NONLINEAR COMPENSATION AND HETEROGENEOUS DATA MODELING FOR ROBUST SPEECH RECOGNITION

A Dissertation
Presented to
The Academic Faculty

by

Yong Zhao

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Electrical and Computer Engineering

Georgia Institute of Technology
May 2013

NONLINEAR COMPENSATION AND HETEROGENEOUS DATA MODELING FOR ROBUST SPEECH RECOGNITION

Approved by:

Professor Mark A. Clements,
Committee Chair
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Biing-Hwang (Fred) Juang,
Advisor
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Chin-Hui Lee
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor David G. Taylor
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Yajun Mei
School of Industrial and Systems Engineering
Georgia Institute of Technology

Date Approved: February 2013

Dedicated to my parents:

Lifeng Che and Zhongyi Zhao

and to my family:

Yuan, Harry, and Aiden

ACKNOWLEDGEMENTS

When five years ago, I decided to chase my dream of obtaining a Ph.D. degree after working for many years in the industry, I did not imagine how hard it would be to fight my way out to the destination. Now as my dream becomes much closer to reality, I strongly apprehend that the doctorate implies more than the academic achievements. It would not have been possible for me to finish this work without support and encouragement from many people.

First of all, I would like to express my most sincere gratitude to Dr. Biing-Hwang (Fred) Juang. His visionary thought and insightful guidance walked me through difficult times. He gave me a great deal of freedom to research what fascinates me, while also holding me the highest standards of scholarship. It is my best fortune to have been working with him.

I would express my gratitude to Prof. Mark A. Clements and Prof. Chin-Hui Lee for their knowledge and experience in my interactions with them. I am grateful to Prof. David G. Taylor and Prof. Yajun Mei for serving on my dissertation committee.

I would thank my previous colleagues in Microsoft Research Asia, especially Dr. Frank K. Soong and Dr. Min Chu. I have begun to enjoy the research and development of speech technologies when I worked with Dr. Frank K. Soong. Successfully transferring text-to-speech technologies into Microsoft products with Dr. Min Chu and other colleagues gave me confidence in my ability to succeed in research.

I owe great acknowledge to Dr. Xiaodong He in Microsoft Research, Dr. Andrej Ljolje and Dr. Diamantino A. Caseiro in AT&T Labs Research, and Dr. Shinji Watanabe in Mitsubishi Electric Research Labs (MERL). The collaboration and communication with them broadened my horizon in the research area of spoken language processing. I would especially thank Dr. Jinyu Li, who gave precious advices in my research of nonlinear compensation models.

I extend my thanks to my colleagues: Chengyuan Ma, Qiang Fu, Ted S. Wada, Antonio Moreno-Danniel, Chao Weng, Sunghwan Shin, Dwi Sianto Mansjur, Wung Jason, Umair

Altaf, Mingyu Chen, and Yu Tsao, for their great support over my Ph.D. Study.

Finally, I am greatly indebted to my parents for their great love and continuous support. I want to express my deepest gratitude to my wife, Yuan, who has always been with me in this long journey.

TABLE OF CONTENTS

PART II HETEROGENEOUS DATA MODELING

LIST OF TABLES

LIST OF FIGURES

SUMMARY

The goal of robust speech recognition is to maintain satisfactory recognition accuracy under mismatched operating conditions. This dissertation addresses the robustness issue from two directions.

In the first part of the dissertation, we propose the Gauss-Newton method as a unified approach to estimating noise parameters for use in prevalent nonlinear compensation models, such as vector Taylor series (VTS), data-driven parallel model combination (DPMC), and unscented transform (UT), for noise-robust speech recognition. While iterative estimation of noise means in a generalized EM framework has been widely known, we demonstrate that such approaches are variants of the Gauss-Newton method. Furthermore, we propose a novel noise variance estimation algorithm that is consistent with the Gauss-Newton principle. The formulation of the Gauss-Newton method reduces the noise estimation problem to determining the Jacobians of the corrupted speech parameters. For sampling-based compensations, we present two methods, sample Jacobian average (SJA) and cross-covariance (XCOV), to evaluate these Jacobians.

The Gauss-Newton method is closely related to another noise estimation approach, which views the model compensation from a generative perspective, giving rise to an EM-based algorithm analogous to the ML estimation for factor analysis (EM-FA). We demonstrate a close connection between these two approaches: they belong to the family of gradient-based methods except with different convergence rates. Note that the convergence property can be crucial to the noise estimation in many applications where model compensation may have to be frequently carried out in changing noisy environments to retain desired performance.

Furthermore, several techniques are explored to further improve the nonlinear compensation approaches. To overcome the demand of the clean speech data for training acoustic models, we integrate nonlinear compensation with adaptive training. We also investigate

the fast VTS compensation to improve the noise estimation efficiency, and combine the VTS compensation with acoustic echo cancellation (AEC) to mitigate issues due to interfering background speech.

The proposed noise estimation algorithm is evaluated for various compensation models on two tasks. The first is to fit a GMM model to artificially corrupted samples, the second is to perform speech recognition on the Aurora 2 database, and the third is on a speech corpus simulating the meeting of multiple competing speakers. The significant performance improvements confirm the efficacy of the Gauss-Newton method to estimating the noise parameters of the nonlinear compensation models.

The second research work is devoted to developing more effective models to take full advantage of heterogeneous speech data, which are typically collected from thousands of speakers in various environments via different transducers. The proposed synchronous HMM, in contrast to the conventional HMMs, introduces an additional layer of substates between the HMM state and the Gaussian component variables. The substates have the capability to register long-span non-phonetic attributes, such as gender, speaker identity, and environmental condition, which are integrally called speech scenes in this study. The hierarchical modeling scheme allows an accurate description of probability distribution of speech units in different speech scenes. To address the data sparsity problem in estimating parameters of multiple speech scene sub-models, a decision-based clustering algorithm is presented to determine the set of speech scenes and to tie the substate parameters, allowing us to achieve an excellent balance between modeling accuracy and robustness. In addition, by exploiting the synchronous relationship among the speech scene sub-models, we propose the multiplex Viterbi algorithm to efficiently decode the synchronous HMM within a search space of the same size as for the standard HMM. The multiplex Viterbi can also be generalized to decode an ensemble of isomorphic HMM sets, a problem often arising in the multi-model systems. The experiments on the Aurora 2 task show that the synchronous HMMs produce a significant improvement in recognition performance over the HMM baseline at the expense of a moderate increase in the memory requirement and computational complexity.

CHAPTER 1

SCIENTIFIC GOALS

State-of-the-art speech recognition systems can achieve high recognition rates. However, their performance may suffer substantial degradation if they are operated under mismatched operating conditions. Sources of mismatch between the training and the test conditions include speaker differences, interfering noise, channel and microphone variations, and other environmental effects. The goal of robust speech recognition is to maintain satisfactory recognition accuracy under mismatched operating conditions. This dissertation addresses the robustness issue from two directions.

The first part of the dissertation is devoted to developing unified optimization approaches to estimate noise parameters of the nonlinear compensation models for noise-robust speech recognition. Structured adaptation and compensation are a broad class of approaches to combating the mismatch problems, in which some transformation structures accounting for the mismatch condition are assumed, and either the speech features or the acoustic models are adjusted in response to an estimated transformation for the test condition. Normally, the transformation structure is prescribed and parameterized by the system designer, and the values of the parameters are to be optimized according to a chosen criterion given the limited adaptation data collected in the unknown or adverse conditions.

The simplest form of the transformation is a linear one, such as the structured affine transformations used in maximum-likelihood linear regression (MLLR) [52], [25] and constrained MLLR (CMLLR) [15]. One advantage of linear transformation is its simplicity: it is capable of providing a performance gain without incurring exorbitant cost and effort. With the increase of the available adaptation data, the system can also scale up the number of linear transformations through a regression tree, resulting in a piecewise-linear transformation to cope with the non-homogeneous mismatch characteristics.

With increased sophistication, nonlinear transformations that are specifically prescribed

to tackle the additive noise and convolutional distortion have been proposed. Representative methods in this category include vector Taylor series (VTS) [63], [3], data-driven parallel model combination (DPMC) [24], and unscented transform (UT) [37].

To establish proper model compensation, the transformation parameters attributed to the additive noise and the convolutional distortion must be estimated. Unlike MLLR or CMLLR, nonlinear compensation models are parsimonious and their associated parameters bear direct physical meanings. The representational parsimony means advantageously that the compensation parameters can be estimated with only a few utterances, thus allowing rapid adaptation to changing conditions. Numerous approaches for estimating the distortion parameters have been proposed. For example, the mean cepstral vector of an utterance can be regarded as an estimate of the convolutional distortion parameter and has been used in a simple scheme like cepstral mean normalization (CMN) [7]. Also, in the presence of a voice activity detector (VAD), it is possible to estimate the additive noise parameter from non-speech frames.

Accurate estimation of all the noise parameters can be formulated in an expectation-maximization (EM) [13] fashion using the maximum-likelihood (ML) criterion. The induced complexity to the noise estimation procedure by the nonlinear compensation model is what this study will overcome. In particular, the maximization of the EM auxiliary function is a rather complex optimization problem that needs to be addressed with rigor.

In the literature, these estimation algorithms can be grouped into two categories. The first approach seeks a generalized EM algorithm to progressively improve the conventional EM auxiliary function. In [63], Moreno established this EM framework to estimate both the additive noise and the convolutional channel for the VTS compensation in the feature domain. In [54] and [57], this approach was extended to the model-domain VTS compensation with additional compensation for the dynamic features and Gaussian variances, which have been shown to further boost the recognition performance. Estimating the noise variance in the same EM fashion is difficult. In [57], Liao proposed a gradient-descent method to obtain the noise variance estimate. The main drawback of this method is that it does not guarantee increase in the auxiliary function with the gradient-based adjustment, and

thus requires a heuristic back-off step to avoid divergence. . In [54], Newton's method was presented for estimating noise variances. However, the Hessian matrix of the auxiliary function leads to a complicated computation, and it also needs to be properly regularized to be negative-definite such that the re-estimated noise variances would converge to a stationary point solution.

The noise estimation method has also been extended to the UT compensation in [55], by making use of the fact that the corrupted mean of the UT model is a weighted sum of the mismatch function over the sampled points. Furthermore, several variations of the estimation method have been proposed to allow for the integration of nonlinear compensation with uncertainty decoding [57], adaptive training [43], and other advanced noise-robust techniques.

The second approach views the compensation models from a generative perspective. This gives rise to an EM-based algorithm analogous to the ML estimation for factor analysis (EM-FA) [74], [73]. In the EM-FA algorithm [73], clean speech and noise observations are regarded as latent variables, which lead to an auxiliary function different from that specified in the Gauss-Newton method and make the M step relatively simple to solve. Though [73] did not account for the nonlinear compensation models such as VTS, it forms a foundation for the estimation methods in this category. The EM-FA method was explicitly formulated for the VTS compensation in [45], and was extended to support adaptive training [38] and the UT compensation [19].

The optimization of the nonlinear compensation models has been known to play a crucial role in high-performance robust speech recognition. However, in spite of a large number of studies on the optimization methods in various contexts of the nonlinear compensation, a couple of issues are still open and have not yet been fully addressed in the literature.

First, as for the first noise estimation method, there is a lack of a general framework for optimizing the compensation models. The noise means are optimized with the Gauss-Newton method, as will be shown in Chapter 3, whereas the noise variances are optimized with others. Also, the principle behind optimizing the VTS compensation and the sampling-based compensation models is vague, and how the method can be generalized to optimize

other compensation models has not been properly addressed.

Second, there has been little effort in the literature to compare the two ML noise estimation methods in a rigorous manner. This is at odds with the fact that the noise estimation method is often the contrasting component among some of the works, with other setups being similar. Consider adaptive training using the VTS compensation as an instance. Following a similar compensation scheme, the noise parameters have been estimated using the EM-FA method in [38], and a hybrid of Gauss-Newton and Newton's methods in [43], respectively. Different noise estimation methods have been developed by different research groups. Even though they reported experimental results on the same benchmark databases like Aurora 2 [34], what contributes to the performance difference has not yet been clearly understood.

The objective of the first research work is to construct a unified framework to optimize a range of nonlinear compensation models for robust speech recognition. We begin by considering the problem of estimating the noise parameters for the VTS compensation model. While iterative optimization of the noise means in a generalized EM framework has been widely known, we demonstrate that such approaches are variants of the Gauss-Newton method. Furthermore, we propose a novel noise variance estimation algorithm that is consistent with the Gauss-Newton principle.

The formulation of the Gauss-Newton method in the EM framework reduces the noise estimation problem to the determination of the Jacobians that relate the parameters of the corrupted speech distribution with those of the clean speech and noise distributions. For the VTS compensation, the estimation of such Jacobians is straightforward. For the sampling-based compensation, we present two methods, sample Jacobian average (SJA) and cross-covariance (XCOV), to numerically evaluate the Jacobians [6].

Moreover, we conduct a systematic comparison between the Gauss-Newton and the EM-FA methods. We provide a complete formulation for estimating the static and dynamic parameters of the compensation models, and derive the channel mean estimation in a more intuitive way than the work originally presented in [45]. Remarkably, we show that the EM-FA algorithm is a particular instance of the gradient-based method. As such, we

demonstrate a close connection between these two methods: they belong to the family of gradient-based methods except with different update directions. This connection is pervasively manifested through the re-estimation formulas for noise means and variances, and for the VTS and sampling-based compensation. Building on this relationship, we present an in-depth discussion on the advantages and limitations of the two approaches.

In addition, we illustrate how to extend the algorithms to incorporate adaptive training, where both the compensation transforms and the canonical speech model are sought by jointly maximizing the likelihood of the training data. We also investigate a fast VTS compensation method to improve the noise estimation efficiency, including estimation from non-speech areas and incremental adaptation.

Furthermore, several techniques are explored to improve the performance of the nonlinear compensation methods. To overcome the dependence of the compensation approach on the clean speech data for training clean acoustic models, we incorporate adaptive training into the nonlinear compensation.

The proposed nonlinear compensation framework is evaluated on several tasks. The first is to fit a GMM model to synthetically generated samples. The second is to perform speech recognition on the Aurora 2 noise-corrupted connected-digit database. Moreover, to assess the potential of the compensation models in more realistic situations, we collect a speech database simulating the conversation between multiple competing speakers. A series of robust speech recognition experiments are carried out on the database.

The second research work is devoted to developing more effective models to take full advantage of heterogeneous speech data and to achieve an improved recognition performance under unknown or adverse conditions. A common practice to address the speaker and environmental variabilities for the speech recognition system is to estimate the parameters of the acoustic models from speech data that cover a large variety of acoustic conditions. However, the multistyle training may not fully realize its performance potential as the conventional HMM-based acoustic models are excessively diffused by the heterogeneity of the multistyle data.

One class of approaches to achieve a more accurate representation of heterogeneous data

is to generate multiple models by dividing the training corpus into a number of homogeneous blocks, and then training an HMM set for each block. Recognition can be performed by running multiple recognizers of these models in parallel. The recognition hypothesis is obtained by either combining the decoding outputs of the multiple recognizers through majority voting in a ROVER-like paradigm [20], or choosing the one with the highest likelihood. An alternative way of combining multiple models is to preselect one model set that best matches the operating condition for recognition.

The multi-model approach is an attractive scheme to address heterogeneous data sources for speech recognition. However, a number of problems may limit their usefulness. The first problem is the data sparsity in estimating parameters of multiple models. Typically, the speech data are divided into a number of subsets for training multiple models. As the number of the models increases, there will be fewer data available for providing reliable estimation for each individual model. As a result, only simple division of speech data has been explored in large vocabulary recognition systems.

The second problem is the heavy computational load in combining multiple models during recognition. Following the classical ensemble learning theory, it is expected that the best performance should be obtained by applying the constituent models in parallel to produce a plurality of candidate hypotheses for the majority voting. Unfortunately, this introduces multiple decoding with dramatically increased computational complexity and memory requirements. Though alternative methods such as model pre-selection can alleviate this drawback, they are at the expense of compromising the recognition accuracy.

The objective of the second research work is thus to present a novel acoustic modeling framework, named synchronous HMM, which takes full advantage of the capacity of the diversified speech data and achieves an excellent balance between modeling accuracy and robustness. In contrast to the conventional HMMs, the synchronous HMM introduces an additional layer of latent variables, referred to as substates, between the HMM state and the Gaussian component variables. The substates have the capability to register long-span non-phonetic attributes, such as gender, speaker identity, and environmental condition, which are integrally called speech scenes in this study. The hierarchical modeling scheme allows

an accurate description of probability distribution of speech units in different speech scenes.

To overcome the data sparsity problem, a decision tree-based algorithm is presented to determine the set of speech scenes and to tie the substate parameters, allowing us to achieve an excellent balance between modeling accuracy and robustness. Moreover, we propose a novel multiplex Viterbi decoding algorithm that performs an effective decoding on the synchronous HMM by keeping the search space of the same size as for the standard HMM. Remarkably, the multiplex Viterbi can be generalized to decode an ensemble of isomorphic standard HMM sets, a problem often arising in multi-model systems. A major advantage of the multiplex Viterbi is that it significantly reduces the memory requirement and the computational complexity in comparison with the standard Viterbi algorithm for decoding multiple HMM sets.

One special case of the synchronous HMM is the stranded HMM, which explicitly models the dependence among the mixture components. In other words, each mixture component is assumed to depend on the previous mixture component in addition to the state that supports it.

Experimental results on the Aurora 2 database demonstrate that the synchronous HMMs achieve the lowest WER of 6.27%, 17% relative reduction over the baseline HMMs. By jointly applying the speech scenes decision tree, multiplex Viterbi, and the speech scene pruning, the decoding time of the 18-scene synchronous models is reduced to 2.0 times the HMM baseline decoding time, saving the computational cost with a factor of 9 compared with the simple multi-model approach.

This dissertation is organized as follows: Chapter 2 introduces the nonlinear compensation models and the relevant techniques, and gives a brief overview of the Gauss-Newton method. Chapter 3 presents the Gauss-Newton method as a unified approach to estimating noise parameters of the prevalent nonlinear compensation models, extends it for addressing noise adaptive training and fast VTS compensation, and discusses the implementation issues. Chapter 4 reviews the EM-FA method and demonstrates that it is a gradient-based method. Based on that, we make an in-depth comparison between the Gauss-Newton and

the EM-FA methods. Chapter 5 experimentally investigates the effectiveness of the nonlinear compensation models and the proposed noise estimation algorithms. Chapter 6 presents the experimental results on speech that is collected in a meeting scenario. Chapter 7 and Chapter 8 presents the stranded HMMs and the synchronous HMMs, respectively. Chapter 9 concludes the study of this dissertation.

NONLINEAR COMPENSATION AND HETEROGENEOUS DATA MODELING FOR ROBUST SPEECH RECOGNITION

PART I

Nonlinear Compensation

by

Yong Zhao

CHAPTER 2

BACKGROUND

Nonlinear compensation approaches typically utilize a nonlinear noise mismatch function that characterizes the joint effects of the additive and convolutional noise [2]. The representative methods in this category include vector Taylor series (VTS) [63], [3], data-driven parallel model combination (DPMC) [24], and unscented transform (UT) [37]. This chapter provides an introduction to the nonlinear compensation approaches and briefly describes relevant techniques in the literature regarding nonlinear compensation from four perspectives: the compensation model, the noise estimation method, the compensation domain, and the training speech variability. Moreover, we give a brief overview of the Gauss-Newton method [66], which lays the basis for the proposed noise estimation framework as will be discussed in the next chapter.

2.1 Noise Mismatch Function

Assume that a time-domain clean speech signal $x(t)$ is corrupted by both additive noise $n(t)$ and convolutional distortion $h(t)$. The resulting corrupted speech $y(t)$ can be expressed as

$$y(t) = x(t) * h(t) + n(t). \tag{2.1}$$

With the filterbank analysis, the magnitude spectrum of the corrupted speech can be approximated by [2]

$$|Y[l]| \approx |X[l]||H[l]| + |N[l]| \tag{2.2}$$

where $Y[l]$, $X[l]$, $H[l]$, and $N[l]$ are the spectra of $y(t)$, $x(t)$, $h(t)$, and $n(t)$ at the Mel-scale filterbank bin l, respectively. The relationship can be rewritten in the log-spectral domain as

$$\boldsymbol{y}^{\mathrm{l}} = \boldsymbol{x}^{\mathrm{l}} + \boldsymbol{h}^{\mathrm{l}} + \mathbf{log}(1 + \mathbf{exp}(\boldsymbol{n}^{\mathrm{l}} - \boldsymbol{x}^{\mathrm{l}} - \boldsymbol{h}^{\mathrm{l}})) \tag{2.3}$$

where $\boldsymbol{x}^{\mathrm{l}}$, $\boldsymbol{n}^{\mathrm{l}}$, $\boldsymbol{h}^{\mathrm{l}}$, and $\boldsymbol{y}^{\mathrm{l}}$ are L-dimensional vectors comprising the log magnitude-spectra of the corresponding filterbank outputs, e.g., $\boldsymbol{y}^{\mathrm{l}} = \left[\log|Y[1]|, \log|Y[2]|, ..., \log|Y[L]|\right]^{\mathrm{T}}$. The

bold $\mathbf{log}(\cdot)$ and $\mathbf{exp}(\cdot)$ functions indicate element-wise operations to the input vector. Usually, a speech recognizer operates on Mel-frequency cepstral coefficient (MFCC) features of the cepstral domain, and so the mismatch function accordingly takes the form

$$\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{h} + \boldsymbol{C}\,\mathbf{log}(1 + \mathbf{exp}(\boldsymbol{C}^{\dagger}(\boldsymbol{n} - \boldsymbol{x} - \boldsymbol{h}))) \equiv \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{n}, \boldsymbol{h}) \tag{2.4}$$

where $\boldsymbol{x}$, $\boldsymbol{n}$, and $\boldsymbol{h}$, and $\boldsymbol{y}$ denote the N-dimensional MFCC feature vectors of the clean speech, additive noise, channel distortion, and corrupted speech, respectively; $\boldsymbol{C}$ denotes the $N \times L$ truncated discrete cosine transform (DCT) matrix and $\boldsymbol{C}^{\dagger}$, its pseudo-inverse.

2.2 Taxonomy of Nonlinear Compensation Techniques

In the past two decades, research efforts that attempt to exploit the nonlinear noise mismatch function (2.4) for improved robust speech recognition have led to many interesting algorithms. Interestingly, a large subset of these algorithms can roughly be identified in term of the following four dimensions: the compensation domain, the compensation model, the noise estimation method, and the training speech variability, as shown in Figure 2.1. Various combinations of categories from different dimensions have been addressed in the literature. For example, in [63], the VTS compensation was performed on the feature domain with the acoustic model trained from clean speech and noise parameters estimated using the Gauss-Newton method. In this chapter, relevant works in the literature regarding nonlinear noise compensation will be introduced with respect to these dimensions.

The designated four dimensions do not represent all the variabilities for nonlinear compensation techniques. A number of aspects are not discussed here, such as alternative mismatch functions (domain-based [24] or phase-sensitive [14]), the mode of parameter estimation (batch-mode or incremental [23]), the manner to combine with other adaptation and compensation models [23], and the manner to handle non-stationary noise, just to name a few. These aspects, no doubt of great importance, may deserve more research efforts.

As we shall see shortly, many techniques for handling nonlinear models are originally proposed for speaker adaptation with linear transformations, such as adaptive training [6] and the use of regression classes. The analogy between two transformation forms indeed boosts the progress of nonlinear compensation.

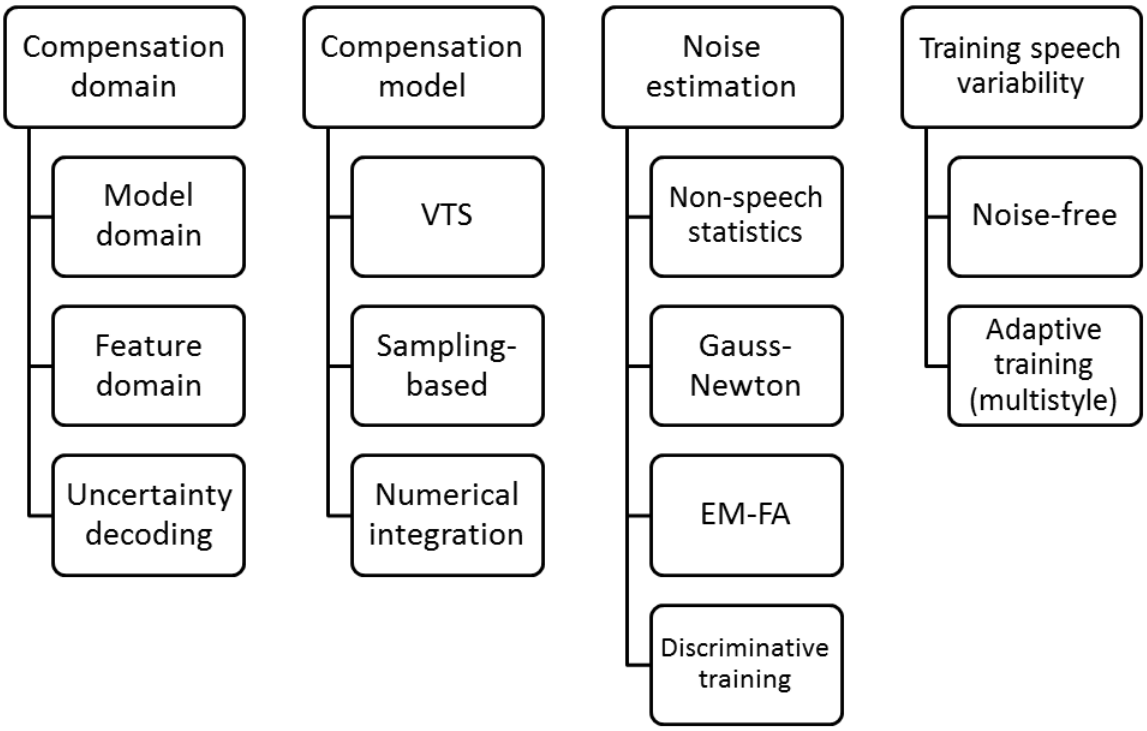


Figure 2.1: Hierarchy of noise compensation techniques that make use of the nonlinear mismatch function (2.4).

2.3 Nonlinear Noise Compensation

Due to the nonlinearity of the mismatch function, it is difficult to derive, in a closed form, the distribution of the corrupted speech feature vector. To simplify the problem, two assumptions are commonly used in the literature [28]: given each Gaussian mixture component of the clean speech models,

1. the clean speech observation and the noise observation are independent and Gaussian distributed;

2. the resulting corrupted speech observation is also Gaussian distributed.

It is clear that even with the first assumption, the distribution of the corrupted speech is still non-Gaussian. Especially at low SNRs, the corrupted speech may follow a bimodal distribution [68]. Nevertheless, the Gaussian assumption of the corrupted speech is widely employed, since it leads to dramatic savings in computational cost.

As such, the problem is reduced to obtaining the mean $\boldsymbol{\mu}_y$ and the covariance matrix $\boldsymbol{\Sigma}_y$ of the corrupted speech for each pair of the Gaussian distributions of the clean speech and the noise. A number of compensation models can be derived with additional approximation assumptions, among which we will review two such forms: VTS and sampling-based compensation.

2.3.1 Vector Taylor Series Compensation

Given the mismatch function (2.4) and the distributions of the clean speech feature $\boldsymbol{x}$, the additive noise $\boldsymbol{n}$, and the channel distortion $\boldsymbol{h}$, the distribution of the noisy observation $\boldsymbol{y}$ can be obtained via change or transformation of variables. However, the complexity of (2.4) may be prohibitive in implementation, and thus call for much simplification. The vector Taylor series (VTS) compensation method [63], [3] proposes to obtain the corrupted speech distribution by approximating the mismatch function with its first-order Taylor series expansion. Assuming that $\boldsymbol{x}$ and $\boldsymbol{n}$ are independent and Gaussian distributed as $\mathcal{N}(\boldsymbol{\mu}_x, \boldsymbol{\Sigma}_x)$ and $\mathcal{N}(\boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n)$, respectively, and $\boldsymbol{h} = \boldsymbol{\mu}_h$ is a constant, the first-order VTS approximation of the static corrupted speech $\boldsymbol{y}$ can be expressed as

$$\boldsymbol{y} \approx \boldsymbol{y}|_{\mu^{(0)}} + \boldsymbol{G}_x^{(0)}(\boldsymbol{x} - \boldsymbol{\mu}_x) + \boldsymbol{G}_n^{(0)}(\boldsymbol{n} - \boldsymbol{\mu}_n) \tag{2.5}$$

where $\boldsymbol{G}_x^{(0)}$ and $\boldsymbol{G}_n^{(0)}$ are Jacobian matrices given by

$$\boldsymbol{G}_x^{(0)} = \frac{\partial \boldsymbol{y}}{\partial \boldsymbol{x}}\Big|_{\boldsymbol{\mu}^{(0)}} = \boldsymbol{C}\boldsymbol{G}_x^{\mathrm{l},(0)}\boldsymbol{C}^{\dagger} \tag{2.6}$$

$$\boldsymbol{G}_n^{(0)} = \frac{\partial \boldsymbol{y}}{\partial \boldsymbol{n}}\Big|_{\boldsymbol{\mu}^{(0)}} = \boldsymbol{I} - \boldsymbol{G}_x^{(0)} \tag{2.7}$$

and $|_{\mu^{(0)}}$ denotes the Taylor series expansion point at $\boldsymbol{\mu}_x$, $\boldsymbol{\mu}_n$, and $\boldsymbol{\mu}_h$. The Jacobian $\boldsymbol{G}_x^{\mathrm{l},(0)}$ in the log-spectral domain is a diagonal matrix, diagonal entries of which are given by

$$\mathrm{diagv}(\boldsymbol{G}_x^{\mathrm{l},(0)}) = \frac{1}{1 + \mathbf{exp}(\boldsymbol{C}^{\dagger}(\boldsymbol{\mu}_n - \boldsymbol{\mu}_x - \boldsymbol{\mu}_h))} \tag{2.8}$$

where $\mathrm{diagv}(\cdot)$ denotes the vector containing the diagonal elements of the input matrix.

From the approximation in (2.5), the corrupted mean and variance for the VTS compensation can be obtained as [63], [3]

$$\boldsymbol{\mu}_y^{\text{vts}} = \boldsymbol{y}|_{\mu^{(0)}} = \boldsymbol{g}(\boldsymbol{\mu}_x, \boldsymbol{\mu}_n, \boldsymbol{\mu}_h) \tag{2.9}$$

$$\boldsymbol{\Sigma}_y^{\text{vts}} = \boldsymbol{G}_x^{(0)} \boldsymbol{\Sigma}_x \boldsymbol{G}_x^{(0)\text{T}} + \boldsymbol{G}_n^{(0)} \boldsymbol{\Sigma}_n \boldsymbol{G}_n^{(0)\text{T}}. \tag{2.10}$$

In practice, speech recognition systems use diagonal covariance matrices, and so we may also diagonalize the transformed covariance matrices $\boldsymbol{\Sigma}_y^{\text{vts}}$ to retain the same form.

For the dynamic portion of the MFCC feature, the following compensation formulas can be obtained using the continuous-time approximation [33]:

$$\boldsymbol{\mu}_{\Delta y}^{\text{vts}} = \boldsymbol{G}_x^{(0)} \boldsymbol{\mu}_{\Delta x} \tag{2.11}$$

$$\boldsymbol{\Sigma}_{\Delta y}^{\text{vts}} = \boldsymbol{G}_x^{(0)} \boldsymbol{\Sigma}_{\Delta x} \boldsymbol{G}_x^{(0)\text{T}} + \boldsymbol{G}_n^{(0)} \boldsymbol{\Sigma}_{\Delta n} \boldsymbol{G}_n^{(0)\text{T}}. \tag{2.12}$$

The transformation of the delta-delta parameter takes a similar form as

$$\boldsymbol{\mu}_{\Delta^2 y}^{\text{vts}} = \boldsymbol{G}_x^{(0)} \boldsymbol{\mu}_{\Delta^2 x} \tag{2.13}$$

$$\boldsymbol{\Sigma}_{\Delta^2 y}^{\text{vts}} = \boldsymbol{G}_x^{(0)} \boldsymbol{\Sigma}_{\Delta^2 x} \boldsymbol{G}_x^{(0)\text{T}} + \boldsymbol{G}_n^{(0)} \boldsymbol{\Sigma}_{\Delta^2 n} \boldsymbol{G}_n^{(0)\text{T}}. \tag{2.14}$$

2.3.2 Sampling-Based Compensation

One limitation of the VTS compensation is that it ignores the higher-order effect of the nonlinear mismatch function. Higher-order Taylor expansions [83], [97], [17] may be used to alleviate the problem, but will lead to an unduly complicated model with much increased number of adaptation parameters. One alternative approach is to use sampling-based compensation methods. The sampling-based method randomly draws samples from the clean speech and noise distributions, simulates the corrupted speech observations through the governing mismatch function (2.4), and then estimates the distribution of the corrupted speech using the sample mean and covariance. Let $\boldsymbol{y}^{(m)}$ be the corrupted speech observation corresponding to the mth sample pair $\{\boldsymbol{x}^{(m)}, \boldsymbol{n}^{(m)}\}$. We have the distribution of the

corrupted speech with parameters

$$\boldsymbol{\mu}_{\boldsymbol{y}}^{\text{smp}} = \frac{1}{M}\sum_{m=1}^{M} \boldsymbol{y}^{(m)} \tag{2.15}$$

$$\boldsymbol{\Sigma}_{y}^{\text{smp}} = \frac{1}{M}\sum_{m=1}^{M} (\boldsymbol{y}^{(m)} - \boldsymbol{\mu}_{y}^{\text{smp}})(\boldsymbol{y}^{(m)} - \boldsymbol{\mu}_{y}^{\text{smp}})^{\text{T}}. \tag{2.16}$$

An instance of the sampling-based compensation is data-driven parallel model combination (DPMC) [24], where the samples are drawn using the Monte Carlo sampling technique. The advantage of DPMC is that as the number of sampled observations increases, the noise-compensated models converge to the assumed distribution asymptotically. However, DPMC is computationally prohibitive; normally, 25–1000 sample points need to be generated per Gaussian component.

Another sampling approach is to use unscented transform (UT) [42], [37]. UT draws a limited number of deterministically chosen samples, called sigma points, to approximate the statistics of the transformed distribution. The advantage of UT is that it can achieve an approximation accuracy at least up to the second-order Taylor series expansion of the nonlinear function, with a moderate increase in computational cost over the VTS approach.

Let $\boldsymbol{z}$ denote a $2N$-dimensional vector by joining the clean speech feature and noise feature, $\boldsymbol{z} = [\boldsymbol{x}^{\text{T}}, \boldsymbol{n}^{\text{T}}]^{\text{T}}$. The following $4N$ sigma points are chosen:

$$\boldsymbol{z}^{(m)} = \begin{cases} \boldsymbol{\mu}_z + (\sqrt{2N\boldsymbol{\Sigma}_z})_m, & \text{if } 1 \le m \le 2N \\ \boldsymbol{\mu}_z - (\sqrt{2N\boldsymbol{\Sigma}_z})_{m-2N}, & \text{if } 2N < m \le 4N \end{cases} \tag{2.17}$$

where $(\sqrt{\boldsymbol{\Sigma}})_m$ indicates the mth column of the square root matrix of $\boldsymbol{\Sigma}$.

It is not straightforward to extend the sampling-based methods to the dynamic parameters in a principled way, though possible [46], [88]. In this work, we retain the continuous-time approximation used in the VTS compensation for dynamic features (2.11) and (2.12), except that the Jacobians are replaced with those for the sampling-based models, which will be described in Section 3.4.

In addition to the above sampling-based models, there are other compensation models that aim to boost the approximation accuracy beyond the VTS assumption. For example,

instead of the first-order VTS expansion, the second-order or higher-order Taylor expansion can be applied to the mismatch function, from which the distribution of the corrupted speech is derived [83], [97], [17]. Alternatively, corrupted speech statistics can be estimated using numerical integration techniques [29], [40], [4]. One advantage of the numerical integration is that it may lift the Gaussian assumption of the corrupted speech model and improve the parameter estimates of the corrupted speech. However, this method is usually computationally intensive.

2.4 Noise Parameter Estimation

One main issue in this compensation procedure is that the noise parameters are often not known and need to be estimated from the input utterances. Accurate estimation of the noise parameters has been known to play a crucial role in high-performance robust speech recognition. A simple solution is to estimate the additive noise parameters using the statistics of the non-speech frames in an utterance [3], [30], [37]. This method however requires a reliable voice activity detector (VAD) and can not estimate the convolutional distortion.

A better approach to estimating the complete set of noise parameters is often formulated in an expectation-maximization (EM) [13] framework using the maximum-likelihood (ML) criterion. Nevertheless, the nonlinearity of the compensation model induces a remarkable complexity to the noise estimation procedure, which will be the focus of the first part of this thesis. In the literature, these ML estimation algorithms can be roughly classified into two categories.

The first approach is to directly differentiate the conventional EM auxiliary function and iteratively approximate the root of the resulting nonlinear derivative function. In [63], Moreno established this generalized EM framework to estimate both the additive noise and the convolutional channel for the VTS compensation in the feature domain. In [54] and [57], this approach was extended to the model-domain VTS compensation with additional compensation for the dynamic features and Gaussian variances, which have been shown to further boost the recognition performance. Estimating the noise variance in the same

EM fashion is difficult. In [57], Liao proposed a gradient-descent method to obtain a noise variance estimate. The main drawback of this method is that it does not guarantee increase in the auxiliary function with the gradient-based adjustment, and thus requires a heuristic back-off step to avoid divergence.. In [54], Newton's method was presented for estimating noise variances. However, the Hessian matrix of the auxiliary function leads to a complicated computation, and it also needs to be properly regularized to be negative-definite such that the re-estimated noise variances would converge to a stationary point solution.

The noise estimation method has been extended for the UT compensation model in [55], by thinking that the UT model utilizes a mismatch function that is a weighted sum of the original mismatch function (2.4) over the sample points. Also, several variations of the estimation method have been proposed to allow for the integration of the nonlinear compensation with uncertainty decoding [57], adaptive training [43], and other advanced noise-robust techniques.

In Chapter 3, we will demonstrate that the above iterative methods for estimating noise means are variants of the Gauss-Newton method, and we will propose a novel approach for the estimation of noise variances that is consistent with the Gauss-Newton principle. This leads to a unified Gauss-Newton approach for optimizing various nonlinear compensation models.

The second approach views the model compensation from a generative perspective. This gives rise to an EM-based algorithm analogous to the ML estimation for factor analysis (EM-FA) [74], [73]. In the EM-FA algorithm [73], clean speech and noise observations are regarded as latent variables, which lead to an auxiliary function different from that specified in the Gauss-Newton method and make the M step relatively simple to solve. Though [73] did not account for the nonlinear compensation models such as VTS, it forms a foundation for the estimation methods in this category. The EM-FA method was explicitly formulated for the VTS compensation in [45], and was extended to support adaptive training [38] and the UT compensation [19]. Chapter 4 will discuss in detail the EM-FA method.

There has been interest in estimating the noise parameters using discriminative training methods. Other than finding the ML estimate, discriminative training aims to explicitly

minimize objective functions that are more closely related to the recognition error rate [9], [41], [69]. Most of state-of-the-art speech recognition systems have been applied with discriminative training to boost the recognition performance. In [22], minimum phone error (MPE) training [69] has been used to refine the canonical speech model in VTS adaptive training. Estimating the noise parameters under the discriminative criteria is still an open problem, as discriminative training requires proper supervision (i.e., availability of the data labels) in training, a need that cannot be easily satisfied in unsupervised adaptation [92].

2.5 Compensation Domain

One common way to categorize robust speech recognition techniques is bound on the domain in which the compensation is performed. Compensation methods applied in the model domain modify the parameters of the acoustic models to match the speech in the noisy test environment. The model-domain compensation was proposed in [3] for the VTS model, in [30] for the DPMC model, and in [37] for the UT model, respectively. The model-domain methods usually achieve high recognition accuracy, but are computationally expensive. Most of the work in this thesis is carried out in the model domain.

By contrast, the feature-domain approaches attempt to clean and adjust the incoming speech features such that the cleaned features resemble the original training environment. To make use of the compensation models, a separate small-sized speech model is embedded in the front-end of the recognition system. Since these methods do not rewrite the acoustic models, they are more efficient to implement. If the clean speech $\boldsymbol{x}$ and the corrupted speech $\boldsymbol{y}$ are jointly Gaussian distributed, the minimum mean square error (MMSE) estimate of the clean speech given $\boldsymbol{y}$ is given by

$$\boldsymbol{\mu}_{x|y} = \boldsymbol{\mu}_x + \boldsymbol{\Sigma}_{xy}\boldsymbol{\Sigma}_y^{-1}(\boldsymbol{y} - \boldsymbol{\mu}_y). \tag{2.18}$$

The feature-domain compensation with the VTS model has been proposed in [63] that uses a GMM frond-end speech model. It was extended to have an ergodic HMM in the front-end, called model-based feature enhancement (MBFE) [84][1]. In [79], the VTS model was

[1]The term "model-based" may cause confusion in the literature of robust speech recognition. In the context of feature enhancement or compensation, it refers to the use of an embedded front-end speech model

replaced with the UT model for improved modeling accuracy. The compensation models in the feature domain were often optimized using the Gauss-Newton method [63], [84].

The feature-domain compensation techniques are less robust than the model-domain techniques, as the errors associated with the front-end estimation of the clean speech are irreversible and will propagate to the recognizer. Recently, uncertainty decoding (UD) [16], [57] has been proposed to help alleviate this problem. The estimation uncertainty at the front end is passed on as variance biases to augment the acoustic model variances. In other words, the recognizer takes into account the posterior distribution, rather than as a point estimate, of the clean speech. UD with the VTS model was presented in [99] and [57]. In the model-based UD [57], the posterior distributions are associated with the regression classes of the back-end acoustic models. In this regard, the UD can be viewed as a model-domain compensation technique with the transforms shared over the Gaussian components within each regression class, similar to CMLLR. Newton's method was used in [57] for noise estimation. To simplify the calculation of the Hessian matrix, the optimization was performed on separate feature dimensions by diagonalizing the Jacobian matrices. In principle, the Gauss-Newton and the EM-FA algorithms can be modified to estimate the model parameters under the UD scheme.

2.6 Adaptive Training

One drawback with standard compensation approaches is that it assumes the availability of the clean speech data for training the acoustic models. If such data do not exist, the performance will be impaired. This restriction can be eliminated with the use of adaptive training [6], [38], [43], [57], which incorporates noise compensation during training. A set of the compensation transforms are estimated in response to different environmental conditions of the training data. The speech model is then estimated based on these transforms, leading to a canonical model that is expected to represent the intrinsic variability of the speech. While originally developed to remove speaker variations from the acoustic models [6], adaptive training has been successfully applied to the VTS compensation in [38], [43].

in the feature domain. In other cases, it indicates the use of the standard acoustic models, also known as in the model domain. Its meaning in MBFE falls into the former case.

These two works differ mainly in the choice of the optimization approach. The EM-FA method was used in [38], and a hybrid of Gauss-Newton and Newton's methods was used in [43]. Adaptive training with the UD-based VTS compensation has been proposed in [57].

2.7 Gauss-Newton Method

This section gives a brief introduction to the general Gauss-Newton method [66], which plays a central role in the proposed approach for optimizing the aforementioned compensation models. The Gauss-Newton method is a widely used iterative optimization technique for solving nonlinear least squares problems. Consider N data points, $(x_1, y_1), \ldots, (x_N, y_N)$, and a nonlinear model $f(x, \boldsymbol{\theta})$ prescribed with an unknown parameter vector $\boldsymbol{\theta}$. We wish to find the model parameter $\boldsymbol{\theta}$ such that the model fits the given data in the sense of minimum sum of square errors

$$\psi(\boldsymbol{\theta}) = \frac{1}{2}\sum_{i=1}^{N}(y_i - f(x_i, \boldsymbol{\theta}))^2 = \frac{1}{2}\sum_{i=1}^{N} r_i(\boldsymbol{\theta})^2 \tag{2.19}$$

where we define the ith residual as $r_i(\boldsymbol{\theta}) = y_i - f(x_i, \boldsymbol{\theta})$.

The error function can be minimized by Newton's method. Recursively, given an existing estimate $\boldsymbol{\theta}$, the new estimate is updated as follows:

$$\hat{\boldsymbol{\theta}} = \boldsymbol{\theta} - \Big(\frac{\partial^2 \psi}{\partial \boldsymbol{\theta}^2}\Big)^{-1} \frac{\partial \psi}{\partial \boldsymbol{\theta}} \tag{2.20}$$

where the gradient and Hessian of $\psi(\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$ are given by

$$\frac{\partial \psi}{\partial \boldsymbol{\theta}} = \sum_{i=1}^{N} r_i \frac{\partial r_i}{\partial \boldsymbol{\theta}} \tag{2.21}$$

$$\frac{\partial^2 \psi}{\partial \boldsymbol{\theta}^2} = \sum_{i=1}^{N} \frac{\partial r_i}{\partial \boldsymbol{\theta}} \Big(\frac{\partial r_i}{\partial \boldsymbol{\theta}}\Big)^{\mathrm{T}} + \sum_{i=1}^{N} r_i \frac{\partial^2 r_i}{\partial \boldsymbol{\theta}^2} \tag{2.22}$$

where $\frac{\partial r_i}{\partial \boldsymbol{\theta}}$ and $\frac{\partial^2 r_i}{\partial \boldsymbol{\theta}^2}$ denote the first and second derivatives of the residual $r_i(\boldsymbol{\theta})$ at the current estimate $\boldsymbol{\theta}$, respectively. The problem with Newton's method is that the second summation term in (2.22) involves the second derivative $\frac{\partial^2 r_i}{\partial \boldsymbol{\theta}^2}$, which is usually either expensive or impractical to calculate.

Unlike Newton's method, the Gauss-Newton method uses an approximated Hessian

matrix by ignoring the second term, yielding the following update formula:

$$\hat{\boldsymbol{\theta}} = \boldsymbol{\theta} - \Big[\sum_{i=1}^{N} \frac{\partial r_i}{\partial \boldsymbol{\theta}} \big(\frac{\partial r_i}{\partial \boldsymbol{\theta}}\big)^{\mathrm{T}}\Big]^{-1} \sum_{i=1}^{N} r_i(\boldsymbol{\theta}) \frac{\partial r_i}{\partial \boldsymbol{\theta}}. \tag{2.23}$$

This approximation gives a number of advantages over the plain Newton's method. First, the Gauss-Newton method saves the cumbersome calculation of the second-order derivatives. Second, for most applications, the first term in (2.22) dominates over the second term, so that the Hessian approximation is reliable and the Gauss-Newton method achieves an approximately quadratic convergence rate, similar to that of Newton's method. Third, the Gauss-Newton method ensures a descent direction whenever the existing estimate is not a stationary point. To see this, we note that the approximated Hessian is positive-semidefinite, and thus the inner product between the gradient and the update direction is nonnegative. This is also an important advantage of the Gauss-Newton method over Newton's method. Since the Hessian matrix of the general Newton's method is indefinite, additional efforts to regularize the Hessian must be taken to prevent divergence.

The Gauss-Newton method can be alternatively motivated by linearizing the model $f(x, \boldsymbol{\theta})$ around the existing estimate $\boldsymbol{\theta}$

$$f(x_i, \hat{\boldsymbol{\theta}}) \approx f(x_i, \boldsymbol{\theta}) + \big(\frac{\partial f(x_i, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\big)^{\mathrm{T}} (\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}). \tag{2.24}$$

Substituting (2.24) into the error function (2.19), noting that $\frac{\partial f(x_i, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = -\frac{\partial r_i}{\partial \boldsymbol{\theta}}$, results in a linear least squares problem

$$\min_{\Delta\boldsymbol{\theta}} \frac{1}{2} \sum_{i=1}^{N} \left(r_i + \big(\frac{\partial r_i}{\partial \boldsymbol{\theta}}\big)^{\mathrm{T}} \Delta\boldsymbol{\theta} \right)^2 \tag{2.25}$$

where the update direction $\Delta\boldsymbol{\theta} = \hat{\boldsymbol{\theta}} - \boldsymbol{\theta}$. The optimum of this intermediate problem coincides with (2.23).

Interested readers are referred to [66] for a detailed account of the Gauss-Newton method about the convergence performance, numerical stability and extensions.

2.8 Summary

This chapter describes the nonlinear compensation methods for robust speech recognition. A nonlinear noise mismatch function is introduced that characterizes the effects of the clean

speech corrupted by additive noise and convolutional channel distortion. Various compensation models that make use of the mismatch function to predict the distributions of the corrupted speech are given. Moreover, relevant techniques regarding nonlinear compensation are discussed from three perspectives: the noise estimation method, the compensation domain, and the training speech variability. By drawing a clear picture of various compensation methods that have been proposed in the literature, we maintain important insights that help to guide the research in this area. Finally, we give a brief overview of the general Gauss-Newton method, which lays the basis for the proposed noise estimation framework.

CHAPTER 3

GAUSS-NEWTON METHOD FOR NOISE ESTIMATION

The compensation models introduced in the previous chapter characterize the change of speech distribution due to the presence of noise and channel distortions. The main issue in this compensation procedure is that the noise parameters are often unknown and need to be estimated from the input utterances. In this chapter, we begin by formulating the noise estimation problem using the maximum-likelihood (ML) criterion, and we then present a unified approach based on the Gauss-Newton method [8], [6], [1] for the optimization of various nonlinear compensation models. We shall demonstrate that the widely used methods for estimating the noise means are variants of the Gauss-Newton method. Furthermore, we propose a novel noise variance estimation algorithm that is consistent with the Gauss-Newton principle. This principled optimization framework differentiates it from all of the noise estimation methods that have been proposed in the literature. The formulation of the Gauss-Newton method reduces the noise estimation problem to the determination of the Jacobians of the corrupted speech distributions with respect to the clean speech and noise distributions. For the VTS compensation, the estimation is straightforward. For the sampling-based compensation, we present two methods, the sample Jacobian average (SJA) and the cross-covariance (XCOV), to numerically evaluate such Jacobians. We shall also consider some extensions to the standard noise compensation, and in particular we describe noise adaptive training and fast VTS compensation. Finally, we discuss some implementation issues of the Gauss-Newton method.

3.1 Maximum-Likelihood Noise Estimation

Consider the acoustic models trained with clean speech, $\mathbf{\Lambda}_x$, in which the kth Gaussian component of the jth state is composed of three portions distributed as $\mathcal{N}(\boldsymbol{\mu}_{x,jk}, \mathbf{\Sigma}_{x,jk})$, $\mathcal{N}(\boldsymbol{\mu}_{\Delta x,jk}, \mathbf{\Sigma}_{\Delta x,jk})$, and $\mathcal{N}(\boldsymbol{\mu}_{\Delta^2 x,jk}, \mathbf{\Sigma}_{\Delta^2 x,jk})$, respectively. Given an estimate of the noise parameters $\boldsymbol{\theta} = \{\boldsymbol{\mu}_n, \boldsymbol{\mu}_h, \mathbf{\Sigma}_n, \mathbf{\Sigma}_{\Delta n}, \mathbf{\Sigma}_{\Delta^2 n}\}$, we can compensate each Gaussian component

of $\mathbf{\Lambda}_x$ to generate a new set of the speech models $\mathbf{\Lambda}_y$. The transformed models should characterize better the speech in the target environment and yield an improved recognition accuracy.

The main issue in this compensation procedure is that the noise parameters are often unknown and need to be estimated from the input utterances. One can simply estimate the additive noise parameters from non-speech frames. This method however requires a reliable voice activity detector (VAD) and can not estimate the convolutional distortion.

The ML estimation of the whole noise parameter set can be formulated in an EM framework [63]. Given the corrupted observation sequence $\boldsymbol{O}$ and its hypothesized transcription $\mathcal{W}$, the noise parameters $\boldsymbol{\theta}$ are estimated by maximizing the likelihood of $\boldsymbol{O}$ given $\boldsymbol{\theta}$ in combination with the speech models $\mathbf{\Lambda}_x$

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \log p(\boldsymbol{O}|\mathbf{\Lambda}_x, \boldsymbol{\theta}, \mathcal{W}). \tag{3.1}$$

The EM auxiliary function can be expressed as

$$Q(\hat{\boldsymbol{\theta}}|\boldsymbol{\theta}) = \sum_t \sum_{j,k} \gamma_{jk}(t) \log p(\boldsymbol{o}_t|\mathbf{\Lambda}_x, j, k, \hat{\boldsymbol{\theta}}) \tag{3.2}$$

where $\boldsymbol{\theta}$ and $\hat{\boldsymbol{\theta}}$ are the existing and the new parameter sets, respectively; $\gamma_{jk}(t)$ denotes the posterior probability of being in the kth Gaussian component of the jth state at time t given the existing corrupted speech models $\mathbf{\Lambda}_y$; $\boldsymbol{o}_t$ denotes the corrupted speech observation, $\boldsymbol{o}_t = [\boldsymbol{y}_t^{\mathrm{T}}, {}_{\Delta}\boldsymbol{y}_t^{\mathrm{T}}, {}_{\Delta^2}\boldsymbol{y}_t^{\mathrm{T}}]^{\mathrm{T}}$.

To simplify the M step, the auxiliary function is often decomposed into three portions corresponding to static, delta, and delta-delta dimensions, respectively, and each portion is independently maximized with respect to different noise parameters. We follow this divide-and-conquer strategy. The dependence of the corrupted speech distribution on the noise parameters being optimized can be summarized using function notations. For the static corrupted speech parameters, the dependence can be expressed as:

$$\boldsymbol{\mu}_{y,jk} = \boldsymbol{g}_{\mu}(\boldsymbol{\mu}_{x,jk}, \boldsymbol{\mu}_n, \boldsymbol{\mu}_h) \tag{3.3}$$

$$\mathbf{\Sigma}_{y,jk} = \boldsymbol{g}_{\Sigma}(\mathbf{\Sigma}_{x,jk}, \mathbf{\Sigma}_n) \tag{3.4}$$

where (3.3) and (3.4) represent (2.9) and (2.10) for the VTS compensation, and (2.15) and (2.16) for the sampling-based compensation, respectively. For the delta (similar for delta-delta) dimensions, we have

$$\boldsymbol{\mu}_{\Delta y,jk} = \boldsymbol{g}_{\Delta\mu}(\boldsymbol{\mu}_{\Delta x,jk}) \tag{3.5}$$

$$\boldsymbol{\Sigma}_{\Delta y,jk} = \boldsymbol{g}_{\Delta\Sigma}(\boldsymbol{\Sigma}_{\Delta x,jk}, \boldsymbol{\Sigma}_{\Delta n}) \tag{3.6}$$

where (3.5) and (3.6) represent (2.11) and (2.12) for both the VTS and sampling-based compensation.

Note that the decomposition of the M step embodies two approximations, which tend to be overlooked. First, for the VTS compensation, since the Jacobians $\boldsymbol{G}_x^{(0)}$ and $\boldsymbol{G}_n^{(0)}$ are functions of $\boldsymbol{\mu}_n$ and $\boldsymbol{\mu}_h$, (3.4)–(3.6) should also depend on $\boldsymbol{\mu}_n$ and $\boldsymbol{\mu}_h$. Here, we consider the Jacobians as constant in (3.4)–(3.6), not being optimized. Second, for the sampling-based compensation, either $\boldsymbol{\mu}_{y,jk}$ or $\boldsymbol{\Sigma}_{y,jk}$ depends on both the static noise means and variances, because the simulation samples are drawn based on the static noise distributions. We simplify the dependence of $\boldsymbol{\mu}_{y,jk}$ and $\boldsymbol{\Sigma}_{y,jk}$ as (3.3) and (3.4), respectively, similar to the VTS compensation.

To update the noise parameters in the M step, we may separately differentiate the decomposed Q functions with respect to the corresponding noise parameters, and solve for their zeros. Unfortunately, the derivatives of Q are nonlinear functions of the noise parameters, and the maximization problem does not have a closed-form solution. One may seek a generalized EM [13] scheme to progressively change $\boldsymbol{\theta}$ to increase the auxiliary function, other than directly maximizing it.

In [63], Moreno has established the generalized EM framework to estimate both the additive noise and the convolutional channel for the VTS compensation in the feature domain. In [57], [54], this approach was extended to the model-domain VTS compensation by incorporating the compensation of dynamic features and HMM variances. Note that the noise variances are estimated using the gradient ascent method in [57] and Newton's method in [54]. The optimization method has also been modified to allow for the UT compensation [55] and adaptive training [43].

In this chapter, we will demonstrate that the iterative approach for estimating noise means is a variant of the Gauss-Newton method. Furthermore, we will present a novel noise variance estimation method that is consistent with the Gauss-Newton principle. The generalization of the Gauss-Newton method for various compensation models is also discussed.

3.2 Estimating Noise and Channel Means

To estimate the noise and channel means, we need to maximize the auxiliary function (3.2), which, by absorbing terms independent of the static noise means and variance into "const," can be rewritten as

$$Q(\hat{\boldsymbol{\theta}}|\boldsymbol{\theta}) = -\frac{1}{2}\sum_{t}\sum_{j,k}\gamma_{jk}(t)\Big[\log|\hat{\boldsymbol{\Sigma}}_{y,jk}| + (\boldsymbol{y}_t - \hat{\boldsymbol{\mu}}_{y,jk})^{\mathrm{T}}\hat{\boldsymbol{\Sigma}}_{y,jk}^{-1}(\boldsymbol{y}_t - \hat{\boldsymbol{\mu}}_{y,jk})\Big] + \text{const.} \quad (3.7)$$

We note that if $\hat{\boldsymbol{\Sigma}}_{y,jk}$ is fixed, say $\hat{\boldsymbol{\Sigma}}_{y,jk} = \boldsymbol{\Sigma}_{y,jk}$, the Q function takes the form of weighted nonlinear least squares for $\{\hat{\boldsymbol{\mu}}_n, \hat{\boldsymbol{\mu}}_h\}$. The observations $\boldsymbol{y}_t$ are fitted by $\hat{\boldsymbol{\mu}}_{y,jk}$, a nonlinear function of $\{\hat{\boldsymbol{\mu}}_n, \hat{\boldsymbol{\mu}}_h\}$, and the squared residues are scaled by $\hat{\boldsymbol{\Sigma}}_{y,jk}^{-1}$. Thus, the optimization of the noise and channel means is a straightforward application of the Gauss-Newton method.

In the literature, the optimization is often formulated by linearizing $\hat{\boldsymbol{\mu}}_{y,jk}$, and then finding the solutions to the resulting linear least squares problem as a new estimate of the noise means [63], [57], [54]. This approach is analogous to the linearization of (2.24) for solving the generic nonlinear least squares problem. Taking the first-order Taylor series expansion at the existing estimate of the noise and channel means yields

$$\hat{\boldsymbol{\mu}}_{y,jk} \approx \boldsymbol{\mu}_{y,jk} + \boldsymbol{G}_{n,jk}(\hat{\boldsymbol{\mu}}_n - \boldsymbol{\mu}_n) + \boldsymbol{G}_{h,jk}(\hat{\boldsymbol{\mu}}_h - \boldsymbol{\mu}_h) \quad (3.8)$$

where $\boldsymbol{G}_{n,jk}$ and $\boldsymbol{G}_{h,jk}$ are the Jacobian matrices of $\boldsymbol{\mu}_y$ with respect to $\boldsymbol{\mu}_n$ and $\boldsymbol{\mu}_h$ for the Gaussian component (j,k), respectively

$$\boldsymbol{G}_n = \frac{\partial \boldsymbol{\mu}_y}{\partial \boldsymbol{\mu}_n} \quad (3.9)$$

$$\boldsymbol{G}_h = \boldsymbol{G}_x = \frac{\partial \boldsymbol{\mu}_y}{\partial \boldsymbol{\mu}_x}. \quad (3.10)$$

Note that the Taylor expansion (3.8) and its associated Jacobians should not be confused with those in (2.5) used for the VTS compensation. In (3.8), the Jacobians are defined

over the model parameters ($\boldsymbol{\mu}_y$, $\boldsymbol{\mu}_n$, and $\boldsymbol{\mu}_h$) to drive an iterative optimization procedure, whereas the Jacobians in (2.5) are defined in the observation space to relate the corrupted and clean speech observations for the VTS model. We may refer to $\boldsymbol{G}_x$ and $\boldsymbol{G}_n$ as the model Jacobians, and $\boldsymbol{G}_x^{(0)}$ in (2.6) and $\boldsymbol{G}_n^{(0)}$ in (2.7) as the sample Jacobians, when necessary to distinguish them. The two sets of Jacobians become equal in value in the special case of the VTS compensation. Their connection for the sampling-based compensation will be made clear in Section 3.4.

Differentiating the Q function with respect to $\hat{\boldsymbol{\mu}}_n$ and $\hat{\boldsymbol{\mu}}_h$, respectively, and equating them to zero, we obtain [1]

$$\frac{\partial Q}{\partial \hat{\boldsymbol{\mu}}_n} = \sum_t \sum_{j,k} \gamma_{jk} \boldsymbol{G}_{n,jk}^{\mathrm{T}} \boldsymbol{\Sigma}_{y,jk}^{-1} (\boldsymbol{y}_t - \hat{\boldsymbol{\mu}}_{y,jk}) = 0 \tag{3.11}$$

$$\frac{\partial Q}{\partial \hat{\boldsymbol{\mu}}_h} = \sum_t \sum_{j,k} \gamma_{jk} \boldsymbol{G}_{x,jk}^{\mathrm{T}} \boldsymbol{\Sigma}_{y,jk}^{-1} (\boldsymbol{y}_t - \hat{\boldsymbol{\mu}}_{y,jk}) = 0. \tag{3.12}$$

Substituting (3.8) into (3.11) and (3.12) leads to a system of linear equations, from which $\hat{\boldsymbol{\mu}}_n$ and $\hat{\boldsymbol{\mu}}_h$ can be solved simultaneously. This formulation has been used in [63] and [57].

In a slightly different way, the noise and channel means can be solved sequentially by fixing the other parameters at their existing values, as described in [54]. By substituting (3.8) into (3.11) with $\hat{\boldsymbol{\mu}}_h = \boldsymbol{\mu}_h$, the noise mean can be updated as

$$\hat{\boldsymbol{\mu}}_n = \boldsymbol{\mu}_n + \Big[\sum_{j,k} \gamma_{jk} \boldsymbol{G}_{n,jk}^{\mathrm{T}} \boldsymbol{\Sigma}_{y,jk}^{-1} \boldsymbol{G}_{n,jk} \Big]^{-1} \sum_{j,k} \boldsymbol{G}_{n,jk}^{\mathrm{T}} \boldsymbol{\Sigma}_{y,jk}^{-1} \boldsymbol{c}_{y,jk} \tag{3.13}$$

where we define the following sufficient statistics:

$$\gamma_{jk} = \sum_t \gamma_{jk}(t) \tag{3.14}$$

$$\boldsymbol{c}_{y,jk} = \sum_t \gamma_{jk}(t) (\boldsymbol{y}_t - \boldsymbol{\mu}_{y,jk}). \tag{3.15}$$

Similarly, by substituting (3.8) into (3.12) with $\hat{\boldsymbol{\mu}}_n = \boldsymbol{\mu}_n$, the channel mean can be estimated as

$$\hat{\boldsymbol{\mu}}_h = \boldsymbol{\mu}_h + \Big[\sum_{j,k} \gamma_{jk} \boldsymbol{G}_{x,jk}^{\mathrm{T}} \boldsymbol{\Sigma}_{y,jk}^{-1} \boldsymbol{G}_{x,jk} \Big]^{-1} \sum_{j,k} \boldsymbol{G}_{x,jk}^{\mathrm{T}} \boldsymbol{\Sigma}_{y,jk}^{-1} \boldsymbol{c}_{y,jk}. \tag{3.16}$$

[1] In (3.12), we use $\boldsymbol{G}_{x,jk}$ in place of $\boldsymbol{G}_{h,jk}$, though the latter appears more suitable in this equation. The two terms are identical, and for simplicity, we will retain the use of $\boldsymbol{G}_{x,jk}$ when applicable.

Estimating the noise and the channel means sequentially may slow the convergence rate compared to solving them simultaneously. However, the re-estimation formulas become relatively simple[2] and are favorable for the discussion of their optimization properties. In this work, we adopt the sequential re-estimation formulas.

As we have mentioned, the above re-estimation formulas are instances of the Gauss-Newton method, and can be derived from Newton's method via an approximation. Taking estimation of $\boldsymbol{\mu}_n$ as an example, we can write the gradient and Hessian of the Q function with respect to $\boldsymbol{\mu}_n$ as

$$\frac{\partial Q}{\partial \boldsymbol{\mu}_n} = \sum_{j,k} \boldsymbol{G}_{n,jk}^{\mathrm{T}} \boldsymbol{\Sigma}_{y,jk}^{-1} \boldsymbol{c}_{y,jk} \tag{3.17}$$

$$\frac{\partial^2 Q}{\partial \boldsymbol{\mu}_n^2} = -\sum_{j,k} \gamma_{jk} \boldsymbol{G}_{n,jk}^{\mathrm{T}} \boldsymbol{\Sigma}_{y,jk}^{-1} \boldsymbol{G}_{n,jk} - \sum_{j,k} \boldsymbol{\Sigma}_{y,jk}^{-1} \boldsymbol{c}_{y,jk} \frac{\partial^2 \boldsymbol{\mu}_{y,jk}}{\partial \boldsymbol{\mu}_n^2} \tag{3.18}$$

where the second summation term on the right-hand side of (3.18), involving a 3-dimensional array $\frac{\partial^2 \boldsymbol{\mu}_{y,jk}}{\partial \boldsymbol{\mu}_n^2}$, is not exact and is presented here for illustration.

The Gauss-Newton method is formed by ignoring the second term in the Hessian, which is rather expensive to evaluate due to $\frac{\partial^2 \boldsymbol{\mu}_{y,jk}}{\partial \boldsymbol{\mu}_n^2}$, resulting in the same update formula (3.13).

By deriving the widely used algorithm for estimating the noise means from the Gauss-Newton perspective, we gain a deeper understanding of the approach. One conclusion is that it saves the cumbersome calculation of the second-order derivatives, while achieving an approximately quadratic convergence rate. More importantly, the formulation of the Gauss-Newton method offers a unified treatment for optimizing various noise compensation models. The problem of estimating the noise means is reduced to determining the model Jacobians $\boldsymbol{G}_x$ and $\boldsymbol{G}_n$, which will be described in Section 3.4.

3.3 Estimating Noise Variances

Estimating the noise variances in the EM framework is a nontrivial problem and has been generally avoided in most of the earlier work. This is due to the fact that the compensated variances, which are affine functions of the noise variances, appear in the form of determinant

[2] Also, the sequential estimation involves the inversion of two $N \times N$ matrices, slightly cheaper to solve than the simultaneous estimation, which inverts a $2N \times 2N$ matrix.

and matrix inverse in the auxiliary function. In the literature, two methods have been proposed to estimate the noise variances: the gradient ascent method [57] and Newton's method [54]. Here, we present an approach to recursively estimating the noise variances, promising a better performance and less computational complexity. First, we describe this optimization procedure through linear approximation of the derivative of the Q function, and then show that the optimization conforms to the Gauss-Newton principle.

Consider the case of estimating the static noise variance in VTS compensation. Fixing $\hat{\boldsymbol{\mu}}_n = \boldsymbol{\mu}_n$, the derivative of the Q function (3.7) with respect to $\hat{\boldsymbol{\Sigma}}_n$ is given by (see Appendix A)

$$\frac{\partial Q}{\partial \hat{\boldsymbol{\Sigma}}_n} = \frac{1}{2} \sum_{j,k} \boldsymbol{G}_{n,jk}^{\mathrm{T}} \hat{\boldsymbol{\Sigma}}_{y,jk}^{-1} (\boldsymbol{S}_{y,jk} - \gamma_{jk} \hat{\boldsymbol{\Sigma}}_{y,jk}) \hat{\boldsymbol{\Sigma}}_{y,jk}^{-1} \boldsymbol{G}_{n,jk} \tag{3.19}$$

where we define the sufficient statistic

$$\boldsymbol{S}_{y,jk} = \sum_t \gamma_{jk}(t) (\boldsymbol{y}_t - \boldsymbol{\mu}_{y,jk})(\boldsymbol{y}_t - \boldsymbol{\mu}_{y,jk})^{\mathrm{T}}. \tag{3.20}$$

The resulting derivative function is a nonlinear function of the noise variance and has no closed-form solutions for its roots. Nevertheless, the derivative function can be thought of as a sum of rational functions, where the numerator $(\boldsymbol{S}_{y,jk} - \gamma_{jk} \hat{\boldsymbol{\Sigma}}_{y,jk})$ is an affine function of $\hat{\boldsymbol{\Sigma}}_n$ and the denominator is $\hat{\boldsymbol{\Sigma}}_{y,jk}(\cdots)\hat{\boldsymbol{\Sigma}}_{y,jk}$, a square of $\hat{\boldsymbol{\Sigma}}_{y,jk}$. By fixing $\hat{\boldsymbol{\Sigma}}_n$ in the denominator to the existing estimate of the noise variance, we obtain a linear approximation to the derivative function, whose zeros can be solved in the following equation[3]:

$$\sum_{j,k} \gamma_{jk} \boldsymbol{A}_{jk} (\hat{\boldsymbol{\Sigma}}_n - \boldsymbol{\Sigma}_n) \boldsymbol{A}_{jk}^{\mathrm{T}} = \sum_{j,k} \boldsymbol{B}_{jk} \tag{3.21}$$

where

$$\boldsymbol{A}_{jk} = \boldsymbol{G}_{n,jk}^{\mathrm{T}} \boldsymbol{\Sigma}_{y,jk}^{-1} \boldsymbol{G}_{n,jk} \tag{3.22}$$

$$\boldsymbol{B}_{jk} = \boldsymbol{G}_{n,jk}^{\mathrm{T}} \boldsymbol{\Sigma}_{y,jk}^{-1} (\boldsymbol{S}_{y,jk} - \gamma_{jk} \boldsymbol{\Sigma}_{y,jk}) \boldsymbol{\Sigma}_{y,jk}^{-1} \boldsymbol{G}_{n,jk}. \tag{3.23}$$

The explicit solution to the linear equation (3.21) can be obtained by rewriting $\hat{\boldsymbol{\Sigma}}_n$ in a

[3]We may move the term involving $\boldsymbol{\Sigma}_n$ on the left-hand side of (3.21) to the right-hand side for a compact expression. The form (3.21) is adopted to highlight its gradient-based update structure.

vector form as

$$\mathrm{vec}(\hat{\boldsymbol{\Sigma}}_n) = \mathrm{vec}(\boldsymbol{\Sigma}_n) + \Big[\sum_{j,k} \gamma_{jk} \boldsymbol{A}_{jk} \otimes \boldsymbol{A}_{jk}\Big]^{-1} \sum_{j,k} \mathrm{vec}(\boldsymbol{B}_{jk}) \tag{3.24}$$

where $\mathrm{vec}(\cdot)$ transforms a matrix into a vector by stacking the columns of the matrix into a single column vector, and $\otimes$ denotes the Kronecker product of two matrices [36].

The re-estimation formula (3.21) gives a general form for estimating the noise variances in the VTS compensation, even in the case of full covariance matrices. For the estimation of the dynamic noise variances, one needs to replace the static parameters with the corresponding dynamic parts.

Furthermore, the formula reflects the physical meaning of the noise variance. The term $(\boldsymbol{S}_{y,jk} - \gamma_{jk}\boldsymbol{\Sigma}_{y,jk})$ in $\boldsymbol{B}_{jk}$ represents the residual between the sample variance and the variance estimate of the corrupted speech. Then a weighted sum of the residual variances over all Gaussian mixtures becomes a correction term to the existing estimate of the noise variance.

We can show that the above optimization algorithm conforms to the Gauss-Newton method. To simplify the discussion, we consider first the case of a 1-dimensional noise variance and then extend the conclusion to the multidimensional case by inspection. Let σ_n^2, $\sigma_{x,jk}^2$, $\sigma_{y,jk}^2$ and $s_{n,jk}$ be the 1-dimensional counterparts of $\boldsymbol{\Sigma}_n$, $\boldsymbol{\Sigma}_{x,jk}$, $\boldsymbol{\Sigma}_{y,jk}$ and $\boldsymbol{S}_{y,jk}$, respectively. The key is to think that the Q function depends on σ_n^2 via the precision parameter of the corrupted speech $\beta_{y,jk} = \sigma_{y,jk}^{-2}$. Apply the chain rule to find the first and second derivatives:

$$\frac{\partial Q}{\partial \sigma_n^2} = -\frac{1}{2}\sum_{j,k}(s_{y,jk} - \gamma_{jk}\beta_{y,jk}^{-1})\frac{\partial \beta_{y,jk}}{\partial \sigma_n^2} \tag{3.25}$$

$$\frac{\partial^2 Q}{\partial^2 \sigma_n^2} = -\frac{1}{2}\sum_{j,k}\gamma_{jk}\beta_{y,jk}^{-2}\Big(\frac{\partial \beta_{y,jk}}{\partial \sigma_n^2}\Big)^2 - \frac{1}{2}\sum_{j,k}(s_{y,jk} - \gamma_{jk}\beta_{y,jk}^{-1})\frac{\partial^2 \beta_{y,jk}}{\partial^2 \sigma_n^2}. \tag{3.26}$$

As we have seen for the estimation of the noise means, $\frac{\partial^2 Q}{\partial^2 \sigma_n^2}$ is also composed of two terms, which involve the first and second derivatives of $\beta_{y,jk}$ with respect to σ_n^2, respectively. For the same reason, the Gauss-Newton method can be formulated by ignoring the second term in (3.26). Substituting $\beta_{y,jk} = \sigma_{y,jk}^{-2}$ into (3.25) and (3.26), the re-estimation formula is

given by

$$\hat{\sigma}_n^2 = \sigma_n^2 + \Big[\sum_{j,k} \gamma_{jk}\sigma_{y,jk}^{-4}\Big(\frac{\partial \sigma_{y,jk}^2}{\partial \sigma_n^2}\Big)^2\Big]^{-1}\Big[\sum_{j,k} \sigma_{y,jk}^{-4}(s_{y,jk} - \gamma_{jk}\sigma_{y,jk}^2)\frac{\partial \sigma_{y,jk}^2}{\partial \sigma_n^2}\Big]. \tag{3.27}$$

The re-estimation formula (3.27) gives a general form for the estimation of the noise variance, where $\sigma_{y,jk}^2$ is an arbitrary 1-dimensional function of σ_n^2. When the corrupted variance depends linearly on the noise variance as (2.10) in a multidimensional context, the re-estimation formula becomes (3.24), where $\sum_{j,k} \gamma_{jk}\boldsymbol{A}_{jk} \otimes \boldsymbol{A}_{jk}$ corresponds to the approximated Hessian, and $\sum_{j,k} \mathrm{vec}(\boldsymbol{B}_{jk})$ to the gradient.

To estimate the noise variance in the sampling-based compensation models, one needs to determine $\frac{\partial \boldsymbol{\Sigma}_y}{\partial \boldsymbol{\Sigma}_n}$. A rigorous derivation of such a Jacobian is considerably complicated, though not impossible. To simplify it, we assume that $\boldsymbol{\Sigma}_y$ can also be approximated in a linear form of $\boldsymbol{\Sigma}_n$ as $\boldsymbol{\Sigma}_y = \boldsymbol{G}_n\boldsymbol{\Sigma}_n\boldsymbol{G}_n^{\mathrm{T}} + \boldsymbol{K}$, where $\boldsymbol{K}$ is a term independent of $\boldsymbol{\Sigma}_n$, and $\boldsymbol{G}_n$ is defined in (3.9). Since the term involving $\boldsymbol{\Sigma}_n$ retains the same structure as the one of the corrupted variance (2.10) for the VTS compensation, the re-estimation formula (3.21) applies equally to the sampling-based compensation models. The determination of $\boldsymbol{G}_n$ will be described in the next section.

3.4 Jacobians of Compensation Models

As we have seen, when we employ the Gauss-Newton method to iteratively update the nose parameters in the M step, the main question left is to find the model Jacobian matrices, $\boldsymbol{G}_x$ and $\boldsymbol{G}_n$. For the VTS compensation, it is straightforward:

$$\boldsymbol{G}_x^{\mathrm{vts}} = \frac{\partial \boldsymbol{\mu}_y}{\partial \boldsymbol{\mu}_x} = \frac{\partial \boldsymbol{y}}{\partial \boldsymbol{x}}\Big|_{\boldsymbol{\mu}^{(0)}} = \boldsymbol{G}_x^{(0)} \tag{3.28}$$

$$\boldsymbol{G}_n^{\mathrm{vts}} = \boldsymbol{G}_n^{(0)}. \tag{3.29}$$

For the sampling-based compensation, direct determination of the model Jacobians is problematic, because the corrupted mean is not a simple closed-form function of the clean speech and noise means as (2.15). To tackle this problem, we first relate the model Jacobians to the expected value of the sample Jacobians of the mismatch function, and then present two alternatives for the evaluation of $\boldsymbol{G}_x$ and $\boldsymbol{G}_n$.

Consider a random vector $\boldsymbol{z} = [\boldsymbol{x}^{\mathrm{T}}, \boldsymbol{n}^{\mathrm{T}}]^{\mathrm{T}}$, a realization of which is obtained by first choosing a value from the zero-mean unit-covariance Gaussian distribution and then transforming it to the given distribution. Thus, $\boldsymbol{z}$ is defined as

$$\boldsymbol{z} = \sqrt{\boldsymbol{\Sigma}_z}\,\tilde{\boldsymbol{z}} + \boldsymbol{\mu}_z \tag{3.30}$$

where $\tilde{\boldsymbol{z}}$ is distributed as $\mathcal{N}(0, \boldsymbol{I})$. This mapping procedure separates out the effect of the distribution parameters from the Gaussian randomness, as $\boldsymbol{\mu}_z$ and $\boldsymbol{\Sigma}_z$ are independent of $\tilde{\boldsymbol{z}}$. Thus, it follows from the chain rule that[4]

$$\frac{\partial \boldsymbol{\mu}_y}{\partial \boldsymbol{\mu}_z} = \mathcal{E}\Big[\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{\mu}_z}\Big] = \mathcal{E}\Big[\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{z}}\frac{\partial \boldsymbol{z}}{\partial \boldsymbol{\mu}_z}\Big] = \mathcal{E}\Big[\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{z}}\Big] \tag{3.31}$$

where we use $\frac{\partial \boldsymbol{z}}{\partial \boldsymbol{\mu}_z} = \boldsymbol{I}$ due to (3.30).

3.4.1 Sample Jacobian Average (SJA) Method

By using the identity (3.31), we can determine $\frac{\partial \boldsymbol{\mu}_y}{\partial \boldsymbol{\mu}_z}$ through the numerical evaluation over $\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{z}}$. We have

$$\begin{aligned}\boldsymbol{G}_x^{\mathrm{sja}} &= \frac{1}{M}\sum_{m=1}^{M}\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{x}}\Big|_{x^{(m)},n^{(m)}} = \frac{1}{M}\sum_{m=1}^{M}\boldsymbol{G}_x^{(m)} \\ &= \frac{1}{M}\boldsymbol{C}\Big[\sum_{m=1}^{M}\boldsymbol{G}_x^{\mathrm{l},(m)}\Big]\boldsymbol{C}^{\dagger} \qquad (3.32)\end{aligned}$$

$$\boldsymbol{G}_n^{\mathrm{sja}} = \frac{1}{M}\sum_{m=1}^{M}\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{n}}\Big|_{x^{(m)},n^{(m)}} = \boldsymbol{I} - \boldsymbol{G}_x^{\mathrm{sja}}. \tag{3.33}$$

where $\boldsymbol{G}_x^{(m)}$ denotes the Jacobian matrix of $\boldsymbol{y}$ with respect to $\boldsymbol{x}$ at the sampling point $\{\boldsymbol{x}^{(m)}, \boldsymbol{n}^{(m)}\}$, and $\boldsymbol{G}_x^{\mathrm{l},(m)}$, its log-spectral version. The second line in (3.32) follows from (2.6) and uses the fact that $\boldsymbol{G}_x^{(m)}$ depends linearly on $\boldsymbol{G}_x^{\mathrm{l},(m)}$. Since $\boldsymbol{G}_x^{\mathrm{l},(m)}$ is diagonal, the degrees of freedom in evaluating $\boldsymbol{G}_x^{\mathrm{sja}}$ are reduced to the size of the filterbank, L. The reduction in degrees of freedom is important to the sampling method based on Monte Carlo, as the number of samples needed to attain a given approximation accuracy increases rapidly with the degrees of freedom.

[4]It is possible to proceed with the method to exactly determine the Jacobian $\frac{\partial \boldsymbol{\Sigma}_y}{\partial \boldsymbol{\Sigma}_n}$ for the sampling-based models, except in a more complicated form.

In [55], this approach has been used for the noise estimation in the UT compensation. Here, we provide a more succinct derivation under the Gauss-Newton framework.

3.4.2 Cross-Covariance (XCOV) Method

There is an underlying assumption in the SJA method that the mismatch function characterizing the corrupted speech observations is in a closed form. However, in some instances, the dependence of the mismatch function on its control parameters is so complicated that reliably evaluating its gradients is difficult. An alternative to determining the Jacobians of the compensation models can be expresses as

$$\boldsymbol{G}_x^{\text{xcov}} = \boldsymbol{\Sigma}_{yx}\boldsymbol{\Sigma}_x^{-1} \tag{3.34}$$

$$\boldsymbol{G}_n^{\text{xcov}} = \boldsymbol{\Sigma}_{yn}\boldsymbol{\Sigma}_n^{-1}. \tag{3.35}$$

The above equations come from the following theorem:

Theorem 1. *Assume $\boldsymbol{x}$ to be a multivariate Gaussian random variable distributed as $p(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_x, \boldsymbol{\Sigma}_x)$, and $\boldsymbol{y} = \boldsymbol{g}(\boldsymbol{x})$ is a function of $\boldsymbol{x}$. If $p(\boldsymbol{x})|\boldsymbol{g}(\boldsymbol{x})| \to 0$, as $|\boldsymbol{x}| \to \infty$, we have*

$$\mathcal{E}\left[\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{x}}\right] = \boldsymbol{\Sigma}_{yx}\boldsymbol{\Sigma}_x^{-1}. \tag{3.36}$$

Proof. First, we show the theorem holds for 1-dimensional cases. Then x is distributed as $\mathcal{N}(x|\mu_x, \sigma_x^2)$. It follows from integration by parts

$$\begin{aligned}
\Sigma_{yx} &= \frac{1}{\sqrt{2\pi\sigma_x^2}} \int_{-\infty}^{\infty} g(x)(x-\mu_x) e^{-\frac{(x-\mu_x)^2}{2\sigma_x^2}} \,\mathrm{d}x \\
&= \frac{-\sigma_x^2}{\sqrt{2\pi\sigma_x^2}} \Big[g(x) e^{-\frac{(x-\mu_x)^2}{2\sigma_x^2}} \Big|_{-\infty}^{\infty} - \int_{-\infty}^{\infty} g'(x) e^{-\frac{(x-\mu_x)^2}{2\sigma_x^2}} \,\mathrm{d}x \Big] \\
&= \sigma_x^2 \, \mathcal{E}\Big[\frac{\partial y}{\partial x}\Big]
\end{aligned} \tag{3.37}$$

and we have 1-dimensional case of (3.36).

With the notation of vector calculus, the above derivation can be extended to the case of multivariate Gaussian of $\boldsymbol{x}$. The result for vector-valued function $\boldsymbol{y}$ is trivially obtained by stacking all such results for elements of $\boldsymbol{y}$. □

The theorem is related to Bussgang's theorem [12]. Furthermore, it can be relaxed to the case where $\boldsymbol{y}$ depends on $\boldsymbol{x}$ as well as other random variables. Let $\boldsymbol{y}$ be some function

$\boldsymbol{y} = \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{n})$, where $\boldsymbol{x}$ and $\boldsymbol{n}$ are Gaussian and independently distributed. Substituting $(\boldsymbol{x}, \boldsymbol{n})$ for $\boldsymbol{x}$ in (3.36) yields

$$\begin{aligned}\left[\mathcal{E}\left[\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{x}}\right] \quad \mathcal{E}\left[\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{n}}\right]\right] &= \begin{bmatrix}\boldsymbol{\Sigma}_{yx} & \boldsymbol{\Sigma}_{yn}\end{bmatrix} \begin{bmatrix}\boldsymbol{\Sigma}_x & 0 \\ 0 & \boldsymbol{\Sigma}_n\end{bmatrix}^{-1} \\ &= \begin{bmatrix}\boldsymbol{\Sigma}_{yx}\boldsymbol{\Sigma}_x^{-1} & \boldsymbol{\Sigma}_{yn}\boldsymbol{\Sigma}_n^{-1}\end{bmatrix}.\end{aligned} \tag{3.38}$$

and we have the identities as (3.34) and (3.35).

The cross-covariance terms in (3.34) and (3.34) can be numerically calculated as

$$\boldsymbol{\Sigma}_{yx}^{\text{smp}} = \frac{1}{M}\sum_{m=1}^{M}(\boldsymbol{y}^{(m)} - \boldsymbol{\mu}_y^{\text{smp}})(\boldsymbol{x}^{(m)} - \boldsymbol{\mu}_x^{\text{smp}})^{\text{T}} \tag{3.39}$$

$$\boldsymbol{\Sigma}_{yn}^{\text{smp}} = \frac{1}{M}\sum_{m=1}^{M}(\boldsymbol{y}^{(m)} - \boldsymbol{\mu}_y^{\text{smp}})(\boldsymbol{n}^{(m)} - \boldsymbol{\mu}_n^{\text{smp}})^{\text{T}}. \tag{3.40}$$

The main advantage of the XCOV method is that it does not require any explicit gradient information. Interestingly, we see that in the whole noise compensation procedure, it is sufficient to characterize the nonlinear compensation model with four quantities for each Gaussian component: the mean $\boldsymbol{\mu}_y$, the variance $\boldsymbol{\Sigma}_y$, and the cross-covariances $\boldsymbol{\Sigma}_{yx}$ and $\boldsymbol{\Sigma}_{yn}$. This treatment may facilitate the study of more complicated distortion models.

The number of degrees of freedom in evaluating $\boldsymbol{G}_x^{\text{xcov}}$ and $\boldsymbol{G}_n^{\text{xcov}}$ is N^2, much larger than that of the SJA method, as (3.34) and (3.35) treat the mismatch function as a black box. As a consequence, the XCOV method may require more sample points to achieve an accurate evaluation of the model Jacobians.

3.5 Discussion

We have addressed the problem of noise parameter estimation for nonlinear compensation models. Due to the nonlinearity of the compensation models, the M step for re-estimating the noise parameters is intractable, and thus the original EM scheme may require some generalization. We propose the Gauss-Newton method as a unified approach to iteratively optimize the auxiliary function. The optimization problem is decomposed into two kinds of subproblems: optimizing the Gaussian means that are (nonlinear) functions of the optimization parameters such as (3.3) and optimizing the Gaussian variances that are (linear

or nonlinear) functions of the parameters as (3.4). The presented Gauss-Newton solutions are sufficiently general, provided the two subproblems rely on disjoint sets of optimization parameters.

In either case, the Gauss-Newton method brings two substantial advantages compared with generic gradient-based methods. First and the foremost important, the Gauss-Newton method can approach a quadratic convergence rate, while saving the calculation of second-order derivatives. Second, the Gauss-Newton method ensures an ascending direction whenever the existing estimate is not a stationary point.

The optimization of the Gaussian means using the Gauss-Newton method is straightforward, because the auxiliary function can be viewed as a quadratic function of the Gaussian means. The extension of the Gauss-Newton method to the optimization of the Gaussian variances is nontrivial. To the best of the author's knowledge, there has not been such a general optimization scheme reported in the literature.

The problem of Gaussian variance optimization has been primarily studied in the area of structured covariance and precision modeling, which aims to represent the full covariance matrices in a data efficient fashion. Typical schemes include semi-tied covariance (STC) models [26], extended maximum-likelihood linear transform (EMLLT) [67], subspace for precision and mean (SPAM) models [8], and factor analyzed covariance models [76]. Often, the full covariance (or precision) matrices are parameterized as a weighted superposition of a set of basis matrices. If the basis matrices have rank 1, they can be expressed as an outer product of the basis vectors. In most cases, there exists no closed-form solution for updating the basis matrices and the corresponding coefficients. Various optimization methods have been proposed depending on the complexities of the models, such as Newton's method [89], the conjugate gradient algorithm [8], and the factor analysis method [76]. The proposed Gauss-Newton framework is promising and warrants further investigation in optimizing these structured covariance matrices in a general and more effective way.

It is interesting to examine the situations when the Gauss-Newton and Newton's method

become equivalent. For optimizing the Gaussian means, we see from (3.18) that the equivalence arises when $\frac{\partial^2 \boldsymbol{\mu}_{y,jk}}{\partial \boldsymbol{\mu}_n^2}$ is zero. Hence, the means $\boldsymbol{\mu}_{y,jk}$ are affine functions of the parameters, and the optimization problem degenerates to linear least squares. For optimizing the Gaussian variances, the equivalence of the two optimization methods occurs when the Gaussian precision matrices are affine functions of the optimization parameters as can be seen from (3.26). This time, however, the exact solution is still intractable. A good example of this case is to model the full precision matrix by a weighted sum of the basic precision matrices [89]. In that work, both the basis matrices and the coefficients were optimized using Newton's method. It turns out that this gives rise to the same update equations as does the Gauss-Newton method.

In [2], we have conducted a comparative study between the Gauss-Newton method and another popular noise estimation approach, which views the model compensation from a generative perspective, where noise and clean speech are latent variables to generate corrupted observations. This gives rise to an EM-based algorithm analogous to the ML estimation for factor analysis (EM-FA) [73], [45], [38]. Both methods belong to the family of gradient-based methods except with different convergence rates: the Gauss-Newton method possesses an approximately quadratic convergence rate, superior to the first-order EM-FA method. Readers are referred to [2] for further discussions of the Gauss-Newton method in its comparison with the EM-FA method.

3.6 Noise Adaptive Training

One straightforward extension of the noise model estimation methods being discussed is to incorporate adaptive training of compensation models. One drawback with standard compensation approaches is that it assumes the availability of the clean speech data for training the acoustic models. If such data do not exist, the performance will be impaired. This restriction can be eliminated with the use of adaptive training [6], [38], [43], [57], which incorporates noise compensation during training. A set of the compensation transforms are estimated in response to different environmental conditions of the training data. The speech model is then estimated based on these transforms, leading to a canonical model that is

expected to represent the intrinsic variability of the speech. While originally developed to remove speaker variations from the acoustic models [6], adaptive training has been successfully applied to the VTS compensation in [38], [43]. These two efforts differ mainly in the choice of the optimization approach. The EM-FA method was used in [38], and a hybrid of the Gauss-Newton and Newton's methods was used in [43].

Here we briefly describe how to extend the Gauss-Newton method to allow adaptive training with noise compensation. Suppose we have a multistyle training set of R sentences, where $\boldsymbol{O}^{(r)}$ is the acoustic observations of the rth training utterance with the reference transcription $\mathcal{W}^{(r)}$. We need to maximize the log-likelihood of the training data with respect to the canonical model $\boldsymbol{\Lambda}_x$ and the set of compensation transforms $\Theta = \{\boldsymbol{\theta}^{(1)}, ..., \boldsymbol{\theta}^{(R)}\}$, so that

$$(\hat{\boldsymbol{\Lambda}}_x, \hat{\Theta}) = \arg\max_{\boldsymbol{\Lambda}_x, \Theta} \sum_r \log p(\boldsymbol{O}^{(r)} | \boldsymbol{\Lambda}_x, \boldsymbol{\theta}^{(r)}, \mathcal{W}^{(r)}). \tag{3.41}$$

The EM algorithm in a similar fashion as described in Section 3.1 is used. The auxiliary function can be written as

$$Q(\hat{\boldsymbol{\Lambda}}_x, \hat{\Theta} | \boldsymbol{\Lambda}_x, \Theta) = \sum_r \sum_{t,j,k} \gamma_{jk}^{(r)}(t) \log p(\boldsymbol{o}_t^{(r)} | j, k, \hat{\boldsymbol{\Lambda}}_x, \hat{\boldsymbol{\theta}}^{(r)}). \tag{3.42}$$

We can interleave the re-estimation of the canonical model and the compensation transforms to simplify the optimization. Based on the existing canonical model $\boldsymbol{\Lambda}_x$, the new set of the compensation transforms $\hat{\Theta}$ is estimated. Then the new acoustic model $\hat{\boldsymbol{\Lambda}}_x$ is obtained given all of the new transforms. This interleaving process is repeated until the likelihood converges.

The re-estimation formulas for the canonical model parameters are similar to those for estimating the noise parameters, but the sufficient statistics are accumulated over all the training utterances. The update equation for the clean speech means is given by

$$\hat{\boldsymbol{\mu}}_{x,jk} = \boldsymbol{\mu}_{x,jk} + \Big[\sum_r \gamma_{jk}^{(r)} \boldsymbol{G}_{x,jk}^{(r)\,\mathrm{T}} \boldsymbol{\Sigma}_{y,jk}^{(r)\,-1} \boldsymbol{G}_{x,jk}^{(r)} \Big]^{-1} \sum_r \boldsymbol{G}_{x,jk}^{(r)\,\mathrm{T}} \boldsymbol{\Sigma}_{y,jk}^{(r)\,-1} \boldsymbol{c}_{y,jk}^{(r)}. \tag{3.43}$$

For the clean speech variance, we have

$$\sum_r \gamma_{jk}^{(r)} \boldsymbol{A}_{x,jk}^{(r)} \left(\hat{\boldsymbol{\Sigma}}_{x,jk} - \boldsymbol{\Sigma}_{x,jk} \right) \boldsymbol{A}_{x,jk}^{(r)\,\mathrm{T}} = \sum_r \boldsymbol{B}_{x,jk}^{(r)} \tag{3.44}$$

where the following notations are defined

$$\boldsymbol{A}_{x,jk}^{(r)} = \boldsymbol{G}_{x,jk}^{(r)\,\mathrm{T}} \boldsymbol{\Sigma}_{y,jk}^{(r)\,-1} \boldsymbol{G}_{x,jk}^{(r)} \tag{3.45}$$

$$\boldsymbol{B}_{x,jk}^{(r)} = \boldsymbol{G}_{x,jk}^{(r)\,\mathrm{T}} \boldsymbol{\Sigma}_{y,jk}^{(r)\,-1} \left(\boldsymbol{S}_{y,jk}^{(r)} - \gamma_{jk}^{(r)} \boldsymbol{\Sigma}_{y,jk}^{(r)} \right) \boldsymbol{\Sigma}_{y,jk}^{(r)\,-1} \boldsymbol{G}_{x,jk}^{(r)}. \tag{3.46}$$

3.7 Fast VTS Compensation

One major drawback of the nonlinear compensation approach is the expensive computational load in estimating the compensation transforms. The standard compensation procedure requires to perform multiple rounds of recognition passes, EM re-estimations and model transformations for each utterance. All of these aspects may considerably raise the computational complexity of the compensation approach and hinder its popularization in practical applications.

One solution is to have a properly-initialized noise estimate such that the re-estimation iterations are significantly reduced. Originally, the noise mean and variance are initialized using the sample average over non-speech frames of the utterance. A more elaborate variant is to apply the same VTS noise estimation process on non-speech areas of an utterance, which is referred to as the fast VTS. If the silence of clean speech is modeled by a single Gaussian component (this model may be separate from the HMM set), the estimation equations (3.13) and (3.21) are simplified to

$$\hat{\boldsymbol{\mu}}_n = \boldsymbol{\mu}_n + \frac{1}{T_{\mathrm{sil}}} \boldsymbol{G}_{n,\mathrm{sil}}^{-1} \boldsymbol{c}_{y,\mathrm{sil}} \tag{3.47}$$

$$\hat{\boldsymbol{\Sigma}}_n = \boldsymbol{\Sigma}_n + \boldsymbol{G}_{n,\mathrm{sil}}^{-1} \left(\frac{\boldsymbol{S}_{y,\mathrm{sil}}}{T_{\mathrm{sil}}} - \boldsymbol{\Sigma}_{y,\mathrm{sil}} \right) \boldsymbol{G}_{n,\mathrm{sil}}^{-\mathrm{T}} \tag{3.48}$$

where the subscript $_{\mathrm{sil}}$ indicates the silence model. The sufficient statistics $\boldsymbol{c}_{y,\mathrm{sil}}$ and $\boldsymbol{S}_{y,\mathrm{sil}}$ are accumulated over T_{sil} frames as (3.15) and (3.20) with the posterior probability $\gamma_{jk}(t)$ set to 1.

The fast VTS in principle is analogous to the Jacobian approach described in [75]. In both schemes, the difference between the reference and observed noise cepstra is exploited to predict the compensation of the acoustic model set.

3.8 Noise Compensation Procedure

The acoustic models of the recognition system are compensated in an unsupervised, utterance-by-utterance manner, similar to [54]. The standard compensation procedure for each input utterance is illustrated in Figure 3.1 and summarized as follows:

Step 1: Initialize the additive noise parameters using the first and last several frames, and set the channel mean vector to 0.

Step 2: Transform the clean acoustic models using the noise compensation model and decode the utterance.

Step 3: Refine the noise estimate with respect to the decoded hypothesis using the Gauss-Newton/EM-FA method, and then transform the models. Multiple re-estimation iterations may be used.

Step 4: Decode the utterance.

Step 5: If the stopping criterion is met, output the recognition transcription; otherwise go to Step 3.

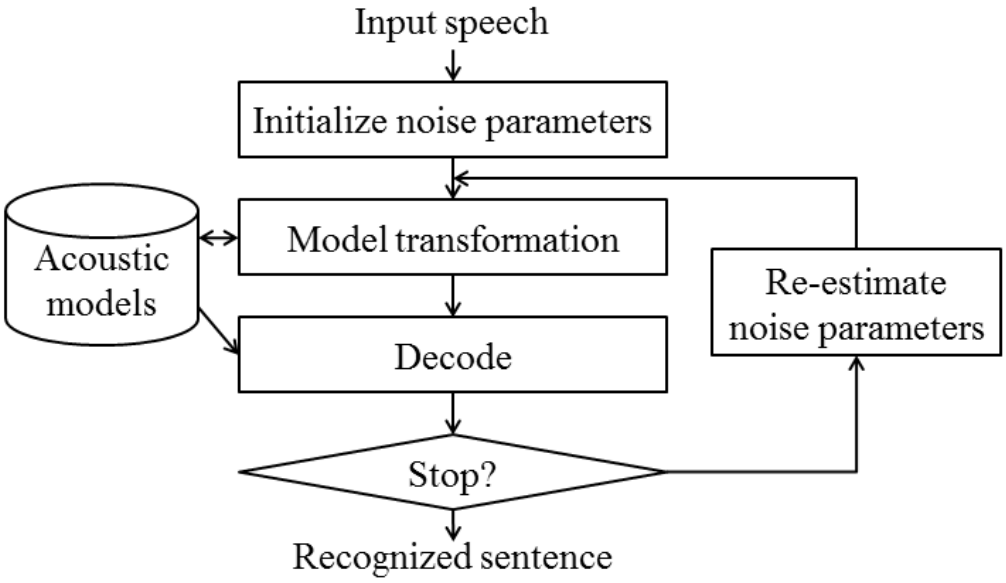


Figure 3.1: The noise compensation procedure.

If the acoustic models are adaptively trained in the presence of the multistyle training data, as described in Section 3.6, the models are obtained on the basis of the conventional acoustic models using the following procedure:

Step 1: Start with the multistyle acoustic models and initialize the noise parameters for each training utterance.

Step 2: Re-estimate the noise parameters for each utterance given the existing acoustic models.

Step 3: Re-estimate the acoustic models given the new noise parameters of the training utterances.

Step 4: Stop, if the stopping criterion is met; otherwise go to Step 2.

3.9 Implementation Issues

There are a number of issues we have to deal with when applying the Gauss-Newton method to the noise parameter estimation, such as computational complexity and numerical stability. Before giving a detailed discussion, we introduce an identity that will be repeatedly used in practical implementations. For a diagonal matrix $\boldsymbol{D}$ and two matrices $\boldsymbol{X}$ and $\boldsymbol{Y}$, the diagonal elements of the product $\boldsymbol{X}\boldsymbol{D}\boldsymbol{Y}^{\mathrm{T}}$ can be efficiently computed as a multiplication between a matrix and a vector

$$\mathrm{diagv}(\boldsymbol{X}\boldsymbol{D}\boldsymbol{Y}^{\mathrm{T}}) = (\boldsymbol{X} \circ \boldsymbol{Y})\,\mathrm{diagv}(\boldsymbol{D}) \tag{3.49}$$

where the operator $\circ$ denotes an element-wise product of two matrices. For matrices of size $N \times N$, this requires $\mathcal{O}(2N^2)$ operations compared to $\mathcal{O}(2N^3)$ for the naive matrix multiplication. A direct application of (3.49) is to compute the diagonal covariance matrix $\boldsymbol{\Sigma}_{y,jk}$ of the VTS compensation in (2.10).

3.9.1 Noise and Channel Means

If the Hessian approximations in the re-estimation formulas (3.13) and (3.16) are ill-conditioned, the parameter update may become unstable. The problem is specially observed at high SNR

levels, in which case $\boldsymbol{G}_n$ approaches zero and then the Hessian approximation in (3.13) tends to zero. To improve the stabilities of the Gauss-Newton method, the Levenberg-Marquardt method [66] can be used. The Hessian approximation is revised as

$$\boldsymbol{H}^{\mathrm{lm}} = \boldsymbol{H} + \lambda \operatorname{diag}(\boldsymbol{H}) \tag{3.50}$$

where $\operatorname{diag}(\cdot)$ sets all of the non-diagonal elements of a matrix to zero, and $\boldsymbol{H}$ denotes the approximated Hessian in (3.13) and (3.16). The damping factor λ controls the impact of the diagonal matrix $\operatorname{diag}(\boldsymbol{H})$, and also plays a role as the learning rate, since as λ becomes large, we have $\boldsymbol{H}^{\mathrm{lm}} \approx (1+\lambda)\operatorname{diag}(\boldsymbol{H})$. Thus, it is desirable to adjust λ at each iteration to maintain a rapid and yet stable update. Here λ is determined by enforcing the least degree of the diagonal dominance of the resulting matrix $\boldsymbol{H}^{\mathrm{lm}}$, such that

$$(1+\lambda)\big|[\boldsymbol{H}]_{ii}\big| \geq \rho \sum_{j \neq i} \big|[\boldsymbol{H}]_{ij}\big| \quad \text{for all } i \tag{3.51}$$

where $[\,\cdot\,]_{ij}$ denotes the element in row i and column j of a matrix, and the diagonal dominance factor ρ is empirically set. Specifically, when $\rho = 1$, the resulting matrix $\boldsymbol{H}^{\mathrm{lm}}$ becomes diagonally dominant [35].

Furthermore, we may directly take the diagonal of $\boldsymbol{H}$ as an approximation

$$\boldsymbol{H}^{\mathrm{diag}} = \operatorname{diag}(\boldsymbol{H}). \tag{3.52}$$

An advantage of this diagonalization is that by using (3.49), we can significantly reduce the computational cost of the matrix-matrix multiplications and eliminate the cost of the matrix inversion of $\boldsymbol{H}$.

3.9.2 Noise Variances

For the full covariance matrix case of the noise variances, solving the re-estimation equation (3.21) is computationally very expensive. The computational cost can be greatly reduced if the covariance matrices $\boldsymbol{\Sigma}_n$, $\boldsymbol{\Sigma}_{y,jk}$, $\boldsymbol{\Sigma}_{y,jk}$, and $\boldsymbol{S}_{y,jk}$ are assumed to be diagonal. If $\boldsymbol{\Sigma}_n$ is diagonal, the optimization with respect to $\boldsymbol{\Sigma}_n$ only needs to consider its diagonal entries. Thus, the derivative of the Q function with respect to $\boldsymbol{\Sigma}_n$, (3.19), is diagonalized, and the same for both sides of (3.21). Furthermore, by approximating $\boldsymbol{A}_{jk}$ with their diagonals, we

can apply (3.49) to both $\boldsymbol{A}_{jk}$ and $\boldsymbol{B}_{jk}$ and simplify all of matrix-matrix multiplications in (3.21). Therefore, (3.21) can be solved in separate dimensions, and the noise variance at the ith dimension is given by

$$[\hat{\boldsymbol{\Sigma}}_n]_{ii} = [\boldsymbol{\Sigma}_n]_{ii} + \big(\sum_{j,k} \gamma_{jk} [\boldsymbol{A}_{jk}]_{ii}^2\big)^{-1} \sum_{j,k} [\boldsymbol{B}_{jk}]_{ii}. \tag{3.53}$$

The stand-alone re-estimation of $\boldsymbol{\Sigma}_n$ has a cost of $\mathcal{O}(4N^2M)$, where M is the total number of the mixture components. Remarkably, $\boldsymbol{A}_{jk}$ is the only approximated variable in solving the diagonal covariance $\boldsymbol{\Sigma}_n$. By noting that $\boldsymbol{A}_{jk}$ appears in the noise mean re-estimation (3.13), we may use the full matrix $\boldsymbol{A}_{jk}$ for free. However, informal experiments found no observable improvement in performance, when $\boldsymbol{A}_{jk}$ is full.

To prevent the noise variance estimate from being negative, which may happen at high SNR levels, simple variance flooring is used. Also, to stabilize the re-estimation iterations, the increment of the variance estimate at each iteration is constrained. Thus, we have

$$0 \le [\hat{\boldsymbol{\Sigma}}_n]_{ii} \le \eta [\boldsymbol{\Sigma}_n]_{ii} \tag{3.54}$$

where η is empirically set at 3.

3.10 Summary

In this chapter, we propose the Gauss-Newton method as a unified approach to optimize various nonlinear noise compensation models. Specifically, we present a novel noise variance estimation algorithm that conforms to the Gauss-Newton principle. The formulation of the Gauss-Newton method reduces the noise estimation problems to the determination of the Jacobians of the corrupted speech parameters with respect to the clean speech and noise parameters, which turns out to be the expectation of the sample Jacobians of the mismatch function. We present two methods, SJA and XCOV, to evaluate such Jacobians for the sampling-based compensation. From the perspective of XCOV, we show that in the noise compensation procedure, the nonlinear compensation model can be completely characterized by four statistics for each Gaussian component, the mean $\boldsymbol{\mu}_y$, the variance $\boldsymbol{\Sigma}_y$, and the cross-covariances $\boldsymbol{\Sigma}_{yx}$ and $\boldsymbol{\Sigma}_{yn}$. This observation coincides with what other researchers have observed from the feature-domain noise compensation schemes [4]. In addition, we

illustrate how to extend the proposed noise estimation algorithms for the incorporation of adaptive training. Though we describe the algorithms by assuming compensation in the model domain, they can be generalized to allow for the compensation in the feature-domain [63], [84] and the study of more complicated compensation models.

CHAPTER 4

EM-FA METHOD FOR NOISE ESTIMATION

In Chapter 3, we proposed the Gauss-Newton method as a unified approach for optimizing various nonlinear noise compensation models. This chapter will describe another popular noise estimation approach for optimizing the nonlinear compensation models. It views the model compensation from a generative perspective, giving rise to an EM-based algorithm analogous to the ML estimation for factor analysis (EM-FA) [74], [73]. In the EM-FA algorithm [73], clean speech and noise observations, as well as HMM states and mixture indices, are all regarded as latent variables, which make the M step of the EM algorithm relatively simple to solve. The method has been extended for the VTS compensation [45], VTS with adaptive training [38], and the UT compensation [19].

Our description of the algorithm is based on [73], [45], [38], but going beyond those works, we will show that the EM-FA algorithm is a particular instance of the gradient-based method. Moreover, we provide a complete formulation for estimating the static and dynamic parameters of the compensation models, and derive the channel mean estimation in a more intuitive way than the work originally presented in [45]. we shall also generalize the EM-FA algorithm for the optimization of sampling-based compensation models.

One of the main goals of this chapter is to demonstrate a close connection between the Gauss-Newton and the EM-FA methods: they both belong to the family of gradient-based methods except with different convergence rates. Building on this connection, an in-depth comparison of the two methods will be given in the end of this chapter. This comparison offers an insight into the relationships between the two algorithms, and demonstrates trade-off factors and feasibility of switching between these algorithms for a specific application. We believe that such a comparison could benefit the overall understanding of the nonlinear compensation methods for robust speech recognition.

Most of the discussion in this chapter assumes the VTS compensation model. To keep

notation simple, the superscripts of the Jacobians $\boldsymbol{G}_{n,jk}^{(0)}$ and $\boldsymbol{G}_{x,jk}^{(0)}$ are dropped with the understanding that they are the VTS versions of $\boldsymbol{G}_{n,jk}$ and $\boldsymbol{G}_{x,jk}$, respectively.

4.1 Estimating Static Noise Mean and Variance

The compensation models can be viewed from a generative prospective as represented by a dynamic Bayesian network (DBN) [64] shown in Figure 4.1. The model consists of two independent Markov chains: the clean speech process, which outputs $\boldsymbol{x}_t$ conditioned on state s_t at time t, and the noise process, which outputs $\boldsymbol{n}_t$ on state s'_t. Since a single Gaussian component is used to represent the noise process, the Markov chain of the noise degenerates to only one state. The corrupted speech observation $\boldsymbol{y}_t$ at time t depends only on the latent values of $\boldsymbol{x}_t$ and $\boldsymbol{n}_t$ at that time. To make the generative model tractable, the dependence of $\boldsymbol{y}_t$ on $\boldsymbol{x}_t$ and $\boldsymbol{n}_t$ can be linearized using the following VTS approximation, similar to one described in (2.5)

$$\boldsymbol{y}_t = \boldsymbol{G}_{x,jk}\boldsymbol{x}_t + \boldsymbol{G}_{n,jk}\boldsymbol{n}_t + \boldsymbol{g}(\boldsymbol{\mu}_{x,jk}, \boldsymbol{\mu}_n, \boldsymbol{\mu}_h) - \boldsymbol{G}_{x,jk}\boldsymbol{\mu}_{x,jk} - \boldsymbol{G}_{n,jk}\boldsymbol{\mu}_n \tag{4.1}$$

where $\boldsymbol{x}_t \sim \mathcal{N}(\hat{\boldsymbol{\mu}}_{x,jk}, \hat{\boldsymbol{\Sigma}}_{x,jk})$ and $\boldsymbol{n}_t \sim \mathcal{N}(\hat{\boldsymbol{\mu}}_n, \hat{\boldsymbol{\Sigma}}_n)$.

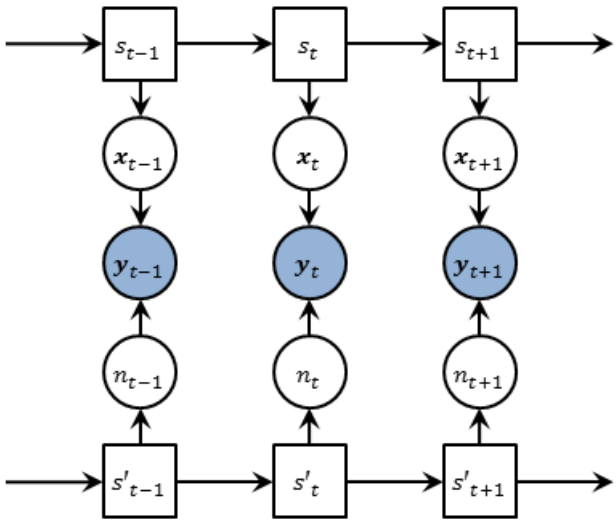


Figure 4.1: Dynamic Bayesian network representation of the compensation models. Square nodes denote discrete variables, circles continuous. Observed variables are shaded, latent variables unshaded.

The generative viewpoint motivates an EM-like algorithm for estimating parameters of

the compensation models, analogous to the ML estimation for factor analysis (EM-FA) [73]. The latent variables in the EM-FA algorithm consist of clean speech and noise outputs, as well as HMM states and mixture indices. The auxiliary function, retaining terms dependent on the static noise parameters, takes the form

$$Q(\hat{\boldsymbol{\theta}}|\boldsymbol{\theta}) = \sum_{t,j,k} \gamma_{jk}(t)\, \mathcal{E}\Big[\log p(\boldsymbol{x}_t, \boldsymbol{n}_t|j,k,\boldsymbol{\Lambda}_x, \hat{\boldsymbol{\theta}}) \Big| \boldsymbol{y}_t, j, k, \boldsymbol{\Lambda}_x, \boldsymbol{\theta} \Big] \tag{4.2}$$

where $\mathcal{E}[\,\cdot\,|\boldsymbol{y}_t, j, k, \boldsymbol{\Lambda}_x, \boldsymbol{\theta}]$ denotes the expectation with respect to $p(\boldsymbol{x}_t, \boldsymbol{n}_t|\boldsymbol{y}_t, j, k, \boldsymbol{\Lambda}_x, \boldsymbol{\theta})$, which is the joint posterior distribution of the two latent variables $\boldsymbol{x}_t$ and $\boldsymbol{n}_t$ given the corrupted speech observation $\boldsymbol{y}_t$ and Gaussian component (j,k) evaluated using the existing parameter values. Assuming $\boldsymbol{x}_t$ and $\boldsymbol{n}_t$ are independent, the Q function for estimating the noise mean and variance can be reduced to

$$\begin{aligned} Q(\hat{\boldsymbol{\theta}}|\boldsymbol{\theta}) = -\frac{1}{2} \sum_{t,j,k} \gamma_{jk}(t)\, \mathcal{E}\Big[& \log |\hat{\boldsymbol{\Sigma}}_n| \\ & + (\boldsymbol{n}_t - \hat{\boldsymbol{\mu}}_n)^{\mathrm{T}} \hat{\boldsymbol{\Sigma}}_n^{-1} (\boldsymbol{n}_t - \hat{\boldsymbol{\mu}}_n) \Big| \boldsymbol{y}_t, j, k, \boldsymbol{\Lambda}_x, \boldsymbol{\theta} \Big]. \end{aligned} \tag{4.3}$$

In the E step, we note that the Q function requires finding the sufficient statistics $\mathcal{E}[\boldsymbol{n}_t|\boldsymbol{y}_t, j, k, \boldsymbol{\Lambda}_x, \boldsymbol{\theta}]$ and $\mathcal{E}[\boldsymbol{n}_t\boldsymbol{n}_t^{\mathrm{T}}|\boldsymbol{y}_t, j, k, \boldsymbol{\Lambda}_x, \boldsymbol{\theta}]$. Since $\boldsymbol{n}_t$ and $\boldsymbol{y}_t$ are jointly Gaussian, the conditional distribution of $\boldsymbol{n}_t$ conditioned on $\boldsymbol{y}_t$ is again Gaussian. We have

$$p(\boldsymbol{n}_t|\boldsymbol{y}_t, j, k, \boldsymbol{\Lambda}_x, \boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{n}_t|\boldsymbol{\mu}_{n|y,jk}(t), \boldsymbol{\Sigma}_{n|y,jk}) \tag{4.4}$$

where

$$\boldsymbol{\mu}_{n|y,jk}(t) = \boldsymbol{\mu}_n + \boldsymbol{\Sigma}_{ny,jk}\boldsymbol{\Sigma}_{y,jk}^{-1}(\boldsymbol{y}_t - \boldsymbol{\mu}_{y,jk}) \tag{4.5}$$

$$\boldsymbol{\Sigma}_{n|y,jk} = \boldsymbol{\Sigma}_n - \boldsymbol{\Sigma}_{ny,jk}\boldsymbol{\Sigma}_{y,jk}^{-1}\boldsymbol{\Sigma}_{yn,jk} \tag{4.6}$$

$$\boldsymbol{\Sigma}_{yn,jk} = \boldsymbol{\Sigma}_{ny,jk}^{\mathrm{T}} = \boldsymbol{G}_{n,jk}\boldsymbol{\Sigma}_n. \tag{4.7}$$

Then

$$\mathcal{E}[\boldsymbol{n}_t|\boldsymbol{y}_t, j, k, \boldsymbol{\Lambda}_x, \boldsymbol{\theta}] = \boldsymbol{\mu}_{n|y,jk}(t) \tag{4.8}$$

$$\mathcal{E}[\boldsymbol{n}_t\boldsymbol{n}_t^{\mathrm{T}}|\boldsymbol{y}_t, j, k, \boldsymbol{\Lambda}_x, \boldsymbol{\theta}] = \boldsymbol{\Sigma}_{n|y,jk} + \boldsymbol{\mu}_{n|y,jk}(t)\boldsymbol{\mu}_{n|y,jk}^{\mathrm{T}}(t). \tag{4.9}$$

In the M step, the new estimate of the noise parameters can be obtained by setting the derivatives of Q function with respect to $\hat{\boldsymbol{\mu}}_n$ and $\hat{\boldsymbol{\Sigma}}_n$, respectively, to zero. This, by making use of (4.8) and (4.9), yields the following update formulas:

$$\begin{aligned}\hat{\boldsymbol{\mu}}_n &= \frac{1}{T}\sum_t \sum_{j,k} \gamma_{jk}(t)\mathcal{E}[\boldsymbol{n}_t|\boldsymbol{y}_t, j, k] \\ &= \frac{1}{T}\sum_{t,j,k} \gamma_{jk}(t)\boldsymbol{\mu}_{n|y,jk}(t) \qquad (4.10)\\ \hat{\boldsymbol{\Sigma}}_n &= \frac{1}{T}\sum_{t,j,k} \gamma_{jk}(t)\mathcal{E}[(\boldsymbol{n}_t - \hat{\boldsymbol{\mu}}_n)(\boldsymbol{n}_t - \hat{\boldsymbol{\mu}}_n)^{\mathrm{T}}|\boldsymbol{y}_t, j, k] \\ &= \frac{1}{T}\sum_{t,j,k} \gamma_{jk}(t)\Big[\boldsymbol{\Sigma}_{n|y,jk} + \boldsymbol{\mu}_{n|y,jk}(t)\boldsymbol{\mu}_{n|y,jk}^{\mathrm{T}}(t)\Big] - \hat{\boldsymbol{\mu}}_n\hat{\boldsymbol{\mu}}_n^{\mathrm{T}}. \qquad (4.11)\end{aligned}$$

It is remarkable that the above update formulas can be converted into forms based on the gradients of the standard HMM auxiliary function (3.7). Substituting (4.5) into (4.10) gives the following update equation for the noise mean

$$\hat{\boldsymbol{\mu}}_n = \boldsymbol{\mu}_n + \frac{1}{T}\boldsymbol{\Sigma}_n\Big[\sum_{j,k} \boldsymbol{G}_{n,jk}^{\mathrm{T}}\boldsymbol{\Sigma}_{y,jk}^{-1}\boldsymbol{c}_{y,jk}\Big]. \qquad (4.12)$$

For the noise variance, we substitute (4.5), (4.6), and (4.7) into (4.11), and obtain

$$\begin{aligned}\hat{\boldsymbol{\Sigma}}_n &= \frac{1}{T}\sum_{j,k} \gamma_{jk}\boldsymbol{\Sigma}_{n|y,jk} + \frac{1}{T}\sum_t\sum_{j,k} \gamma_{jk}(t)(\boldsymbol{\mu}_{n|y,jk}(t) - \hat{\boldsymbol{\mu}}_n)(\boldsymbol{\mu}_{n|y,jk}(t) - \hat{\boldsymbol{\mu}}_n)^{\mathrm{T}} \\ &= \frac{1}{T}\sum_{j,k} \gamma_{jk}\boldsymbol{\Sigma}_{n|y,jk} - (\hat{\boldsymbol{\mu}}_n - \boldsymbol{\mu}_n)(\hat{\boldsymbol{\mu}}_n - \boldsymbol{\mu}_n)^{\mathrm{T}} \\ &\quad + \frac{1}{T}\sum_{j,k} \gamma_{jk}(t)\boldsymbol{\Sigma}_{ny,jk}\boldsymbol{\Sigma}_{y,jk}^{-1}\boldsymbol{S}_{y,jk}\boldsymbol{\Sigma}_{y,jk}^{-1}\boldsymbol{\Sigma}_{yn,jk} \\ &= \boldsymbol{\Sigma}_n - (\hat{\boldsymbol{\mu}}_n - \boldsymbol{\mu}_n)(\hat{\boldsymbol{\mu}}_n - \boldsymbol{\mu}_n)^{\mathrm{T}} \\ &\quad + \frac{1}{T}\boldsymbol{\Sigma}_n\Big[\sum_{j,k} \boldsymbol{G}_{n,jk}^{\mathrm{T}}\boldsymbol{\Sigma}_{y,jk}^{-1}(\boldsymbol{S}_{y,jk} - \gamma_{jk}\boldsymbol{\Sigma}_{y,jk})\boldsymbol{\Sigma}_{y,jk}^{-1}\boldsymbol{G}_{n,jk}\Big]\boldsymbol{\Sigma}_n. \qquad (4.13)\end{aligned}$$

Compared to the original update formulas (4.10) and (4.11), the gradient-based equations are more compact by relating to the sufficient statistics γ_{jk}, $\boldsymbol{c}_{y,jk}$, and $\boldsymbol{S}_{y,jk}$, allowing considerable reduction in the computational time. More importantly, through the gradient-based form, we observe a close similarity between the EM-FA and the Gauss-Newton methods for addressing the noise estimation. Both methods belong to the family of

gradient-based methods, except with different update directions. The final section of this chapter will make a detailed comparison between these two methods.

We note that the convergence rate of the EM-FA algorithm is sensitive to the noise variance $\boldsymbol{\Sigma}_n$. The step sizes in (4.12) and (4.13) depend on $\boldsymbol{\Sigma}_n$. When $\boldsymbol{\Sigma}_n$ is initialized to a small value, the convergence is slow. If $\boldsymbol{\Sigma}_n$ is set to 0, there will be no update at all. The sensitivity of the convergence to the initial values will be examined in Section 5.1.

Finally, unlike the standard EM algorithm, which guarantees a monotonic convergence of the likelihood, the EM-FA method for optimizing the compensation models does not hold such a property in a strict sense, because the generative model defined by (2.5) is obtained through approximation and varies with each iteration. However, experiments show that the EM-FA method still exhibits a good convergence behavior in terms of either the model likelihood or the speech recognition accuracy.

4.2 Estimating Dynamic Noise Variance

For the estimation of the dynamic noise variances, one can view the dynamic noise, say delta noise, as Gaussian distributed with mean 0 and variance $\boldsymbol{\Sigma}_{\Delta n}$. Similar to the derivation for the static noise variance, the delta noise variance is estimated as

$$\hat{\boldsymbol{\Sigma}}_{\Delta n} = \boldsymbol{\Sigma}_{\Delta n} + \frac{1}{T}\boldsymbol{\Sigma}_{\Delta n}\Big[\sum_{j,k} \boldsymbol{G}_{n,jk}^{\mathrm{T}} \boldsymbol{\Sigma}_{\Delta y,jk}^{-1} (\boldsymbol{S}_{\Delta y,jk} - \gamma_{jk}\boldsymbol{\Sigma}_{\Delta y,jk}) \boldsymbol{\Sigma}_{\Delta y,jk}^{-1} \boldsymbol{G}_{n,jk}\Big]\boldsymbol{\Sigma}_{\Delta n}. \quad (4.14)$$

4.3 Estimating Channel Mean

Estimating the channel mean in the EM-FA scheme is a tricky issue. The channel is assumed to be a constant quantity in a given utterance. Suppose that we estimate the channel mean in a straightforward EM-FA fashion by regarding the channel observations as an additional set of latent variables. As mentioned in Section 4.1, this method would cause the channel mean not to be updated, as the channel variance is 0. If we, instead, assume a nonzero channel variance, and accordingly append a term like $\boldsymbol{G}_x\boldsymbol{\Sigma}_h\boldsymbol{G}_x^{\mathrm{T}}$ to (2.10), the variance of the channel may still be small, leading to a slow convergence.

An alternative approach is to think that the clean speech in the noisy environment becomes $\boldsymbol{x}'_t = \boldsymbol{x}_t + \boldsymbol{\mu}_h$ and is re-distributed as $\mathcal{N}(\boldsymbol{\mu}_{x,jk} + \boldsymbol{\mu}_h, \boldsymbol{\Sigma}_{x,jk})$, where $\boldsymbol{\mu}_{x,jk}$ and $\boldsymbol{\Sigma}_{x,jk}$

are the known prior parameters. This strategy is similar in spirit to the estimation of the convolutional channel in the signal bias removal (SBR) algorithm [71]. The corresponding auxiliary function, retaining the terms dependent on the channel, is given by

$$Q(\hat{\boldsymbol{\theta}}|\boldsymbol{\theta}) = \sum_{t,j,k} \gamma_{jk}(t)\, \mathcal{E}\Big[(\boldsymbol{x}'_t - \boldsymbol{\mu}_{x,jk} - \hat{\boldsymbol{\mu}}_h)^{\mathrm{T}} \boldsymbol{\Sigma}_{x,jk}^{-1} (\boldsymbol{x}'_t - \boldsymbol{\mu}_{x,jk} - \hat{\boldsymbol{\mu}}_h) \Big| \boldsymbol{y}_t, j, k, \boldsymbol{\Lambda}_x, \boldsymbol{\theta}\Big]. \quad (4.15)$$

Differentiating the auxiliary function with respect to $\hat{\boldsymbol{\mu}}_h$ and equating it to 0, we have

$$\hat{\boldsymbol{\mu}}_h = \Big[\sum_{j,k} \gamma_{jk} \boldsymbol{\Sigma}_{x,jk}^{-1}\Big]^{-1} \sum_{t,j,k} \gamma_{jk}(t) \boldsymbol{\Sigma}_{x,jk}^{-1} (\boldsymbol{\mu}_{x'|y,jk}(t) - \boldsymbol{\mu}_{x,jk}) \quad (4.16)$$

where

$$\boldsymbol{\mu}_{x'|y,jk}(t) = \boldsymbol{\mu}_{x,jk} + \boldsymbol{\mu}_h + \boldsymbol{\Sigma}_{xy,jk} \boldsymbol{\Sigma}_{y,jk}^{-1} (\boldsymbol{y}_t - \boldsymbol{\mu}_{y,jk}) \quad (4.17)$$

$$\boldsymbol{\Sigma}_{yx,jk} = \boldsymbol{\Sigma}_{xy,jk}^{\mathrm{T}} = \boldsymbol{G}_{x,jk} \boldsymbol{\Sigma}_{x,jk}. \quad (4.18)$$

Similarly, we can convert the update formula for $\boldsymbol{\mu}_h$ into a gradient-based form by substituting (4.17) into (4.16)

$$\hat{\boldsymbol{\mu}}_h = \boldsymbol{\mu}_h + \Big[\sum_{j,k} \gamma_{jk} \boldsymbol{\Sigma}_{x,jk}^{-1}\Big]^{-1} \sum_{j,k} \boldsymbol{G}_{x,jk}^{\mathrm{T}} \boldsymbol{\Sigma}_{y,jk}^{-1} \boldsymbol{c}_{y,jk}. \quad (4.19)$$

It is remarkable that the above channel re-estimation formula is the same as the one used in [45], [38] for updating the channel convolution, except with a much intuitive derivation. The channel re-estimation equation presented in [45], [38] is given by

$$\begin{aligned}\hat{\boldsymbol{\mu}}_h = &\Big[\sum_{t} \sum_{j,k} \gamma_{jk} \boldsymbol{G}_{x,jk}^{\mathrm{T}} \boldsymbol{\Sigma}_{y|n,jk}^{-1} \boldsymbol{G}_{x,jk}\Big]^{-1} \sum_{t} \sum_{j,k} \gamma_{jk} \boldsymbol{G}_{x,jk}^{\mathrm{T}} \boldsymbol{\Sigma}_{y|n,jk}^{-1} \\ &\times \big[\boldsymbol{y}_t + \boldsymbol{G}_{x,jk} \boldsymbol{\mu}_h - \boldsymbol{\mu}_{y,jk} - \boldsymbol{\Sigma}_{yn,jk} \boldsymbol{\Sigma}_n^{-1} (\boldsymbol{\mu}_{n|y,jk}(t) - \boldsymbol{\mu}_n))\big]. \end{aligned} \quad (4.20)$$

where

$$\boldsymbol{\Sigma}_{y|n,jk} = \boldsymbol{\Sigma}_{y,jk} - \boldsymbol{\Sigma}_{yn,jk} \boldsymbol{\Sigma}_n^{-1} \boldsymbol{\Sigma}_{ny,jk} = \boldsymbol{G}_{x,jk} \boldsymbol{\Sigma}_{x,jk} \boldsymbol{G}_{x,jk}^{\mathrm{T}}. \quad (4.21)$$

Substituting (4.21), (4.5), and (4.7) into (4.20) yields

$$\begin{aligned}\hat{\boldsymbol{\mu}}_h &= \boldsymbol{\mu}_h + \Big[\sum_{j,k} \gamma_{jk} \boldsymbol{\Sigma}_{x,jk}^{-1}\Big]^{-1} \sum_{t} \sum_{j,k} \gamma_{jk}(t) \boldsymbol{\Sigma}_{x,jk}^{-1} \boldsymbol{G}_{x,jk}^{-1} \\ &\quad \times \big[(\boldsymbol{y}_t - \boldsymbol{\mu}_{y,jk}) - \boldsymbol{G}_{n,jk} \boldsymbol{\Sigma}_n \boldsymbol{G}_{n,jk}^{\mathrm{T}} \boldsymbol{\Sigma}_{y,jk}^{-1} (\boldsymbol{y}_t - \boldsymbol{\mu}_{y,jk})\big] \\ &= \boldsymbol{\mu}_h + \Big[\sum_{j,k} \gamma_{jk} \boldsymbol{\Sigma}_{x,jk}^{-1}\Big]^{-1} \sum_{j,k} \boldsymbol{G}_{x,jk}^{\mathrm{T}} \boldsymbol{\Sigma}_{y,jk}^{-1} \boldsymbol{c}_{y,jk}. \end{aligned} \quad (4.22)$$

The second equality merges the terms dependent on $(\boldsymbol{y}_t - \boldsymbol{\mu}_{y,jk})$ and uses (2.10).

4.4 EM-FA for Sampling-Based Compensation

So far, the EM-FA method has been confined to the VTS compensation model. The derivation can be generalized, however, to allow for the more complicated models introduced in Section 2.3. This generalization has been described in [19] to estimate noise parameters of the UT compensation. Here we summarize it in the more general case of sampling-based compensation. In retrospect of the EM-FA derivation, we note that the representation as a factor analysis model is equally applicable to the sampling-based compensation. The difficulty arises from evaluating $\mathcal{E}[\boldsymbol{n}_t|\boldsymbol{y}_t, j, k, \boldsymbol{\theta}]$ and $\mathcal{E}[\boldsymbol{n}_t\boldsymbol{n}_t^{\mathrm{T}}|\boldsymbol{y}_t, j, k, \boldsymbol{\theta}]$ for sampling-based compensation. If we again assume $\boldsymbol{n}_t$ and $\boldsymbol{y}_t$ to be jointly Gaussian[1] and use the results about joint Gaussian variables, (4.5) and (4.6), the problem is reduced to determining the cross-covariance $\boldsymbol{\Sigma}_{yn,jk}$. This term can be numerically evaluated using (3.40). It turns out that the formulation of the EM-FA algorithm is applicable to the sampling-based compensation, except that we need to reversely solve for $\boldsymbol{G}_{n,jk}$ using $\boldsymbol{\Sigma}_{yn,jk}$ through (4.7).

Moreover, we observe a further correspondence between the EM-FA and the Gauss-Newton methods: they involve identical expressions of $\boldsymbol{G}_{n,jk}$ and $\boldsymbol{G}_{x,jk}$ for both VTS and sampling-based compensations.

4.5 Comparing Gauss-Newton with EM-FA

The Gauss-Newton and the EM-FA methods described in the previous and current chapters represent two major techniques for estimating noise parameters of the nonlinear compensation models. We have demonstrated that both techniques belong to the family of the gradient-based approach. Building on this relationship, we here give a more detailed comparison between these two methods and address their respective advantages and limitations.

At first glance, it appears that the comparison is more concerned with the maximization of the auxiliary function (3.2) than the maximization of the overall likelihood. However, for unsupervised compensation, as is often the case in practice, the E step of the first EM re-estimation will produce the sufficient statistics that are accurate enough for the following parameter re-estimations. Hence, maximizing the auxiliary function (3.2) is nearly

[1] The joint Gaussian statement is not a natural consequence of the sampling-based compensation models.

equivalent to maximizing the overall likelihood in terms of convergence properties.

The main challenge of the noise estimation is that the maximization of the standard auxiliary function (3.2) does not have a closed-form solution. The Gauss-Newton method tackles the problem by embedding itself in a generalized EM framework and iteratively maximizing the standard auxiliary function. By contrast, the EM-FA method views the model compensation from a generative perspective and employs a different auxiliary function. The corresponding M step turns out to be the gradient-based update with respect to the original auxiliary function (3.2). Both approaches can be expressed in a gradient-based update structure

$$\hat{\boldsymbol{\theta}} = \boldsymbol{\theta} - \boldsymbol{H}^{-1} \frac{\partial Q}{\partial \boldsymbol{\theta}} \tag{4.23}$$

where Q denotes the auxiliary function (3.2) and $\boldsymbol{H}$ denotes a Hessian-like matrix used in the two methods. The inverse matrix $\boldsymbol{H}^{-1}$ acts as the step size of the update equations. Table 4.1 presents the gradients and the Hessian approximations for these two optimization methods with respect to various parameters. Note that the noise variance estimation is expressed in a vector form of $\boldsymbol{\Sigma}_n$ to fit the update equation (4.23).

The two methods exhibit a significant correspondence with the same gradient term for re-estimating means and variances of speech and noise models. This is not surprising as the iterative updating procedure should move the solutions toward its stationary points. Also because of this correspondence, when we extend both methods to tackle the sampling-based compensation models, some equivalence of the Jacobian matrices is anticipated; this is indeed the case. Moreover, the approximated Hessian $\boldsymbol{H}$ in both methods are guaranteed to be negative-semidefinite, which ensures an ascending search direction whenever the existing estimate is not a stationary point. This property differentiates the two methods, with a great advantage, from the generic gradient-based methods. Finally, it is truly surprising to observe an aesthetic connection between $\boldsymbol{H}$ for updating means and $\boldsymbol{H}$ for updating variances. That is, if we represent $\boldsymbol{H}$ for updating the mean as $\sum_{j,k} \gamma_{jk} \boldsymbol{A}_{jk}$, then $\boldsymbol{H}$ for updating the corresponding variance can be written as $\sum_{j,k} \gamma_{jk} \boldsymbol{A}_{jk} \otimes \boldsymbol{A}_{jk}$. Though we notice this coincidence in both the Gauss-Newton and the em-FA methods, it has not been

Table 4.1: Choice of the gradient and Hessian approximation in the Gauss-Newton and the EM-FA Methods. The update equations for static parameters are tabulated. The re-estimation of the dynamic parameters, which include $\hat{\boldsymbol{\Sigma}}_{\Delta n}$, $\hat{\boldsymbol{\Sigma}}_{\Delta^2 n}$, $\hat{\boldsymbol{\mu}}_{\Delta x,jk}$, $\hat{\boldsymbol{\mu}}_{\Delta^2 x,jk}$, $\hat{\boldsymbol{\Sigma}}_{\Delta x,jk}$, and $\hat{\boldsymbol{\Sigma}}_{\Delta^2 x,jk}$, can be formulated in a similar way as for their corresponding static parameters. Moreover, for the EM-FA method, $\hat{\boldsymbol{\Sigma}}_n$ should be additionally subtracted by $(\hat{\boldsymbol{\mu}}_n - \boldsymbol{\mu}_n)(\hat{\boldsymbol{\mu}}_n - \boldsymbol{\mu}_n)^{\mathrm{T}}$, due to the change in the noise mean estimate. Similarly, $\hat{\boldsymbol{\Sigma}}_{x,jk}$ should be subtracted by $(\hat{\boldsymbol{\mu}}_{x,jk} - \boldsymbol{\mu}_{x,jk})(\hat{\boldsymbol{\mu}}_{x,jk} - \boldsymbol{\mu}_{x,jk})^{\mathrm{T}}$, and the same for $\hat{\boldsymbol{\Sigma}}_{\Delta x,jk}$ and $\hat{\boldsymbol{\Sigma}}_{\Delta^2 x,jk}$.

Parameter	Gradient ($\frac{\partial Q}{\partial \boldsymbol{\theta}}$)	Minus Hessian approximation ($-\boldsymbol{H}$)	
		Gauss-Newton	EM-FA
$\hat{\boldsymbol{\mu}}_n$	$\sum_{j,k} \boldsymbol{G}_{n,jk}^{\mathrm{T}} \boldsymbol{\Sigma}_{y,jk}^{-1} \boldsymbol{c}_{y,jk}$	$\sum_{j,k} \gamma_{jk} \boldsymbol{G}_{n,jk}^{\mathrm{T}} \boldsymbol{\Sigma}_{y,jk}^{-1} \boldsymbol{G}_{n,jk}$	$T \boldsymbol{\Sigma}_n^{-1}$
$\hat{\boldsymbol{\mu}}_h$	$\sum_{j,k} \boldsymbol{G}_{x,jk}^{\mathrm{T}} \boldsymbol{\Sigma}_{y,jk}^{-1} \boldsymbol{c}_{y,jk}$	$\sum_{j,k} \gamma_{jk} \boldsymbol{G}_{x,jk}^{\mathrm{T}} \boldsymbol{\Sigma}_{y,jk}^{-1} \boldsymbol{G}_{x,jk}$	$\sum_{j,k} \gamma_{jk} \boldsymbol{\Sigma}_{x,jk}^{-1}$
$\mathrm{vec}(\hat{\boldsymbol{\Sigma}}_n)$	$\mathrm{vec}\Big(\sum_{j,k} \boldsymbol{G}_{n,jk}^{\mathrm{T}} \boldsymbol{\Sigma}_{y,jk}^{-1} (\boldsymbol{S}_{y,jk} - \gamma_{jk} \boldsymbol{\Sigma}_{y,jk}) \boldsymbol{\Sigma}_{y,jk}^{-1} \boldsymbol{G}_{n,jk} \Big)$	$\sum_{j,k} \gamma_{jk} \Big(\boldsymbol{G}_{n,jk}^{\mathrm{T}} \boldsymbol{\Sigma}_{y,jk}^{-1} \boldsymbol{G}_{n,jk} \Big) \otimes \Big(\boldsymbol{G}_{n,jk}^{\mathrm{T}} \boldsymbol{\Sigma}_{y,jk}^{-1} \boldsymbol{G}_{n,jk} \Big)$	$T \boldsymbol{\Sigma}_n^{-1} \otimes \boldsymbol{\Sigma}_n^{-1}$
$\hat{\boldsymbol{\mu}}_{x,jk}$	$\sum_r \boldsymbol{G}_{x,jk}^{(r)\,\mathrm{T}} \boldsymbol{\Sigma}_{y,jk}^{(r)\,-1} \boldsymbol{c}_{y,jk}^{(r)}$	$\sum_r \gamma_{jk}^{(r)} \boldsymbol{G}_{x,jk}^{(r)\,\mathrm{T}} \boldsymbol{\Sigma}_{y,jk}^{(r)\,-1} \boldsymbol{G}_{x,jk}^{(r)}$	$\Big(\sum_r \gamma_{jk}^{(r)} \Big) \boldsymbol{\Sigma}_{x,jk}^{-1}$
$\mathrm{vec}(\hat{\boldsymbol{\Sigma}}_{x,jk})$	$\mathrm{vec}\Big(\sum_r \boldsymbol{G}_{x,jk}^{(r)\,\mathrm{T}} \boldsymbol{\Sigma}_{y,jk}^{(r)\,-1} (\boldsymbol{S}_{y,jk}^{(r)} - \gamma_{jk}^{(r)} \boldsymbol{\Sigma}_{y,jk}^{(r)}) \boldsymbol{\Sigma}_{y,jk}^{(r)\,-1} \boldsymbol{G}_{x,jk}^{(r)} \Big)$	$\sum_r \gamma_{jk}^{(r)} \Big(\boldsymbol{G}_{x,jk}^{(r)\,\mathrm{T}} \boldsymbol{\Sigma}_{y,jk}^{(r)\,-1} \boldsymbol{G}_{x,jk}^{(r)} \Big) \otimes \Big(\boldsymbol{G}_{x,jk}^{(r)\,\mathrm{T}} \boldsymbol{\Sigma}_{y,jk}^{(r)\,-1} \boldsymbol{G}_{x,jk}^{(r)} \Big)$	$\Big(\sum_r \gamma_{jk}^{(r)} \Big) \boldsymbol{\Sigma}_{x,jk}^{-1} \otimes \boldsymbol{\Sigma}_{x,jk}^{-1}$

clear what mechanism is responsible for this phenomenon.

The different update directions of the two methods give rise to distinct convergence properties. As a typical EM method, the convergence rate of EM-FA is linear [61], which is inferior to the Gauss-Newton update whose convergence rate can approach quadratic. Note that the convergence property can be crucial to the noise estimation in many applications where model compensation may have to be frequently carried out in changing noisy environments to retain desired performance.

Whilst the Gauss-Newton method achieves a super-linear convergence rate, it saves the calculation of the second-order derivatives. Also, by exploiting the problem structure, we have shown in Section 3.9 the significant reduction of the computational overhead in evaluating the approximated Hessian matrix. As such, the Gauss-Newton method outperforms the EM-FA method in rate of convergence at the expense of a moderate increase in computational cost.

The EM-FA method, though converging linearly, differentiates itself from the common first-order optimization methods like gradient ascent, as its step size changes dynamically with the existing parameter values, and its iteration sequence exhibits a good convergence property.

A more subtle difference comes from the treatment to the optimization constraints. The EM algorithm can naturally embed within it the probabilistic constraints of the optimization parameters. Specifically, it guarantees the estimated noise variance as (4.11) to be positive definite. The Gauss-Newton method, however, addresses unconstrained optimization problems in its plain form, which means it needs to check and maintain the positiveness of the noise variances. Since the noise covariance matrix is assumed to be diagonal, simple variance flooring can be used to prevent it from being negative.

Nevertheless, the noise variance can be viewed as a bias to the speech variances, and thus possible to be negative-valued in some scenarios, as long as the compensated speech variances are positive. One specific application is to adapt the multistyle acoustic models to a specific noise condition by narrowing the variances, such that the adapted models become sharp and optimal to the test condition [58]. The Gauss-Newton method can handle this

loosened variance constraint directly other than the EM-FA method.

Finally, the EM-FA method has an advantage over the Gauss-Newton method in that it is relatively easy to incorporate discriminative training criteria [22]. Discriminative training methods such as maximum mutual information (MMI) [9], minimum classification error (MCE) [41], and minimum phone/word error (MPE/MWE) [69], have been shown to significantly boost the recognition performance on large-vocabulary continuous speech recognition (LVCSR) tasks. Other than maximizing the model likelihood, discriminative criteria aim to explicitly minimize objective functions that are more closely related to the recognition error rate. Since the EM-FA method is essentially an EM method, it can be readily extended to support discriminative training, in an analogous way to that used to obtain the update formulas for discriminative training of standard acoustic models. In [22], the EM-FA method has been modified to discriminatively refine the canonical speech models in the adaptive training scheme using the MPE criterion. Extending the Gauss-Newton method to support discriminative training is problematic. We may derive the Gauss-Newton update formulas for discriminative training by applying the concept of weak-sense auxiliary function [69]. However, the second-order nature of the Gauss-Newton method substantially complicates the settings of the smoothing constant used in the auxiliary function and other control constants, which are known to be important for stable convergence of discriminative training. There has not been such work reported in the literature.

The advantages and disadvantages of the Gauss-Newton and the em-FA methods in the context of solving the noise estimation problem are summarized in Table 4.2. We emphasize that the two methods are sufficiently general and can be applied to the optimization of a variety of structured HMM models, where Gaussian means and variances are (nonlinear) functions of optimization variables. This comparison will help us to determine the best scheme for the application of interest.

4.6 Summary

This chapter describes the EM-FA method for estimating noise parameters of the nonlinear compensation models. The EM-FA method views the model compensation from a generative

Table 4.2: Properties of the Gauss-Newton and the em-FA methods for optimizing nonlinear compensation models.

Gauss-Newton
1. Approaches quadratic convergence rate.
2. Saves the calculation of the second-order derivative.
3. Guarantees a descent search direction because the approximated Hessian is negative-semidefinite.
4. Is capable of unconstrained optimization in its plain form.
5. Can be used to estimate a negative-valued variance bias.
EM-FA
1. Converges linearly.
2. Exhibits good convergence behavior in terms of either likelihood or recognition accuracy.
3. Does not guarantee monotonic convergence because the generative model is approximated at each iteration.
4. Is sensitive to initial values of noise variances.
5. Naturally embeds within it the probabilistic constraints.
6. Can be extended to incorporate discriminative training criteria.

perspective, giving rise to an EM algorithm for factor analysis. Specifically, we show that the EM-FA method is an instance of the gradient-based method. Therefore, both the EM-FA method and the Gauss-Newton method, which was proposed in the previous chapter, belong to the family of gradient-based methods and possess a pervasive correspondence in the estimation of different model parameters for various compensation models. A detailed comparison between the Gauss-Newton and the em-FA methods from a general optimization prospective is also presented. The major advantages of the Gauss-Newton method consist of achieving the super-linear convergence rate and saving the cumbersome computation of the second-order derivatives. In contrast, the EM-FA method, as fully derived in an EM framework, inherits many properties from EM, such as linear and stable convergence,

relatively simple maximization step, and embedding the probabilistic constraints. Also, the EM-FA method can be readily employed for discriminative training.

CHAPTER 5

EXPERIMENTS WITH NONLINEAR COMPENSATION

This chapter presents the experimental results pertaining to the effectiveness of the nonlinear compensation models and the proposed noise estimation algorithms. The noise estimation algorithms are evaluated for various compensation models on two tasks. The first is to fit a GMM model to artificially corrupted signals that are generated through a Monte Carlo simulation. The second is to perform speech recognition on the Aurora 2 database [34].

To simplify the experimental procedure and analysis, VTS compensation with the Gauss-Newton method (VTS-GN) is used as a reference system. It is compared to other noise estimation techniques under the same VTS compensation settings, or other compensation models by fixing the noise estimation algorithm to Gauss-Newton. The effect of an arbitrary combination of a compensation model and a noise estimation algorithm can be reasonably reflected through the impacts of its composing modules.

5.1 Simulation on a GMM Fitting Task

In the GMM fitting task, the mismatch function is simplified to the form

$$\boldsymbol{y} = \boldsymbol{x} + \mathbf{log}(1 + \mathbf{exp}(\boldsymbol{n} - \boldsymbol{x})) \tag{5.1}$$

which can be viewed as a mismatch in the log-spectrum domain without the convolutional noise and the dynamic features.

The test data are generated through Monte Carlo simulation[1] as follows. The clean signal $\boldsymbol{x}$ is 8-dimensional and consists of eight Gaussian components. The Gaussian components are chosen with mean values uniformly drawn from -20 to 20 and variance values from 1/4 to 16 along each dimension. Then, 125 observations are drawn from each of the

[1]The Monte Carlo invoked here is just for the sake of generating the test data, irrespective of the sampling-based compensation. In this section, test data are specifically referred to as "observations", and the term "sample" indicates samples generated by the sampling-based compensation.

eight components, totaling 1000 observations of $\boldsymbol{x}$. The clean signal is then contaminated through the mismatch function (5.1) with noise $\boldsymbol{n}$ that is drawn from a Gaussian distribution with mean 0 and variance 4 along each dimension. Eight test sets are generated by repeating the above procedure.

The noise estimation algorithms are carried out for each test set with different initial noise means and variances. The initial means range from -2 to 2 at a step size of 0.5, and the initial variances range from 1/8 to 32 increasing by a factor of 2. The EM iterations are stopped whenever the log-likelihood of the transformed GMM fails to change (increase or decrease) by a certain threshold (0.1%) over the previous iteration. The Kullback-Leibler (KL) divergence [49] of the noise estimate to the true noise distribution is also used to measure the algorithmic accuracy. Not all iterations will converge to the true noise distributions. Some noise estimation methods, when beginning from a specific initial noise values, may diverge, or stop before they converge due to a slow change. These iterations are counted separately from the converged iterations in the result analysis. It is admitted that the non-converged iterations can be remedied through a number of ways, such as backing-off to the gradient ascent method. However, in this GMM fitting experiment, we are more concerned with the properties of the individual noise estimation methods.

5.1.1 Comparison of Noise Estimation Methods

In Table 5.1, we provide a comparison of four noise estimation algorithms, Gauss-Newton, Newton, EM-FA, and gradient ascent (denoted by *Gradient* in the table), for the VTS compensation in terms of the convergence and algorithmic accuracy. Newton's method is implemented in a similar way to the Gauss-Newton method except with an exact Hessian matrix. The step size for the gradient ascent method is set to $1/T$, where T is the number of observations. The high rate (43.36%) of excluded runs for Newton's method is mainly due to diverged iterations, as the Hessian matrix of Newton's method is indefinite. The excluded runs for the gradient ascent and EM-FA methods are mainly attributed to the slow increment in the log -likelihood. For the converged runs, the Gauss-Newton method converges fastest among the four algorithms. All noise estimation algorithms except for the

gradient ascent achieve the same level of accuracy in the end.

Table 5.1: Algorithmic accuracy and convergence for four noise estimation methods with the VTS compensation in a GMM fitting task.

System	Excluded runs (%)	# of iterations mean ± std. dev.	Average log-likeli.	KL divergence
VTS-GN	0.15	3.29±0.75	-17.970	0.446
VTS-Newton	43.36	5.09±1.92	-17.973	0.372
VTS-EM-FA	13.43	8.07±3.66	-17.998	0.511
VTS-Gradient	36.11	12.27±7.24	-18.261	1.083

To gain some insight into the behavior of the optimization algorithms, Figure 5.1 shows the changes in the average log-likelihood and KL divergence of the noise estimate at each iteration in the re-estimation procedure, where the stopping criterion is disabled. Apart from what can be observed from Table 5.1, we see that all algorithms except Newton's method show a monotonic decrease in the log-likelihood with different convergence rates. The convergence curve of Newton's method varies acutely because its Hessian is indefinite. This observation provides an account for why the Gauss-Newton method converges faster than Newton's method, though both methods may converge quadratically if the starting point is close to the optimum.

We have noted in Chapter 4 that the convergence rate of the EM-FA method is sensitive to the initial values. This effect can be seen by plotting the number of iterations required with respect to the initial values of the noise mean and variance, as shown in Figure 5.2. The true distribution of the noise signals is with mean 0 and variance 4. However, the linear approximation of VTS introduces a systematic bias in the estimation of the noise mean, which shifts from 0 to around 0.5. This bias can also be observed in Figure 5.2, where the point with minimum iterations is around mean 0.5 and variance 4. It is observed that in regions centered around this limit point, the change in the iteration number is roughly proportional to the change in the initial means and the change in the logarithm of initial variances. Moreover, when both initial noise means and variances are small, the convergence becomes substantially slow.

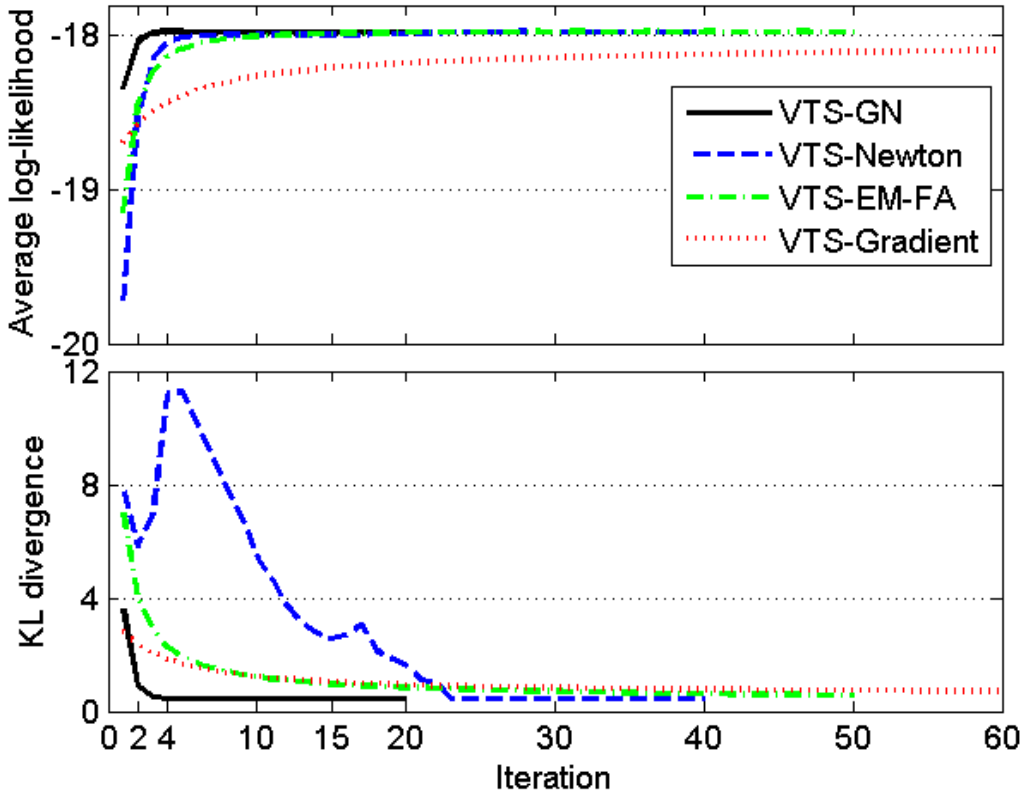


Figure 5.1: Log-likelihood and KL divergence of the noise estimate as a function of the iterations for three noise estimation methods using the VTS compensation in a GMM fitting task.

5.1.2 Comparison of Compensation Models

Table 5.2 compares the performance of three compensation models, VTS, UT, and DPMC, using the Gauss-Newton method. Two Jacobian evaluation methods, SJA and XCOV, are also examined for UT and DPMC models, respectively. Figure 5.3 shows the changes in the number of iterations and the KL divergence with respect to the number of Monte Carlo samples per Gaussian in DPMC, where several selective sample numbers (50, 100, and 200) are detailed in Table 5.2. We see that DPMC with more than 100 samples per Gaussian produces significantly lower KL divergence than the VTS and UT models. This confirms that the sampling-based compensation yields more accurate models than the VTS compensation, which ignores the higher-order effect of the nonlinear mismatch function.

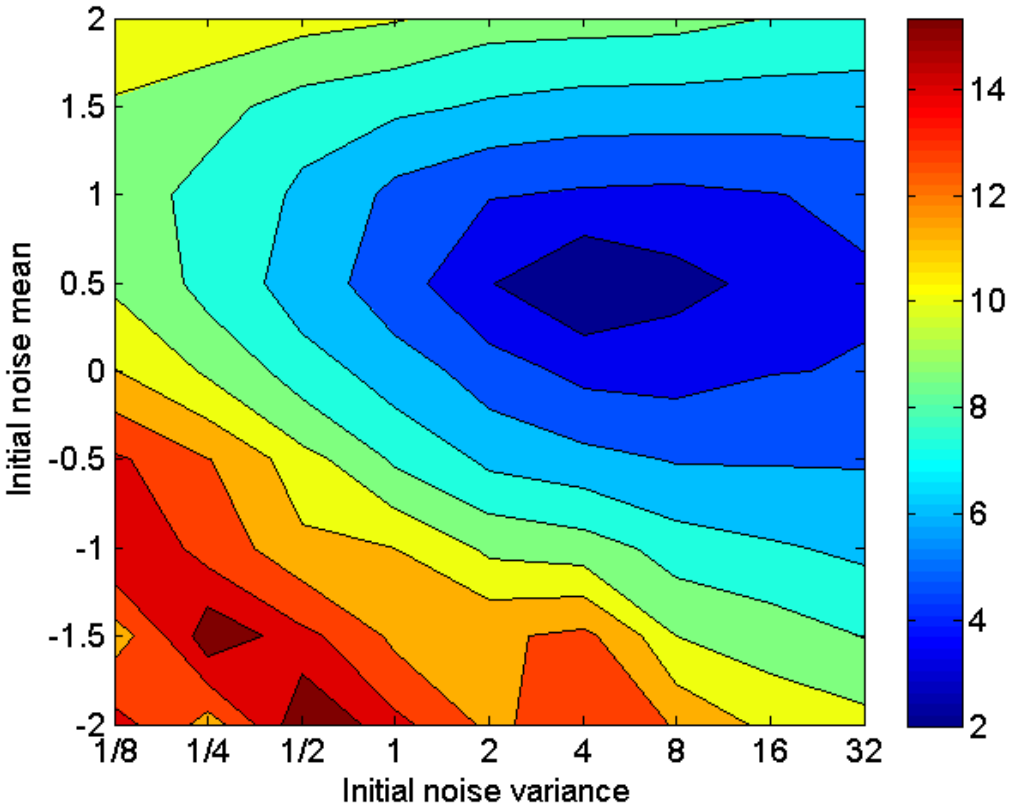


Figure 5.2: Contour of the number of iterations with respect to the initial values for the VTS compensation using the EM-FA method in a GMM fitting task. Note that the horizontal axis (i.e., initial noise variance) is in log-scale.

Table 5.2: Algorithmic accuracy and convergence of the sampling-based compensation using the Gauss-Newton method in a GMM fitting task.

System	# of samples	Excluded runs (%)	# of iterations mean ± std. dev.	Average log-likeli.	KL divergence
UT-SJA	32	0.31	3.62± 1.08	-17.925	0.572
UT-XCOV	32	8.33	3.27± 0.77	-17.900	0.348
DPMC-SJA	50	2.01	13.18± 8.57	-17.997	0.110
	100	3.86	4.46± 1.99	-17.916	0.072
	200	3.86	4.49± 1.69	-17.880	0.050
DPMC-XCOV	50	2.16	13.37±10.56	-17.993	0.276
	100	3.24	8.48± 6.73	-17.921	0.063
	200	4.78	4.72± 1.79	-17.874	0.052

The VTS model achieves a similar likelihood to the sampling-based models at the expense of a biased noise estimate[2]. It should be noted that from a classification perspective,

[2]The bias in the noise mean estimate of the VTS compensation can be observed from Figure 5.2, where

the log-likelihood is more pertinent to the system performance than the KL divergence. Thus, the VTS model may not perform worse than the sampling-based model for the speech recognition task, as we shall see in the next section.

It is unexpected that the UT compensation does not show gains in KL divergence over VTS or DPMC with a similar size of samples. Figure 5.3 may provide some insights. We observe that when the samples are small, DPMC takes more iterations to converge. This is possibly because many iterations of random sampling may produce the effect of stochastically approximating the true model parameters [11]. In contrast, samples in UT are chosen in a deterministic fashion, which may introduce a bias to the noise estimate.

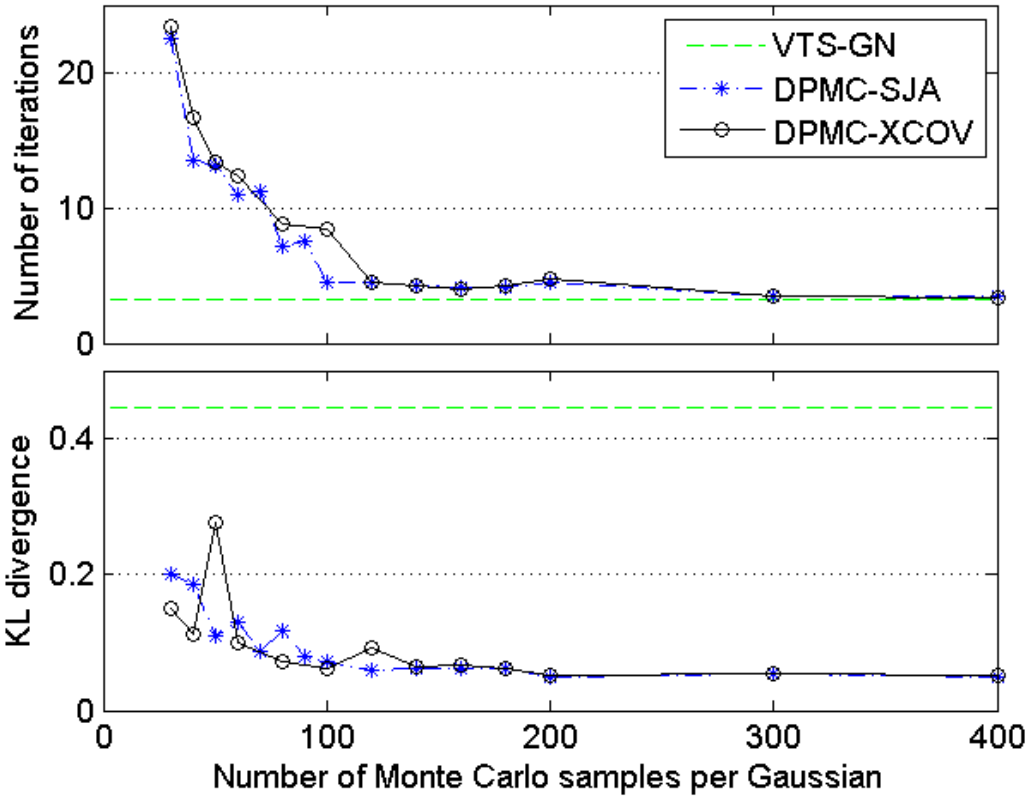


Figure 5.3: The number of iterations used and the KL divergence with respect to the number of Monte Carlo samples per Gaussian for the DPMC compensation in a GMM fitting task. VTS-GN is plotted as a reference.

There is no significant difference between the SJA and XCOV methods. This can be

the stationary point is around 0.5, not 0, on the vertical axis.

attributed to the fact that the Jacobian matrix $\boldsymbol{G}_n$ in the GMM fitting task is diagonal, and has 8 degrees of freedom in both Jacobian evaluation methods.

5.2 Recognition on Aurora 2 Database

In this task, the proposed algorithm was evaluated on the Aurora 2 database [34] of connected digits. Aurora 2 provides two training sets, clean and multistyle, each containing 8,440 utterances. Specifically, the multistyle training set consists of noisy data involving four types of noise (subway, babble, car, and exhibition hall) at four SNRs (20, 15, 10, and 5 dB), along with clean data. The test set consists of three different parts. Test Set A contains four types of noise as in the multistyle training data; Set B comprises four different noises: restaurant, street, airport, and station; the data in Set C are contaminated with two additive noises (subway and street) as well as channel distortion. For each noise type, a subset of 1001 clean speech utterances is contaminated at SNRs ranging from 20 to -5 dB at a 5 dB step size, which, including the clean condition, result in seven different SNR levels. The word error rate (WER) averaged over SNRs between 20 and 0 dB of the three test sets is used to measure the system performance as suggested in [34].

The acoustic models are trained following the standard Aurora 2 recipe for the simple back-end. Each digit is modeled by a whole-word left-to-right HMM, consisting of 16 states and three Gaussian components per state. Besides, a 3-state silence model and a 1-state short pause model with six Gaussian components per state are used. Each feature frame is characterized by 39-dimensional MFCCs with the zeroth cepstral coefficient as the energy term. The cepstra are computed based on spectral magnitude.

For most of the compensation experiments, the standard compensation scheme as described in Section 3.8 is performed, where the acoustic models are built using the Aurora 2 clean training data. The clean baseline system produces a WER of 41.57%. For the experiments with adaptive training, the Aurora 2 multistyle training data are used for obtaining the canonical speech model.

During the compensation, the first and last 20 frames of each utterance, assumed to be non-speech signals, are used for initializing the means and variances of the additive noise.

The channel bias vector is initialized to 0. For the Gauss-Newton method, the Hessian form (3.50) with $\rho = 0.4$ is used. We will examine how variations of the factor ρ affect the recognition results later.

5.2.1 Comparison of Noise Estimation Methods

The first experiment compares the recognition performance of two noise estimation methods, Gauss-Newton and EM-FA, for the VTS compensation. Figure 5.4 shows the WER evolution with respect to the total number of re-estimation iterations, which is calculated as

$$\text{total \# of re-est.} = (\text{\# of dec. passes} - 1) \times (\text{\# of re-est/pass}).$$

Note that the models for the first decoding pass (12.86% in WER) come from the initial noise estimate, thereby avoiding the re-estimation.

It is observed that the Gauss-Newton method converges significantly faster than the EM-FA method, though the two approaches achieve similar performance improvement given sufficient iteration steps. The figure also shows that additional decoding passes, which interleave with the parameter re-estimations, do not substantially benefit the performance.

Table 5.3 shows the recognition performance and convergence of the noise estimation algorithms in two stopping criteria. The first case is the minimum WER that the algorithms can achieve in a long run. The second case compares the performance in a more practical setup, i.e., we limit the algorithms to two decoding passes and stop the EM iterations when WER fails to decrease by a certain threshold (1%). The Gauss-Newton approach significantly outperforms the EM-FA approach in both recognition accuracy and convergence. In the experiments henceforward, we retain the second configuration for the noise estimation algorithms. For example, the Gauss-Newton method is configured by default to perform two decoding passes and two re-estimations in the second pass. Table 5.4 gives detailed WER for the VTS compensation using the Gauss-Newton method on Test Set A, B, and C with respect to different SNRs.

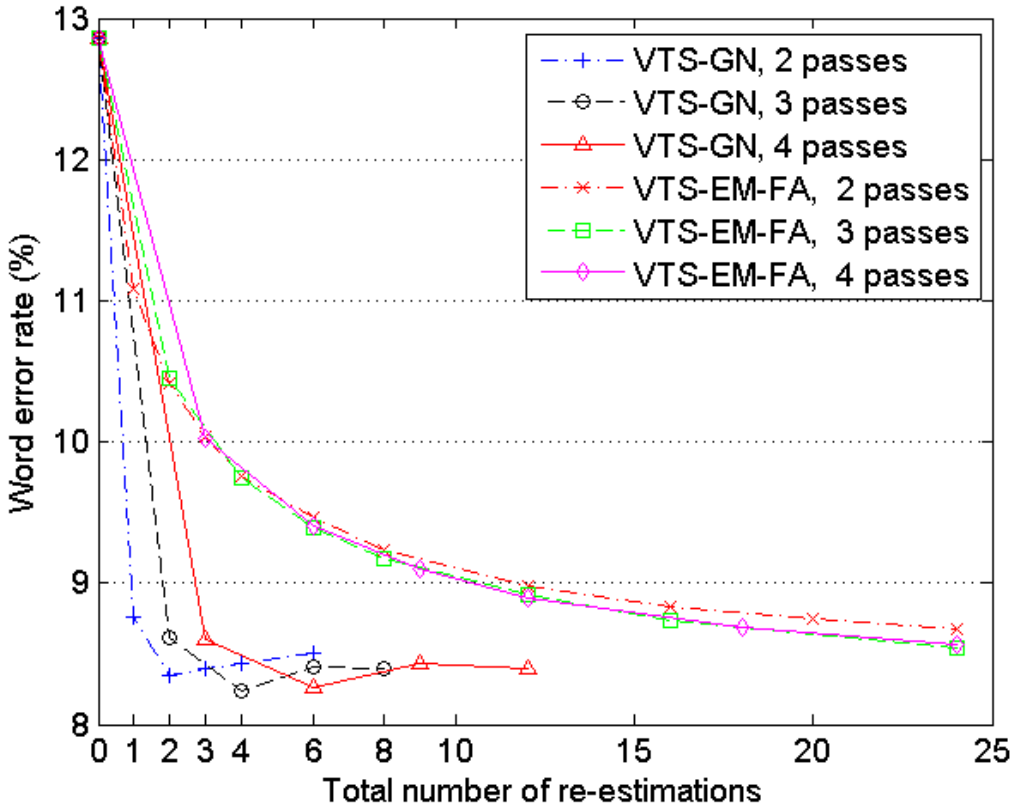


Figure 5.4: WER (%) against the total number of re-estimation iterations for the VTS compensation.

Table 5.3: WER (%) and iterations for two noise estimation methods with the VTS compensation in two stopping criteria using models trained on Aurora 2 clean data.

System	Stop at min. WER		Stop at 1% WER reduction	
	WER (%)	# of re-est.	WER (%)	# of re-est.
VTS, 1 pass	12.86	0	—	—
VTS-GN	8.24	4	8.35	2
VTS-EM-FA	8.32	96	9.24	8

5.2.2 Comparison of Compensation Models

A comparison of three compensation models, VTS, UT, and DPMC, using the Gauss-Newton method is given in Table 5.5. The one-pass systems obtain the noise estimate from the non-speech portion of the signal. Similar setups have been used in [3] for VTS, [37] for UT, and [24] for DPMC. The two-pass systems re-estimate the noise parameters based on the ML criterion, which may correspond to [54] for VTS and [55] for UT with a different

Table 5.4: Detailed WER (%) for the VTS compensation using the Gauss-Newton method on the Aurora 2 task.

SNR	Set A	Set B	Set C	Avg.
Clean	1.00	1.00	2.17	1.23
20 dB	1.28	1.13	1.29	1.22
15 dB	2.09	1.97	2.07	2.03
10 dB	3.76	3.63	3.99	3.75
5 dB	9.15	8.76	8.77	8.91
0 dB	26.32	25.04	26.59	25.86
-5 dB	62.26	60.75	60.21	61.24
Avg. (0–20 dB)	8.52	8.10	8.54	8.35

noise estimation method. In DPMC compensation, 100 Monte Carlo samples per Gaussian are generated for SJA, and 800 for XCOV. The one-pass results may reflect the intrinsic efficiency of the compensation models, because the noise parameters have not been treated with optimization. From this perspective, we can say UT has an advantage over VTS and DPMC in terms of the modeling capability. All of the compensation schemes produce a similar performance after two passes, though UT-SJA is slightly better than other systems. This implies that the noise estimation algorithm substantially diminishes the difference of the modeling powers in these compensation models.

Table 5.5: WER (%) comparison for three noise compensation models using the Gauss-Newton method.

		UT		DPMC	
	VTS	SJA	XCOV	SJA	XCOV
1 pass	12.86	12.60	12.49	13.94	14.15
2 passes	8.35	8.22	8.35	8.43	8.53

Figure 5.5 shows the performance variations with respect to the number of Monte Carlo samples per Gaussian for the DPMC compensation. DPMC-SJA requires considerably fewer sample points than DMPC-XCOV, though both the approaches achieve a similar performance given sufficient samples. This can be attributed to the fewer degrees of freedom used in the SJA method than the XCOV method, as discussed in Section 3.4.

Comparing the results obtained from the speech recognition experiments and the GMM fitting experiments described in the previous section, we note there exists a discrepancy in

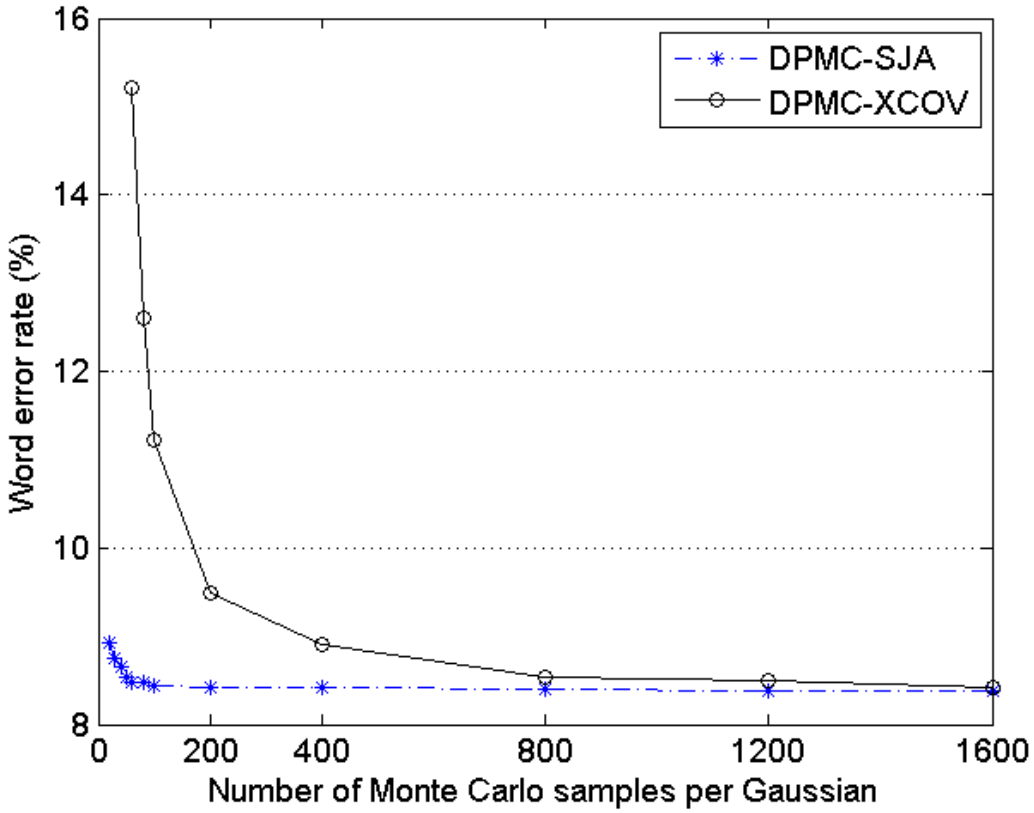


Figure 5.5: WER (%) against the number of Monte Carlo samples per Gaussian for the DPMC compensation using the Gauss-Newton method.

the conclusions concerning the compensation models. The DPMC compensation produced more accurate noise estimate than the VTS model in the GMM fitting, but did not improve the recognition accuracy over VTS in speech recognition. The speech recognition results also seem inconsistent with what have been reported elsewhere [37], [55], [79], [96], [88]. Our conjectures on this discrepancy are as follows.

First, the difference of the sampling-based models from the VTS model as presented in this article is moderate, i.e. they differ in the transformation forms for the static parameters, but have the same number of unknown parameters and share the same form for the dynamic parameters. Note that dynamic parameters are also known to play an important role for robust speech recognition [102], [54].

Second, the sampling-based models have been reported with gains over VTS in various recognition tasks [37], [79], [55]. It is however worth noting in [55] that the performance

gap between the two models varies with the phase factor α of the mismatch function, and may vanish at some values of α. The experiments in this section compute the cepstra in the magnitude domain due to its high performance, which roughly corresponds to $\alpha = 1$ in [55]. The conclusion regarding the comparison of the compensation models is actually in accordance with those reported in [55].

Third, the experimental difference between the GMM fitting and the speech recognition suggests that the sampling-based models improve more effectively in the accuracy of the noise estimate (for the parameter estimation and tracking problem) than in the discriminating power of the compensated models (for the classification problem).

5.2.3 Noise Adaptive Training

We compare the performance of the two estimation methods under the VTS adaptive training scheme. Two acoustic models were built using the Aurora 2 multistyle training data. The first is the standard multistyle acoustic models, which yields a WER of 14.56%. Starting from the multistyle models, we obtain the second, canonical speech models by adaptive training.

Table 5.6 shows the results for the VTS compensation, with and without adaptive training. For either noise estimation algorithm, VTS adaptive training performs better than the standard VTS without adaptive training. It also gives additional gains over the standard VTS using the clean acoustic models. VTS using the multistyle acoustic models does not yield consistent improvements over VTS using the clean acoustic models. This indicates that adaptive training is necessary for the compensation-based approaches to be effective in the presence of multistyle training data. Moreover, the Gauss-Newton method always outperforms the EM-FA method, for example, 7% relative reduction in WER for adaptive training, similar to the standard compensation using the clean acoustic models.

Table 5.6: WER (%) comparison for two noise estimation methods with the VTS compensation using models trained on Aurora 2 multistyle data.

	VTS	Adaptive training
GN	8.70	8.15
EM-FA	9.11	8.75

Table 5.7 gives detailed WER for the adaptively trained VTS system using the Gauss-Newton method on the Aurora 2 task. Compared with Table 5.4, it is observed that adaptive training provides additional benefits only in more severe noise conditions (0 and -5 dB), which present the greatest mismatch to the clean-trained acoustic models for the standard VTS compensation.

Table 5.7: Detailed WER (%) for the VTS adaptive training using the Gauss-Newton method on the Aurora 2 task.

SNR	Set A	Set B	Set C	Avg.
Clean	1.22	1.22	1.32	1.24
20 dB	1.72	1.73	1.70	1.72
15 dB	2.38	2.33	2.54	2.39
10 dB	4.22	3.83	4.46	4.11
5 dB	9.07	8.24	9.14	8.75
0 dB	24.58	22.77	24.41	23.82
-5 dB	58.82	57.03	57.81	57.90
Avg. (0–20 dB)	8.39	7.77	8.44	8.15

Figure 5.6 shows the performance of the VTS adaptive training over the number of iterations used to train canonical models. It is shown that two or three iterations of adaptive training is enough to obtain a good canonical model. The difference in the convergence rate between the two estimation methods vanishes. One possible reason is that minor changes in noise parameters take effects to all of the acoustic models, whereas minor changes in the canonical models take local effects to individual Gaussian components. Thus, the system performance is more sensitive to the optimality of the noise parameters than to the optimality of the canonical models. Moreover, other experiments (not shown in the thesis) indicate that adaptive training does not speed up the convergence of the runtime estimation of noise parameters.

5.2.4 Comparison with Other Techniques

In Table 5.8, we provide a comparison of the performance of the VTS compensation and several popular noise-robust speech recognition techniques, including CMN, MLLR, and ETSI advanced front-end (AFE) [1], in which the acoustic models are trained with both the clean training set and the multistyle training set. This is helpful in indicating how well

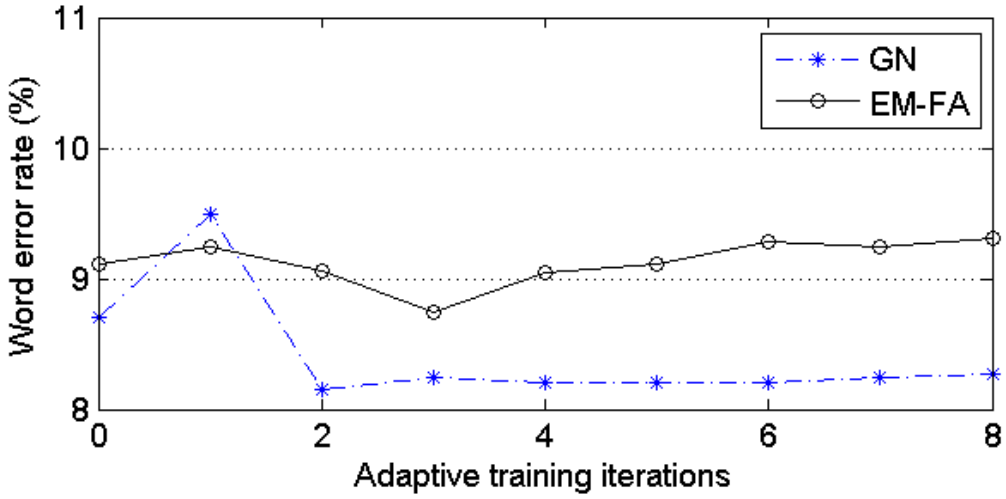


Figure 5.6: WER (%) over the adaptive training iterations for two noise estimation methods with the VTS adaptive training on Aurora 2 multistyle data.

the investigated compensation systems perform. The unsupervised MLLR adaptation is performed at the speaker level. Two regression classes corresponding to speech and non-speech states are used given the limited adaptation data (around ten utterances per speaker in each test condition), to allow for a reliable estimation of the transformation parameters.

For the acoustic models trained with clean data, the VTS compensation produces a WER significantly lower than the other methods. Also, the proposed VTS compensation yields comparable results to those state-of-the-art robust recognition systems [54] reported in the literature on the Aurora 2 task.

When multistyle training data are available, adaptively trained acoustic models are obtained for the VTS compensation. The VTS adaptive training yields the same WER as AFE, significantly lower than the other methods.

Table 5.8: WER (%) comparison between the VTS compensation and several other robust speech recognition techniques.

System	Clean	Multistyle
Baseline	41.57	14.56
CMN	33.79	10.71
MLLR+CMN	24.77	8.97
AFE	12.89	8.15
VTS	8.35	8.15

5.2.5 Detailed Analysis of the Gauss-Newton Method

Here we report two diagnostic experiments on the proposed Gauss-Newton method. The first experiment is designed to quantify the effect of the noise variance estimation method presented in Section 3.3. Table 5.9 provides a comparison of VTS-GN and the second (sample variance) system, which differs from VTS-GN only in that the noise variance estimate is fixed to the sample variance from non-speech segments. The second system has been used in [53]. As different portions of HMM variance parameters are gradually added for the compensation, the Gauss-Newton method reduces WER to 8.35%, 12% relative improvement over the sample variance method.

Table 5.9: WER (%) for two variants of the Gauss-Newton method with the VTS compensation. Two systems differ only in the way of noise variance estimation.

Parameters	Gauss-Newton	Sample variance
$\boldsymbol{\mu}_y, \boldsymbol{\mu}_{\Delta y}, \boldsymbol{\mu}_{\Delta^2 y}$	17.55	
$+\ \boldsymbol{\Sigma}_y$	13.06	12.81
$+\ \boldsymbol{\Sigma}_{\Delta y}$	9.40	9.56
$+\ \boldsymbol{\Sigma}_{\Delta^2 y}$	8.35	9.50

In the second experiment, we investigate the different forms of the Hessian approximation in the Gauss-Newton method, as discussed in Section 3.9. Table 5.10 shows the performance of VTS-GN with the Hessian form $\boldsymbol{H}^{\text{lm}}$ in (3.50) with different factors ρ, and $\boldsymbol{H}^{\text{diag}}$ in (3.52), respectively. Since the singularity of the Hessian might occur when noise is small, the results for the clean test conditions are also enclosed. It is shown that without the treatment of the Levenberg-Marquardt method ($\rho = 0$), the performance degrades substantially at the clean test conditions, especially for Set C where only the channel mismatch exists. Good performance in both the clean conditions and 0–20 dB regions is obtained when $0.4 \leq \rho \leq 0.8$. Using a diagonal Hessian $\boldsymbol{H}^{\text{diag}}$ incurs a 0.6% absolute loss in WER, in exchange for the reduced computational overhead.

[3]Set A and Set B contain the same clean speech data and produce the same recognition result on the clean test condition, whereas the clean speech in Set C is convoluted with the channel distortion.

Table 5.10: WER (%) for different Hessian approximations in the Gauss-Newton method with the VTS compensation.

Hessian approximation	Clean[3]		
	Set A/B	Set C	0–20 dB
Clean baseline	0.66	0.80	41.57
$\boldsymbol{H}^{\text{lm}}$ $\rho = 0$	3.70	23.58	8.44
$\rho = 0.2$	2.99	18.75	8.41
$\rho = 0.4$	1.00	2.17	8.35
$\rho = 0.8$	0.67	0.64	8.48
$\rho = 1.6$	0.66	0.64	8.86
$\boldsymbol{H}^{\text{diag}}$	0.77	0.63	8.95

5.2.6 Fast VTS Compensation

In Table 5.11, we provide a performance comparison of the fast VTS compensation approach proposed in Section 3.7 with the standard VTS compensation. The two systems run one-pass decoding for each utterance, and the noise parameters are estimated from non-speech areas using a sample average and VTS, respectively. As can be seen, the fast VTS achieves 14% relative improvement in WER over the standard VTS method.

Table 5.11: WER (%) comparison for Aurora 2 between the standard and fast VTS compensation methods.

Noise initialization	WER
Sample average	12.86
Fast VTS	11.06

5.3 Summary

We intentionally select two tasks to demonstrate the effectiveness of the nonlinear compensation models and the associated noise estimation methods. The first is to fit a GMM model to artificially corrupted samples, and the second is to perform speech recognition on the Aurora 2 database.

Experimental results verify that the Gauss-Newton method is effective in optimizing various nonlinear noise compensation models. Moreover, both Gauss-Newton and EM-FA methods, in the long run, can achieve a similar recognition performance. However, the Gauss-Newton method is superior to the EM-FA method in terms of the convergence

property. In the practical experimental setups, this difference in convergence leads to the result that the Gauss-Newton method obtains 7%-12% relative reduction in WER over the EM-FA method for standard compensation and adaptive training.

It is shown that the sampling-based compensation techniques produce more accurate noise estimate in GMM fitting, but do not yield the expected gain in WER over the VTS model in speech recognition. Although the VTS model has been criticized for its deficiency of linear approximation, we feel that for the classification problem, the linear approximation of VTS is not as crucial as other fundamental assumptions, like the one-to-one Gaussian mapping between the clean speech and the corrupted speech. Actually, as the level of noise increases, the distribution of the corrupted speech tends toward bimodal [68]. The one-to-one Gaussian mapping places a substantial cap on the accuracy of the following model approximation strategies.

CHAPTER 6

EXPERIMENTS ON OVERLAPPING SPEECH

In Chapter 5, the compensation approaches are primarily evaluated through the speech recognition experiments on the Aurora 2 noise-corrupted database. Though the nonlinear compensation models have yielded superior recognition accuracy on such a database, it is desired to evaluate the proposed approach in more realistic conditions and on more challenging recognition tasks.

This chapter presents the initial experiments conducted on speech that is collected in a meeting scenario. A database, Overlapping TIMIT (OTIMIT), is constructed to simulate a meeting of three participants, where the source speech is from the TIMIT database [21] and re-recorded in a conference room with realistic noise and reverberation levels. It should be noted that the database has originally been designed for the purpose of studying the interfering speech problem that we encounter in realizing immersive acoustic communication over distributed transducer networks [91]. Since the noise compensation method is not intended for addressing the interfering speech, in this study, we also examine several other robust recognition techniques such as CMN and multi-channel acoustic echo cancellation (AEC). We seek to identify the optimal configurations of combining these techniques to combat the distorted speech.

6.1 OTIMIT Corpus

The OTIMIT corpus was constructed by playing the clean TIMIT corpus through loudspeakers, simulating a meeting of multiple competing speakers, as shown in Figure 6.1. The placement of the audio playback and recording devices is similar to that used in the multi-channel overlapping numbers corpus (MONC) [62]. Three loudspeakers, S1, S2, and S3, were placed at 90° spacings by a round conference table. S1 is designated as the primary speaker, and others are competing speakers. A close-talking microphone is mounted

about 20 cm right in front of each speaker. In addition, an 8-component circular microphone array, equally spaced with 3 cm diameter, is placed in the center of the table. All microphones are omnidirectional. The reverberation time of the room during recordings is approximately 200ms. The ambient acoustic noise was primarily due to the AC air flow and pipe-transmitted vibrations. The acoustic noise level measured between 33–36 dBA at the location of the microphones. Additional calibration recordings were provided that can be used for the estimation of the noise spectrum and the acoustic transfer functions.

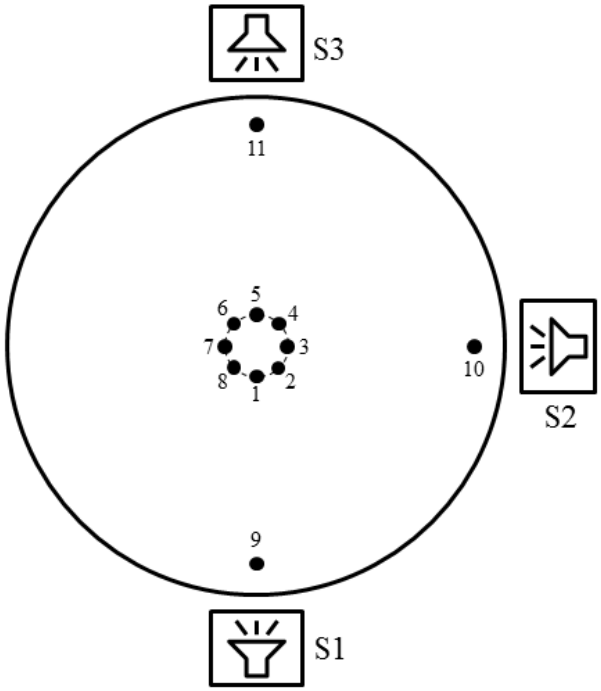


Figure 6.1: Recording setup for OTIMIT corpus.

The OTIMIT corpus considers two recording scenarios: *solo* and *ovlp*. In Scenario *solo*, only the primary speaker S1 is talking. Scenario *ovlp* consists of speech from several concurrent speakers, where the primary speaker S1 speaks continuously and the competing speakers S2 and S3 intervene sporadically. In each session, the speech signal from S1 is generated by concatenating the utterances from each of the TIMIT speakers (*sa* utterances excluded), with a fixed short pause (100ms) placed between the utterances. Meanwhile, S2 and S3 randomly choose different TIMIT speakers and play back the corresponding utterances in a sporadic fashion. The occurrence of the competing speech from S2 and S3

is managed by two separate queue systems, similar to an M/M/1 model. Utterances are dispatched to the speaker S2/S3 following the Poisson process with parameter $\lambda = 0.12$. When an utterance arrives, if the speaker is idle, the utterance will be played immediately. Otherwise, the utterance will be stacked in the queue and await its turn. The session will be drawn to an end, as long as S1 finishes all utterances, and both S2 and S3 play out complete utterances.

Table 6.1 shows the overlapping rates of the resulting *ovlp* dataset. Note that we intentionally raise the rate of overlapping speech as high as 55%, though unrealistic in usual meetings, to make effective use of the limited TIMIT data. In [80], it was shown the overlapping rate varies depending on the context of meetings and telephone conversations, and can reach up to 15% of words. Nevertheless, once the performance of a recognition system on OTIMIT is evaluated, we can predict its performance in a task of less overlapping speech through regression analysis, as shown in the next section.

Table 6.1: Overlapping rates (%) of the competing speakers over the primary speaker S1.

	S1-S2	S1-S3	S1-S2-S3
Training set	33.2	32.3	11.4
Test set	33.0	34.0	11.9

6.2 Experimental Setup

The OTIMIT corpus consists of a large assortment of signal acquisition devices allowing a flexible configuration for various application scenarios. In this initial study, we consider a meeting setup in which the master participant S1 speaks with the other two participants at the remote site. The signals from the remote participants are transmitted and played back through separate loudspeakers in the meeting room. The speech from S1 is picked up by the close-talking microphone M9, which inevitably mixes the competing speech from S2 and S3. This teleconference scenario is in line with a series of studies we have carried out to realize immersive acoustic communication [91], [72], [9]. To enhance and successfully recognize the speech from S1, we consider the integration of multi-channel acoustic echo cancellation (AEC) and the VTS compensation. The multi-channel AEC uses the approach

proposed in [95], which applies an improved residual echo enhancement (REE) procedure to further improve the AEC system performance. Therefore, using the reference signals of S2 and S3, we can cancel both S2 and S3 from the mixed signal at M9 and produce an estimate of S1

We evaluated the performance of the proposed method on the OTIMIT corpus for phoneme recognition. All phones are modeled as 3-state strict left-to-right context-independent HMMs. Each state observation density is modeled by a 32-component Gaussian mixture density with diagonal covariance matrices. A phone bigram language model is used in decoding. Following [50], the 61 phones in TIMIT are mapped to 48 phones for model training, which are then folded to 39 phones when evaluating the results. Each feature frame consists of 13-dimensional MFCCs plus their first and second order derivatives, with zeroth cepstral coefficient for the energy term. Features are normalized by CMN at the sentence level.

For VTS, we conducted noise compensation in two decoding passes for each utterance. In the first pass, the noise parameters are initialized with the first and the last 20 frames, and the channel mean is set to 0. The second pass refines the noise estimate using the Gauss-Newton methods and performs the final decoding.

6.3 Experimental Results

Tables 6.2 provides some baseline experiments for matched and mismatched condition training on the OTIMIT corpus. Three sets of acoustic models were trained with speech from the original TIMIT (*clean*), OTIMIT *solo*, and OTIMIT *ovlp*, respectively, and tested against speech from these conditions. As shown, the *clean* baseline produces a phone accuracy of 69.37% in the matched condition, and suffers a degradation of performance in the mismatched conditions. Specifically, there is a significant increase (20% absolute) in error rate on the overlapping speech. This indicates that the primary source of distortion in OTIMIT comes from the interfering speech, rather than the environmental mismatch. This observation is confirmed by the results of the matched conditions for *solo* and *ovlp*, where *solo* almost regains the drop in performance, but *ovlp* still remains at a high error rate.

One may quantize the effect of the overlapping speech through linear regression analysis,

Table 6.2: Phone accuracy (%) comparison for the matched and mismatched condition training on OTIMIT task.

Model	clean	solo	ovlp
clean	69.37	63.99	48.96
solo	—	68.10	51.35
ovlp	—	—	57.24

so as to predict the performance of the recognition system for a meeting setup with a realistic overlapping rate. Suppose that the phone accuracy of the overlapping speech depends linearly on the proportions of four patterns of interfering speech, as listed in the right-hand side of (6.1). To compute the regression coefficients, we randomly choose a subset of N (20) sessions from the *ovlp* test set and measure the interference proportions and the phone accuracy of the subset. By repeatedly sampling the test set, sufficient regression observations can be collected to estimate the coefficients. For example, the regression for the recognition on the *ovlp* test set using the *solo* model yields the following regression:

$$\begin{aligned} \text{Accuracy} =& 0.67 \times \% \text{ S1 solely} \\ &+ 0.39 \times \% \text{ Interference from S2 only} \\ &+ 0.44 \times \% \text{ Interference from S3 only} \\ &+ 0.32 \times \% \text{ Interference from both S2 and S3.} \end{aligned} \tag{6.1}$$

The regression coefficients can be regarded as the phone accuracies of the respective interference patterns that occur exclusively. We see that the coefficient of the *S1 solely* portion (0.67) is quite close to the performance of the *solo* model evaluated in the matched condition (68.10%). Moreover, the coefficients of the other interference patterns agree with our intuition about the influence of the interfering sources, that is, S2 is located closer to S1 and thus introduces higher distortion to S1 than does S3. When both S2 and S3 interject, the degradation in performance becomes more severe. All these evidences in turn validate the linear regression model.

Table 6.3 shows the performance of the VTS compensation and multi-channel AEC with the models trained from the TIMIT data. On the *solo* set, the standard VTS compensation gives a gain over the baseline system, and VTS adaptive training further improves the

recognition accuracy. However, the VTS compensation does not effectively reduce the error rate on the overlapping speech. This is because the overlapping speech is highly non-stationary and correlated to the speech being recognized, obviously violating the underlying assumption of the compensation approaches that the additive noise should be relatively stationary across the whole utterance and independent of the speech. The last two rows of Table 6.3 show the effect of the multi-channel AEC on the overlapping speech. Obviously, the multi-channel AEC significantly improves the recognition accuracy by 10% absolute over the baseline system. The combination of the multi-channel AEC and VTS obtains the best performance of 60.75% on the overlapping speech.

Table 6.3: Phone accuracy (%) for the VTS compensation and AEC on OTIMIT task.

Systems	solo	ovlp
baseline	63.99	48.96
VTS	65.96	50.16
VTS adaptive training	66.90	49.86
AEC	—	58.96
AEC+VTS	—	60.75

The standard model-domain VTS does not use the features normalized by CMN, which may partially invalidates the formulation of the mismatch function. However, on the OTIMIT corpus, we experimentally identified that the VTS in combination with CMN performs the same and sometimes better than the stand-alone VTS, and thus we choose the CMN-normalized features as default. The similar observation was also reported in [17] that a combination of CMN and the feature-domain VTS yields performance gain over the stand-alone VTS in most cases. CMN is considered to reduce the sensitivity to the channel variation, and thus it appears redundant in functionality with the VTS compensation of the convolutional channel. It is worth investigating the effect of combining CMN and VTS.

Table 6.4 summarizes the performance of the experiment involving the CMN and the VTS channel compensation on the OTIMIT *solo* task. As can be seen, without any treatment to mitigate the channel mismatch, the system performs quite poorly. Either CMN or the VTS channel compensation can greatly remedy this problem. Nevertheless, the combination of the two leads to only a slight further improvement, confirming that the effects of

CMN and the VTS compensation of the convolutional channel are almost identical.

Table 6.4: Phone accuracy (%) of the experiment probing the CMN and the VTS channel compensation on the OTIMIT *solo* task.

Systems	no CMN	CMN
VTS, no channel compensation	59.48	65.79
VTS	65.24	65.96

6.4 Summary

In this chapter, we present results examining the integration of the VTS compensation and other robust speech recognition techniques on overlapping speech. The OTIMIT database was constructed to simulate the meeting of three concurrent participants. Since the OTIMIT speech contains low additive noise, the VTS compensation and the VTS adaptive training achieve a moderate gain over the baseline system on the single-talker speech. However, the VTS compensation performs poorly on the overlapping speech, indicating that simply regarding the interfering speech as additive noise is fundamentally problematic. The multi-channel AEC significantly improves the results on the overlapping speech by making use of the reference signals of the interfering speech. Based on that, the VTS gives additional gains.

One interesting finding from this study is that the VTS in combination with CMN outperforms the stand-alone VTS in most cases. Though the effects of CMN and the VTS channel compensation are almost identical, CMN appears to be a better choice. CMN is computationally more efficient and can be estimated without the need of the EM re-estimation. Moreover, in the VTS compensation, the channel mean is simply initialized to zero, which, in a severe channel mismatch condition, may lead to a poor hypothesis generated from the initial decoding pass and degrade the subsequent noise model re-estimation. In this regard, the VTS compensation in combination with CMN can be used to remedy its weakness.

NONLINEAR COMPENSATION AND HETEROGENEOUS DATA MODELING FOR ROBUST SPEECH RECOGNITION

PART II

Heterogeneous Data Modeling

by

Yong Zhao

CHAPTER 7

STRANDED HMM

The Gaussian mixture HMMs, though being the predominant modeling technique for speech recognition, are often criticized as being inaccurate to model heterogeneous data sources. In this chapter, we propose the stranded HMM, an extension of the conventional HMM, to explicitly model the dependence among the mixture components, i.e., each mixture component is assumed to depend on the previous mixture component in addition to the state that generates it. The learning procedure and the decoding algorithm for the stranded HMMs are described. The performance of the proposed models is evaluated on the Aurora 2 database.

7.1 Background and Motivation

State-of-the-art speech recognition systems assume the availability of sufficient speech data to achieve an accurate and robust recognition performance. Efficient modeling techniques that are highly scalable to the data volume consist of N-gram language models to maintain accurate word prediction, context-dependent phoneme models to represent pronunciation variations, and multiple mixtures of Gaussians to account for the large variability of the speech features. One particular observation can be made with the last technique: when dealing with heterogeneous data sources, the feature sequence of some speech units can be matched at a high probability with the competing model by concatenating its mixture components that are obtained in different acoustic conditions. This problem is referred to as trajectory folding [32], and it may be attributed to the observation independence assumption made by the HMM that successive observations are related only through the underlying states that generate them.

One approach to improve the modeling accuracy is to relax the HMM conditional-independence assumption, and condition the distribution of each observation on the previous observations in addition to the state that generates it [93], [10]. This method is known

as conditional-Gaussian HMMs or autoregressive HMMs. The dependency properties of the conditional-Gaussian HMMs can be graphically shown in Figure 7.1. However, it has been shown that the conditional-Gaussian HMMs do not provide a benefit if the dynamic features are used [44], [10]. Another class of methods explores the use of more complex HMM structures, such as multi-path HMM [85], [47]. The HMM is composed of multiple parallel paths, each of which may account for the acoustic variability from a specific source. The multi-path HMM may over-correct the trajectory folding problem associated with the Gaussian mixture HMM because the allowable mixture paths are exponentially reduced. Most of such systems have been only evaluated on some simple recognition tasks using a small number of parallel paths. How to achieve a model that is intrinsically robust to speaker and environmental changes is still a challenging and interesting problem, though we have observed less efforts being attempted along this direction in recent years.

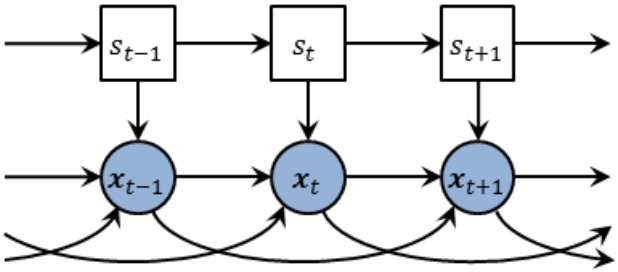


Figure 7.1: Dynamic Bayesian network representation of the conditional-Gaussian HMM. In this example, the distribution of $\boldsymbol{x}_t$ depends on two previous observations $\boldsymbol{x}_{t-1}$ and $\boldsymbol{x}_{t-2}$ as well as state $\boldsymbol{s}_t$.

In this chapter, we propose the stranded HMM, an extension of the Gaussian mixture HMM, to explicitly model the dependence among the mixture components. In other words, each mixture component is assumed to depend on the previous mixture component in addition to the state that generates it. Thus the conditional-independence assumption for observations in the conventional HMMs is implicitly relaxed. Another motivation for the stranded model comes from the hope to make use of the discriminating power that would be possessed by the mixture weights. It has often been noted that the acoustic models using Gaussian mixture HMMs produce approximately equal mixture weights for each state.

Hence mixture weights are regarded as little informative and receive less attention than do Gaussian means and variances in developing effective modeling techniques. However, the mixture weights in the stranded HMMs evolve as the mixture transition probabilities with a significantly wider dynamic range, which makes it worth investigating the potentials of the weight-like parameters to improve the system performance.

7.2 Stranded HMM

As opposed to the conventional HMM, the stranded HMM aims to explicitly model the relationships among the mixture components. The distribution of the mixture component is assumed to depend on the previous mixture component in addition to the state that generates it. The model can be illustrated by a DBN as shown in Figure 7.2. Note that additional links between successive mixture variables are added in comparison with the conventional Gaussian mixture HMM.

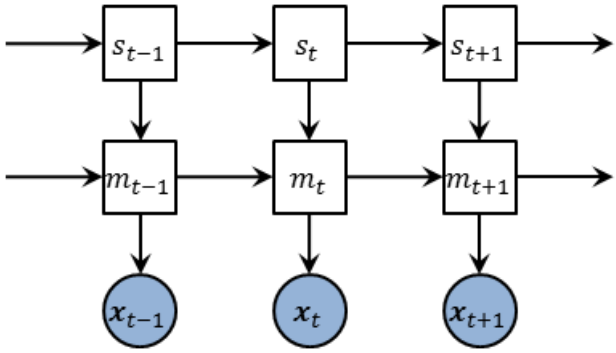


Figure 7.2: Dynamic Bayesian network representation of the stranded HMM.

Let $\boldsymbol{x}_1^T = \boldsymbol{x}_1, ..., \boldsymbol{x}_T$ be a sequence of observations of length T, and $s_1^T = s_1, ..., s_T$ and $m_1^T = m_1, ..., m_T$ are the hypothesized state and mixture sequences, respectively. The joint probability of the three sequences in the stranded model is given by

$$p(\boldsymbol{x}_1^T, s_1^T, m_1^T) = \prod_{t=1}^{T} p(s_t|s_{t-1}) p(m_t|s_{t-1}, m_{t-1}, s_t) p(\boldsymbol{x}_t|s_t, m_t) \tag{7.1}$$

The stranded HMM consists of the following elements: the state transition probability

$$p(s_t = j|s_{t-1} = j') = a_{j'j} \tag{7.2}$$

the Gaussian observation probability given state j and mixture l

$$p(\boldsymbol{x}_t|s_t = j, m_t = l) = b_{jl} = \mathcal{N}(\boldsymbol{x}_t; \boldsymbol{\mu}_{jl}, \boldsymbol{\Sigma}_{jl}) \tag{7.3}$$

and the mixture transition probabilities defined as

$$p(m_t = l|s_{t-1} = j', m_{t-1} = l', s_t = j) = \begin{cases} c_{l'l}^{(j'j)} & \text{if } a_{j'j} > 0 \\ 0 & \text{if } a_{j'j} = 0 \end{cases}. \tag{7.4}$$

Note that from the definition of the mixture transition probability, the DBN in Figure 7.2 should include a link from s_{t-1} to m_t to indicate the dependence of m_t on s_{t-1} as well as m_{t-1} and s_t. However, because the state $s_{t-1} = j'$ can be inferred from the mixture component $m_{t-1} = l'$, the DBN neglects this link for simplicity. The mixture transition probabilities for each state transition from j' to j of a non-zero probability $a_{j'j}$ form a matrix $C^{(j'j)} = \left[c_{l'l}^{(j'j)}\right]$. Each mixture transition matrix satisfies the following statistical constraint

$$\sum_l c_{l'l}^{(j'j)} = 1, \quad \text{for any feasible } j', j, l'. \tag{7.5}$$

We see that the mixture components in state j' have multiple matrices of mixture transitions. Which transition matrix is activated at a particular time depends on the mastering state transitions. Also, we may refer to $C^{(j'j)}, j' = j$, as within-state mixture transitions, and $C^{(j'j)}, j' \neq j$, as cross-state mixture transitions.

7.2.1 Advantages of the Stranded HMM

This section describes a number of properties of the stranded HMM. In particular, the advantage of using the stranded HMM over the conventional flattened HMM is discussed.

The stranded HMM can be portrayed in a state transition graph as in Figure 7.3. The transitions between the states and the transitions between the corresponding mixture components (or substates) constitute a two-layer diagram, and are synchronized with each other. At a first glance, it appears that the stranded HMM can be converted to an HMM by regarding each state/mixture pair as an augmented state. The resulting flat HMM has the same model topology as the lower-layer transition graph in Figure 7.3, except that its

transition probabilities are the product of the corresponding state and mixture transition probabilities in the stranded model.

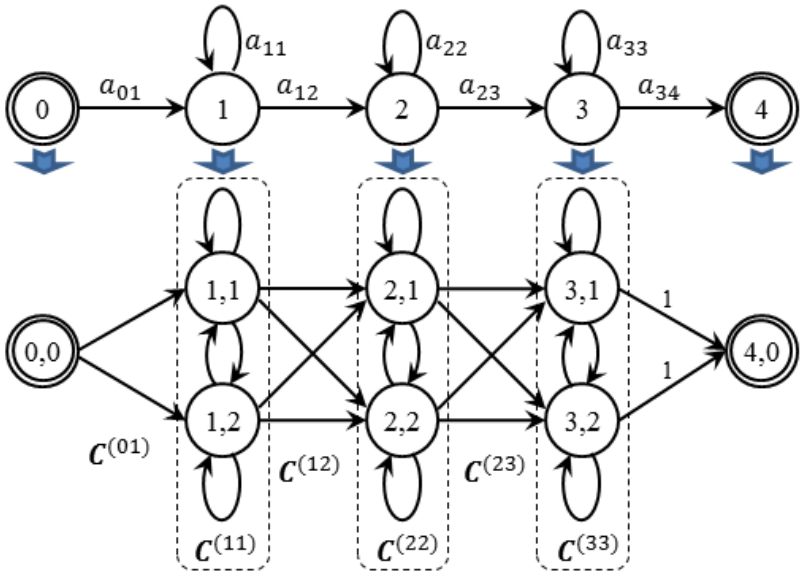


Figure 7.3: Example of the two-layer state transition diagram for a 3-state 2-mixture left-to-right stranded HMM. The top layer consists of a Markov chain, in which each state corresponds to a column of the mixture components in the lower layer. The transitions between the mixture components are synchronized with the state transitions. The initial and final states are non-emitting, and represented by double circles.

However, the stranded HMM is different and has several advantages over the flat HMM. First, unlike the flat HMM, the two-layer structure of the stranded HMM enforces the synchronization among different HMM paths. This extra constraint has great practical importance in modeling. When we learn the model parameters in the presence of numerous observation sequences, we hope that one observation sequence might be matched by one such HMM path. Through synchronization, other less likely paths have to go with the dominant path, and not to warp themselves to repeatedly match the current observation sequence. Thus, the synchronization prevents the path repetition problem, which might greatly discount the modeling power of the multi-path model.

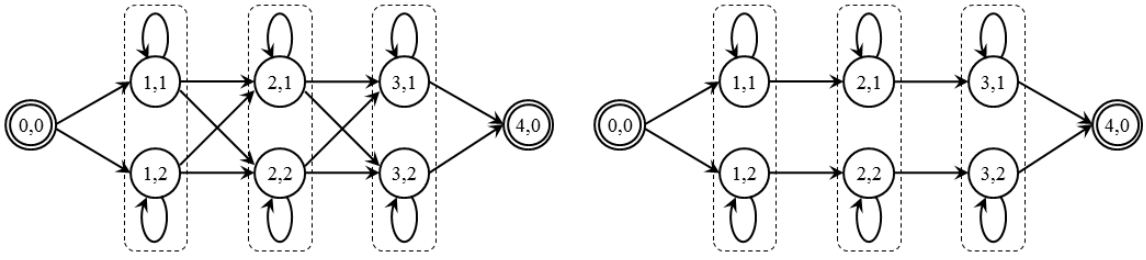
Second, the two-layer decomposition of the stranded HMM retains the essential interpretation of the state transitions, and allows the manipulation on the mixture transitions with great flexibility. In particular, the type of the model, such as ergodic or left-to-right,

is decided by the state transition matrix, regardless of the mixture transitions. This means that we can modify or prune the mixture transitions at ease, only if the statistical constraint (7.5) is satisfied. Such operations pose a challenge to the flat HMM, where arbitrary pruning of the transitions might, for example, cause some states trapped in a dead loop.

Finally, in many applications of HMMs, it is often of interest to find the most likely state sequence, excluding the mixture component sequence. In Section 7.2.3, we propose a modified Viterbi algorithm to find the best state sequence through the stranded model by integrating out the mixture variables. This choice is infeasible for the flat HMM, which can only find the best sequence of the augmented states.

Another property of the stranded HMM is that the HMM conditional-independence assumption for observations is implicitly relaxed. This can be verified through the d-separation rule [10] on the graphical representation of the model in Figure 7.2: the observation variables are not d-separated by the state sequence due to the connection of the mixture variables.

Furthermore, the stranded HMM contains multi-path HMMs as special cases as illustrated in Figure 7.4. If we set the within-state transition matrices to the identity matrix, it results in a model composed of parallel HMM paths with cross-coupled connections [70]. Further forcing the cross-state transition matrices to be a permutation matrix gives rise to a mixture of separate parallel paths [85], [47]. Since the stranded HMM still imposes the synchronization between the HMM paths, to be precise, we should say that the stranded HMM can represent parallel and synchronous HMM paths.



(a) (b)

Figure 7.4: State transition diagrams of two multi-path HMMs. (a) parallel paths with cross-coupled connections; (b) separate parallel paths.

7.2.2 Estimating Parameters of Stranded HMMs

The parameters of the stranded HMM can be learned with th expectation-maximization (EM) algorithm, similar to a regular HMM. As both the states and the mixture components are latent variables, we need to maximize the following EM auxiliary function

$$Q(\hat{\Lambda}|\Lambda) = \sum_{s_1^T} \sum_{m_1^T} p(s_1^T, m_1^T|\boldsymbol{x}_1^T, \Lambda) \log p(\boldsymbol{x}_1^T, s_1^T, m_1^T|\hat{\Lambda}) \tag{7.6}$$

where Λ and $\hat{\Lambda}$ denote the existing and new estimates of the model parameters, respectively. In the E step, the Q function requires finding the following sufficient statistics: the posterior probability of being in mixture l of state j at time t, $\gamma_t(j,l)$; the joint posterior probability of two successive state/mixture pairs, $\xi_t(j',l',j,l)$; and the joint posterior probability of two successive state variables, $\zeta_t(j',j)$, so that

$$\gamma_t(j,l) = p(s_t = j, m_t = l|\boldsymbol{x}_1^T, \Lambda) = \frac{\alpha_t(j,l)\beta_t(j,l)}{p(\boldsymbol{x}_1^T|\Lambda)} \tag{7.7}$$

$$\begin{aligned}\xi_t(j',l',j,l) &= p(s_{t-1} = j', m_{t-1} = l', s_t = j, m_t = l|\boldsymbol{x}_1^T, \Lambda) \\ &= \frac{\alpha_{t-1}(j',l')a_{j'j}c_{l'l}^{(j'j)}b_{jl}\beta_t(j,l)}{p(\boldsymbol{x}_1^T|\Lambda)}\end{aligned} \tag{7.8}$$

$$\xi_t(j',j) = p(s_{t-1} = j', s_t = j|\boldsymbol{x}_1^T, \Lambda) = \sum_{l'} \sum_{l} \xi_t(j',l',j,l) \tag{7.9}$$

where we have defined the forward and backward probabilities as

$$\alpha_t(j,l) = p(\boldsymbol{x}_1^t, s_t = j, m_t = l|\Lambda) \tag{7.10}$$

$$\beta_t(j,l) = p(\boldsymbol{x}_{t+1}^T|s_t = j, m_t = l, \Lambda) \tag{7.11}$$

The two quantities can be evaluated recursively in the forward-backward algorithm.

In the M step, we maximize the Q function with respect to the model parameters which

yields

$$\hat{a}_{j'j} = \frac{\sum_{t=1}^{T} \xi_t(j', j)}{\sum_{t=1}^{T} \sum_j \xi_t(j', j)} \tag{7.12}$$

$$\hat{c}_{l'l}^{(j'j)} = \frac{\sum_{t=1}^{T} \xi_t(j', l', j, l)}{\sum_{t=1}^{T} \sum_l \xi_t(j', l', j, l)} \tag{7.13}$$

$$\mu_{jl} = \frac{\sum_{t=1}^{T} \gamma_t(j, l) \boldsymbol{x}_t}{\sum_t \gamma_t(j, l)} \tag{7.14}$$

$$\boldsymbol{\Sigma}_{jl} = \frac{\sum_{t=1}^{T} \gamma_t(j, l)(\boldsymbol{x}_t - \boldsymbol{\mu}_{jl})(\boldsymbol{x}_t - \boldsymbol{\mu}_{jl})^T}{\sum_t \gamma_t(j, l)} \tag{7.15}$$

One problem in learning the stranded HMM is how to gradually increase the mixture components to achieve a model with an optimal performance. In this work, the model is learned in two steps. First, a regular multiple-mixture HMM is achieved by gradually increasing the number of the mixtures to the required number; then, the stranded model is re-estimated by initializing the mixture transition probabilities to the weights of the Gaussian mixtures. The training procedure is summarized in Figure 7.5. It is admitted that there may exist more efficient methods to train the stranded model. For example, we can run some sequential-data clustering algorithm to find a suitable initialization for multiple HMM paths, then establish connections among these paths for a complete stranded HMM, followed by the model re-estimation.

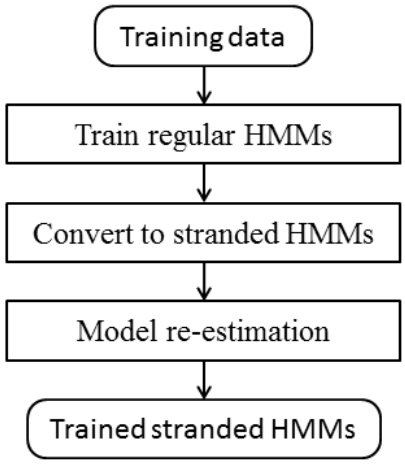


Figure 7.5: Training procedure of the stranded HMMs.

7.2.3 Decoding Algorithm

A direct method to decode the stranded HMM is to simultaneously find the most likely state/mixture pair sequence $\{s_1^T, m_1^T\}$ for a given observations sequence. This method may not be optimal, as in most cases we are only interested in the governing state sequence. Here, we propose a modified Viterbi algorithm to find the best state sequence through the stranded model. The proposed algorithm embeds the forward algorithm in the dynamic programming procedure to integrate out the latent mixture variables. Let $\tilde{\delta}_t(j)$ be the highest probability of observing the partial sequence $\boldsymbol{x}_1^t$ and being in state j at time t

$$\delta_t(j) = \max_{s_1^{t-1}} p(\boldsymbol{x}_1^t, s_1^{t-1}, s_t = j) \tag{7.16}$$

and $\tilde{\delta}_t(j,l)$, its associated portion on each mixture component l. Obviously, we have $\tilde{\delta}_t(j) = \sum_l \tilde{\delta}_t(j,l)$. The best state sequence for $\tilde{\delta}_t(j)$ can be found using dynamic programming

$$i^* = \arg\max_i \; a_{ij} \sum_l \sum_{l'} \tilde{\delta}_{t-1}(i,l') c_{l'l}^{(ij)} b_{jl}(\boldsymbol{x}_t) \tag{7.17}$$

$$\tilde{\delta}_t(j,l) = a_{i^*j} \sum_{l'} \tilde{\delta}_{t-1}(i^*,l') c_{i^*l}^{(i^*j)} b_{jl}(\boldsymbol{x}_t) \tag{7.18}$$

where i^* is the preceding state to achieve the best path ending in state j at time t. Note that strictly i^* should be written as $i_t^*(j)$ to indicate its dependence on time t and state$j$. Nevertheless, we apply the shorter notations for ease of understanding. It should be noted that the above recursive procedure is an approximate inference, because the maximized quantity $\delta_{t-1}(j')$ may not guarantee to be the highest after a re-weighted sum of its portions, as $\sum_{j'} \delta_{t-1}(j',l') c_{l'l}^{(j'j)}$ in (7.18). The approximation will be accurate enough when, as is often the case, the probability $\delta_t(j)$ is dominated by one or a few of its mixtures.

7.3 Experimental Results

The proposed algorithm is evaluated on the Aurora 2 database [34] of connected digits. The multistyle training set is used to learn the stranded HMM systems. Following the standard Aurora 2 recipe for acoustic model training, each digit is modeled by a 16-state left-to-right HMM, and the silence and the short pause are modeled by three and one states, respectively. The number of mixtures per state for the silence model is roughly 1.5 times the size for the

digit models. Each feature vector consists of 13 mel-cepstral coefficients (including zeroth order for the energy term), and their delta and delta-delta coefficients[1]. The 20-mixture HMM baseline yields a word error rate (WER) of 7.53% by averaging over SNRs between 20 and 0 dB of three test sets.

In Figure 7.6, we compare the recognition accuracy of the proposed stranded HMMs with the regular HMMs in different numbers of mixture components per state. The significant improvements over the regular system are observed at all levels of the model complexities. With more than 4 mixtures, the stranded system can reduce the WER by 7%–11% relative.

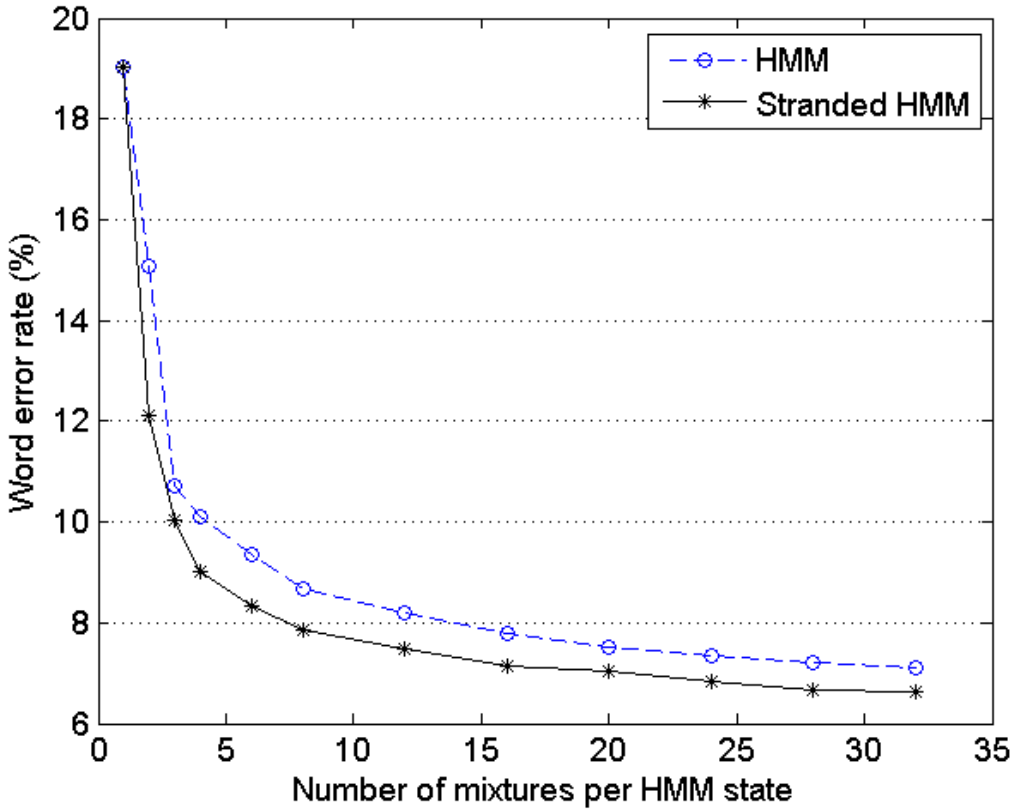


Figure 7.6: WER (%) as a function of the number of mixtures per state using the stranded HMMs on the Aurora 2 test set.

[1]The features are different from the ones used in the experiments described in [4], where the log energy replaces the zeroth order cepstral coefficient. This modification constitutes the major difference of two experimental setups and interestingly, results in more than 1% absolute WER improvement on the Aurora 2 task for the 20-mixture acoustic models.

In the second experiment, we investigate the 20-mixture stranded system with different configurations, as shown in Table 7.1. First, the stranded model yield WER of 7.03%, 7% relative improvement over the regular HMM system. To quantify the incremental contribution of different model parameters in the course of refining the stranded system based on the regular HMM system, we produce two systems by fixing some parameters to the baseline HMM system, as shown in the third part of Table 7.1. We can see that the further refinement of the Gaussian means and variances is helpful in achieving a good performance for the stranded system, whereas refining the state transition probabilities has little effect. This is not unexpected as the Gaussian parameters possess much more discriminating power than do the other parameter in an HMM system.

Table 7.1: WER (%) of the 20-mixture stranded system with various configurations.

System	WER
Regular HMM	7.53
Stranded HMM	7.03
Fixing state transitions	7.02
Fixing Gaussian means & variances	7.25
Diagonalizing within-state mixture transitions	7.20
Diagonalizing cross-state mixture transitions	8.13
Diagonalizing both within-state & cross-state	9.57

Table 7.2: Detailed WER (%) for the 20-mixture stranded HMMs on the Aurora 2 task.

SNR	Set A	Set B	Set C	Avg.
Clean	0.42	0.42	0.44	0.43
20 dB	0.71	0.82	0.91	0.79
15 dB	1.22	1.24	1.28	1.24
10 dB	2.55	2.63	2.80	2.64
5 dB	6.87	7.25	7.74	7.02
0 dB	24.35	23.25	21.46	23.33
-5 dB	63.06	61.75	60.65	62.05
Avg. (0–20 dB)	7.14	7.04	6.83	7.03

The last part of Table 7.1 shows the performance of the stranded system configured as several multi-path HMMs. We modify the mixture transition matrices of the well-trained stranded system, such that each row has 1 in one entry and 0 everywhere else. This operation is (loosely) referred to as diagonalizing. For the within-state transition matrices, ones are

assigned to the main diagonals. For the cross-state transition matrices, ones are assigned to those entries with probabilities as high as possible, provided the resulting matrix is a permutation matrix. After diagonalizing the mixture transition matrices, the re-estimation is then repeated for several times until convergence. Hence, the first row of the last part of Table 7.1 represents the system of cross-coupled parallel HMM paths [70], and the third row for a mixture of separate parallel paths [85], [47]. It is shown that these multi-path HMMs do not produce higher recognition accuracy than the stranded system, and the models whose cross-state transition matrices are diagonalized perform even worse than the regular HMM system. This observation may indicate that the multi-path HMMs, in a simplistic setup as described in this chapter, over-correct the trajectory folding problem associated with the conventional HMMs.

Finally, we analyze the distributions of the mixture transition probabilities for the 20-mixture stranded system. Usually, the more peaked the transition probabilities, the more discriminability they may hold. The outgoing transition probabilities of each mixture are sorted in a descending order. Then the within-state and cross-state transitions of all mixtures at separate order levels are pooled, respectively, and their statistics are illustrated with the box plots in Figure 7.7. It is observed that the ordered probabilities decay dramatically along with the level of orders. In fact, if we place a threshold of 10^{-5} on the effective outgoing transitions, the average fan-out will be 5.0 for within-state, and 6.7 for cross-state, respectively. Moreover, the first order bar of the within-state transitions, which mainly consists of the self-loop transitions, is more prominent than the first order bar of the cross-state transitions.

7.4 Summary

In this chapter, we propose the stranded HMM to explicitly model the relationship among the mixture components, and achieve more accurate representations of heterogeneous data. In the stranded HMM, the observations are assumed to be generated through two layers of Markov chains, where the transitions between mixture components are synchronized with the transitions between states. This synchronous topology can be exploited to greatly

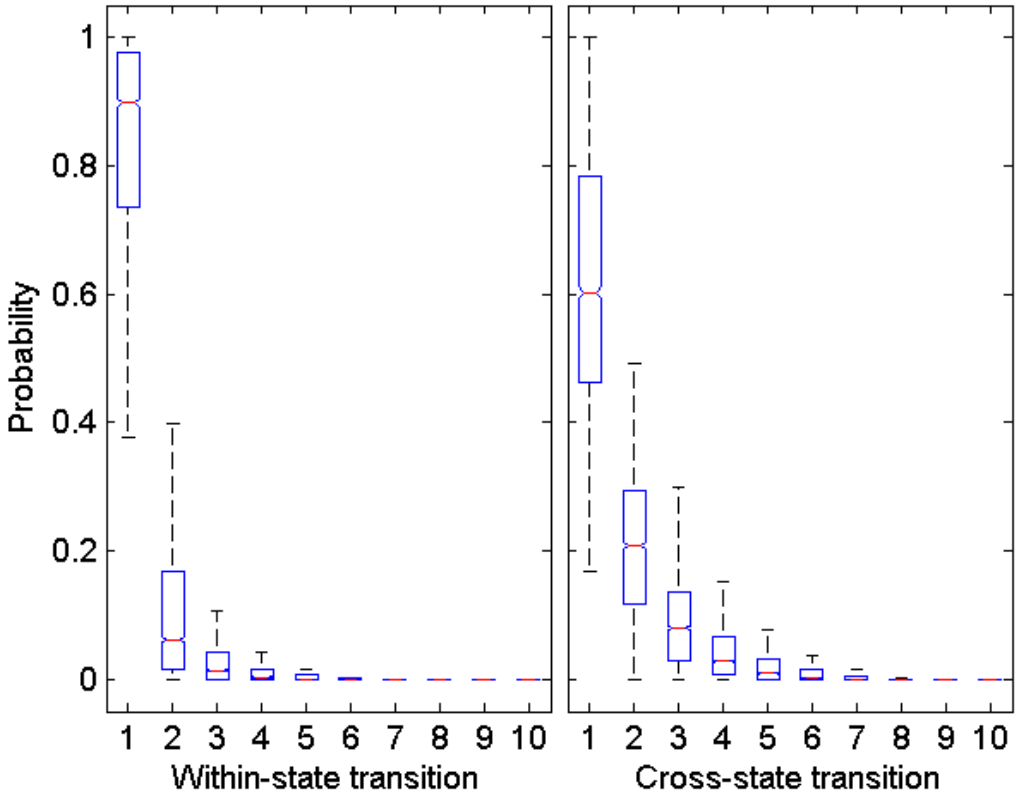


Figure 7.7: Box plots of the ordered outgoing transition probabilities for the 20-mixture stranded system. Top 10 transitions for within-state and cross-state are enclosed, respectively.

simplify the model re-estimation and inference procedure. Due to its close similarity to the Gaussian mixture HMM, the stranded system can be learned starting from an existing GMM system to achieve a further improvement. Moreover, the stranded system is less expensive and can work readily with many acoustic modeling techniques established in the literature, like MLLR and HLDA. Our initial experiments on the Aurora 2 database have shown the significant gain with the standard HMM system, encouraging further investigation on more challenging tasks.

CHAPTER 8

SYNCHRONOUS HMM

A common practice to address the speaker and environmental variabilities for the speech recognition system is to estimate the parameters of the acoustic models from speech data that cover a large variety of acoustic conditions. However, the multistyle training may not fully realize its performance potential as the conventional HMM-based acoustic models are excessively diffused by the heterogeneity of the multistyle data. In this chapter, we consider a novel acoustic modeling framework, named synchronous HMM, which takes full advantage of the capacity of the diversified speech data and achieves an excellent balance between modeling accuracy and robustness. In contrast to conventional HMMs, the synchronous HMM introduces an additional layer of latent variables, referred to as substates, between the HMM state and the Gaussian component variables. The substates have the capability to register long-span non-phonetic attributes, such as gender, speaker identity, and environmental condition, which are integrally called speech scenes in this study. This hierarchical modeling scheme allows an accurate description of the probability distribution of speech units in different speech scenes. To address the data sparsity problem, a decision-based clustering algorithm is presented to determine the set of speech scenes and to tie the substate parameters. Moreover, we propose the multiplex Viterbi algorithm to efficiently decode the synchronous HMMs within a search space of the same size as for the standard HMMs.

The synchronous HMM contains the stranded HMM described in the previous chapter as a special case. When the observation distribution of the substates degenerates to a single Gaussian, the synchronous HMM is equivalent to the stranded HMM.

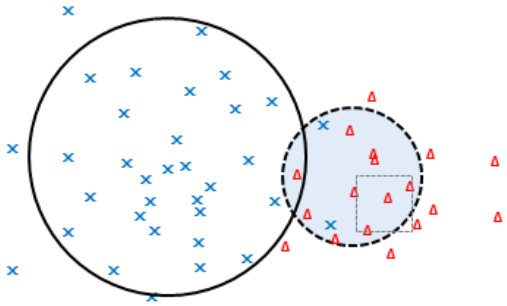
8.1 Background and Motivation

It is widely known that the performance of a speech recognition system often degrades dramatically if it is operated under mismatched operating conditions. A common practice to

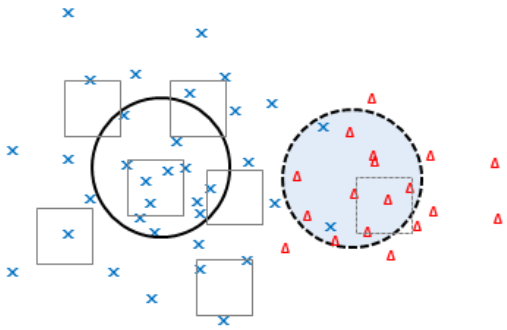
ameliorate this mismatch problem, known as multistyle training, is to collect large amounts of speech data from a variety of acoustic conditions for training the acoustic models. In this way, the space of the resulting acoustic models is large enough to partially overlap with the target albeit unknown condition, and the recognition performance could be retained. The concept behind multistyle training is illustrated in Figure 8.1a. However, the multistyle training may be saturated as the acoustic models are excessively diffused to accommodate the extraneous variabilities introduced by the tremendous amounts of speech data. Many approaches have been proposed to take full advantage of the capacity of the diversified speech data and achieve a more accurate representation of speech from highly heterogeneous sources.

One way is to incorporate the feature normalization in the feature extraction process to reduce the extraneous variability. The most popular form is cepstral mean normalization (CMN) [7], which removes the mean of cepstral features to address the convolutional effect. In addition, cepstral variance normalization (CVN) [90] normalizes the cepstral features to unit variance, which has been shown to improve the robustness to additive noise. In [51], [18], vocal tract length normalization (VTLN) reduces the inter-speaker variability by scaling the frequency axis for each speaker. The feature normalization approaches can not remove all of the factors due to speaker and environmental variations. The remaining variability should be taken into account by the acoustic models.

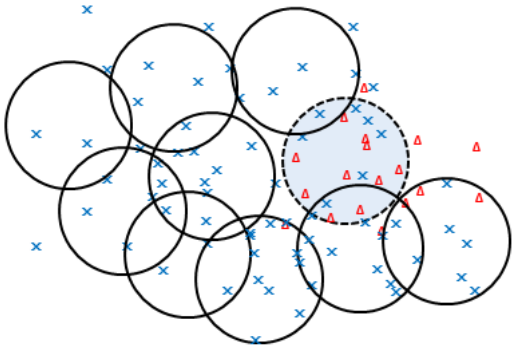
One approach to handing the variability in the back-end model domain is to use adaptive training [6]. The basic idea of adaptive training, as illustrated in Figure 8.1b, is to incorporate adaptation transforms during training. A set of transforms are estimated in response to different speaker and environmental conditions of the training data. The speech model is then estimated based on these transforms, leading to a canonical model that is expected to represent the intrinsic variability of the speech. Adaptive training schemes can be classified in terms of the functional form of the transformations. One popular form of the transformation is a linear one, such as the structured affine transformations used in maximum-likelihood linear regression (MLLR) [52], [25] and constrained MLLR (CMLLR)



(a) Multistyle training



(b) Adaptive training



(c) Multiple modeling

Figure 8.1: Three schemes of modeling heterogeneous data sources for speech recognition. Training and test data points are denoted in cyan (×) and red (△), respectively. Circles denote speech models, squares transforms. Models estimated from the training data are drawn with solid lines, and models from the test data with dashed lines.

[15]. VTS utilizes a nonlinear transformation that is specifically prescribed to tackle the additive noise and convolutional distortion. The use of VTS transforms for adaptive training is explored in [38], [43], and have been described in Chapter 3.

There are some limitations with adaptive training schemes. First, the decomposition

into a canonical model and its associated transforms is a simplified way to characterize the real-world speech. In particular, some transforms like the ones in MLLR are statistically driven and lack the strong physical justification. Because of this, and also because the transforms are usually compact to be robustly estimated, the canonical model still contain considerable variations, though less severe than do the multistyle-trained acoustic models. Second, the transforms that are estimated in the training stage are discarded, failing to take full advantage of the diversified training data. An example of utilizing these transforms is acoustic sniffing for adaptive training [78]. Consider that the multistyle training data contains a subset of utterances collected in a similar condition to the test utterances. If we can identify this training subset through the knowledge from the acoustic conditions or by some automatic classification methods, we may save the time-consuming adaptation step by applying the corresponding known transform to generate the adapted models.

Another class of approaches that address the modeling of heterogeneous speech data is to use an ensemble of models, each focusing on a particular acoustic condition, as illustrated in Figure 8.1c. The motivation is that the conventional HMM represents a speech unit by a string of HMM states, which is insufficient to describe accurately the diversified speech features. A simplistic way to generate multiple models is to divide the training corpus into a number of homogeneous blocks, and then train an HMM set for each block. These homogeneous blocks can be determined through prior knowledge such as the gender, the speaker group identity, the speaking rate, the environmental condition, and the noise level, or by statistical clustering methods. Multiple models can be generated in other ways, such as the use of different feature sets [111], [86], randomized decision trees [82], [101], and different feature normalization and model adaptation strategies [81].

Recognition can be performed by running multiple recognizers of these models in parallel. The recognition hypothesis is obtained by either combining the decoding outputs of the multiple recognizers through majority voting in a ROVER-like paradigm [20], or choosing the one with the highest likelihood. An alternative way of combining multiple models is to preselect one model set that best matches the operating condition for recognition. The model pre-selection procedure significantly reduces the computational complexity as

compared to running parallel recognizers; however, if an error occurs in the pre-selection procedure, the recognition hypothesis would be sub-optimal.

Moreover, multiple models can be combined at the frame level to achieve a more granular form of combination. The most common forms are cluster adaptive training (CAT) [27] and eigenvoice [48], where the parameters of the target model used for recognition are obtained as a linear interpolation of multiple speaker/cluster-dependent models. Models can also be combined by their likelihoods through linear or log-linear combination functions [101], [56]. In general, the combination weights are estimated from the adaptation data using the ML criterion. For likelihood combination, the combination weights have implications for the priorities of the corresponding models. Hence, they may be also determined based on the recognition accuracy or some confident measures of individual models.

The multi-model approach is an attractive scheme to address the issue of data heterogeneity for speech recognition. However, the use of multiple models is not as widespread as might be expected. A number of problems may limit their usefulness. The first problem is the data sparsity in estimating parameters of multiple models. Typically, the speech data are divided into a number of subsets for training multiple models. As the number of the models increases, there will be fewer data available for providing reliable estimation for each individual model. As a result, only simple division of speech data has been explored in large vocabulary recognition systems. The second problem is the heavy computational load in combining multiple models. Following the classical ensemble learning theory, it is expected that the best performance should be obtained by applying the constituent models in parallel to produce a plurality of candidate hypotheses for the majority voting. Unfortunately, this introduces multiple decoding with dramatically increased computational complexity and memory requirements. A similar situation applies to CAT, which usually requires two decoding passes. The first pass generates the initial transcription hypothesis for predicting the interpolation weighs, and the second pass outputs the final hypothesis by operating on the adapted model. Though alternative methods such as model pre-selection can alleviate this drawback, they work at the expense of compromising the recognition accuracy.

In this chapter, we consider a novel acoustic modeling framework, named synchronous

HMM, which takes full advantage of the capacity of the diversified speech data and achieves an excellent balance between modeling accuracy and robustness. In contrast to conventional HMMs, the synchronous HMM introduces an additional layer of latent variables, referred to as substates, between the HMM state and the Gaussian component variables. By specifying a tight dependence of substates on the previous substates, we establish a meaningful interpretation of the sequence of substates, which can be used to capture the long-span non-phonetic attributes. Examples of such attributes are speaker identity, gender, speaking rate, environmental condition, and channel convolution. These attributes are integrally referred to as speech scenes in this study.

The acoustic models based on the synchronous HMMs can be thought of as a collection of multiple acoustic models, each corresponding to a specific speech scene. In this regard, it is related to the multi-model approaches [94], [5], [98]. However, the synchronous HMM offers a number of advantages over the conventional multi-model approaches. First, the hierarchical modeling scheme allows an accurate description of the probability distribution of speech units in different speech scenes. Second, by closely incorporating the models of speech scenes as sub-models of the synchronous HMM, we can determine the model structure and estimate the model parameters in an integral and consistent manner. Furthermore, by exploiting the synchronous relationship among the speech scene sub-models, we propose the multiplex Viterbi algorithm to efficiently decode the synchronous HMM within a search space of the same size as for the standard HMM. The multiplex Viterbi can also be generalized to decode an ensemble of isomorphic HMM sets, a problem often arising in the multi-model systems.

8.2 Synchronous HMM

In contrast to conventional Gaussian mixture HMM, the synchronous HMM introduces an additional layer of latent variables, referred to as substates, between the HMM state and the Gaussian component variables. A substate depends on the previous substate in addition to the state that generates it. Accordingly, the model consists of a quadruple of stochastic processes $(\boldsymbol{x}_1^T, s_1^T, z_1^T, m_1^T)$, where $\boldsymbol{x}_1^T = \boldsymbol{x}_1, ..., \boldsymbol{x}_T$ is a sequence of observations of length T,

and $s_1^T = s_1, ..., s_T$, $z_1^T = z_1, ..., z_T$, and $m_1^T = m_1, ..., m_T$ are sequences of latent variables of HMM states, substates, and mixture indices, respectively. The statistical dependencies between these variables can be represented by a DBN [64] as shown in Figure 8.2.

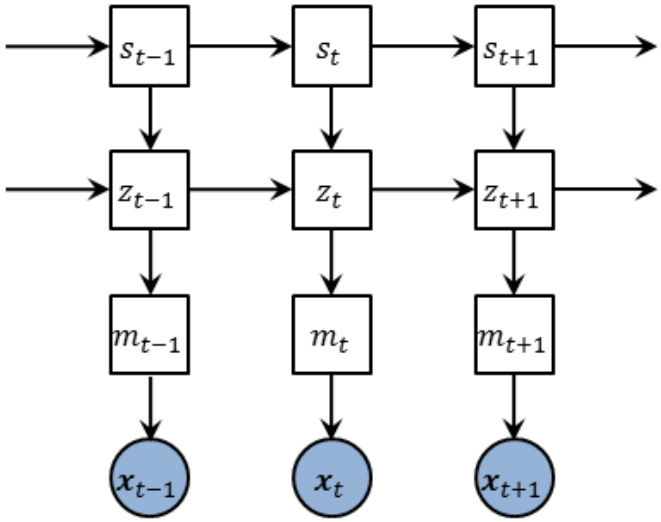


Figure 8.2: Dynamic Bayesian network representation of the synchronous HMM.

The synchronous HMM is motivated by the need to accurately characterize the data from highly heterogeneous sources. Herein, the generation of sequential observations is described in a progressive rendering manner. The Markov chain of the state layer represents the evolution of the primary factor of interest. The dynamics of the state layer is then propagated to the subordinate substate layer while under the influence of secondary, possibly slowly varying, factors. The rendering proceeds to the third layer of mixture components, which control the generation of the final observed sequences. For speech recognition, the state layer represents the process of the canonical speech, and the substate layer represents the process of a variety of real-world speech due to speaker and environmental variations, which are referred to as speech scenes in this study.

One key property of the synchronous HMMs is that the evolution of the substate layer is synchronous with the evolution of the state layer. This effectively eliminates the possible explosion of the state space caused by introducing multiple Markov chains, as for the case of factorial HMM [31]. Suppose that the model consists of N states and each state corresponds

to K substates, which leads to NK substates in the substate layer. Naively, the state space of the synchronous model, which is composed of the direct product of states and substates, would be of size N^2K. However, by imposing the synchronous constraint on the two Markov chains, the state space retains a size of NK.. Moreover, the synchronous HMM can be interpreted as synchronization among substates of different speech scenes. This will lead to substantial computational savings in learning and decoding the model as will be discussed later.

From the DBN in Figure 8.2, the joint probability of these sequences in the synchronous model can be factored as

$$p(\boldsymbol{x}_1^T, s_1^T, z_1^T, m_1^T) = \prod_{t=1}^{T} p(s_t|s_{t-1})p(z_t|z_{t-1}, s_t)p(m_t|s_t, z_t)p(\boldsymbol{x}_t|s_t, z_t, m_t) \tag{8.1}$$

A synchronous HMM consists of the following elements:

- state transition probability

$$a_{j'j} = p(s_t = j|s_{t-1} = j') \tag{8.2}$$

- substate transition probability $p(z_t|z_{t-1}, s_t)$ (to be discussed shortly).

- prior of Gaussian component l from state $s_t = j$ and substate $z_t = k$

$$w_{jkl} = p(m_t = l|s_t = j, z_t = k) \tag{8.3}$$

- likelihood of Gaussian component l from state $s_t = j$

$$p(\boldsymbol{x}_t|s_t = j, m_t = l) = \mathcal{N}(\boldsymbol{x}_t; \boldsymbol{\mu}_{jl}, \boldsymbol{\Sigma}_{jl}) \tag{8.4}$$

Note that, to make effective use of data, the Gaussian components for each state are shared among all substates of that state. Thus the substates from the same state differ only in the mixture weights, analogous to the method used in the semi-continuous HMMs [39]. The substate output distribution thus is given by

$$b_{jk} = P(\boldsymbol{x}_t|s_t = j, z_t = k) = \sum_l w_{jkl}\mathcal{N}(\boldsymbol{x}_t; \boldsymbol{\mu}_{jl}, \boldsymbol{\Sigma}_{jl}) \tag{8.5}$$

The substate transition probability $p(z_t|z_{t-1}, s_t)$ is conditioned on the previous substate z_{t-1} and the current state s_t. Depending on the form of the substate transition probabilities, there are several variants of the synchronous HMM.

When the observation distribution of the substates degenerates to a single Gaussian, the synchronous HMM is equivalent to the stranded HMM as described Chapter 7. One limitation of the stranded HMMs is that the transitions between the substates (i.e., mixture components in the stranded HMM) are bounded inside an HMM. For speech recognition problems, the inference is accomplished in a search network compose of a large amount of subword HMMs. Given the tremendous pairs of the HMMs, it is impossible to estimate the mixture transitions across HMMs. This limits the stranded HMMs to capture the local mixture dynamics inside speech units.

An alternative of the synchronous HMM, which aims to take into account the long-span temporal dependency of speech feature sequence, is to deterministically specify the transitions between substates. Consider that the kth substates of all the states represent speech from a particular speech scene, and the speech scene keeps unchanged during an utterance. The substate transition probability can be written as

$$p(z_t = k|z_{t-1} = k', s_t = j) = \delta(k', k) \tag{8.6}$$

where $\delta(\cdot, \cdot)$ denotes the Kronecker delta function. This means that once the process enters a substate indexed by k, it will be confined to the kth substates of the model until the process terminates. In addition, we can maintain the speech scene dependency across the models by creating multiple dummy substates for each dummy state and drawing links between the corresponding ones. Thus, the kth substates of all the states form a separate sub-model representing the kth speech scene. For the separate-scene synchronous HMMs, the joint probability can be reduced as

$$p(\boldsymbol{x}_1^T, s_1^T, m_1^T, k) = p(k) \prod_{t=1}^{T} p(s_t|s_{t-1}) p(m_t|s_t, k) p(\boldsymbol{x}_t|s_t, m_t) \tag{8.7}$$

The substate transition diagram of the synchronous HMM with separate speech scenes is illustrated in Figure 8.3.

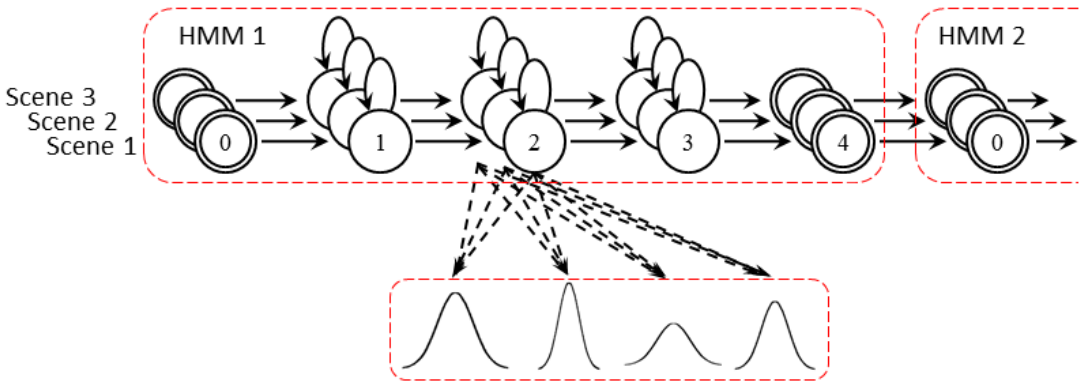


Figure 8.3: Illustration of substate transitions and observation distributions for the synchronous HMM. The substates of three speech scenes are connected separately within and across the models. The processes of different speech scenes are synchronous with each other. The substates from the same state share the pool of Gaussian components and differ in the mixture weights.

Moreover, the speech scenes can be allowed to switch at boundaries of HMMs or words, as given by

$$p(z_t = k|z_{t-1} = k', s_t) = \begin{cases} c_{k'k} & \text{if } s_t \text{ begins a phone/word} \\ \delta(k', k) & \text{otherwise} \end{cases} \tag{8.8}$$

The synchronous HMM with switching speech scenes can capture the speech under the influence of slowly varying factors, such as non-stationary environmental noise.

In this study, we will focus on investigating the synchronous HMM with separate speech scenes.

8.2.1 Estimating Parameters of Synchronous HMMs

The parameters of the synchronous HMM can be learned with an EM algorithm similar to the regular HMM, except that we need to account for scene-dependent mixture weights $p(m_t|s_t, k)$. Since both the states and the mixture components are latent variables, we need to maximize the following EM auxiliary function

$$Q(\hat{\Lambda}|\Lambda) = \sum_k \sum_{s_1^T} \sum_{m_1^T} p(s_1^T, m_1^T, k|\boldsymbol{x}_1^T, \Lambda) \log p(\boldsymbol{x}_1^T, s_1^T, m_1^T, k|\hat{\Lambda}) \tag{8.9}$$

where Λ and $\hat{\Lambda}$ denote the existing and new estimates of the model parameters, respectively. In the E step, the Q function requires finding the following sufficient statistics: the posterior

probability of being in substate k of state j at time t, $\gamma_t(j,k)$; the posterior probability of being in mixture l of substate k of state j at time t, $\gamma_t(j,k,l)$; the joint posterior probability of two successive states for scene k, $\xi_t(j',j,k)$, so that

$$\gamma_t(j,k) = p(s_t = j, z = k|\boldsymbol{x}_1^T, \Lambda) = \frac{\alpha_t(j,k)\beta_t(j,k)}{p(\boldsymbol{x}_1^T|\Lambda)} \tag{8.10}$$

$$\gamma_t(j,k,l) = p(s_t = j, z = k, m_t = l|\boldsymbol{x}_1^T, \Lambda) = \gamma_t(j,k)\frac{w_{jkl}\mathcal{N}(\boldsymbol{x}_t; \boldsymbol{\mu}_{jl}, \boldsymbol{\Sigma}_{jl})}{b_{jk}(\boldsymbol{x}_t)} \tag{8.11}$$

$$\begin{aligned}\xi_t(j',j,k) &= p(s_{t-1} = j', s_t = j, z = k|\boldsymbol{x}_1^T, \Lambda) \\ &= \frac{\alpha_{t-1}(j',k)a_{j'j}b_{jk}(\boldsymbol{x}_t)\beta_t(j,k)}{p(\boldsymbol{x}_1^T|\Lambda)}\end{aligned} \tag{8.12}$$

where we have defined the forward and backward probabilities as

$$\begin{aligned}\alpha_t(j,k) &= p(\boldsymbol{x}_1^t, s_t = j, z = k|\Lambda) \\ &= \Big[\sum_{j'} \alpha(j',k)a_{j'j}\Big] b_{jk}(\boldsymbol{x}_t)\end{aligned} \tag{8.13}$$

$$\begin{aligned}\beta_t(j',k) &= p(\boldsymbol{x}_{t+1}^T|s_t = j', z = k, \Lambda) \\ &= \sum_j a_{j'j}b_{jk}(\boldsymbol{x}_{t+1})\beta_{t+1}(j,k)\end{aligned} \tag{8.14}$$

In the M step, we need to maximize the auxiliary function 8.9. The re-estimation formulas for Λ are similar to those for regular HMMs, except that we sometimes need to sum over the scenes as well as the time indices. We have

$$a_{j'j} = \frac{\sum_t \sum_k \xi_t(j',j,k)}{\sum_t \sum_j \sum_k \xi_t(j',j,k)} \tag{8.15}$$

$$w_{jkl} = \frac{\sum_t \gamma_t(j,k,l)}{\sum_t \sum_l \gamma_t(j,k,l)} \tag{8.16}$$

$$c_k = \frac{\sum_t \sum_j \gamma_t(j,k)}{T} \tag{8.17}$$

$$\mu_{jl} = \frac{\sum_t \sum_k \gamma_t(j,k,l)\boldsymbol{x}_t}{\sum_t \sum_k \gamma_t(j,k,l)} \tag{8.18}$$

$$\boldsymbol{\Sigma}_{jl} = \frac{\sum_t \sum_k \gamma_t(j,k,l)(\boldsymbol{x}_t - \boldsymbol{\mu}_{jl})(\boldsymbol{x}_t - \boldsymbol{\mu}_{jl})^T}{\sum_t \sum_k \gamma_t(j,k,l)} \tag{8.19}$$

To accomplish the basic training process of the synchronous HMM, we still need to determine the speech scenes and properly address the estimation of the scene-specific model parameters (i.e., mixture weights). Basically, we start with a set of well-trained Gaussian

mixture HMMs. We then divide the training corpus into a number of homogeneous subsets based on some criterion, and generate multiple speech scene sub-models by updating the mixture weights of a sub-model based on each subset. Finally, several runs of full-scale re-estimation are carried out without explicitly associating the subsets to the speech scene sub-models. In the next section, we will present a decision tree-based method to automatically generate the set of speech scenes.

8.3 Speech Scene Decision Tree

One issue in applying the synchronous HMM for speech recognition is the data sparsity. The acoustic models based on the synchronous HMMs are essentially composed of multiple sets of the standard acoustic models. When the number of the speech scenes increases, there will be fewer data available for providing reliable estimation of the scene-specific model parameters. The problem becomes particularly severe when many factors related to speaker and environmental variations are accounted for in the speech recognition task. A combination of all these factors may quickly deplete the amount of available training data. In speech recognition, one widely used method for handling the data sparsity problem is to tie the model parameters using clustering techniques. By sharing parameters between the model components that are acoustically or phonetically similar, we can maintain the balance between model complexity and data availability.

In fact, we have applied the principle of parameter tying in defining the synchronous HMM. Inspired by semi-continuous HMM, all substates of a state share the pool of Gaussian components for that state, but differ in the mixture weights. Sharing Gaussian components exploits the fact that there is a high degree of redundancy among different speech scenes. As a consequence, the problem is reduced to the determination of the number of speech scenes and the robust estimation of the scene-specific mixture weights.

We present a decision tree-based algorithm to address this problem, analogous to the phonetic decision tree used for clustering context-dependent phone models [104]. Suppose that the utterances are tagged with the acoustic conditions, such as gender, speaker identity,

and environmental condition. We first produce the synchronous HMMs with as many sub-models as the distinct acoustic conditions in the training corpus. Then the decision tree-based clustering is applied to tie the parameters between those similar model components.

Consider that a set of substates, Z, from state j is tied in a substate cluster. If we assume that during clustering, the Gaussian components remain unchanged, then only the mixture weights of the substate cluster need to be calculated, which can be expressed as

$$w_{j\cdot l}(Z) = \frac{\sum_t \sum_{k\in Z} \gamma_t(j,k,l)}{\sum_t \sum_{k\in Z} \sum_l \gamma_t(j,k,l)}. \tag{8.20}$$

The log-likelihood of Z generating the training data can be approximated by the corresponding auxiliary function given by

$$\begin{aligned} L_j(Z) &= \sum_t \sum_{k\in Z} \sum_l \gamma_t(j,k,l) \log \Big(\log w_{j\cdot l}(Z) \mathcal{N}(\boldsymbol{x}_t; \boldsymbol{\mu}_{jl}, \boldsymbol{\Sigma}_{jl}) \Big) \\ &= \sum_t \sum_{k\in Z} \sum_l \gamma_t(j,k,l) \log w_{j\cdot l}(Z) + \sum_t \sum_{k\in Z} \sum_l \gamma_t(j,k,l) \log \mathcal{N}(\boldsymbol{x}_t; \boldsymbol{\mu}_{jl}, \boldsymbol{\Sigma}_{jl}). \end{aligned} \tag{8.21}$$

Applying (8.20), the first term in (8.21) can be rewritten as

$$\sum_t \sum_{k\in Z} \sum_l \gamma_t(j,k,l) \log w_{j\cdot l}(Z) = \sum_l \Big[\sum_t \sum_{k\in Z} \gamma_t(j,k,l) \Big] \log w_{j\cdot l}(Z) \tag{8.22}$$

$$= \gamma_j(Z) \sum_l w_{j\cdot l}(Z) \log w_{j\cdot l}(Z) \tag{8.23}$$

$$= -\gamma_j(Z) H(w_j(Z)) \tag{8.24}$$

where $H(w)$ denotes the entropy of the mixture weight distribution w.

The decision tree clustering is constructed based on the maximum increase in the log-likelihood. If we split node Z with a question into nodes Z^+ and Z^-, the increase in the log-likelihood $\triangle L_j(Z)$ is given by

$$\begin{aligned} \triangle L_j(Z) &= L(Z^+) + L(Z^-) - L(Z) \\ &= \gamma_j(Z) H(w_j(Z)) - \gamma_j(Z^+) H(w_j(Z^+)) - \gamma_j(Z^-) H(w_j(Z^-)). \end{aligned} \tag{8.25}$$

Note that the second terms in the log-likelihood function (8.21) due to the Gaussian components are canceled out. This leaves the terms involving the mixture weights in the splitting

criterion. It turns out that the criterion of maximum increase in the log-likelihood is equivalent to the maximum reduction of the weighted entropy of the mixture weights.

The speech scene decision tree can be applied globally or at the substates of individual states. The global decision tree is aimed to cluster the speech scene sub-models, and thus to determine the set of sub-models in the final synchronous HMMs. The tree is built in a top-down fashion with the questions relating to the acoustic condition tags. A node in the tree denotes a set of speech scene sub-models, whose log-likelihood is the sum of the log-likelihoods over all the substates of those speech scene sub-models given by

$$\triangle L(Z) = \sum_j \triangle L_j(Z) \tag{8.26}$$

Nodes are iteratively split at each iteration by finding a node and an associated question that jointly produce the maximum increase in the log-likelihood on the training data given by (8.26). On completion of the clustering, the speech scene sub-models in the same cluster can be merged. This leads to a compact set of speech scenes in the synchronous HMM. Merging the equivalent speech scenes also saves the memory requirement and computational cost for decoding the model. Figure 8.4 shows an example of the decision tree which results in six speech scenes.

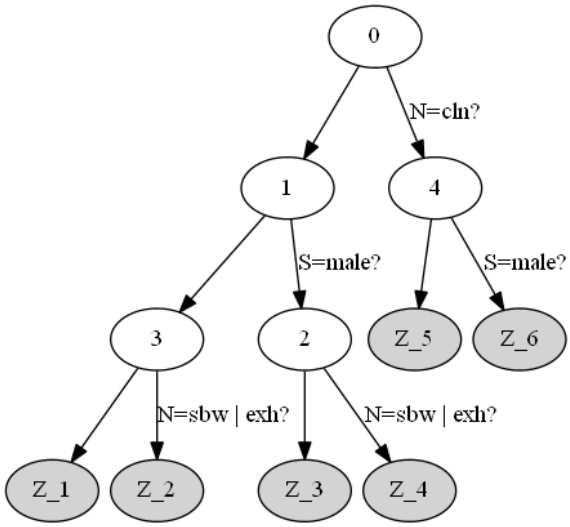


Figure 8.4: Example of a speech scene decision tree. The questions used to split the nodes relate to speaker gender (S), noise type (N), and noise levels (SNR), separately. The non-leaf nodes are indexed in the order of splitting precedence.

Alternatively, we can cluster and tie the substates of individual states for improved

efficiency. We may apply the decision tree for each state following the same top-down clustering procedure. However, there are some problems associated with this approach. First, the training data associated with some speech scenes may be phonetically highly unbalanced, because the speech data in these speech scenes are typically limited to a small set of vocabulary words. This produces a large amount of rarely seen substates, which would be inappropriately clustered just based on the fuzzy clues of the question set. Moreover, building separate trees for different states hinders the possible merging of the speech scene sub-models, because the sub-models can be merged only if they share common clusters for all their substates.

To address this problem, we cluster the substates of individual states by trimming the global decision tree in a bottom-up fashion. We initialize a decision tree of the substates for each state by cloning the topology of the global decision tree. The trimming process starts with pairs of sibling leaf nodes, which are merged if the log-likelihood reduction is less than some threshold, or some leaf nodes lack sufficient data to support themselves. After iteratively merging all of such sibling pairs, we note that some rarely seen leaf nodes have not yet been trimmed, because their siblings are non-leaf nodes. We then merge these dangling leaf nodes with other leaf nodes that result in the minimum reduction in the log-likelihood.

8.4 Multiplex Viterbi Decoding

The decoding process using the synchronous HMM is to find the best path that matches the given observation sequence, through the search space spanned by states and substates. Because of the synchronization between states and substates, finding this best path is equivalent to finding the best substate sequence, or simultaneously finding the best state sequence and speech scene. A straightforward decoding method is to perform the Viterbi algorithm through the search space comprising the substates of the model, where the substate transition probabilities are scaled by the state transition probabilities. Let $\delta_t(j,k)$ be the likelihood of the best substate path ending in substate k of state j at time t. By

induction, we have

$$\delta_t(j,k) = \delta_{t-1}(i^*,k)a_{i^*j}b_{jk}(\boldsymbol{x}_t) \qquad 2{\le}t{\le}T,\ 1{\le}j{\le}N,\ 1{\le}k{\le}K \tag{8.27}$$

$$i^* = \arg\max_i \delta_{t-1}(i,k)a_{ij}b_{jk}(\boldsymbol{x}_t) \qquad 2{\le}t{\le}T,\ 1{\le}j{\le}N,\ 1{\le}k{\le}K \tag{8.28}$$

where i^* is the preceding state that leads to the best path ending in substate k of state j at time t. Note that in a strict sense i^* should be written as $i_t^*(j,k)$ to indicate its dependence on time t, state j, and substate k. Though the standard Viterbi algorithm is guaranteed to find the optimal substate sequence, it leads to a dramatic increase of memory requirements and computational complexity, which roughly correspond to K times of decoding the standard HMM.

We propose a novel multiplex Viterbi algorithm that performs an effective decoding on the synchronous HMM by keeping the search space of the same size as for the standard HMM. The multiplex Viterbi exploits the synchronous topology of the model. The search space is constructed based on the model states, except that each state node is compounded by all the substates of that state. At each time step, the substates of a state share the same path, but keep individual records of the sub-path scores, which are cumulated over the substate sequence following the shared path. The path score of the state takes the highest sub-path score from the constituent substates, and is used to represent the fitness of that state in the Viterbi decoding. Figure 8.5 shows the trellis diagram for the multiplex Viterbi algorithm.

More formally, let $\tilde{\delta}_t(j,k)$ be the sub-path likelihood for substate k following the best partial path ending in state j at time t. By induction, we have the recursion formula

$$\tilde{\delta}_t(j,k) = \tilde{\delta}_{t-1}(i^*,k)a_{i^*j}b_{jk}(\boldsymbol{x}_t) \qquad 2{\le}t{\le}T,\ 1{\le}j{\le}N,\ 1{\le}k{\le}K \tag{8.29}$$

$$i^* = \arg\max_i \left\{ \max_k \tilde{\delta}_{t-1}(i,k)a_{ij}b_{jk}(\boldsymbol{x}_t) \right\} \qquad 2{\le}t{\le}T,\ 1{\le}j{\le}N \tag{8.30}$$

where i^* is the preceding state that leads to the best path ending in state j at time t. Note that in a strict sense i^* should be written as $i_t^*(j)$to indicate their dependence on time t and state j. Nevertheless, we apply the shorter notations for ease of understanding (8.29). i^* is also used to keep track of the Viterbi path, where the longer notation $i_t^*(j)$ would be appropriate.

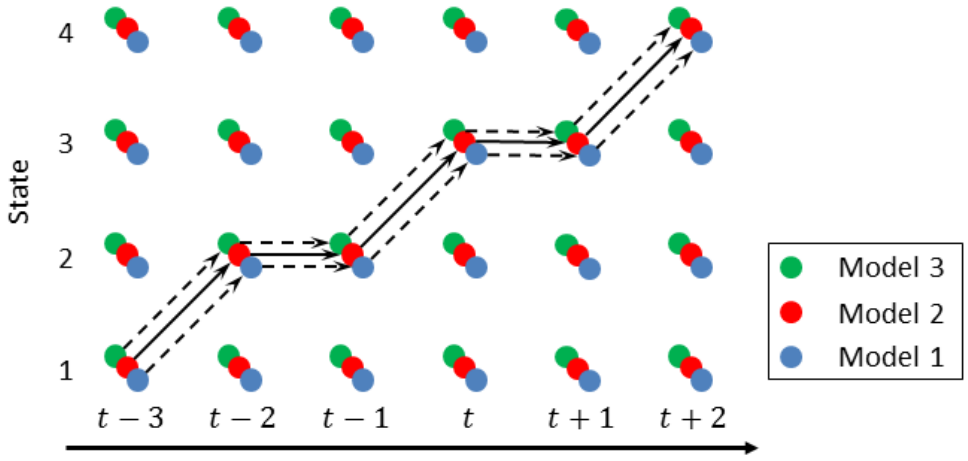


Figure 8.5: Illustration of the multiplex Viterbi algorithm. The substates of a state share the same path, but keep individual sub-path scores.

It should be noted that the multiplex Viterbi finds an approximate solution to the best state sequence, because the state path is jointly determined by the constituent sub-paths. We here assume that if a substate sequence can successfully match the observation sequence, the resulting state/observation alignment should also be appropriate for aligning the observations with other substate sequences following the same state sequence. On the other hand, if a speech scene prefers to a substate alignment that significantly deviates from the reference alignment of the best substate sequence, it may indicate that this speech scene is not appropriate for the acoustic condition of the given utterance and thus can be reasonably ignored. Informal experiments have shown that the multiplex Viterbi does not increase the search errors measurably.

Remarkably, the multiplex Viterbi can be generalized to decode an ensemble of isomorphic standard HMM sets, a problem often arising in the multi-model systems. These HMM sets share the same search space, and equal in the state transition probabilities, but differ in the observation probability distributions (and possibly the features being used). Typically, we are only interested in the best HMM set and the best state sequence that jointly achieve the highest likelihood given the observation sequence. As we can see, the multiplex Viterbi can efficiently address this decoding problem.

Moreover, we can apply the beam search strategy [65], [103] to prune less likely speech

scenes during the multiplex Viterbi to accelerate the decoding speed. We first determine, at each time step, the highest sub-path score for each speech scene. Then the speech scenes whose highest scores fall short of the score of the best speech scene by more than a fixed factor are pruned from further consideration. This pruning strategy is applied to the speech scene level. More effective pruning can be applied to the substates of individual states by comparing the sub-path score of the substates to the path score (which equals to the highest sub-path score) of that state.

The speech scenes can be pruned even before the decoding begins using some environmental classification methods. Environmental classification has been widely used in the multi-model systems [5], [98], [100], [59]. To obviate decoding all of the model sets in parallel, an environmental classifier is learned to detect the acoustic condition given the test utterance or several frames of it, and the HMM set that best matches the acoustic condition is selected for recognition. One limitation of the environmental classification method is that if the environmental classifier makes an error, the recognition hypothesis would be sub-optimal. This limitation can be easily lifted in the multiplex Viterbi. Rather than selecting only one HMM, the environmental classification module can indicate several HMM sets that are relevant to the test condition. Then the multiplex Viterbi is performed to search the best state sequence within a narrowed set of speech scenes. Since the environmental classification module only needs to rule out those highly unlikely speech scenes, a simple GMM classifier would suffice the task.

The major advantage of the multiplex Viterbi is that it significantly reduces the memory requirements and computational complexities in comparison with the standard Viterbi algorithm for decoding K HMM sets. First, by constructing a search space of the same size as for the standard HMM, the multiplex Viterbi eliminates the memory and computational overhead in constructing and maintaining a K-times increased search space. Moreover, the memory overhead in decoding can be further reduced. The multiplex Viterbi needs an extra array per state to store the sub-path scores than the standard Viterbi of one HMM set. Since the sub-path scores are much less important than the path scores, the sub-path scores at time t can be discarded after the Viterbi search proceeds to time $t+1$. Therefore, the

multiplex Viterbi of K HMM sets only causes a slight increase in memory over the standard Viterbi of one HMM set. We note that now the computational overhead of the multiplex Viterbi is mainly due to calculating the acoustic scores. In a naive implementation of the multiplex Viterbi, this overhead could not be reduced. However, we can apply the speech scene pruning strategy to effectively reduce this computational load.

8.5 Experimental Results

The proposed algorithm is evaluated on the Aurora 2 database [34] of connected digits. The multistyle training set is used to train the acoustic models. It consists of noisy data involving four types of noise (subway, babble, car, and exhibition hall) at four SNRs (20, 15, 10, and 5 dB), along with clean data, totaling 17 noise conditions. We further split the training set by gender and obtain 34 subsets as the basic speech scenes for the synchronous HMMs.

All of the experiments are accomplished with a modified version of the hidden Markov toolkit (HTK) [103]. The baseline HMMs are obtained following the standard Aurora 2 recipe for the complex back end. Each digit is modeled by the whole word left-to-right HMM, consisting of 16 states and 20 Gaussian components per state. Besides, a 3-state silence model and a 1-state short pause model with 36 Gaussian components per state are used. Each feature vector consists of 13 mel-cepstral coefficients (including zeroth order for the energy term), and their delta and delta-delta coefficients. Features are normalized by CMN at the sentence level. The HMM baseline yields word error rate (WER) of 7.53% by averaging over SNRs between 20 and 0 dB of three test sets.

8.5.1 Performance of Synchronous HMMs

Table 8.1 shows the performance of the baseline synchronous HMM systems by manually determining the speech scenes. Three synchronous systems are examined and the corresponding speech scenes are described with the set notation in the second column of the table. The first system consists of two speech scenes corresponding to male and female speech. The second system consists of six scenes by further dividing the data in terms of three noise levels (clean, high SNR, and low SNR). This division is similar to the one used

in [87] for speaker and environmental clustering. The third system uses all combinations of genders, noise types, and noise levels that are contained in the training data. It is observed that as the number of the speech scenes increases, the performance of the synchronous models is gradually improved. Specifically, the 34-scene synchronous HMMs achieve the lowest WER of 6.27%, 17% relative reduction over the baseline HMMs. It is interesting to note that simply clustering scenes by gender can obtain half of the gain. This indicates that the gender difference remains an important factor in recognizing speech in adverse environments.

Moreover, the multiplex Viterbi algorithm greatly improves the decoding speed in comparison with the standard Viterbi for decoding K HMM sets. In particular, the multiplex Viterbi decoding for the 34-scene system takes 7.9 times the HMM baseline decoding time, at least four times faster than the corresponding multi-model approach. Compared with the conventional HMMs, the extra computational load of decoding the synchronous HMMs mainly involves the log-sum operations over the mixture components for calculating the substate output probabilities. By applying a regression analysis on the decoding time of the synchronous HMMs with respect to different numbers of speech scenes, it is shown that the likelihood calculations takes about 20% of the HMM baseline decoding time for each speech scene. We will demonstrate later that the multiplex Viterbi can be further sped up without loss of accuracy using the speech scene decision tree and speech scene pruning.

Table 8.1: WER (%) and decoding time (times the HMM baseline) of the synchronous HMMs with different numbers of speech scenes manually determined on the Aurora 2 task.

# of scenes	Set of scenes	WER	Avg. time
1	—	7.53	1.0
2	{Female, Male}	6.90	1.6
6	{Female, Male} × {cln, 15-20 dB, 5-10 dB}	6.46	2.5
34	{All combinations}	6.27	7.9

Since the speech scenes sub-models in the synchronous HMMs differ only in the mixture weights, it is worth investigating the distribution of the mixture weights. Figure 8.6 shows the mixture weights for various scenes of a particular speech state in the 34-scene system. As expected, we see that the mixture weights are sparse and each substate only relates to

a small number of Gaussian components from the Gaussian pool of that state. This is in contrast to the conventional Gaussian mixture HMMs, where the mixture weights for each state are approximately equal and noninformative. Moreover, the scenes for female and male speech largely involve different Gaussian components. For example, the fourth and fourteenth components are used for the female speech, and the first and the eighth Gaussian components are used for the male speech. Further inspection of the figure reveals that the subway (sbw) and exhibition hall (exh) speech have similar weight distributions, and so do the babble and car speech. This observation is also confirmed in the experiment of the speech scene clustering using the decision tree, as will be seen later.

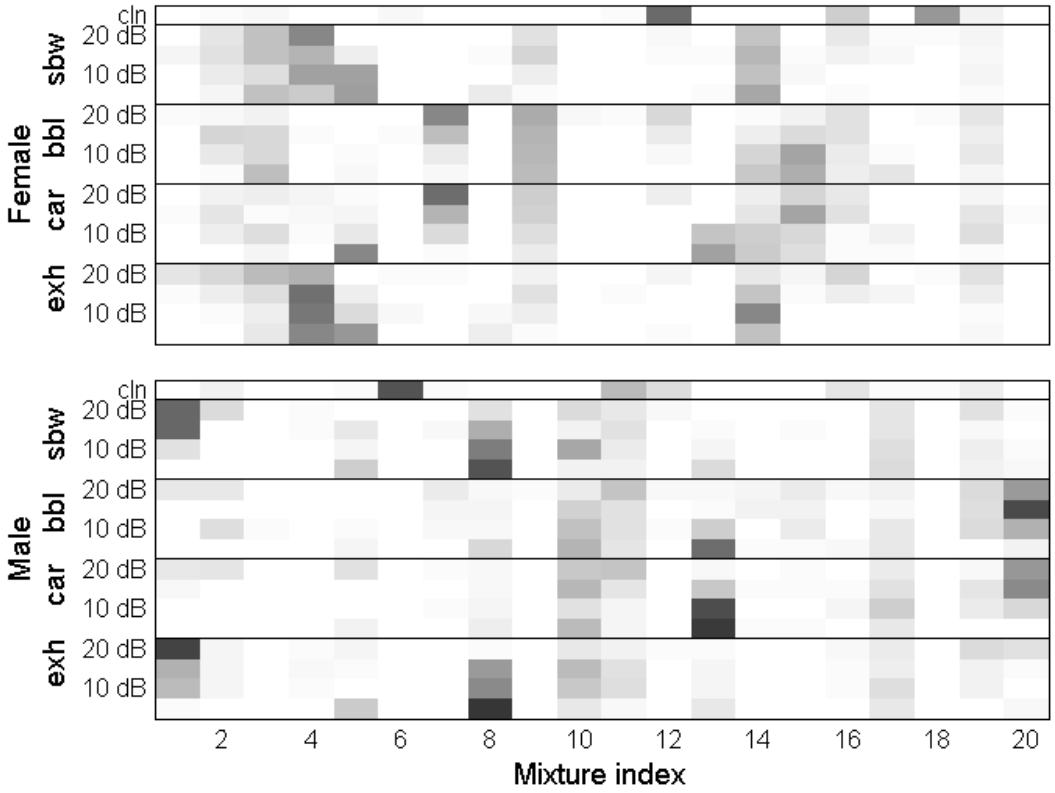


Figure 8.6: Magnitudes of the mixture weights for different scenes of a particular state in the 34-scene system. The rows correspond to the speech scenes, which are sorted according to gender, noise type, and noise level. The darker the color, the more prominent the weight is.

To further quantify the sparsity of the mixture weights in the synchronous HMMs, we

compute the average perplexity of mixture weights for the scene k as

$$PP_k = \frac{1}{N} \sum_j 2^{-\sum_l w_{jkl} \log_2 w_{jkl}} \tag{8.31}$$

Where N denotes the number of states in the synchronous HMM. Obviously, if the weights of L Gaussian components are equal, the perplexity is L. Figure 8.7 shows the average weight perplexities over the speech states for different scenes in the 34-scene system. It is observed that the perplexities for these scenes range from 3.6 to 7.5, given 20 mixture components for each speech state. By comparison, the average weight perplexity for the HMM baseline is 19.5. The very low weight perplexities in the synchronous HMMs confirm that the mixture weights are sparse and the constituent sub-models make use of sparse weights to select the relevant mixture components to represent a specific speaker and environmental condition.

Moreover, the average weight perplexities vary with genders, noise levels, and perhaps noise types. For example, the scenes of the female speech have slightly higher perplexities than do the corresponding scenes of the male speech. This difference may be due to the trade-off between the dynamics of individual speech scenes and their similarities to other scenes.

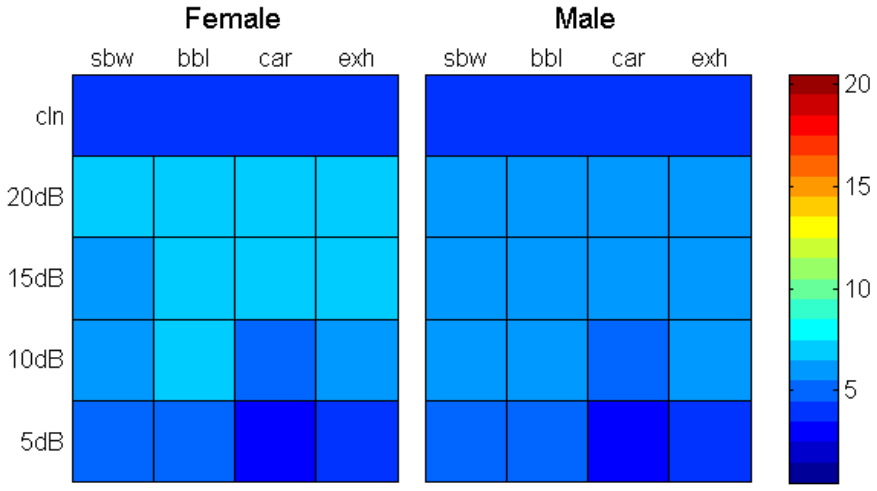


Figure 8.7: Average perplexities of the mixture weights for different scenes in the 34-scene synchronous HMMs.

8.5.2 Speech Scene Decision Tree

We next investigate the effect of the speech scene decision tree to determines the speech scenes and ties the substate parameters in a data-driven manner. Figure 8.8 shows the produced global decision tree which clusters 34 basic speech scenes into 18 scenes for the Aurora 2 task. As can be seen, the decision tree grows in a symmetric fashion. In particular, the left subtree of the root node, which involves all the noise-corrupted speech, is expanded layer by layer. The most useful questions that split the trees are concerned with noise types and speaker genders. Furthermore, we see that the four environmental noises in the training data are grouped in two clusters: 1) subway and exhibition hall; 2) babble and car. This fact has been observed when examining the mixture weight distributions for different scenes in the previous experiment. One interesting issue about this classification is that since the car noise is stationary and the babble noise is non-stationary, what factors play a role in grouping the two environmental noises. We will provide an explanation from the noise spectrum perspective later in Section 8.5.4.

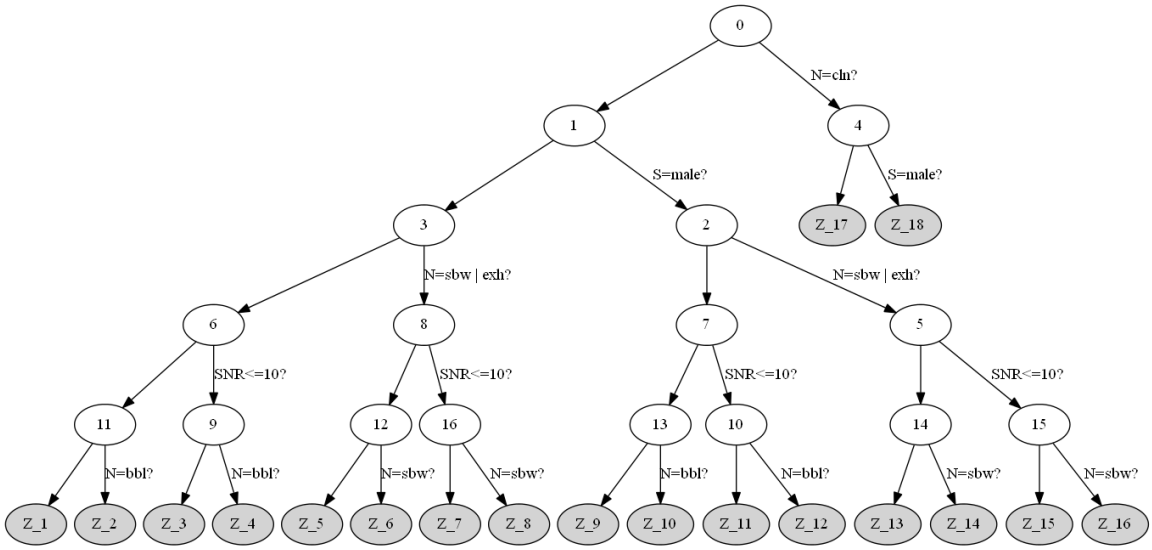


Figure 8.8: A speech scene decision tree built for the Aurora 2 task.

By modifying the stop criterion for the global speech scene clustering, the synchronous HMMs with different numbers of scenes are prepared for the evaluation, as shown in Table 8.2. Due to the symmetric structure of the decision tree, we can still encode the clustering

results for those selective numbers of speech scenes with a shorthand notation. It is observed that the synchronous HMMs with 18 clustered speech scenes maintain the same accuracy as the unclustered 34-scene system, while decoding in a reduced time. This demonstrates that the speech scene decision tree can automatically determine an appropriate set of speech scenes to strike a balance between decoding efficiency and recognition accuracy. We do not see that the speech scene clustering would improve the recognition performance over the unclustered 34-scene system. This implies that the training data in the Aurora 2 corpus are sufficient for reliably estimating the mixture weights of 34 speech scene sub-models. This differs from what have been known for the state tying of triphone models, i.e., the tied-triphone system allows for a more robust estimation of models and thus outperforms the untied counterpart.

Table 8.2: WER (%) and decoding time (times the HMM baseline) of the synchronous HMMs with different numbers of speech scenes clustered using the decision tree on the Aurora 2 task.

# of scenes	Set of scenes	WER	Avg. time
1	—	7.53	1.0
2	{cln, no cln}	7.07	1.6
6	{Female, Male} × {cln, sbw|exh, bbl|car}	6.61	2.5
10	{Female, Male} × {cln, {sbw|exh, bbl|car} × {15-20 dB, 5-10 dB}}	6.38	3.3
18	{Female, Male} × {cln, {sbw, bbl, car, exh} × {15-20 dB, 5-10 dB}}	6.25	5.0
34	{All combinations}	6.27	7.9

Comparing with Table 8.1, we see that manually determining the speech scenes produces a slight advantage over the data-driven approach. However, this may be due to the inconsistency between the training conditions and the test conditions. For example, the Aurora 2 evaluation metric does not account for the performance on the clean speech, and so dividing the training data into clean and corrupted speech may be not as useful as dividing the data by gender.

Table 8.3 gives detailed WER for the synchronous HMMs with 18 clustered speech scenes on Test Set A, B, and C with respect to different SNRs. We see that the system performs much worse on the test speech at -5 and 0 dB SNRs than it does at 5 dB–20 dB SNRs.

This is not surprising as the speech under severe distortion is more formidable to recognize. Nevertheless, the lack of very low-SNR training data aggravates the problem because the synchronous HMMs could not pick a suitable sub-model to accommodate speech from such a severely adverse condition.

Table 8.3: Detailed WER (%) for the 18-scene synchronous HMMs on the Aurora 2 task.

SNR	Set A	Set B	Set C	Avg.
Clean	0.38	0.38	0.41	0.39
20 dB	0.67	0.69	0.85	0.71
15 dB	0.95	1.10	1.06	1.03
10 dB	2.13	2.43	2.46	2.31
5 dB	5.75	6.43	6.42	6.15
0 dB	21.46	21.25	19.89	21.06
-5 dB	60.58	60.18	58.26	59.96
Avg. (0–20 dB)	6.19	6.38	6.14	6.25

Furthermore, the substates are tied for each individual state for improved efficiency, as shown in Table 8.4. It is shown that the proper setting of the substate based clustering can maintain the recognition performance while reducing the model size and accordingly the computational load. On completion of the substate clustering, some speech scene sub-models may become effectively identical, since they share the common clusters for all their substates. Thus, these speech scenes can be merged at a global level. The numbers of distinct speech scenes after clustering are listed in the third column of Table 8.4.

Table 8.4: WER (%) of the synchronous HMMs using the state-based scene clustering on the Aurora 2 task.

# of scenes	Avg. # of tied substates	# of distinct scenes	WER
34	34	34	6.27
34	18.7	34	6.26
34	9.8	34	6.23
34	6.3	18	6.33
18	18	18	6.25
18	8.8	18	6.25
18	6.2	16	6.34
18	4.1	11	6.56

8.5.3 Multiplex Viterbi with Speech Scene Pruning

The multiplex Viterbi search can be accelerated by pruning unlikely speech scenes. Table 8.5 shows the WER and the decoding time of the synchronous system with varying scene pruning thresholds. The system being used represents the most compact and accurate model set we have achieved so far, which consists of 18 speech scenes, 8.8 tied substate per state in average. It is shown that few search errors occur when the scene pruning threshold is 50 or larger. In particular, at the pruning threshold of 100, the decoding search takes 2.0 times the HMM baseline decoding time, saving the computational cost with a factor of 9 compared with the simple multi-model approach.

Table 8.5: WER (%) and decoding time for the multiplex Viterbi with different scene pruning thresholds on the Aurora 2 task. The synchronous HMMs consist of 18 speech scenes, 8.8 tied substates per state in average.

Scene pruning threshold	WER	Avg. time
—	6.25	3.0
200	6.24	2.4
100	6.25	2.0
50	6.35	1.7
20	6.81	1.4

8.5.4 Noise Spectrum Analysis

An interesting observation drawn from the previous experiments with the synchronous HMMs is that the four noise types used in the Aurora 2 multistyle training data can be grouped in two clusters: 1) subway and exhibition hall; 2) babble and car. As the car noise is stationary and the babble noise is non-stationary, it is worth investigating what factors contribute to grouping the two apparently different noises.

By inspecting the spectrogram and spectrum of the noise signals, we may get some clues to interpret this grouping decision. Figure 8.9 shows the spectrograms of the four noise signals, each 20 second length, which visually display the stationarity of the noise signals. we can see that the car noise is stationary and the babble noise consists of non-stationary segments. In addition, the subway train noise appears periodically stationary, interleaved with the vertical lines, which come from the train riding on a railroad switch or

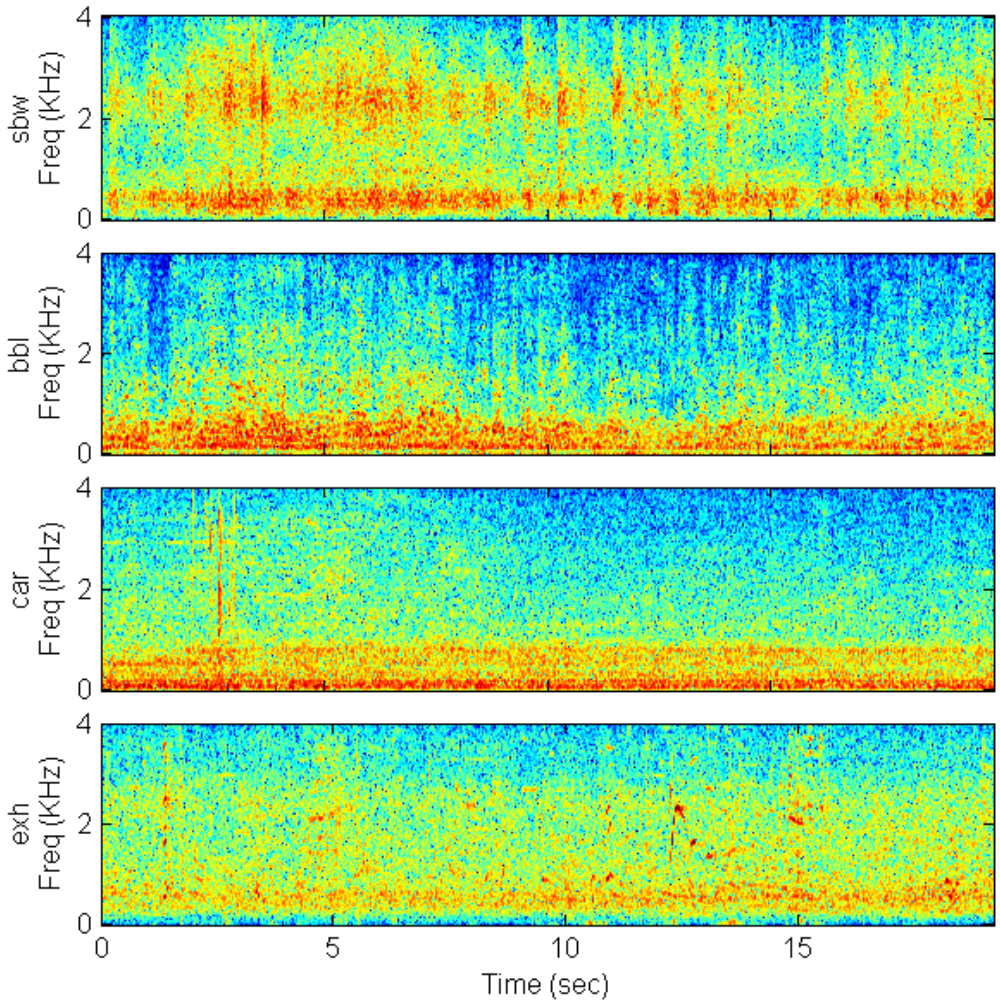


Figure 8.9: Spectrograms of four noise signals in the Aurora 2 multistyle training data.

joint.

On the other hand, Figure 8.10 shows the spectra of the noise signals by averaging the spectral amplitude over frames[1]. As can be seen, the spectral patterns of babble and car are quite similar, both concentrating their energy in the low frequency region. The spectrum of the subway noise has two peaks, and shares one peak with the exhibition hall noise around 500 Hz. Basically, the subway noise looks closer to the exhibition hall noise than the babble and car noises. To quantize the spectral similarity, Table 8.6 gives the distance table of the

[1]The similar spectral images have been plotted in the original paper introducing the Aurora 2 corpus [34].

log-spectral distance of noise signals. It is clear that if we perform clustering on the basis of the table, the results will coincide with the one we have obtained using the synchronous HMMs.

The agreement between the result of the speech scene clustering and the spectrum analysis implies that the HMM-based acoustic models are more sensitive to the shape of noise spectrum than the dynamics of the noise process. It is believed that the short-time spectrum-based features and Gaussian mixture observation probabilities contribute to the resistance of the system to the non-stationarity of interfering noise.

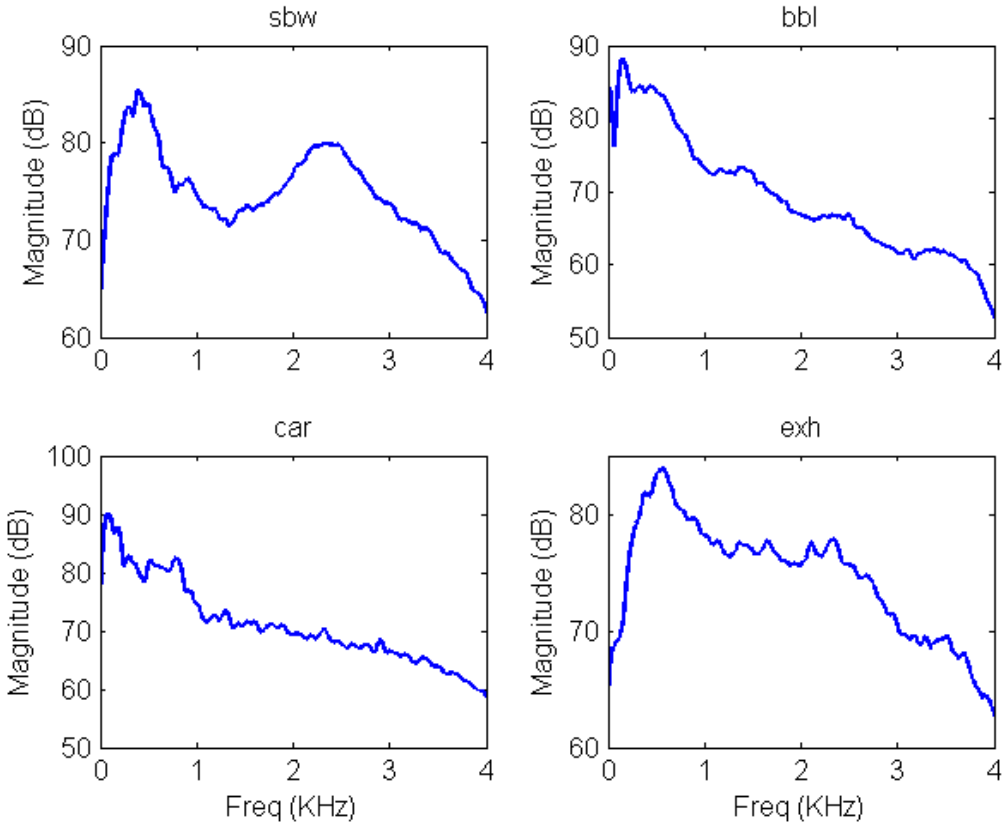


Figure 8.10: Long-term spectra of four noise signals in the Aurora 2 multistyle training data.

8.5.5 Synchronous HMMs with Advanced Frond-End

One substantial advantage of the synchronous HMMs is that it can be readily combined with other techniques used in state-of-the-art speech recognition systems. This section presents

Table 8.6: Log-spectral distance of four noise signals in the Aurora 2 multistyle training data.

	sbw	bbl	car	exh
sbw	0	8.39	6.68	3.22
bbl	8.39	0	3.17	7.93
car	6.68	3.17	0	6.55
exh	3.22	7.93	6.55	0

experiments which combine the synchronous HMMs with ETSI advanced front-end (AFE). The ETSI AFE has achieved great success in robust speech recognition [60] by integrating several noise robustness methods, such as two-stage Wiener filter, SNR-dependent waveform processing, cepstrum calculation, and blind equalization. By substituting the AFE feature for the MFCC feature and keeping other settings the same, the acoustic models are rebuilt using the Aurora 2 multistyle training data. The 20-mixture HMM baseline using the AFE gives a WER of 6.38%, which is significantly better than the system fed with the MFCC feature.

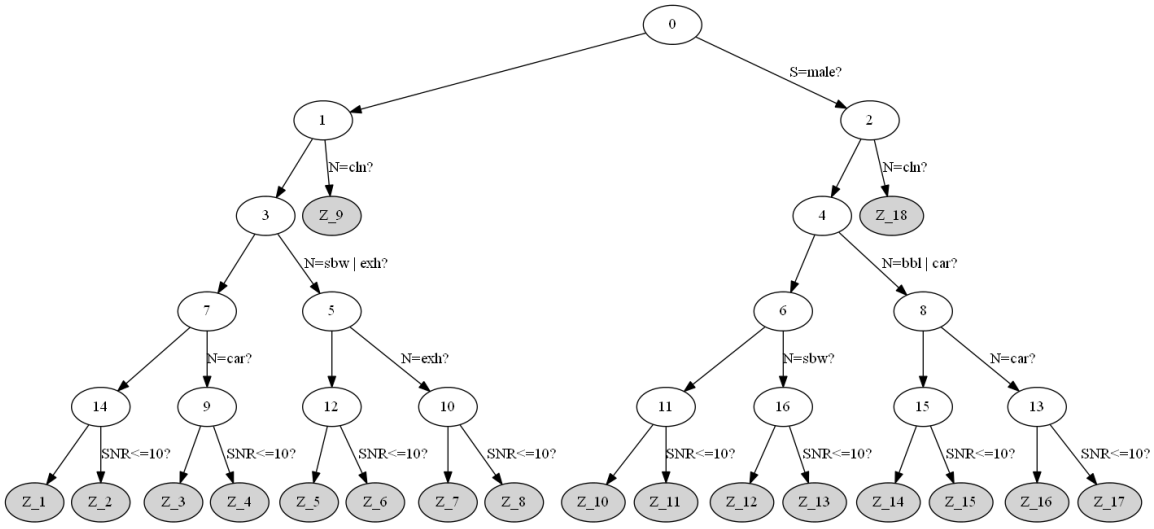


Figure 8.11: Scene decision tree for the synchronous HMMs with the AFE feature.

Figure 8.11 illustrates the speech scene decision tree generated for the synchronous HMMs with the AFE. In comparison with Figure 8.8 for the synchronous HMMs with the MFCC feature, it shows that while the two decision trees consist of the sam leaf nodes of clusters, their structures are different. In the synchronous HMMs with the AFE, the gender

question is asked at the root node of the tree and the noise type questions completely dominate over the noise level questions in growing the tree, whereas in the synchronous HMMs with the MFCC feature, the clean-or-noisy question is used to split the root node and the noise level question (SNR $\leq$ 10dB) is shown to be important in clustering the speech scenes. The structural rearrangement is attributed to the effectiveness of the AFE in noise suppression.

Table 8.7 gives the recognition accuracy using different numbers of scenes. As can be seen, the synchronous HMMs with the AFE also gives a significant performance improvement over the corresponding HMM baseline. Nevertheless, most of the gain comes from dividing the speech data by gender. Additional differentiation between noise types slightly improves the accuracy to a WER of 5.59%, and the differentiation between noise levels does not benefit. These results evidenced that the AFE is effective in removing the noise mismatch and the genders retain the discriminative power after the treatment of the AFE. Moreover, the proposed scene clustering algorithm detects the discriminability automatically and generates individual scenes if necessary. Table 8.8 gives detailed WER for the synchronous HMMs with 10 clustered scenes on Test Set A, B, and C with respect to different SNRs.

Table 8.7: WER (%) of the synchronous HMMs with the AFE features in different numbers of scenes on the Aurora 2 task.

# of scenes	Set of scenes	WER
1	—	6.38
2	{Female, Male}	5.73
10	{Female, Male} $\times$ {cln, sbw, bbl, car, exh}	5.59
18	{Female, Male} $\times$ {cln, {sbw, bbl, car, exh} $\times$ {15-20 dB, 5-10 dB}}	5.63
34	{All combinations}	5.62

8.6 Summary

In this paper, we have proposed the synchronous HMM by introducing an additional substate layer into the standard HMM. The speech scene decision tree is proposed to determine the optimal set of speech scenes and tie the substate parameters. Moreover, we propose a

Table 8.8: Detailed WER (%) for the 10-scene synchronous HMMs with the AFE features on the Aurora 2 task.

SNR	Set A	Set B	Set C	Avg.
Clean	0.25	0.25	0.35	0.27
20 dB	0.58	0.61	0.68	0.61
15 dB	0.98	1.06	1.08	1.03
10 dB	2.04	2.28	2.63	2.25
5 dB	5.01	5.95	6.61	5.70
0 dB	16.79	18.39	21.67	18.40
-5 dB	47.98	49.99	56.09	50.40
Avg. (0–20 dB)	5.08	5.65	6.53	5.59

novel multiplex Viterbi algorithm that performs an effective decoding on the synchronous HMM. Remarkably, the multiplex Viterbi can be generalized to decode an ensemble of standard HMM sets, a problem often arising in the multi-model systems. Our experiments on the Aurora 2 database have showed the synchronous HMMs achieve the lowest WER of 6.27%, 17% relative reduction over the baseline HMMs. By jointly applying the speech scenes decision tree, the multiplex Viterbi, and the speech scene pruning, the decoding time of the 18-scene synchronous models is reduced to 2.0 times the HMM baseline decoding time, saving the computational cost with a factor of 9 compared with the simple multi-model approach. One advantage of the synchronous HMMs is that it can be readily combined with other techniques used in state-of-the-art speech recognition systems. One example is to combine the synchronous HMMs with the AFE, which has been shown to give a significant performance improvement over the corresponding HMM baseline.

CHAPTER 9

CONCLUSION

This dissertation addresses the robustness issue for automatic speech recognition from two directions.

The first part of the dissertation investigates noise estimation for nonlinear compensation models. We propose the Gauss-Newton method as a unified approach to optimize various nonlinear noise compensation models. Specifically, we present a novel noise variance estimation algorithm that conforms to the Gauss-Newton principle. The formulation of the Gauss-Newton method reduces the noise estimation problems to the determination of the Jacobians of the corrupted speech parameters with respect to the clean speech and noise parameters, which turns out to be the expectation of the sample Jacobians of the mismatch function. We present two methods, SJA and XCOV, to evaluate such Jacobians for the sampling-based compensation.

Moreover, we show that the EM-FA method, another popular noise estimation method, is an instance of the gradient-based method. Therefore, both the EM-FA method and the Gauss-Newton method belong to the family of gradient-based methods and possess a pervasive correspondence in the estimation of different model parameters for various compensation models. A detailed comparison between the Gauss-Newton and the em-FA methods from a general optimization prospective is also presented. The major advantages of the Gauss-Newton method consist of achieving the super-linear convergence rate and saving the cumbersome computation of the second-order derivatives. In contrast, the EM-FA method, as fully derived in an EM framework, inherits many properties from EM, such as linear and stable convergence, relatively simple maximization step, and embedding the probabilistic constraints.

The noise estimation algorithms for various compensation models have first been evaluated on two tasks. The first is to fit a GMM model to artificially corrupted samples,

and the second is to perform speech recognition on the Aurora 2 database. Experimental results verified that the Gauss-Newton method is effective in optimizing various nonlinear noise compensation models. Also, the sampling-based compensations produce more accurate noise estimate in the GMM fitting, but do not yield the expected gain in WER over the VTS model in speech recognition. Moreover, both Gauss-Newton and EM-FA methods, in the long run, can achieve a similar recognition performance. However, the Gauss-Newton method is superior to the EM-FA method in terms of the convergence property. In the practical experimental setups, this difference in convergence leads to the result that the Gauss-Newton method obtains 7%-12% relative reduction in WER over the EM-FA method for standard compensation and adaptive training.

We then present experimental results examining the integration of the VTS compensation and other robust speech recognition techniques on overlapping speech. The VTS compensation performs poorly on the overlapping speech, indicating that simply regarding the interfering speech as additive noise is fundamentally problematic. The multi-channel AEC significantly improves the results on the overlapping speech by making use of the reference signals of the interfering speech. Based on that, the VTS gives additional gains.

The second part of the dissertation investigates a novel acoustic modeling framework for modeling speech from heterogeneous sources. First, we propose the stranded HMM to explicitly model the relationship among the mixture components. In other words, each mixture component is assumed to depend on the previous mixture component in addition to the state that generates it. Our initial experiments on the Aurora 2 database have shown the significant gain with the standard HMM system, encouraging further investigation on more challenging tasks.

Second, we generalize the stranded HMM to the synchronous HMM to effectively capture the long-span temporal dependency of speech feature sequence. The synchronous HMM introduces an additional layer of substates between the HMM states and the Gaussian component variables. The speech scene decision tree is used to determine the optimal set of speech scenes and tie the substate parameters. Moreover, we propose a novel multiplex

Viterbi algorithm that performs an effective decoding on the synchronous HMM. Remarkably, the multiplex Viterbi can be generalized to decode an ensemble of standard HMM sets, a problem often arising in the multi-model systems. Our experiments on the Aurora 2 database have showed the synchronous HMMs achieve the lowest WER of 6.27%, 17% relative reduction over the baseline HMMs. By jointly applying the speech scenes decision tree, the multiplex Viterbi, and the speech scene pruning, the decoding time of the 18-scene synchronous models is reduced to 2.0 times the HMM baseline decoding time, saving the computational cost with a factor of 9 compared with the simple multi-model approach.

APPENDIX A

DERIVATIVE OF THE AUXILIARY FUNCTION WITH RESPECT TO NOISE VARIANCES

The derivative of the auxiliary function (3.7) with respect to the static noise variance is

$$\frac{\partial Q}{\partial \boldsymbol{\Sigma}_n} = -\frac{1}{2}\sum_t \sum_{j,k} \gamma_{jk}(t) \frac{\partial}{\partial \boldsymbol{\Sigma}_n}\Big[\log|\boldsymbol{\Sigma}_{y,jk}| + (\boldsymbol{y}_t - \boldsymbol{\mu}_{y,jk})^{\mathrm{T}} \boldsymbol{\Sigma}_{y,jk}^{-1} (\boldsymbol{y}_t - \boldsymbol{\mu}_{y,jk})\Big] \tag{A.1}$$

where $\boldsymbol{\Sigma}_{y,jk}$ is a function of $\boldsymbol{\Sigma}_n$ as (2.10). There are two terms being differentiated, the normalizing determinant in a form of $\frac{\partial}{\partial \boldsymbol{X}} \log|\boldsymbol{C} + \boldsymbol{AXB}|$, and the main probability as $\frac{\partial}{\partial \boldsymbol{X}} \boldsymbol{a}^{\mathrm{T}} (\boldsymbol{C} + \boldsymbol{AXB})^{-1} \boldsymbol{b}$, where we denote $\boldsymbol{\Sigma}_n$ by $\boldsymbol{X}$, $\boldsymbol{G}_x \boldsymbol{\Sigma}_{x,jk} \boldsymbol{G}_x^{\mathrm{T}}$ by $\boldsymbol{C}$, $\boldsymbol{G}_n$ by $\boldsymbol{A}$, $\boldsymbol{G}_n^{\mathrm{T}}$ by $\boldsymbol{B}$, and $(\boldsymbol{y}_t - \boldsymbol{\mu}_{y,jk})$ by $\boldsymbol{a}$ and $\boldsymbol{b}$, respectively. To determine the derivative of the normalizing determinant, we begin with the following equation [77]:

$$\frac{\partial}{\partial x} \log|\boldsymbol{Y}| = \mathrm{tr}\Big(\boldsymbol{Y}^{-1} \frac{\partial \boldsymbol{Y}}{\partial x}\Big) \tag{A.2}$$

where elements of the matrix $\boldsymbol{Y}$ are functions of a scalar x. Substituting $\boldsymbol{C} + \boldsymbol{AXB}$ for $\boldsymbol{Y}$ and applying (A.2) to each element x_{ij} of $\boldsymbol{X}$ gives

$$\begin{aligned}
\frac{\partial}{\partial x_{ij}} \log|\boldsymbol{C} + \boldsymbol{AXB}| &= \mathrm{tr}\Big((\boldsymbol{C} + \boldsymbol{AXB})^{-1} \frac{\partial (\boldsymbol{C} + \boldsymbol{AXB})}{\partial x_{ij}}\Big) \\
&= \mathrm{tr}\Big((\boldsymbol{C} + \boldsymbol{AXB})^{-1} \boldsymbol{A} \boldsymbol{E}^{ij} \boldsymbol{B}\Big) \\
&= \Big[\boldsymbol{A}^{\mathrm{T}} (\boldsymbol{C} + \boldsymbol{AXB})^{-\mathrm{T}} \boldsymbol{B}^{\mathrm{T}}\Big]_{ij}
\end{aligned} \tag{A.3}$$

where $\boldsymbol{E}^{ij}$ is a null matrix except for unity at row i and column j. Arraying all such results of (A.3) into a matrix yields

$$\frac{\partial}{\partial \boldsymbol{X}} \log|\boldsymbol{C} + \boldsymbol{AXB}| = \boldsymbol{A}^{\mathrm{T}} (\boldsymbol{C} + \boldsymbol{AXB})^{-\mathrm{T}} \boldsymbol{B}^{\mathrm{T}}. \tag{A.4}$$

For the derivative of the main probability term, we make use of the identity [77]

$$\frac{\partial}{\partial x} \boldsymbol{Y}^{-1} = -\boldsymbol{Y}^{-1} \frac{\partial \boldsymbol{Y}}{\partial x} \boldsymbol{Y}^{-1}. \tag{A.5}$$

Similarly, we have

$$\begin{aligned}
&\frac{\partial}{\partial x_{ij}}\boldsymbol{a}^{\mathrm{T}}(\boldsymbol{C}+\boldsymbol{AXB})^{-1}\boldsymbol{b} \\
&\quad=-\boldsymbol{a}^{\mathrm{T}}(\boldsymbol{C}+\boldsymbol{AXB})^{-1}\frac{\partial(\boldsymbol{C}+\boldsymbol{AXB})}{\partial x_{ij}}(\boldsymbol{C}+\boldsymbol{AXB})^{-1}\boldsymbol{b} \\
&\quad=-\boldsymbol{a}^{\mathrm{T}}(\boldsymbol{C}+\boldsymbol{AXB})^{-1}\boldsymbol{A}\boldsymbol{E}^{ij}\boldsymbol{B}(\boldsymbol{C}+\boldsymbol{AXB})^{-1}\boldsymbol{b} \\
&\quad=-\left[\boldsymbol{A}^{\mathrm{T}}(\boldsymbol{C}+\boldsymbol{AXB})^{-\mathrm{T}}\boldsymbol{a}\boldsymbol{b}^{\mathrm{T}}(\boldsymbol{C}+\boldsymbol{AXB})^{-\mathrm{T}}\boldsymbol{B}^{\mathrm{T}}\right]_{ij}
\end{aligned} \tag{A.6}$$

and

$$\begin{aligned}
&\frac{\partial}{\partial \boldsymbol{X}}\boldsymbol{a}^{\mathrm{T}}(\boldsymbol{C}+\boldsymbol{AXB})^{-1}\boldsymbol{b}= \\
&\qquad-\boldsymbol{A}^{\mathrm{T}}(\boldsymbol{C}+\boldsymbol{AXB})^{-\mathrm{T}}\boldsymbol{a}\boldsymbol{b}^{\mathrm{T}}(\boldsymbol{C}+\boldsymbol{AXB})^{-\mathrm{T}}\boldsymbol{B}^{\mathrm{T}}.
\end{aligned} \tag{A.7}$$

Hence, applying (A.4) and (A.7) to the corresponding terms in (A.1), the derivative of the auxiliary function with respect to the static noise variance resolves to

$$\frac{\partial Q}{\partial \boldsymbol{\Sigma}_n}=\frac{1}{2}\sum_{j,k}\boldsymbol{G}_{n,jk}^{\mathrm{T}}\boldsymbol{\Sigma}_{y,jk}^{-1}(\boldsymbol{S}_{y,jk}-\gamma_{jk}\boldsymbol{\Sigma}_{y,jk})\boldsymbol{\Sigma}_{y,jk}^{-1}\boldsymbol{G}_{n,jk}. \tag{A.8}$$

REFERENCES

[1] *Speech processing, transmission and quality aspects (STQ); Distributed speech recognition; Advanced front-end feature extraction algorithm; Compression algorithms*, Std. ETSI ES 202 050, Rev. 1.1.1, Oct. 2002.

[2] A. Acero, "Acoustical and environmental robustness in automatic speech recognition," Ph.D. dissertation, Carnegie Mellon Univ., Pittsburgh, PA, 1990.

[3] A. Acero, L. Deng, T. Kristjansson, and J. Zhang, "HMM adaptation using vector Taylor series for noisy speech recognition," in *Proc. ICSLP*, Beijing, China, 2000, pp. 869–872.

[4] M. Afify, "Accurate compensation in the log-spectral domain for noisy speech recognition," *IEEE Trans. Speech Audio Process.*, vol. 13, no. 3, pp. 388–398, May 2005.

[5] M. Akbacak and J. H. L. Hansen, "Environmental sniffing: Noise knowledge estimation for robust speech systems," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 15, no. 2, pp. 465–477, Feb. 2007.

[6] T. Anastasakos, J. McDonough, R. Schwartz, and J. Makhoul, "A compact model for speaker-adaptive training," in *Proc. ICSLP*, Philadelphia, PA, 1996, pp. 1137–1140.

[7] B. Atal, "Effectiveness of linear prediction characteristics of the speech wave for automatic speaker identification and verification," *J. Acoust. Soc. Amer.*, vol. 55, no. 6, pp. 1304–1312, Jun. 1974.

[8] S. Axelrod, R. Gopinath, and P. Olsen, "Modeling with a subspace constraint on inverse covariance matrices," in *Proc. ICSLP*, Denver, CO, 2002, pp. 2177–2180.

[9] L. R. Bahl, P. F. Brown, P. V. De Souza, and R. L. Mercer, "Maximum mutual information estimation of hidden Markov model parameters for speech recognition," in *Proc. ICASSP*, Tokyo, Japan, 1986, pp. 49–52.

[10] J. A. Bilmes, "Graphical models and automatic speech recognition," in *Mathematical Foundations of Speech and Language Processing*, M. Johnson, S. P. Khudanpur, M. Ostendorf, and R. Rosenfeld, Eds. New York: Springer-Verlag, 2003.

[11] J. R. Blum, "Multidimensional stochastic approximation methods," *Ann. Math. Stat.*, vol. 25, no. 4, pp. 737–744, 1954.

[12] J. J. Bussgang, "Crosscorrelation functions of amplitude-distorted Gaussian signals," Mas. Inst. Technol., Cambridge, MA, Tech. Rep. 216, Mar. 1952.

[13] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. R. Statist. Soc. B*, vol. 39, no. 1, pp. 1–38, 1977.

[14] L. Deng, J. Droppo, and A. Acero, "Enhancement of log mel power spectra of speech using a phase-sensitive model of the acoustic environment and sequential estimation of the corrupting noise," *IEEE Trans. Speech Audio Process.*, vol. 12, no. 2, pp. 133–143, Mar. 2004.

[15] V. V. Digalakis, D. Rtischev, and L. G. Neumeyer, "Speaker adaptation using constrained estimation of Gaussian mixtures," *IEEE Trans. Speech Audio Process.*, vol. 3, no. 5, pp. 357–366, Sep. 1995.

[16] J. Droppo, A. Acero, and L. Deng, "Uncertainty decoding with SPLICE for noise robust speech recognition," in *Proc. ICASSP*, Orlando, FL, 2002, pp. 57–60.

[17] J. Du and Q. Huo, "A feature compensation approach using high-order vector Taylor series approximation of an explicit distortion model for noisy speech recognition," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 19, no. 8, pp. 2285–2293, Nov. 2011.

[18] E. Eide and H. Gish, "A parametric approach to vocal tract length normalization," in *Proc. ICASSP*, 1996, pp. 346–348.

[19] F. Faubel, J. McDonough, and D. Klakow, "On expectation maximization based channel and noise estimation beyond the vector Taylor series expansion," in *Proc. ICASSP*, Dallas, TX, 2010, pp. 4294–4297.

[20] J. G. Fiscus, "A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (ROVER)," in *Proc. ASRU*, 1997, pp. 347–354.

[21] W. M. Fisher, G. R. Doddington, and K. M. Goudie-Marshall, "The DARPA speech recognition research database: specifications and status," in *Proc. DARPA Workshop on Speech Recognition*, 1986, pp. 93–99.

[22] F. Flego and M. J. F. Gales, "Discriminative adaptive training with VTS and JUD," in *Proc. ASRU*, Merano, Italy, 2009, pp. 170–175.

[23] ——, "Incremental predictive and adaptive noise compensation," in *Proc. ICASSP*, Taipei, Taiwan, 2009, pp. 3837–3840.

[24] M. J. F. Gales, "Model-based techniques for noise robust speech recognition," Ph.D. dissertation, Univ. of Cambridge, Cambridge, U.K., 1995.

[25] ——, "Maximum likelihood linear transformations for HMM-based speech recognition," *Comput. Speech Lang.*, vol. 12, no. 2, pp. 75–98, Jan. 1998.

[26] ——, "Semi-tied covariance matrices for hidden Markov models," *IEEE Trans. Speech Audio Process.*, vol. 7, no. 3, pp. 272–281, May 1999.

[27] ——, "Cluster adaptive training of hidden Markov models," *IEEE Trans. Speech Audio Process.*, vol. 8, no. 4, pp. 417–428, Jul. 2000.

[28] ——, "Model-based approaches to handling uncertainty," in *Robust Speech Recognition of Uncertain or Missing Data*, D. Kolossa and R. Haeb-Umbach, Eds. New York: Springer-Verlag, 2011.

[29] M. J. F. Gales and S. J. Young, "Robust speech recognition in additive and convolutional noise using parallel model combination," *Comput. Speech Lang.*, vol. 9, pp. 289–307, 1995.

[30] ——, "Robust continuous speech recognition using parallel model combination," *IEEE Trans. Speech Audio Process.*, vol. 4, no. 5, pp. 352–359, Sep. 1996.

[31] Z. Ghahramani and M. I. Jordan, "Factorial hidden Markov models," *Machine Learning*, vol. 29, no. 2, pp. 245–273, 1997.

[32] Y. Gong, "Stochastic trajectory modeling and sentence searching for continuous speech recognition," *IEEE Trans. Speech Audio Process.*, vol. 5, no. 1, pp. 33–44, 1997.

[33] R. A. Gopinath, M. J. F. Gales, P. S. Gopalakrishnan, S. Balakrishnan-Aiyer, and M. A. Picheny, "Robust speech recognition in noise – performance of the IBM continuous speech recogniser on the ARPA noise spoke task," in *Proc. ARPA Workshop on Spoken Lang. Syst. Technol.*, 1995, pp. 127–130.

[34] H. G. Hirsch and D. Pearce, "The Aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions," in *Proc. ISCA ITRW ASR*, Paris, France, 2000, pp. 181–188.

[35] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 1990.

[36] ——, *Topics in Matrix Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 1994.

[37] Y. Hu and Q. Huo, "An HMM compensation approach using unscented transformation for noisy speech recognition," in *Proc. ISCSLP*, Kent Ridge, Singapore, 2006, pp. 346–357.

[38] ——, "Irrelevant variability normalization based HMM training using VTS approximation of an explicit model of environmental distortions," in *Proc. Interspeech*, Antwerp, Belgium, 2007, pp. 1042–1045.

[39] X. D. Huang and M. A. Jack, "Semi-continuous hidden Markov models for speech signals," *Comput. Speech Lang.*, vol. 3, no. 3, pp. 239–251, 1989.

[40] H. Jiang and Q. Wang, "Nonlinear noise compensation in feature domain for speech recognition with numerical methods," in *Proc. ICASSP*, Montreal, Canada, 2004, pp. 985–988.

[41] B. H. Juang, W. Chou, and C. H. Lee, "Minimum classification error rate methods for speech recognition," *IEEE Trans. Speech Audio Process.*, vol. 5, no. 3, pp. 257–265, May 1997.

[42] S. J. Julier and J. K. Uhlmann, "Unscented filtering and nonlinear estimation," *Proc. IEEE*, vol. 92, no. 3, pp. 401–422, Mar. 2005.

[43] O. Kalinli, M. Seltzer, J. Droppo, and A. Acero, "Noise adaptive training for robust automatic speech recognition," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 18, no. 8, pp. 1889–1901, Nov. 2010.

[44] P. Kenny, M. Lennig, and P. Mermelstein, "A linear predictive HMM for vector-valued observations with applications to speech recognition," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 38, no. 2, pp. 220–225, Feb. 1990.

[45] D. Y. Kim, C. K. Un, and N. S. Kim, "Speech recognition in noisy environments using first order vector Taylor series," *Speech Commun.*, vol. 24, pp. 39–49, 1998.

[46] T. Kobayashi, T. Masuko, and K. Tokuda, "HMM compensation for noisy speech recognition based on cepstral parameter generation," in *Proc. Eurospeech*, Rhodes, Greece, 1997, pp. 1583–1586.

[47] F. Korkmazskiy, B. H. Juang, and F. K. Soong, "Generalized mixture of HMMs for continuous speech recognition," in *Proc. ICASSP*, Munich, Germany, 1997, pp. 1443–1446.

[48] R. Kuhn, J. C. Junqua, P. Nguyen, and N. Niedzielski, "Rapid speaker adaptation in eigenvoice space," *IEEE Trans. Speech Audio Process.*, vol. 8, no. 6, pp. 695–707, Nov. 2000.

[49] S. Kullback and R. A. Leibler, "On information and sufficiency," *Ann. Math. Stat.*, vol. 21, no. 1, pp. 79–86, Mar. 1951.

[50] K. F. Lee and H. W. Hon, "Speaker-independent phone recognition using hidden Markov models," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 37, no. 11, pp. 1641–1648, 1989.

[51] L. Lee and R. Rose, "Speaker normalization using efficient frequency warping procedures," in *Proc. ICASSP*, 1996, pp. 353–356.

[52] C. J. Leggetter and P. C. Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models," *Comput. Speech Lang.*, vol. 9, pp. 171–186, 1995.

[53] J. Li, L. Deng, D. Yu, Y. Gong, and A. Acero, "High-performance HMM adaptation with joint compensation of additive and convolutive distortions via vector Taylor series," in *Proc. ASRU*, Kyoto, Japan, 2007, pp. 65–70.

[54] ——, "A unified framework of HMM adaptation with joint compensation of additive and convolutive distortions," *Comput. Speech Lang.*, vol. 23, pp. 389–405, 2009.

[55] J. Li, D. Yu, Y. Gong, and L. Deng, "Unscented transform with online distortion estimation for HMM adaptation," in *Proc. Interspeech*, Makuhari, Japan, 2010, pp. 1660–1663.

[56] X. Li and R. M. Stern, "Training of stream weights for the decoding of speech using parallel feature streams," in *Proc. ICASSP*, 2003, pp. 832–835.

[57] H. Liao, "Uncertainty decoding for noise robust speech recognition," Ph.D. dissertation, Univ. of Cambridge, Cambridge, U.K., 2007.

[58] J. Lu, J. Ming, and R. Woods, "Adapting noisy speech modelsExtended uncertainty decoding," in *Proc. ICASSP*, Dallas, TX, 2010, pp. 4322–4325.

[59] L. Ma, D. J. Smith, and B. P. Milner, "Context awareness using environmental noise classification," in *Proc. Interspeech*, Geneva, Switzerland, 2003, pp. 2237–2240.

[60] D. Macho, L. Mauuary, B. Noé, Y. M. Cheng, D. Ealey, D. Jouvet, H. Kelleher, D. Pearce, and F. Saadoun, "Evaluation of a noise-robust DSR front-end on Aurora databases," in *Proc. ICSLP*, Denver, CO, 2002, pp. 17–20.

[61] G. J. McLachlan and T. Krishnan, *The EM Algorithm and Extensions.* New York: Wiley, 1997.

[62] D. Moore and I. McCowan, *The Multichannel Overlapping Numbers Corpus (MONC)*, http://www.cslu.ogi.edu/corpora.

[63] P. J. Moreno, "Speech recognition in noisy environments," Ph.D. dissertation, Carnegie Mellon Univ., Pittsburgh, PA, 1996.

[64] K. P. Murphy, "Dynamic Bayesian networks: representation, inference and learning," Ph.D. dissertation, Univ. Calif. Berkeley, 2002.

[65] H. Ney and S. Ortmanns, "Dynamic programming search for continuous speech recognition," *IEEE Signal Processing Magazine*, vol. 16, no. 5, pp. 64–83, Sep. 1999.

[66] J. Nocedal and S. J. Wright, *Numerical Optimization.* New York: Springer-Verlag, 1999.

[67] P. A. Olsen and R. A. Gopinath, "Modeling inverse covariance matrices by basis expansion," *IEEE Trans. Speech Audio Process.*, vol. 12, no. 1, pp. 37–46, Jan. 2004.

[68] J. P. Openshaw and J. S. Masan, "On the limitations of cepstral features in noise," in *Proc. ICASSP*, Adelaide, Australia, 1994, pp. 49–52.

[69] D. Povey, "Discriminative training for large vocabulary speech recognition," Ph.D. dissertation, Univ. of Cambridge, Cambridge, U.K., 2004.

[70] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, Feb. 1989.

[71] M. G. Rahim and B. H. Juang, "Signal bias removal by maximum likelihood estimation for robust telephone speech recognition," *IEEE Trans. Speech Audio Process.*, vol. 4, no. 1, pp. 19–30, Jan. 1996.

[72] E. Robledo-Arnuncio and B. H. Juang, "Blind source separation of acoustic mixtures with distributed microphones," in *Proc. ICASSP*, Honolulu, HI, 2007, pp. 949–952.

[73] R. C. Rose, E. M. Hofstetter, and D. A. Reynolds, "Integrated models of signal and background with application to speaker identification in noise," *IEEE Trans. Speech Audio Process.*, vol. 2, no. 2, pp. 245–257, Apr. 1994.

[74] D. B. Rubin and D. T. Thayer, "EM algorithms for ML factor analysis," *Psychometrika*, vol. 47, no. 1, pp. 69–76, 1982.

[75] S. Sagayama, Y. Yamaguchi, S. Takahashi, and J. Takahashi, "Jacobian approach to fast acoustic model adaptation," in *Proc. ICASSP*, Munich, Germany, 1997, pp. 835–838.

[76] L. K. Saul and M. G. Rahim, "Maximum likelihood and minimum classification error factor analysis for automatic speech recognition," *IEEE Trans. Speech Audio Process.*, vol. 8, no. 2, pp. 115–125, Mar. 2000.

[77] S. R. Searle, *Matrix Algebra Useful for Statistics*. New York: Wiley, 1982.

[78] G. Shi, Y. Shi, and Q. Huo, "A study of irrelevant variability normalization based training and unsupervised online adaptation for LVCSR," in *Proc. Interpeech*, Makuhari, Japan, 2010, pp. 1357–1360.

[79] Y. Shinohara and M. Akamine, "Bayesian feature enhancement using a mixture of unscented transformation for uncertainty decoding of noisy speech," in *Proc. ICASSP*, Taipei, Taiwan, 2009, pp. 4569–4572.

[80] E. Shriberg, A. Stolcke, and D. Baron, "Observations on overlap: Findings and implications for automatic processing of multi-party conversation," in *Proc. Interpeech*, Aalborg, Denmark, 2001, pp. 1359–1362.

[81] R. Singh, M. L. Seltzer, B. Raj, and R. M. Stern, "Speech in noisy environments: robust automatic segmentation, feature extraction, and hypothesis combination," in *Proc. ICASSP*, 2001, pp. 273–276.

[82] O. Siohan, B. Ramabhadran, and B. Kingsbury, "Contructing ensembles of ASR systems using randomized decision trees," in *Proc. ICASSP*, 2005, pp. 197–200.

[83] V. Stouten, H. Van hamme, and P. Wambacq, "Effect of phase-sensitive environment model and higher order VTS on noisy speech feature enhancement," in *Proc. ICASSP*, Philadelphia, PA, 2005, pp. 433–436.

[84] ——, "Model-based feature enhancement with uncertainty decoding for noise robust ASR," *Speech Commun.*, vol. 48, pp. 1502–1514, 2006.

[85] J. Su, H. Li, J. P. Haton, and K. T. Ng, "Speaker time-drifting adaptation using trajectory mixture hidden Markov models," in *Proc. ICASSP*, Tokyo, Japan, 1996, pp. 709–712.

[86] S. Tibrewala and H. Hermansky, "Sub-band based recognition of noisy speech," in *Proc. ICASSP*, 1997, pp. 1255–1258.

[87] Y. Tsao and C. H. Lee, "An ensemble speaker and speaking environment modeling approach to robust speech recognition," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 17, no. 5, pp. 1025–1037, Jul. 2009.

[88] R. C. van Dalen and M. J. F. Gales, "Extended VTS for noise-robust speech recognition," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 19, no. 4, pp. 733–743, May 2011.

[89] V. Vanhoucke and A. Sankar, "Mixtures of inverse covariances," *IEEE Trans. Speech Audio Process.*, vol. 12, no. 3, pp. 250–264, May 2004.

[90] O. Viikki and K. Laurila, "Cepstral domain segmental feature vector normalization for noise robust speech recognition," *Speech Commun.*, vol. 25, no. 1, pp. 133–147, 1998.

[91] T. S. Wada, E. Robledo-Arununcio, G. Yue, and B. H. Juang, "Immersive acoustic signal processing for intelligent collaboration," in *Proc. WESPAC*, Seoul, Korea, 2006.

[92] L. Wang and P. Woodland, "Discriminative adaptive training using the MPE criterion," in *Proc. ASRU*, St. Thomas, US Virgin Islands, 2003, pp. 279–284.

[93] C. J. Wellekens, "Explicit time correlation in hidden Markov models for speech recognition," in *Proc. ICASSP*, Dallas, TX, 1987, pp. 384–386.

[94] P. C. Woodland, J. J. Odell, V. Valtchev, and S. J. Young, "Large vocabulary continuous speech recognition using HTK," in *Proc. ICASSP*, Adelaide, Australia, 1994, pp. 125–128.

[95] J. Wung, T. S. Wada, B. H. Juang, B. Lee, M. Trott, and R. W. Schafer, "A system approach to acoustic echo cancellation in robust hands-free teleconferencing," in *Proc. IEEE WASPAA*, New Paltz, NY, 2011, pp. 101–104.

[96] H. Xu and K. K. Chin, "Comparison of estimation techniques in joint uncertainty decoding for noise robust speech recognition," in *Proc. Interspeech*, Brighton, U.K., 2009, pp. 2403–2406.

[97] ——, "Joint uncertainty decoding with the second order approximation for noise robust speech recognition," in *Proc. ICASSP*, Taipei, Taiwan, 2009, pp. 3841–3844.

[98] H. Xu, P. Dalsgaard, Z. H. Tan, and B. Lindberg, "Noise condition-dependent training based on noise classification and SNR estimation," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 15, no. 8, pp. 2431–2443, Nov. 2007.

[99] H. Xu, L. Rigazio, and D. Kryze, "Vector Taylor series based joint uncertainty decoding," in *Proc. Interspeech*, Pittsburgh, PA, 2006, pp. 1125–1128.

[100] J. Xu, Y. Zhang, Z. J. Yan, and Q. Huo, "An i-vector based approach to acoustic sniffing for irrelevant variability normalization based acoustic model training and speech recognition," in *Proc. Interspeech*, Florence, Italy, 2011, pp. 1701–1704.

[101] J. Xue and Y. Zhao, "Random forests of phonetic decision trees for acoustic modeling in conversational speech recognition," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 16, no. 3, pp. 519–528, Mar. 2008.

[102] C. Yang, F. K. Soong, and T. Lee, "Static and dynamic spectral features: Their noise robustness and optimal weights for ASR," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 15, no. 3, pp. 1087–1097, Mar. 2007.

[103] S. Young, G. Evermann, M. J. F. Gales, T. Hain, D. Kershaw, X. Liu, G. Moore, J. J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. C. Woodland, *The HTK Book (for HTK Version 3.4)*. Cambridge, U.K.: Univ. of Cambridge, 2006.

[104] S. J. Young, J. J. Odell, and P. C. Woodland, "Tree-based state tying for high accuracy acoustic modelling," in *Proc. HLT*, 1994, pp. 307–312.

[105] Y. Zhao and B. H. Juang, "On noise estimation for robust speech recognition using vector Taylor series," in *Proc. ICASSP*, Dallas, TX, 2010, pp. 4290–4293.

[106] ——, "Non-linear noise compensation for robust speech recognition using Gauss-Newton method," in *Proc. ICASSP*, Praque, Czech Republic, 2011, pp. 4796–4799.

[107] ——, "Nonlinear compensation using the Gauss-Newton method for noise-robust speech recognition," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 20, no. 8, pp. 2191–2206, Oct. 2012.

[108] ——, "Stranded Gaussian mixture hidden Markov models for robust speech recognition," in *Proc. ICASSP*, Kyoto, Japan, 2012, pp. 4301–4304.

[109] ——, "A comparative study of noise estimation algorithms for nonlinear compensation in robust speech recognition," *Speech Commun.*, submitted for publication.

[110] Y. Zhao, S. Shin, E. Robledo-Arnuncio, and B. H. Juang, "A study on recognizing distorted speech over local distributed transducer networks," in *Proc. ICASSP*, Taipei, Taiwan, 2009, pp. 4181–4184.

[111] A. Zolnay, R. Schlüter, and H. Ney, "Acoustic feature combination for robust speech recognition," in *Proc. ICASSP*, Philadelphia, PA, 2005, pp. 457–460.

VITA

Yong Zhao is currently pursuing a Ph.D. degree under the supervision of Prof. Biing-Hwang (Fred) Juang at the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA. He received the B.Eng. and M.Eng. degrees in electrical engineering from Tsinghua University, Beijing, China, in 1998 and 2001, respectively. From 2001 to 2007, he was an associate researcher at Speech Group, Microsoft Research Asia, Beijing, focusing on speech synthesis and analysis. His current research interests include noise robust speech recognition, acoustic modeling, and machine learning. The publications during his Ph.D. study in Georgia Institute of Technology are listed in the following.

[1] Y. Zhao and B. H. Juang, "Nonlinear compensation using the Gauss-Newton method for noise-robust speech recognition," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 20, no. 8, pp. 2191–2206, Oct. 2012.

[2] Y. Zhao and B. H. Juang, "A comparative study of noise estimation algorithms for nonlinear compensation in robust speech recognition," *Speech Commun.*, submitted for publication.

[3] Q. Fu, Y. Zhao, and B. H. Juang, "Automatic speech recognition based on non-uniform error criteria," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 20, no. 3, pp. 780–793, Mar. 2012.

[4] Y. Zhao and B. H. Juang, "Stranded Gaussian mixture hidden Markov models for robust speech recognition," in *Proc. ICASSP*, Kyoto, Japan, 2012, pp. 4301–4304.

[5] Y. Zhao, A. Ljolje, D. Caseiro, and B. H. Juang, "A general discriminative training algorithm for speech recognition using weighted finite-state transducers," in *Proc. ICASSP*, Kyoto, Japan, 2012, pp. 4217–4220.

[6] Y. Zhao and B. H. Juang, "Non-linear noise compensation for robust speech recognition using Gauss-Newton method," in *Proc. ICASSP*, Praque, Czech Republic, 2011, pp. 4796–4799.

[7] Y. Zhao and B. H. Juang, "A comparative study of noise estimation algorithms for VTS-based robust speech recognition," in *Proc. Interspeech*, Makuhari, Japan, 2010, pp. 2090–2093.

[8] Y. Zhao and B. H. Juang, "On noise estimation for robust speech recognition using vector Taylor series," in *Proc. ICASSP*, Dallas, TX, 2010, pp. 4290–4293.

[9] Y. Zhao, S. Shin, E. Robledo-Arnuncio, and B. H. Juang, "A study on recognizing distorted speech over local distributed transducer networks," in *Proc. ICASSP*, Taipei, Taiwan, 2009, pp. 4181–4184.

[10] Y. Zhao and X. He, "Using n-gram based features for machine translation system combination," in *Proc. NAACL-HLT*, Boulder, Colorado, 2009, pp. 205–208.

[11] Y. Zhao and X. He, "System combination for machine translation using n-gram posterior probabilities," in *NIPS 2008 WORKSHOP on Speech and Language: Learning-based Methods and Systems*, 2008.