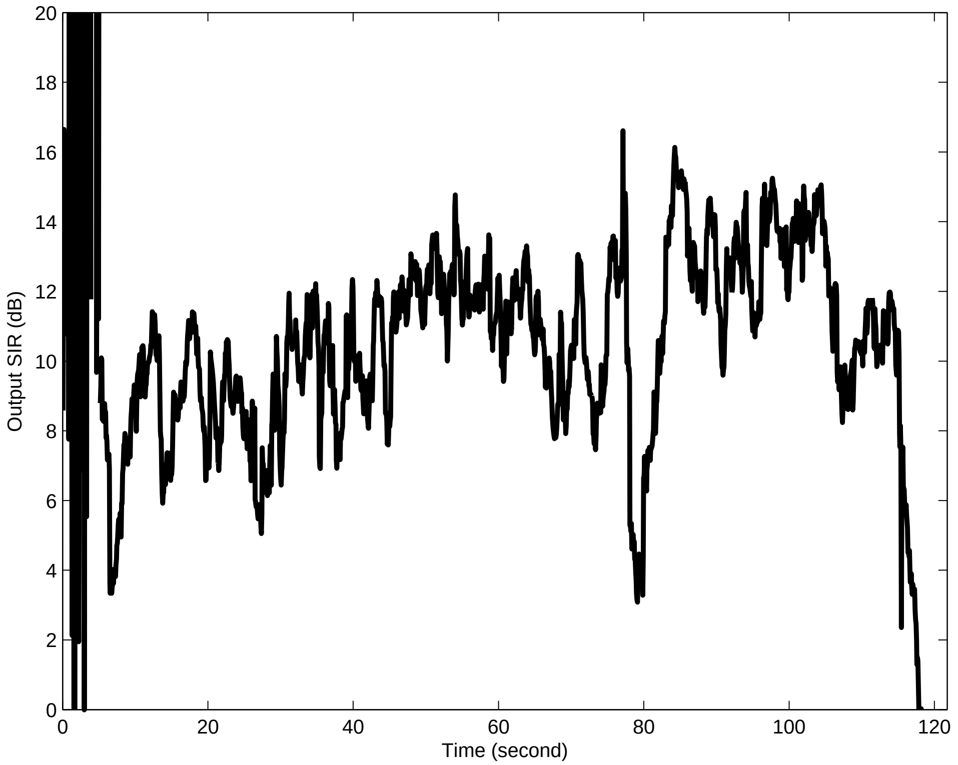

Figure 9.5: The output SIR for music separated by online IVA algorithm. The speech is fixed at $-50°$ and music moves from B_1 to B_2 and back to B_3 as indicated in Figure 9.3 at a speed of $1°$ per second.

speech, SIR's for both music and speech decreased. At the end 110 seconds, the music faded out, the SIR of speech increased and that of music decreased, dramatically. The improved SIR's demonstrated that the online IVA algorithm can track the movement of the source and separate them.

9.6 Results for Noisy IVA

For the noisy case, the signals were mixed using the image method as in the noiseless case and 10 dB white noise was added to the mixed signals. There was 15 states for the GMM. A typical learning curve is plotted in Figure 9.6. The GMM was initialized by training with the first 2 minutes of the signals, and the signals used for testing were 30

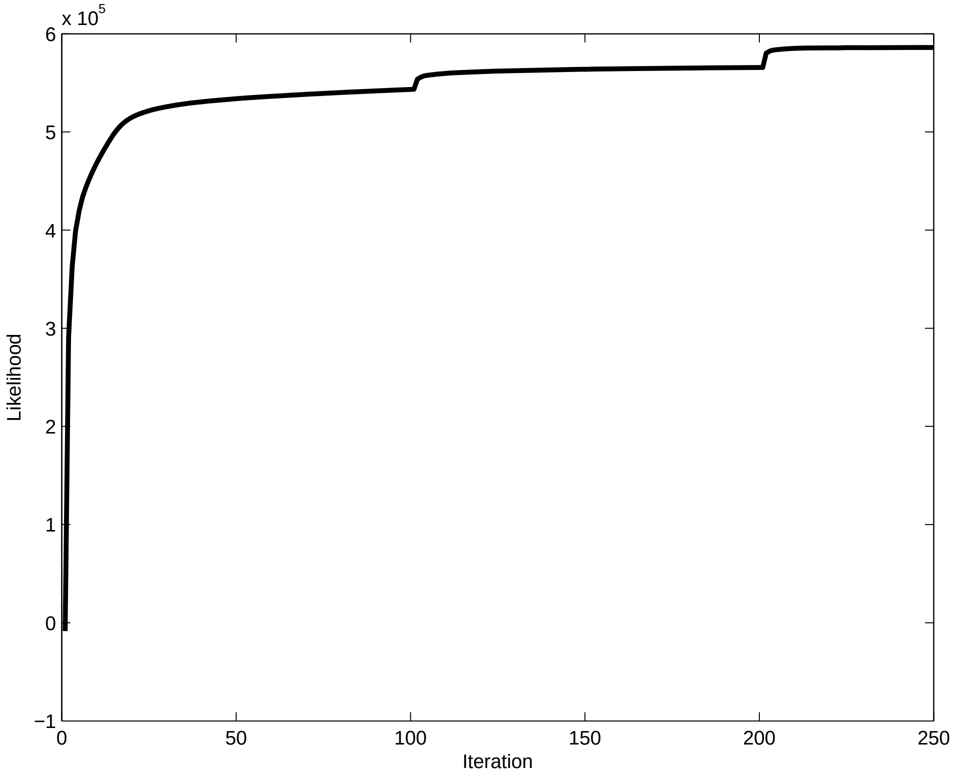

Figure 9.6: The plot of the likelihood value as a function of iterations. The EM algorithm guarantees the increase of the likelihood. When the source priors are updated, the likelihood has a jump.

Table 9.2: Signal to interference ratio for noisy IVA for various source locations. The source priors were estimated. The first number in each cell is the SIR of the speech, and the second number is the SIR of the music.

Source Location	**D,A**	**D,B**	**D,C**	**D,E**	**D,F**	**D,G**
IVA-GMM2	20.8,17.9	11.7,11.7	8.4,8.5	13.5,9.9	19.8,17.0	16.0,19.5

seconds long. There were 250 EM iterations. The convergence rate was slower than in the noiseless case because of the noise. In general it took 250 iterations, each lasting about 2 seconds. The SIR's, shown in Table 9.2 were close to those of the noiseless case for both the speech and music, which demonstrates the effectiveness of the separation. The noise was effectively reduced and the separated signals sounded noise free. Compared to the noiseless case, the separated signals contained no interference because the denoising process also removed the interference as well as the noise. However, they had more noticeable reverberation. The reason is that the unmixing matrices was not assumed to be unitary. The lack of regularization of the unmixing matrices made the algorithm more prone to local optima. Note that the source estimation of the IVA-GMM was nonlinear, since the state probability also depended on the observations. For nonlinear estimation, SIR may not provide a fair comparison. The spectrum compensation is not exact because of the noise, and as a result, the SIR's decreased a little compared to the noiseless case.

Acknowledgement: This chapter contains materials in J. Hao, I. Lee, T.-W. Lee and T. Sejnowski, "Source Separation with Independent Vector Analysis", to be submitted.

10

Conclusions

The goal of this thesis was to propose and develop probabilistic models for speech enhancement and source separation.

1) Speech enhancement algorithms were based upon approximate Bayesian estimation. These approximations made it possible to use the Gaussian Mixture Model (GMM) in the log-spectral domain for speech enhancement. The log-spectral domain Laplace method, which computes the MAP estimator for the log-spectral amplitude, was particularly successful. It offered higher SNR's and smaller recognition error rates. This is consistent with other evidence that the log-spectrum is appropriate for speech processing. The estimation of the log-spectral amplitude was a good fit to the speech recognizer and significantly improved its performance, which makes this approach valuable for recognizing the noisy speech. However, the Laplace method requires iterative optimization and increases the computational cost. Compared to the Laplace method, the Gaussian approximation with a closed-form signal estimation is more efficient and performs comparably well. The fast gain and noise spectrum adaption made this algorithm more flexible. In experiments, the

proposed algorithms demonstrated superior performance over the spectral domain models and were able to reduce the noise of spectral shape similar to that of the speech.

2) Gaussian scale mixture models (GSMM) were applied to speech signals and two approximations were developed for signal estimation: The Laplace method and variational approximation. The GSMM is a more accurate model for speech in the frequency domain and it models probabilistically the dependency between frequency components and log-spectra. The advantage of the GSMM is that it models both the FFT coefficients and the log-spectrum simultaneously. The experimental results shows the algorithms reduced word recognition error rate and improved the SNR. The FFT coefficients estimation gave higher SNR, while the log-spectra estimation produced lower word recognition error rate.

3) A novel probabilistic framework for Independent Vector Analysis (IVA) was proposed. That supported EM algorithms for the noiseless case, the noisy case and the online learning. Because each source was modeled by a different GMM, it could separate different type of signals. For the noiseless case, the derived EM algorithm was rigorous and converged rapidly. It effectively separated speech and music, two different types of signals. A general weighted likelihood cost function was used to derive an online learning algorithm for the moving sources. The parameters were updated sequentially using the single most recent sample. It required no memory of the past samples whose information had already passed through the online updating rules. This adaptation process tracked the source and separated them online, which is essential for non-stationary environments. Further, a noisy IVA algorithm was developed that could separate the signals and reduce the noise. The performance was evaluated by separation of speech and music. The improved SIR demonstrated that the algorithm could effectively separate the sources.

Bibliography

[1] C. M. Bishop, *Neural Networks for Pattern Recognition.* Oxford University Press, 1995.

[2] H. Attias, "A variational bayesian framework for graphical models," in *Proc. NIPS*, vol. 12, 2000, pp. 209–215.

[3] Y. Ephraim and I. Cohen, "Recent advancements in speech enhancement," in *The Electrical Engineering Handbook.* CRC Press, 2006.

[4] H. Attias, J. C. Platt, A. Acero, and L. Deng, "Speech denoising and dereverberation using probabilistic models," in *Proc. NIPS*, 2000, pp. 758–764.

[5] S. Gannot, D. Burshtein, and E. Weinstein, "Signal enhancement using beamforming and nonstationarity with applications to speech," *IEEE Transactions on Signal Processing*, vol. 49, no. 8, pp. 1614–1626, 2001.

[6] I. Cohen, S. Gannot, and B. Berdugo, "An integrated real-time beamforming and postfiltering system for nonstationary noise environments," *EURASIP Journal on Applied Signal Processing*, vol. 11, pp. 1064–1073, 2003.

[7] S. F. Boll, "Suppression of acoustic noise in speech using spectral subtraction," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 27, no. 2, pp. 113–120, 1979.

[8] Y. Ephraim and H. L. V. Trees, "A signal subspace approach for speech enhancement," *IEEE Transactions on Speech and Audio Processing*, vol. 3, no. 4, pp. 251–266, 1995.

[9] J. R. Hopgood and P. J. Rayner, "Single channel nonstationary stochastic signal separation using linear time-varying filters," *IEEE Transactions on Signal Processing*, vol. 51, no. 7, pp. 1739–1752, 2003.

[10] A. Czyzewski and R. Krolikowski, "Noise reduction in audio signals based on the perceptual coding approach," in *Proc. IEEE WASPAA*, 1999, pp. 147–150.

[11] J.-H. Lee, H.-J. Jung, T.-W. Lee, and S.-Y. Lee, "Speech coding and noise reduction using ica-based speech features," in *Workshop on ICA*, 2000, pp. 417–422.

[12] P. Wolfe and S. Godsill, "Towards a perceptually optimal spectral amplitude estimator for audio signal enhancement," in *Proc. ICASSP*, vol. 2, 2000, pp. 821–824.

[13] Y. Ephraim, “Statistical-model-based speech enhancement systems,” *Proceedings of the IEEE*, vol. 80, no. 10, pp. 1526–1555, 1992.

[14] 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*, vol. 32, no. 6, pp. 1109–1121, 1984.

[15] ——, “Speech enhancement using a minimum mean-square error log-spectral amplitude estimator,” *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 33, no. 2, pp. 443–445, 1985.

[16] Y. Ephraim, “A bayesian estimation approach for speech enhancement using hidden markov models,” *IEEE Transactions on Signal Processing*, vol. 40, no. 4, pp. 725–735, 1992.

[17] ——, “Gain-adapted hidden markov models for recognition of clean and noisy speech,” *IEEE Transactions on Signal Processing*, vol. 40, no. 6, pp. 1303–1316, 1992.

[18] D. Burshtein and S. Gannot, “Speech enhancement using a mixture-maximum model,” *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 6, pp. 341–351, 2002.

[19] B. Frey, T. Kristjansson, L. Deng, and A. Acero, “Learning dynamic noise models from noisy speech for robust speech recognition,” in *Proc. NIPS*, 2001.

[20] T. Kristjansson and J. Hershey, “High resolution signal reconstruction,” in *Proc. IEEE Workshop on ASRU*, 2003.

[21] A. Azevedo-Filho and R. D. Shachter, “Laplace’s method approximations for probabilistic inference in belief networks with continuous variables,” in *Proc. UAI*, 1994, pp. 28–36.

[22] I. Cohen, “Noise spectrum estimation in adverse environments: Improved minima controlled recursive averaging,” *IEEE Transactions on Speech and Audio Processing*, vol. 11, no. 5, pp. 466–475, 2003.

[23] T. M. Cover and J. A. Thomas, *Elements of Information Theory.* New York: Wiley-Interscience, 1991.

[24] R. Martin, “Speech enhancement based on minimum mean-square error estimation and supergaussian priors,” *IEEE Transactions on Speech and Audio Processing*, vol. 13, no. 5, pp. 845–856, 2005.

[25] M. Cooke and T.-W. Lee, “Speech separation challenge,” http://www.dcs.shef.ac.uk/~martin/SpeechSeparationChallenge.html.

[26] P. Wolfe, “Example of short-time spectral attenuation,” http://www.eecs.harvard.edu/~patrick/research/stsa.html.

[27] P. Comon, “Independent component analysis, a new concept?” *Neural Computation*, vol. 7, no. 6, pp. 1004–1034, 1995.

[28] A. Hyvärinen and E. Oja, *Independent Component Analysis.* John Wiley and Sons, 2002.

[29] N. Murata, S. Ikeda, and A. Ziehe, "An approach to blind source separation based on temporal structure of speech signals," *Neurocomputing*, vol. 41, pp. 1–24, 2001.

[30] S. Kurita, H. Saruwatari, S. Kajita, K. Takeda, and F. Itakura, "Evaluation of blind signal separation method using directivity pattern under reverberant conditions," in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, 2000, pp. 3140–3143.

[31] P. Smaragdis, "Blind separation of convolved mixtures in the frequency domain," *Neurocomputing*, vol. 22, pp. 21–34, 1998.

[32] T. Kim, H. T. Attias, S.-Y. Lee, and T.-W. Lee, "Blind source separation exploiting higher-order frequency dependencies," *IEEE Trans. on Speech and Audio Processing*, vol. 15, no. 1, pp. 70–79, 2007.

[33] I. Lee and T.-W. Lee, "On the assumption of spherical symmetry and sparseness for the frequency-domain speech model," *IEEE Trans. on Speech, Audio and Language Processing*, vol. 15, no. 5, 2007.

[34] H. Attias, "Independent factor analysis," *Neural Computation*, vol. 11, no. 5, 1999.

[35] K. B. Petersen, O. Winther, and L. K. Hansen, "On the slow convergence of em and vbem in low-noise linear models," *Neural Computation*, vol. 17, pp. 1921–1926, 2005.

[36] J. B. Allen and D. A. Berkley, "Image method for efficiently simulating small room acoustics," *J. Acoust. Soc. Amer.*, vol. 65, pp. 943–950, 1979.

[37] R. B. Stephens and A. E. Bate, *Acoustics and Vibrational Physics.* Edward Arnold Publishers, 1966.

[38] I. Cohen and B. Berdugo, "Noise estimation by minima controlled recursive averaging for robust speech enhancement," *IEEE Signal Processing Letters*, vol. 9, no. 1, pp. 12–15, 2002.

[39] R. McAulay and M. Malpass, "Speech enhancement using a soft-decision noise suppression filter," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 28, no. 2, pp. 137–145, 1980.

[40] R. Martin, "Noise power spectral density estimation based on optimal smoothing and minimum statistics," *IEEE Transactions on Speech and Audio Processing*, vol. 9, pp. 504–512, 2001.

[41] D. Wang and J. Lim, "The unimportance of phase in speech enhancement," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 30, no. 4, pp. 679–681, 1982.

[42] H. Attias, L. Deng, A. Acero, and J. Platt, "A new method for speech denoising and robust speech recognition using probabilistic models for clean speech and for noise," in *Proc. Eurospeech*, 2001.

[43] M. S. Brandstein, “On the use of explicit speech modeling in microphone array applications,” in *Proc. ICASSP*, 1998, pp. 3613–3616.

[44] L. Hong, J. Rosca, and R. Balan, “Independent component analysis based single channel speech enhancement,” in *Proc. ISSPIT*, 2003, pp. 522–525.

[45] C. Beaugeant and P. Scalart, “Speech enhancement using a minimum least-squares amplitude estimator,” in *Proc. IWAENC*, 2001, pp. 191–194.

[46] T. Lotter and P. Vary, “Noise reduction by maximum a posterior spectral amplitude estimation with supergaussian speech modeling,” in *Proc. IWAENC*, 2003, pp. 83–86.

[47] C. Breithaupt and R. Martin, “Mmse estimation of magnitude-squared dft coefficoents with supergaussian priors,” in *Proc. ICASSP*, 2003, pp. 848–851.

[48] J. Benesty, J. Chen, Y. Huang, and S. Doclo, “Study of the wiener filter for noise reduction,” in *Speech Enhancement*, J. Benesty, S. Makino, and J. Chen, Eds. Springer, 2005.

[49] T.-W. Lee, *Independent Component Analysis: Theory and Applications.* Boston: Kluwer Academic Publishers, 1998.

[50] D. Yellin and E. Weinstein, “Multichannel signal separation: methods and analysis,” *IEEE Trans. on Signal Processing*, vol. 44, no. 1, pp. 106–118, 1996.

[51] K. Torkkola, “Blind separation of convolved sources based on information maximization,” in *Proc. IEEE Int. Workshop on Neural Networks for Signal Processing*, 1996, pp. 423–432.

[52] T.-W. Lee, A. J. Bell, and R. Lambert, “Blind separation of convolved and delayed sources,” in *Adv. Neural Information Processing Systems*, 1997, pp. 758–764.

[53] L. Parra and C. Spence, “Convolutive blind separation of non-stationary sources,” *IEEE Trans. on Speech and Audio Processing*, vol. 8, no. 3, pp. 320–327, 2000.

[54] F. Asano, S. Ikeda, M. Ogawa, H. Asoh, and N. Kitawaki, “A combined approach of array processing and independent component analysis for blind separation of acoustic signals,” in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, 2001, pp. 2729–2732.

[55] M. Z. Ikram and D. R. Morgan, “Exploring permutation inconsistency in blind separation of signals in a reverberant environment,” in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, 2000, pp. 1041–1044.

[56] H. Sawada, R. Mukai, S. Araki, and S. Makino, “A robust and precise method for solving the permutation problem of frequency-domain blind source separation,” in *Proc. Int. Conf. on Independent Component Analysis and Blind Source Separation*, 2003, pp. 505–510.

[57] J. Anemueller and B. Kollmeier, “Amplitude modulation decorrelation for convolutive blind source separation,” in *Proc. Int. Conf. on Independent Component Analysis and Blind Source Separation*, 2000, pp. 215–220.

[58] S.-I. Amari, A. Cichocki, and H. H. Yang, “A new learning algorithm for blind signal separation,” in *Adv. Neural Information Processing Systems*, vol. 8, 1996, pp. 757–763.

[59] A. Hyvärinen and P. O. Hoyer, “Emergence of phase and shift invariant features by decomposition of natural images into independent feature subspaces,” *Neural Computation*, vol. 12, no. 7, pp. 1705–1720, 2000.

[60] A. Hyvärinen, P. O. Hoyer, and M. Inki, “Topographic independent component analysis,” *Neural Computation*, vol. 13, no. 7, pp. 1527–1558, 2001.

[61] T.-W. Lee, M. S. Lewicki, and T. Sejnowski, “ICA mixture models for unsupervised classification of non-gaussian classes and automatic context switching in blind separation,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 22, no. 10, 2000.

[62] A. J. Bell and T. J. Sejnowski, “An information maximization approach to blind separation and blind deconvolution,” *Neural Computation*, vol. 7, no. 6, pp. 1129–1159, 1995.

[63] J.-F. Cardoso, “Infomax and maximum likelihood for source separation,” *IEEE Signal Processing Letters*, vol. 4, no. 4, 1997.

[64] A. Hyvärinen and E. Oja, “A fast fixed-point algorithm for independent component analysis,” *Neural Computation*, vol. 9, no. 7, pp. 1483–1492, 1997.

[65] T.-W. Lee, M. Girolami, and T. J. Sejnowski, “Independent component analysis using an extended infomax algorithm for mixed sub-gaussian and super-gaussian sources,” *Neural Computation*, vol. 11, pp. 417–449, 1999.

[66] I. Lee, T. Kim, and T.-W. Lee, “Fast fixed-point independent vector analysis algorithms for convolutive blind source separation,” *Signal Processing*, 2007.

[67] N. Mitianoudis and M. Davies, “Audio source separation of convolutive mixtures,” *IEEE Trans. on Speech and Audio Processing*, vol. 11, no. 5, pp. 489–497, 2003.

[68] D. Obradovic and G. Deco, “Information maximization and independent comonent analysis: Is there a difference?” *Neural Computation*, vol. 10, no. 8, pp. 2085–2101, 1998.

[69] M. Z. Ikram and D. R. Morgan, “A beamforming approach to permutation alignment for multichannel frequency-domain blind speech separation,” in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, 2002, pp. 881–884.

[70] H.-J. Park and T.-W. Lee, “Modeling nonlinear dependencies in natural images using mixture of laplacian distribution,” in *Adv. Neural Information Processing Systems*, 2004.