

UNIVERSITY OF CALIFORNIA

Los Angeles

**Environmental and Speaker Robustness in
Automatic Speech Recognition with Limited
Learning Data**

A dissertation submitted in partial satisfaction

of the requirements for the degree

Doctor of Philosophy in Electrical Engineering

by

Xiaodong Cui

2005

© Copyright by
Xiaodong Cui
2005

The dissertation of Xiaodong Cui is approved.

Yingnian Wu

Mani B. Srivastava

Ingrid Verbauwhede

Abeer Alwan, Committee Chair

University of California, Los Angeles

2005

To my parents

TABLE OF CONTENTS

1	Introduction	1
1.1	Automatic Speech Recognition	1
1.2	Feature Extraction	3
1.3	Hidden Markov Models	5
1.3.1	Forward-Backward Algorithm - Solution to Prob. 1	7
1.3.2	Baum-Welch Algorithm - Solution to Prob. 2	8
1.3.3	Viterbi Algorithm - Solution to Prob. 3	10
1.4	Robustness in ASR	11
1.5	Existing Techniques in Robust ASR	15
1.5.1	Techniques for Environmental Robustness	16
1.5.2	Techniques for Speaker Robustness	19
1.6	Speech Databases	21
1.7	Organization of Dissertation	23
I	Environmental Robust Speech Recognition	24
2	Weighted Viterbi Decoding	26
2.1	System Implementation	26
2.2	Frame SNR Estimation	28
2.3	Experimental Results	31
2.3.1	Aurora 2 database	32
2.3.2	JEITA Database	36

2.4	Summary	38
3	Feature Compensation Based on Polynomial Regression of SNR	39
3.1	Polynomial Regression of Utterance SNR	40
3.1.1	Bias Approximation by SNR Polynomials	40
3.1.2	Feature Compensation	42
3.1.3	Maximum Likelihood Estimation of Regression Polynomials	43
3.2	Utterance SNR Estimation	47
3.3	Experimental Results	48
3.3.1	Experimental Conditions	48
3.3.2	Distribution of Utterance SNRs	50
3.3.3	Regression Polynomial Orders	50
3.3.4	Estimated Regression Polynomials	53
3.3.5	Recognition Results	57
3.4	Summary	60
II	Speaker Robust Speech Recognition	66
4	Adaptation Text Design Based on the Kullback-Leibler Measure	69
4.1	Kullback-Leibler (KL) Measure	70
4.2	Adaptation Text Design Algorithm	71
4.3	Experimental Results	74
5	Rapid Speaker Adaptation by Formant-like Peak Alignment	79
5.1	Relationship between frequency warping and linear transformations	82

5.1.1	Feature Schemes	82
5.1.2	Derivation of the transformation matrix \mathbf{A}	86
5.1.3	Discussion	90
5.1.4	Estimation of the bias vector \mathbf{b}	92
5.1.5	Variance Adaptation	93
5.2	Formant-like Peak Alignment	94
5.3	Experimental Results	97
5.4	Summary and Conclusions	103
6	Speaker Adaptation by Weighted Model Averaging Based on MDL	107
6.1	Structured Transformations	108
6.2	Description of Structured Transformation Based on MDL	112
6.3	Weighted Model Averaging	116
6.4	Choice of Structure Form	117
6.5	Experimental Results	118
6.6	Discussion	120
6.7	Summary	121
7	Summary and Future Work	128
7.1	Summary	128
7.2	Discussion and Future Work	130
	References	133

LIST OF FIGURES

1.1	Block diagram of a typical ASR system	2
1.2	Block diagram of MFCC feature computation.	4
1.3	A schematic representation of a hidden Markov model for the word “zero”.	6
1.4	The Viterbi decoding network (after [YEK01], Pp.10)	11
1.5	Performance of an ASR system on connected digits under subway noise at different SNRs. The system is trained with clean speech.	12
1.6	Performance of an ASR system on connected digits for different gender and age groups. The system is trained with adult male speech.	12
1.7	Normalized spectra of the sound /u:/ from a clean (solid line) and noisy (dotted line) speech signal.	13
1.8	Normalized average LPC spectrum of the sound /i:/ from the speech of a female adult (top panel) and a young girl (bottom panel)	14
1.9	Sampled population and target population.	15
1.10	Parallel Model Combination process (after [GY96])	18
1.11	Frequency warping for VTLN (after [YEK01], Pp.63)	20
2.1	Diagram of weighted Viterbi decoding (WVD).	27
2.2	Noise power(+) estimated by minimum statistics tracking from the noisy speech power spectrum (o) for the utterance “43o6571”. The utterance is labeled 15 dB SNR in the Aurora 2 database.	29

2.3	Estimated SNR (top panel), confidence factor (γ_t) (middle panel) and waveform (bottom panel) of the utterance “0021641” labeled in the Aurora 2 database as having a signal-to-noise ratio of 15 dB.	30
2.4	Estimated SNR (top panel), confidence factor (γ_t) (middle panel) and waveform (bottom panel) of the utterance “4” labeled in the Aurora 2 database as having a signal-to-noise ratio of 0 dB. . . .	31
2.5	γ_t (gamma) for utterance SNRs higher than 10 dB (left) and γ_t (gamma) for utterance SNRs lower than 10 dB (right).	32
3.1	Estimated frame-wise SNR (solid line in top panel), estimated utterance SNR (dotted line in top panel) and waveform (bottom panel) of the utterance “43o6571” labeled 15 dB in the Aurora 2 database.	47
3.2	Training and recognition scheme	49
3.3	Histograms of estimated utterance SNRs labeled as clean (left) and 20 dB SNR (right) data in Set A of Aurora 2 database.	51
3.4	Histograms of estimated utterance SNRs labeled as 15 dB (left) and 10 dB SNR (right) data in Set A of Aurora 2 database. . . .	51
3.5	Histograms of estimated utterance SNRs labeled as 5 dB (left) and 0 dB SNR (right) data in Set A of Aurora 2 database.	51
3.6	Histograms of estimated utterance SNRs in training (left) and testing (right) of the well-matched condition of the Aurora 3 German database.	52
3.7	Histograms of estimated utterance SNRs in training (left) and testing (right) of the medium-mismatched condition of the Aurora 3 German database.	52

3.8	Histograms of estimated utterance SNRs in training (left) and testing (right) of the high-mismatched condition of the Aurora 3 German database.	52
3.9	The left panel shows estimated global polynomials as a function of SNR and iteration number. The right panel shows average likelihood as a function of the number of EM iterations. Both panels use the energy feature component (E) for the airport noise data. .	55
3.10	The left panel shows estimated global polynomials as a function of SNR and iteration number. The right panel shows average likelihood as a function of the number of EM iterations. Both panels use the energy feature component (E) for the station noise data. .	55
3.11	The left and right panels show estimated global polynomials as a function of SNR and number of utterances for energy (E) and the first cepstral coefficient (C1), respectively, under car noise. The number of EM iterations is fixed at 6.	56
3.12	The left and right panels show estimated global polynomials as a function of SNR and number of utterances for the 6th (C6) and 10th (C10) cepstral coefficients, respectively, under subway noise. The number of EM iterations is fixed at 6.	56
4.1	Heuristic text selection algorithm based on the minimum KL measure	73
4.2	Two instances of phoneme distributions of 20 randomly selected sentences from TIMIT.	76
4.3	Two instances of phoneme distributions of 40 randomly selected sentences from TIMIT.	76

4.4	Comparison of the phoneme distributions of 10 sentences chosen randomly(left) and using the minimum KL measure(right) from TIMIT.	77
4.5	Comparison of the phoneme distributions of 40 sentences chosen randomly(left) and using the minimum KL measure(right) from TIMIT.	77
4.6	Comparison of the phoneme distributions of 100 sentences chosen randomly(left) and using the minimum KL measure(right) from TIMIT.	78
4.7	Comparison of the phoneme distributions of 40 sentences chosen with uniform prior(left) and non-uniform prior(right)from TIMIT.	78
5.1	Spectrograms of the digit “9” spoken by a male (left) and a boy (right).	80
5.2	Diagram of the three feature extraction schemes discussed. The feature CEP is computed with no Mel-scale warping. MFCC1 is computed with a Mel-scale warping function, and MFCC2 is computed with Mel-warped triangular filter banks.	83
5.3	Mel-scaled triangular filter bank.	85
5.4	Spectra for the steady part of the sound /uw/ in the digit “two” from an adult male (left) and a boy (right).	95
5.5	Formants estimated (white circles) using Gaussian mixtures for the sound /uw/ in digit “two” from an adult male speaker (left) and a child speaker (right).	96

5.6	Piece-wise linear function (solid line) which aligns the first and third formant-like peaks of the adult and child's speech in Fig. 5.5. The dotted line is the reference line for $y = x$	97
5.7	Original boy's spectrum (solid line) and the re-shaped adult male's spectrum (dotted line) in Fig. 5.5.	98
5.8	Adaptation Algorithm.	99
5.9	Performance of MLLR, VTLN and the peak alignment algorithm using $R(F_3)$ with different numbers of adaptation digits for MFCC2 features.	105
6.1	A comparison of the transformation tying patterns with a regression tree of six base classes using one-diagonal (grey node), and three-diagonal (black node) structures.	113
6.2	Transformation matrices generated based on vocal tract length normalization with scaling factor equal to 1.1 (left) and 0.9 (right). The darker the color, the more significant the element is.	122
6.3	Transformation matrices generated based on vocal tract length normalization with scaling factor equal to 1.2 (left) and 0.8 (right). The darker the color, the more significant the element is.	122
6.4	Flow chart of implementation of weighted model averaging with structured MLLR transformations. The structures are appropriately tied with a regression tree.	123

LIST OF TABLES

2.1	WVD performance (%) with MFCC features on the Aurora 2 database.	33
2.2	WVD performance (%) with LPCC features on the Aurora 2 database.	34
2.3	WVD performance (%) with PLP features on the Aurora 2 database.	34
2.4	Performance (%) of WVD vs. MLLR for eight types of noise from Set A (first four types) and Set B (second four types) of the Aurora 2 database.	35
2.5	10 Japanese digits used in the experiments	36
2.6	WVD performance (%) with MFCC features on the JEITA database.	37
2.7	WVD performance (%) with PLP features on the JEITA database.	37
2.8	WVD performance (%) with LPCC features on the JEITA database.	38
3.1	Average performances of Sets A and B in Aurora 2 with respect to regression polynomial orders. The polynomials are state tied and estimated from 300 utterances.	53
3.2	Word recognition accuracy averaged over all clean conditions in the Aurora 2 database. Feature compensation is performed with 10, 100 and 200 utterances.	54
3.3	Word recognition accuracy for FC and MLLR for 4 types of noise in Set A of the Aurora 2 database. MLLR1 refers to the case where the MLLR transformation matrices are estimated across all SNR levels, while MLLR2 refers to MLLR transformation matrices being SNR-cluster specific. Baseline MFCC results are presented as adaptation with 0 utterances.	61

3.4	Word recognition accuracy for FC and MLLR on 4 types of noise in Set B of the Aurora 2 database. See Table 3.3 caption for the definition of MLLR1 and MLLR2. Baseline MFCC results are presented as adaptation with 0 utterances.	62
3.5	Word recognition accuracy for FC and MLLR for each SNR level in Set A of the Aurora 2 database. See Table 3.3 caption for the definition of MLLR1 and MLLR2. Baseline MFCC results are presented as adaptation with 0 utterances.	63
3.6	Word recognition accuracy for FC and MLLR for each SNR level in Set B of the Aurora 2 database. See Table 3.3 caption for the definition of MLLR1 and MLLR2. Baseline MFCC results are presented as adaptation with 0 utterances.	64
3.7	Word recognition accuracy for FC and MLLR of static tying schemes under high-mismatched (HM), medium-mismatched (MM) and well-matched (WM) conditions of the German part of the Aurora 3 database. Baseline MFCC results are presented as adaptation with 0 utterances.	65
3.8	Word recognition accuracy for FC and MLLR of dynamic tying schemes under high-mismatched (HM), medium-mismatched (MM) and well-matched (WM) conditions of the German part of the Aurora 3 database. Baseline MFCC results are presented as adaptation with 0 utterances.	65
4.1	44 phonemes used in the adaptation text design.	74

5.1	Recognition accuracy of children’s speech with CEP features for (1) baseline, or no adaptation, (2) VTLN1 (speaker-dependent), (3) VTLN2 (utterance-dependent), and (4) three peak alignment schemes ($R(F_1, F_3)$, $R(F_3)$ and $R(\bar{F}_3)$) with and without a bias vector. The results in the parentheses are without a bias. The acoustic models are trained on adult male data and tested on children’s.	103
5.2	Recognition accuracy of children’s speech with MFCC1 features. See Table 5.1 caption for explanation of the test conditions. . . .	104
5.3	Recognition accuracy of children’s speech with MFCC2 features. See Table 5.1 caption for explanation of the test conditions. . . .	104
6.1	Number of parameters of MLLR transformation matrix under different structures.	118
6.2	Word error rate (%) of MLLR with different structured transformations on TIDIGITS database. Acoustic models are trained with male speech and tested on female speech. The performance is the average over the 10 female speakers in the test set. 3D, 7D, 3B and full denote 3-diagonal, 7-diagonal, 3-block and full transformation matrices, respectively.	124
6.3	Word error rate (%) of MLLR with different structured transformations on TIDIGITS database. Acoustic models are trained with female speech and tested on male speech. The performance is the average over the 10 male speakers in the test set. 3D, 7D, 3B and full denote 3-diagonal, 7-diagonal, 3-block and full transformation matrices, respectively.	125

6.4	Word error rate (%) of MLLR with different structured transformations on TIDIGITS database. Acoustic models are trained and tested on adult speech (both male and female). The performance is the average over the 20 adult speakers in the test set. 3D, 7D, 3B and full denote 3-diagonal, 7-diagonal, 3-block and full transformation matrices, respectively.	126
6.5	Word error rate (%) of MLLR with different structured transformations on TIDIGITS database. Acoustic models are trained on adult speech and tested on child speech. The performance is the average over the 10 kid speakers in the test set. 3D, 7D, 3B and full denote 3-diagonal, 7-diagonal, 3-block and full transformation matrices, respectively.	127

ACKNOWLEDGMENTS

I gratefully acknowledge my advisor Prof. Abeer Alwan for her intellectual guidance and encouragement throughout the course of my research. Her invaluable academic and personal support contribute greatly to this dissertation.

I am also indebted to Professor Yingnian Wu for his insightful suggestions to my research. Yingnian is an excellent teacher in statistics and good at explaining complex problems in a clear and intuitive way. Discussions with him at lunch directly lead to some ideas presented in this dissertation. Sincere appreciation also goes to Dr. Yifan Gong for his supervision during my two summer interns at Texas Instruments. His sound knowledge in speech recognition and nice personality not only made my internship a great pleasure but also had an impact on my research in noise robust speech recognition. I also extend my gratitude to Professors Mani Srivastava and Ingrid Verbauwhede who are on my doctoral committee and provided valuable feedback.

I am grateful to our SPAPL labmates, Markus, Panchi, Jintao, Qifeng, Hong, Jianxia, Dian and Jinjin, for their help and encouragement over the last few years. The pleasant experiences with them made my UCLA life most enjoyable. Finally, I would like to thank my family for their love and support.

This work was supported in part by the NSF.

VITA

1974	Born, Qingdao, P.R.China
1992 - 1996	B.S. Shanghai Jiao Tong University, Shanghai, P.R.China
1996 - 1999	M.S. Tsinghua University, Beijing, P.R.China
2000 - 2005	Research Assistant, University of California, Los Angeles, CA

PUBLICATIONS

X. Cui and A. Alwan. “Adaptation of children’s speech with limited data based on formant-like peak alignment”, *Computer Speech and Language*, to appear.

X. Cui and A. Alwan. “Noise robust speech recognition using feature compensation based on polynomial regression of utterance SNR”, *IEEE Trans. on Speech and Audio Processing*, to appear.

X. Cui and A. Alwan. “MLLR-like speaker adaptation based on linearization of VTLN with MFCC features”, *Interspeech*, 2005.

X. Cui and A. Alwan. “Combining feature compensation and weighted Viterbi decoding for noise robust speech recognition with limited adaptation data”, *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, Vol.1, Pp. 969-972, 2004.

X. Cui, A. Bernard and A. Alwan. “A noise-robust ASR back-end technique based on weighted Viterbi recognition”, *European Conf. on Speech Communication and Technology*, Pp.2169-2172, 2003.

X. Cui, M. Iseli, Q. Zhu and A. Alwan. “Evaluation of noise robust features on the Aurora databases”, *Int. Conf. on Spoken Language Processing*, Vol.1, Pp.481-484, 2002.

X. Cui and A. Alwan. “Efficient adaptation text design based on the Kullback-Leibler measure”, *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, Vol.1, Pp. 613-616, 2002.

Q. Zhu, M. Iseli, X. Cui and A. Alwan. “Noise robust feature extraction for ASR using the Aurora 2 database”, *Eurospeech*, Pp.185-188, 2001.

ABSTRACT OF THE DISSERTATION

**Environmental and Speaker Robustness in
Automatic Speech Recognition with Limited
Learning Data**

by

Xiaodong Cui

Doctor of Philosophy in Electrical Engineering

University of California, Los Angeles, 2005

Professor Abeer Alwan, Chair

This dissertation addresses environmental and speaker robust issues in automatic speech recognition with an emphasis on cases where only limited amounts of learning data are available.

The first part of the dissertation is concerned with environmental robustness and consists of two chapters. First, a weighted Viterbi decoding algorithm is discussed where feature observation probabilities are weighted in the Viterbi decoder by a confidence factor which is a function of frame SNR. Second, a feature compensation algorithm based on polynomial regression of SNR is presented. The algorithm approximates the nonlinear bias between noisy and clean speech features by a polynomial of SNR. In the recognition stage, utterance SNR is evaluated from the speech signal and noisy features are compensated accordingly using the regression polynomials which could be tied at various levels of granularity.

The second part of the dissertation is devoted to speaker robustness and contains three chapters. First, an adaptation text design algorithm based on the Kullback-Leibler (KL) measure is introduced. It allows a designer to predefine a target distribution of speech units and selects texts whose speech unit distribu-

tion minimizes the KL measure. Second, a rapid speaker adaptation algorithm by formant-like peak alignment is presented. The algorithm investigates, in the discrete frequency domain, the relationship between frequency warping in the front-end feature domain and linearity of the corresponding transformation in the back-end model domain. Adaptation is conducted by performing the transformation of means deterministically, based on the linear relationship investigated, and estimating biases and variances statistically based on the Expectation-Maximization algorithm. Third, a robust maximum likelihood linear regression technique via weighted model averaging is discussed. A variety of transformation structures is studied and a general form of maximum likelihood estimation of the structures is given. The minimum description length (MDL) principle is applied to account for the compromise between transformation granularity and descriptive ability regarding the tying patterns of structured transformations with a regression tree. Weighted model averaging across the candidate structures is then performed based on the normalized MDL scores.

CHAPTER 1

Introduction

1.1 Automatic Speech Recognition

The past fifty years have witnessed fast progress in automatic speech recognition (ASR) by computer. From the earliest attempts to recognize small vocabulary isolated words to the latest large-vocabulary conversational speech understanding, ASR has become more and more powerful and is impacting our lives in many ways.

The progress in ASR has been a joint effort between several disciplines including acoustics, phonetics, statistics and language modeling, just to name a few. Techniques have been enriched to enable ASR systems to handle more and more sophisticated tasks. The evolution of approaches over the last a few decades to improve ASR is highlighted by several milestones that are worth mentioning.

In its early stage, speech recognition was conducted using the fundamental ideas of acoustic-phonetics which led to several simple systems in the 1950s [DBB52, FF59]. In the 1970s, linear predictive coding (LPC) and dynamic time warping (DTW) were the two major contributions in the speech community which pushed the recognition performance to a new level. Until that time, ASR was based on template-matching strategies and the main research focus was on isolated or connected word recognition. The template-matching approach has its limitations in handling continuous speech. This problem persisted until the 1980s

when the hidden Markov model (HMM) was introduced by several major research institutes (most notably IBM and Bell Labs) to the ASR field with great success. The shift from template-matching to HMM statistical modeling brought in a framework that allowed researchers to incorporate acoustic and language knowledge together in a natural manner. Consequently, large-vocabulary continuous speech recognition became a reality. HMMs quickly became the dominant approach in ASR and nowadays virtually all ASR systems are based on HMMs.

A typical HMM-based ASR system is shown in Fig. 1.1.

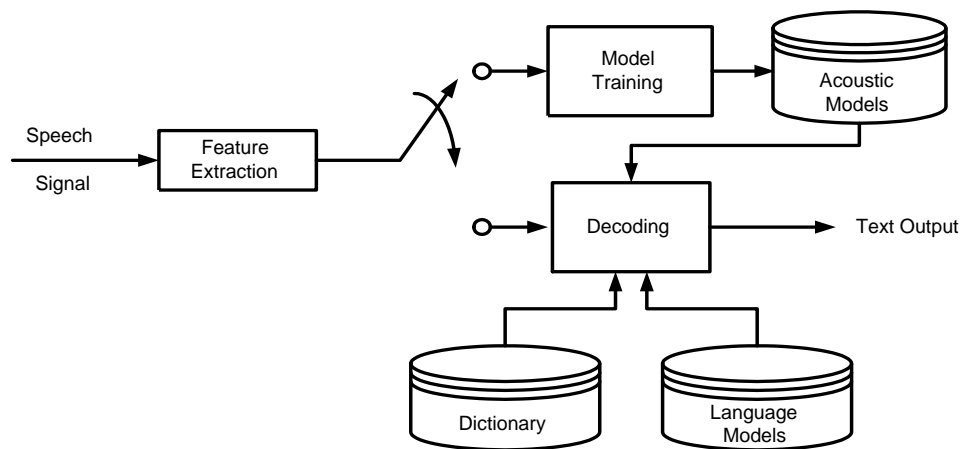


Figure 1.1: Block diagram of a typical ASR system

Speech signals are first converted into features that possess discriminative acoustic information. A training process is required by the system to learn the acoustic characteristics of sounds from the training data. This process results in a set of acoustic models which, in combination with a dictionary and a language model, are used to transcribe the input speech signals into text in the recognition stage.

HMM-based ASR systems deal with speech signals from a probabilistic perspective. Suppose we observed a sequence of speech features $O = \{o_1, o_2, \dots, o_T\}$, and consider all possible word sequences $W = \{w_1, w_2, \dots\}$, the system attempts

to transcribe the utterance using Bayesian rule:

$$W^* = \operatorname{argmax}_W P(W|O) \quad (1.1)$$

which can be rewritten as

$$W^* = \operatorname{argmax}_W P(W|O) \quad (1.2)$$

$$= \operatorname{argmax}_W \frac{P(O|W)P(W)}{P(O)} \quad (1.3)$$

$$= \operatorname{argmax}_W P(O|W)P(W) \quad (1.4)$$

In Eq. 1.4, $P(O)$ is ignored since it is equal for all word sequences. Therefore, the posterior probability in Eq. 1.2 is decoupled into two terms in Eq. 1.4: namely $P(O|W)$ and $P(W)$. $P(O|W)$ is the probability of the speech signal O given the word sequence W and is referred to as the acoustic model; $P(W)$ is the probability of the word sequence W and is referred to as the language model. Both probabilities can be estimated from the training data and are combined together to perform recognition. In later sections, a brief introduction to HMMs is provided.

1.2 Feature Extraction

Since speech signals are nonstationary, speech recognition systems process the signals frame by frame where a frame is about 20ms long within which statistical stationarity is assumed. This is realized in the feature extraction module of Fig. 1.1 which transforms the input speech signals into a sequence of feature vectors. Common speech features include linear predictive coefficients (LPC) [MG76], linear predictive cepstral coefficients (LPCC), perceptual linear prediction (PLP) [Her90] and Mel-frequency cepstral coefficients (MFCC) [DM80]. MFCC is chosen as the baseline feature in most of the experiments conducted in this dissertation considering its good and noise robust performance compared to other features.

Fig. 1.2 shows a diagram of computing MFCC features. Each frame of the input speech signal is first weighted by a Hamming window and the magnitude of its discrete Fourier transform (DFT) is then fed into a triangular Mel-frequency bank. The logarithm is computed on the outputs of the filter bank to compress their dynamic range and further decorrelated via the discrete cosine transform (DCT). To take into account feature dynamics over time, usually first- and second-order derivatives of the features are appended to the feature vector.

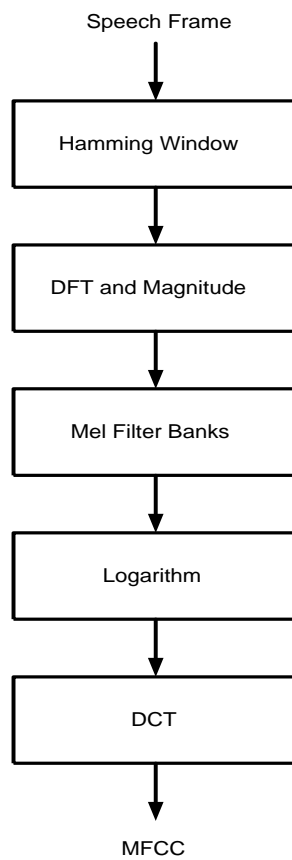


Figure 1.2: Block diagram of MFCC feature computation.

1.3 Hidden Markov Models

Hidden Markov modeling is the mainstream technique that drives modern ASR systems where speech events (e.g. phonemes, syllables or words) are described by a set of HMMs. Fig. 1.3 shows a schematic representation of a hidden Markov model for the word “zero”. The two-dimensional spectrogram illustrates the nonstationarity of acoustic properties of speech sounds. An HMM models such nonstationarity by a Markov chain which is composed of a number of states with transition probabilities. An observation probability density function (PDF) is associated with each state to describe the distribution of speech features in the state. The evolution of states over time can not be directly observed, therefore it is “hidden” from observed features. Typically, an HMM (λ) is characterized by a triplet:

$$\lambda = (A, B, \pi) \tag{1.5}$$

where A , B and π represent state transition probabilities, observation distributions and initial state distributions, respectively.

The theory behind HMMs essentially consists of three fundamental problems [RJ93]:

- Probability Evaluation Problem

Given the observation sequence $O = \{o_1, o_2, \dots, o_T\}$ and the model $\lambda = (A, B, \pi)$, how do we efficiently compute $P(O|\lambda)$?

- Parameter Estimation Problem

How do we adjust the model parameters $\lambda = (A, B, \pi)$ so that $P(O|\lambda)$ could be maximized?

- Optimal State Sequence Problem

Given the observation sequence $O = \{o_1, o_2, \dots, o_T\}$ and the model $\lambda =$

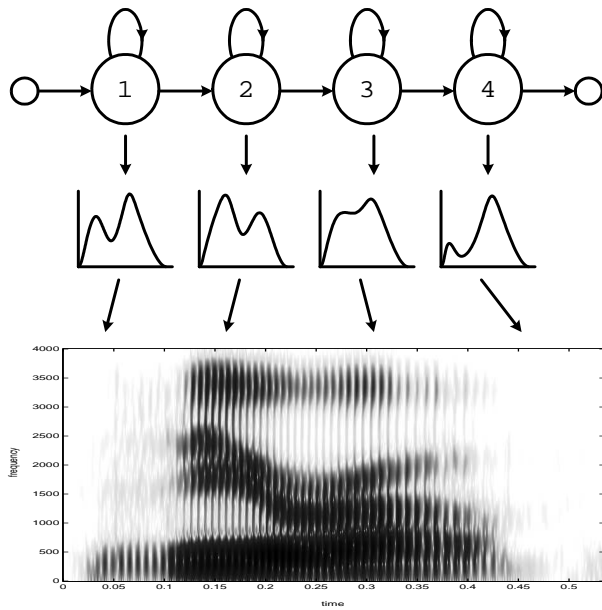


Figure 1.3: A schematic representation of a hidden Markov model for the word “zero”.

(A, B, π) , how do we choose a corresponding state sequence $S = \{s_1, s_2, \dots, s_T\}$ that is optimal in a certain sense?

The solutions to the three problems form the core algorithms in training and recognition of ASR. In particular, Prob.2 addresses the training process where acoustic models are estimated; Prob. 3 searches for the most likely acoustic events corresponding to the given speech signals in the recognition stage; and the efficient probability computation in Prob. 1 facilitates both training and recognition.

Since these three fundamental problems play an essential role in ASR and are the basis for the research work presented in later chapters, a discussion of the solutions in greater detail is necessary.

1.3.1 Forward-Backward Algorithm - Solution to Prob. 1

Direct calculation of the probability $P(O|\lambda)$ involves a summation of all possible state sequences with respect to the observation sequence:

$$\begin{aligned}
 P(O|\lambda) &= \sum_{\text{all } S} P(O, S|\lambda) \\
 &= \sum_{\text{all } S} P(O|S, \lambda)P(S|\lambda) \\
 &= \sum_{s_1, \dots, s_T} \pi_{s_1} b_{s_1}(o_1) a_{s_1 s_2} b_{s_2}(o_2) \cdots a_{s_{T-1} s_T} b_{s_T}(o_T) \quad (1.6)
 \end{aligned}$$

The computational complexity of Eq. 1.6 is prohibitive, especially when T (the number of frames) and N (the number of states) are large. To reduce the computational load, the forward-backward algorithm is used for efficient computation.

The forward-backward algorithm [Bau72] is a recursive algorithm in which the forward variable $\alpha_t(i)$ and the backward variable $\beta_t(i)$ are defined as:

$$\alpha_t(i) = P(o_1, o_2, \dots, o_t, s_t = i|\lambda) \quad (1.7)$$

$$\beta_t(i) = P(o_{t+1}, o_{t+2}, \dots, o_T | s_t = i, \lambda) \quad (1.8)$$

Both $\alpha_t(i)$ and $\beta_t(i)$ could be calculated iteratively. For example, $\alpha_t(i)$ can be computed as follows:

1. Initialization

$$\alpha_1(i) = \pi_i b_i(o_1), \quad 1 \leq i \leq N$$

2. Induction

$$\alpha_{t+1}(j) = [\sum_{i=1}^N \alpha_t(i) a_{ij}] b_j(o_{t+1}), \quad 1 \leq t \leq T-1, \quad 1 \leq j \leq N$$

3. Termination

$$P(O|\lambda) = \sum_{i=1}^N \alpha_T(i)$$

Similarly, $\beta_t(i)$ can be computed as:

1. Initialization

$$\beta_T(i) = 1, \quad 1 \leq i \leq N$$

2. Induction

$$\beta_t(i) = \sum_{j=1}^N a_{ij} b_j(o_{t+1}) \beta_{t+1}(j), \quad t = T-1, \dots, 1, \quad 1 \leq i \leq N$$

The introduction of $\alpha_t(i)$ and $\beta_t(i)$ into the probability calculation dramatically lowers the computational complexity and makes the model parameter estimation in the training process practically feasible.

1.3.2 Baum-Welch Algorithm - Solution to Prob. 2

The training of HMMs is a process of statistical inference where, given an observation sequence, model parameters (A, B, π) are adjusted in accordance to a certain criterion. Since HMMs have hidden variables which could not be observed, common parameter estimation methods directly dealing with the observed data turn out to be difficult. However, this problem has been beautifully solved iteratively thanks to the Expectation-Maximization (EM) algorithm [DLR77]. This iterative maximum likelihood estimation approach is often referred to as the Baum-Welch algorithm after the classic work of Baum and his colleagues [Bau72].

The EM algorithm is designed to deal with so-called “missing” or “unobservable” data. Suppose data O are observed and data S could not be observed directly but are closely related to O through certain mechanism, the EM algorithm deals with the estimation problem by introducing an auxiliary function

$$\begin{aligned} Q(\lambda; \bar{\lambda}) &= \mathbf{E}_S[\log P(O, S|\lambda)|O, \bar{\lambda}] \\ &= \sum_S \log P(O, S|\lambda) p(S|O, \bar{\lambda}) \end{aligned} \tag{1.9}$$

where (O, S) is referred to as complete data and O as incomplete data. It can be proven [DLR77] that by maximizing $Q(\lambda; \bar{\lambda})$ with respect to λ :

$$\lambda^* = \underset{\lambda}{\operatorname{argmax}} Q(\lambda; \bar{\lambda}) \quad (1.10)$$

the likelihood $P(O|\lambda)$ is non-decreasing:

$$P(O|\lambda^*) \geq P(O|\lambda) \quad (1.11)$$

The model parameters obtained this way are substituted back to Eq. 1.9 to start another iteration. This parameter estimation procedure of maximization after expectation proceeds until the likelihood function converges to a local maxima.

When the state observation PDF is represented by Gaussian mixtures:

$$b_i(o_t) = \sum_k w_{ik} \mathcal{N}(o_t; \mu_{ik}, \Sigma_{ik}) \quad (1.12)$$

the re-estimation of the model parameters has a closed-form solution. These closed-form solutions are expressed in terms of the forward and backward variables $\alpha_t(i)$ and $\beta_t(i)$ or another intermediate variable

$$\gamma_t(i, k) = P(s_t = i, \xi_t = k | O, \lambda) \quad (1.13)$$

which represents the probability of being at state i and Gaussian mixture k at time t given the observed feature sequence O and current model parameters λ . It can be computed in terms of $\alpha_t(i)$ and $\beta_t(i)$ by the following relation

$$\gamma_t(i, k) = \frac{\alpha_t(i)\beta_t(i)}{\sum_{i=1}^N \alpha_t(i)\beta_t(i)} \cdot \frac{w_{ik} \mathcal{N}(o_t; \mu_{ik}, \Sigma_{ik})}{\sum_{m=1}^M w_{im} \mathcal{N}(o_t; \mu_{im}, \Sigma_{im})} \quad (1.14)$$

with N and M denoting the number of states and Gaussian mixtures per state in the HMMs.

The re-estimation formulae for the whole set of HMM parameters are listed from Eq. 1.16 to Eq. 1.19:

$$\pi_i = p(s_1 = i | O, \lambda) = \sum_k \gamma_1(i, k) \quad (1.15)$$

$$a_{ij} = \frac{\sum_{t=1}^{T-1} \alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)}{\sum_{t=1}^{T-1} \sum_k \gamma_t^r(i)} \quad (1.16)$$

$$w_{ik} = \frac{\sum_{t=1}^T \gamma_t(i, k)}{\sum_{t=1}^T \sum_k \gamma_t^r(i, k)} \quad (1.17)$$

$$\mu_{ik} = \frac{\sum_{t=1}^T \gamma_t(i, k) \cdot o_t}{\sum_{t=1}^T \gamma_t(i, k)} \quad (1.18)$$

$$\Sigma_{ik} = \frac{\sum_{t=1}^T \gamma_t(i, k) \cdot (o_t - \mu_{ik})(o_t - \mu_{ik})^T}{\sum_{t=1}^T \gamma_t(i, k)} \quad (1.19)$$

Despite the complicated expressions, these re-estimation formulae have straightforward statistical interpretations [RJ93]. In the training process, these formulae are applied in each iteration to update the estimation of model parameters.

1.3.3 Viterbi Algorithm - Solution to Prob. 3

Maximum likelihood is the most widely-used criterion to search for an optimal state sequence which leads to an efficient dynamic programming decoding approach - the Viterbi algorithm. In the recognition stage, a decoding network is created utilizing acoustic HMMs and language models. The Viterbi algorithm searches the network for the state sequence S with the highest likelihood given the observation feature sequence O . Fig. 1.4 shows a decoding network for illustration purposes. Suppose $\phi_j(t)$ represents the maximum likelihood of observing speech feature o_1 to o_t and being in state j at time t , it can be recursively computed as

$$\phi_j(t) = \max_i \{ \phi_i(t-1) \cdot a_{ij} \} [b_j(o_t)], \quad t = 1, \dots, T. \quad (1.20)$$

where

$$\phi_j(1) = \pi_j b_j(o_1), \quad j = 1, \dots, N \quad (1.21)$$

Finally the maximum likelihood is given by

$$\max_j \{\phi_j(T)\}, \quad j = 1, \dots, N \quad (1.22)$$

After the recursion is completed, the maximum likelihood state sequence can be found by tracing back the network.

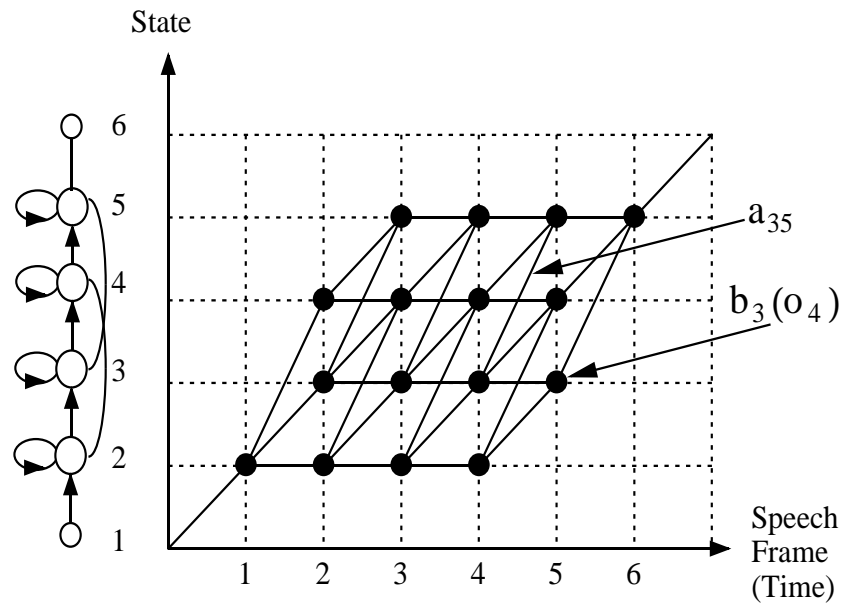


Figure 1.4: The Viterbi decoding network (after [YEK01], Pp.10)

1.4 Robustness in ASR

Although ASR has been studied for quite a long time and impressive progress has been made, it is still haunted by fragility in many aspects. For instance, the performance of an ASR system may be affected by a speaker's gender, age,

speaking rate, accent and emotion. Moreover, noisy environments could further affect performance negatively. The environmental and gender effects are clearly shown in Fig. 1.5 and 1.6. The figures demonstrate an ASR system's behavior on recognizing connected digits in terms of environment and gender groups.

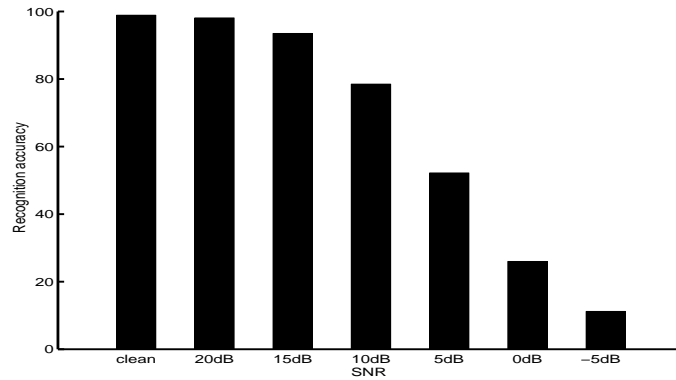


Figure 1.5: Performance of an ASR system on connected digits under subway noise at different SNRs. The system is trained with clean speech.

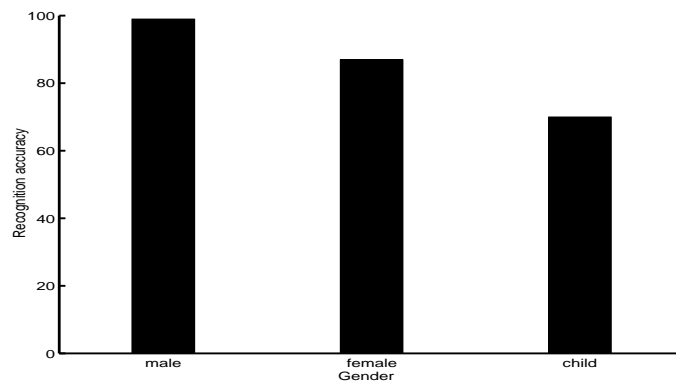


Figure 1.6: Performance of an ASR system on connected digits for different gender and age groups. The system is trained with adult male speech.

Degradation could be observed from both figures if the recognition situation is different from that of training. The degradation is essentially due to the variability of speech signals under noisy conditions and among speakers. Fig. 1.7

demonstrates the spectral difference between a clean and a noisy /u:/ sound and Fig. 1.8 demonstrates two LPC spectra from a female and a young girl for the same sound /i:/. Obvious variations could be observed. This variability is a major factor that hinders the wide deployment of ASR systems. Therefore, robustness is a highly desirable feature and is still the research focus of ASR.

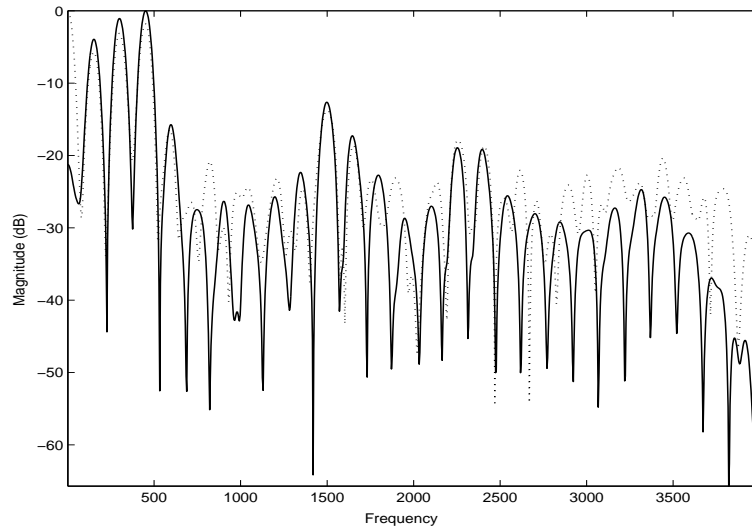


Figure 1.7: Normalized spectra of the sound /u:/ from a clean (solid line) and noisy (dotted line) speech signal.

Robustness in ASR is to address the issue of pattern mismatch. Since current ASR systems are realized through a statistical framework, an analysis of the mismatch from a statistical viewpoint is worth considering. Statistical inference is to evaluate samples from a population and model the population on that basis. The population from which samples are drawn for inference is referred to as the sampled population while the population from which statistical information is desired is referred to as the target population. The statistical statements made from observations of the sampled population are not valid to the target population unless the two populations match. The relationship between the two populations

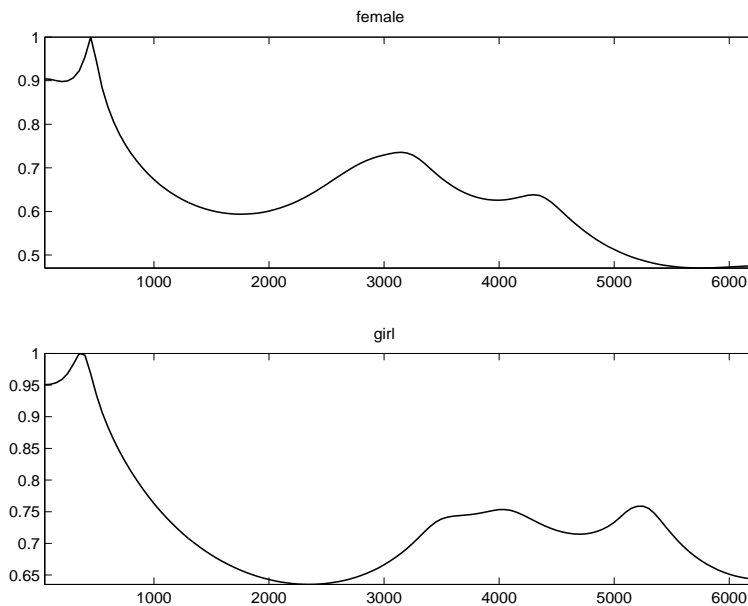


Figure 1.8: Normalized average LPC spectrum of the sound /i:/ from the speech of a female adult (top panel) and a young girl (bottom panel)

is illustrated in Fig. 1.9.

For HMM-based ASR systems, the sampled and target populations correspond to the populations where training and test data are drawn. It is best if the acoustic HMMs could be trained using signals from the target population. Unfortunately, due to the variability of speech signals, mismatch between these two populations almost always exists. For instance, the sampled population in Fig. 1.5 is clean speech while the target population could be noisy signals spoken in a subway. Similarly, the sampled population in Fig. 1.6 is male speech signals while the target population could be speech signals from females or children. To deal with this issue, a common approach is to collect massive amounts of data from a variety of populations. In this way, the sampled population is large enough to overlap with target populations and good performance could be expected. However, this method has some drawbacks. First, since speech signals

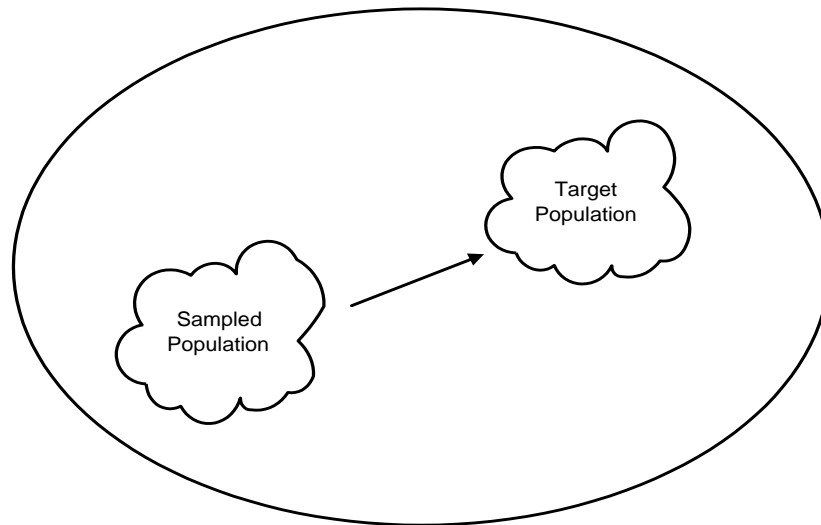


Figure 1.9: Sampled population and target population.

are affected by so many factors it is very hard to collect data from all possible speakers, tasks and backgrounds. Second, pooling speech data across various scenarios may distract the acoustic models from efficiently modeling the speech events themselves, and result in acoustic models with poor discriminative ability. To effectively ameliorate this mismatch, the relationship between the target and sampled populations needs to be investigated, statistically or acoustically, and utilized to improve ASR performance.

1.5 Existing Techniques in Robust ASR

Since robustness is crucial for ASR applications, a large amount of research has been conducted in this area [Gon95][Woo99]. As discussed in the previous section, the key factor to achieve robustness in ASR is being able to handle discrepancies between training and recognition, which originates from variabilities in the signals and is reflected in statistical mismatches. Generally speaking, robust techniques are focused on either the front-end feature domain or back-end acoustic model

domain to either reduce or deal effectively with the mismatch. The techniques are summarized below for environmental and speaker robustness.

1.5.1 Techniques for Environmental Robustness

In the front-end feature domain, spectral subtraction [Bol79] is a common method for noise suppression where the additive noise spectrum $|N(e^{j\omega})|$ is estimated and subtracted from the noisy speech spectrum $|X(e^{j\omega})|$ to recover the clean speech spectrum. The average noise spectrum is typically estimated from the non-speech part of the signal and subtracted from the noisy speech spectrum:

$$|\hat{S}(e^{j\omega})| = \begin{cases} |X(e^{j\omega})| - |N(e^{j\omega})| & \text{if } |X(e^{j\omega})| > |N(e^{j\omega})|, \\ \beta|N(e^{j\omega})| & \text{otherwise,} \end{cases}$$

where $X(e^{j\omega})$, $N(e^{j\omega})$ and $\hat{S}(e^{j\omega})$ are the noisy speech spectrum, noise spectrum and estimated clean speech spectrum, respectively. To deal with non-stationary noise, the noise spectrum may be estimated adaptively. Cepstral mean normalization (CMN)[Ata74] which removes the mean of cepstral features is a popular approach to deal with convolutive channel noise. In [Ace92], an SNR-dependent cepstral normalization (SDCN) algorithm is proposed to compensate noisy speech features in the cepstral domain by removing the compensation vectors from them in a discrete HMM recognizer. The compensation vectors clustered by frame SNRs are estimated using “stereo” data which consist of clean and noisy speech signals recorded simultaneously. Statistical speech feature enhancement based on conditional minimum mean square error estimation is investigated in [DDA04b] where joint prior of static and frame-differential dynamic cepstral features are utilized. Research has also been conducted in search of speech features that model human’s perceptual characteristics and are resistant to noise. Two successful examples are RASTA [HM94] and perceptual linear predictive (PLP) fea-

tures [Her90]. More recently, noise robust front-end feature schemes have been widely investigated for distributed speech recognition (DSR), e.g. [MMN02] and [CIZ02]. In [MMN02], noise robust speech features are obtained using a combination of SNR-dependent waveform processing, two passes of Wiener filtering and blind equalization techniques to achieve impressive performance. In [CIZ02], low-complexity algorithms such as peak isolation [SA97], harmonic demodulation [ZA00a] and variable frame rate [ZA00b] are combined to generate speech features and result in good recognition performance in adverse environments. The algorithms are inspired by the observation that in case of noise, peaks in speech spectra are more robust than valleys, and that formant transitions carry important perceptual information.

In the back-end model domain, parallel model combination (PMC)[GY96] is widely used to adapt the clean acoustic models to match a specific environment. Fig. 1.10 shows a block diagram of PMC with MFCC features. PMC assumes that clean speech and noise are additive in the linear spectral domain. Given acoustic HMMs that are trained on clean speech signals, a single state noise HMM is estimated from background noise. Both acoustic and noise models are first converted from cepstral domain to log-spectral domain by inverse DCT and then to linear spectral domain by exponential operation. They are combined in the linear spectral domain before being transformed back to the cepstral domain by logarithmic operation and DCT. In this way, acoustic models that match the environment are obtained. From reported results, PMC achieves significant improvements in noisy environments. The disadvantages of PMC is its relatively expensive computation since all the acoustic model parameters have to be modified. If the environmental statistics are not stationary, model parameters need to be updated constantly. Since the relationship between the distributions of clean and noisy speech features is statistically nontrivial, vector Taylor series (VTS) and vector polynomial

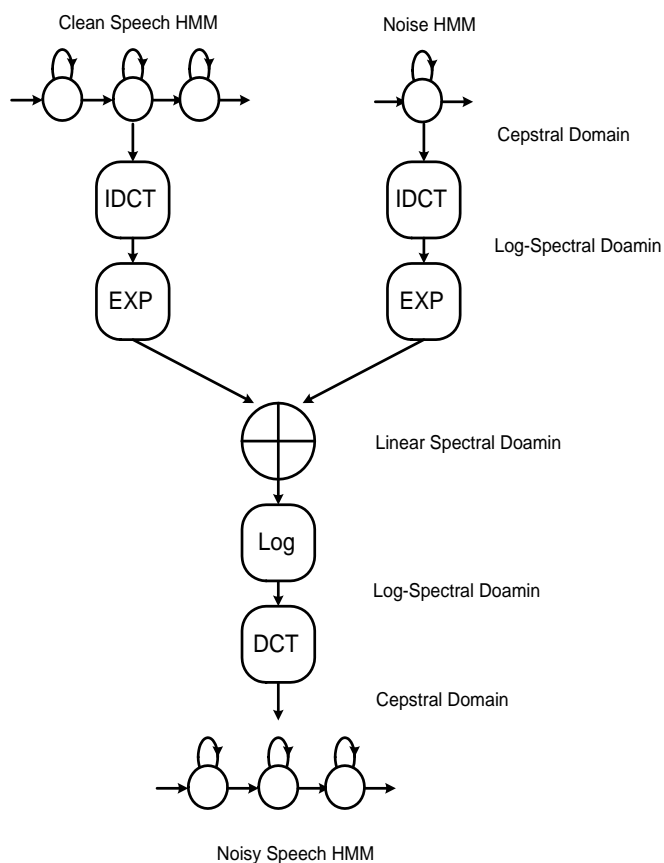


Figure 1.10: Parallel Model Combination process (after [GY96])

approximations (VPS) are proposed in [Mor96] and [RGM96] to approximate the nonlinear compensation relationship, which allows environment adaptation to be carried out in a tractable way. The Jacobian approach in [SYT97] approximates PMC's complicated model compensation by linearization using Jacobian matrices between the initial and test environments. Since the linearization is best performed within a small neighborhood of the initial conditions, the Jacobian method works best for the cases where training and test conditions are not much different. Maximum likelihood linear regression (MLLR) [LW95] is also an effective way to adapt the clean acoustic models to a new environment, although it

was originally developed for speaker adaptation. Without having prior knowledge of the noise, MLLR obtains the environmentally matched models by transforming clean acoustic HMM parameters using linear regression. In comparison with PMC, MLLR is computationally attractive. A modified version of MLLR called piecewise-linear transformation (PLT) is studied in [ZF04] where various types of noise are clustered first based on their spectral properties and acoustic models are trained for each cluster at a variety of SNRs. In recognition, the best matched HMM set is selected and adapted by MLLR.

1.5.2 Techniques for Speaker Robustness

In the front-end feature domain, vocal tract length normalization (VTLN) is a common approach used to reduce the acoustic mismatch originated from a variety of vocal tract shapes. In VTLN, the linear frequency or Mel-frequency axis is scaled by a warping factor as shown in Fig. 1.11 which is obtained via a grid search. In [LR98], a frequency warping approach is investigated. The linear frequency warping factor is estimated from the input speech based on the maximum likelihood criterion and used to re-scale the filter banks when computing MFCC features. An efficient algorithm based on a generic voiced speech model is studied in [WMO96] to simplify the procedure of selecting the frequency warping factor; the technique achieves excellent results using conversational telephone speech. In addition, a variety of frequency axis re-scaling strategies are discussed in [BF96, DNP98, EG96, GS97] to address vocal tract shape variation between children and adults.

In the back-end model domain, speaker adaptation techniques fall into two categories: transform-based and model-based (or direct adaptation methods) approaches. Transform-based approaches relate the original and adapted model parameters by either a linear [LW95] or a nonlinear [PD04] transformation. Most

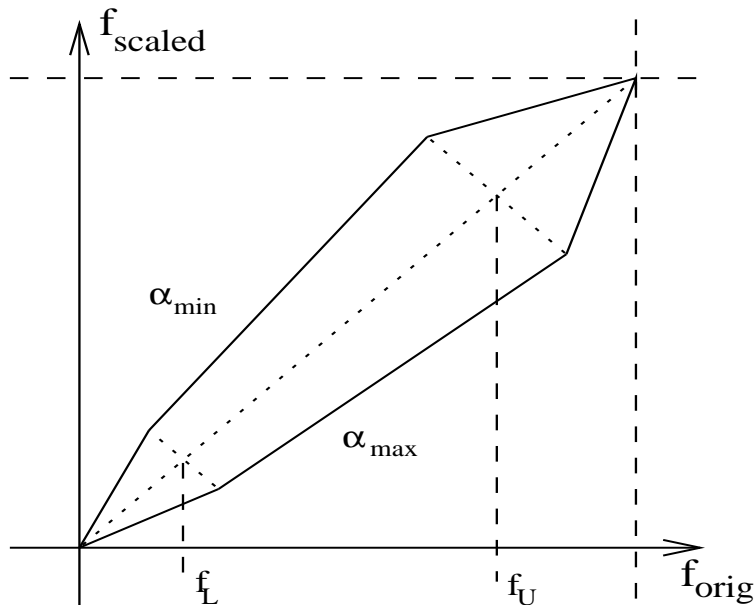


Figure 1.11: Frequency warping for VTLN (after [YEK01], Pp.63)

often the transformations are estimated using the maximum likelihood criterion. The model-based approaches adapt model parameters within a MAP,MMI or MCE framework in which MAP is the most popular. The adapted model parameters, which are considered random variables, are typically estimated based on maximum *a posteriori* criterion with certain prior parameter distribution assumptions [GL94]. In general, unlike VTLN which attempts to compensate for physical (vocal tract) differences, adaptation techniques are statistically driven and do not have clear acoustical justification. Typically, the computational complexity of back-end techniques is higher than VTLN and the techniques require more adaptation data. If the amount of adaptation data is adequate for reliable estimation, adaptation techniques may achieve better performance than VTLN.

If a large amount of adaptation data is available, both transform-based and model-based approaches yield satisfactory performance. In real-world applications, however, situations often occur when only a limited amount of data is

available for adaptation. This may be due to difficulty in collecting more data or a requirement of rapid speaker adaptation. In this situation, data sparseness will affect performance, and can be dealt with using a variety of methods from the two afore-mentioned categories. For the transform-based approaches, a regression class tree is adopted in [LW95] to dynamically tie the transformation parameters while dependencies between acoustic units are studied in [DBB99] and [BDC99] to make effective usage of the data. In the model-based scope, a structural MAP adaptation algorithm is proposed in [KL97] and [SL01] utilizing hierarchical priors and obtains impressive performance. These techniques can yield reliable adaptation for seen or unseen acoustic units by smoothing the adaptation parameters across the sparse data.

Another interesting way to address the sparse-data problem is the eigenvoice method investigated in [KJN00] where the acoustic models are obtained via a linear combination of representative speaker independent models in the eigenvoice (principle components) space and only the linear combination coefficients need to be estimated. The eigenvoice has been extensively studied in the past few years and good performance has been reported (e.g. [Pet01][LR98][MKH04]).

1.6 Speech Databases

The majority of the experiments in this dissertation are conducted using the following databases: TIDIGITS, Aurora 2 and the German part of Aurora 3.

The TIDIGITS database contains speech utterances of connected digit strings recorded in quiet and sampled at 16 kHz. There are 55 males, 57 females, 25 boys and 26 girls in the training corpus and 56 males, 57 females, 25 boys and 25 girls in the test corpus. Each speaker has 77 utterances with string lengths of either 1, 2, 3, 4, 5 or 7 digits (no 6-digit strings in the database).

The Aurora 2 database is a noise-corrupted version of TIDIGITS and the speech signals are downsampled to 8 kHz. Speech data for clean training and multi-condition training are both provided. Only clean training data are utilized in the dissertation which contains 8440 utterances from 55 male and 55 female adult speakers. In the test corpus, three sets of data are provided where Sets A and B are for additive background noise while Set C has both additive and convolutive noise. Since the additive noise is the focus of this dissertation, Sets A and B are chosen for ASR experiments. There are eight types of background noise in the Aurora 2 database, which are subway, babble, car and exhibition noise in Set A and restaurant, street, airport and station noise in Set B. Noisy speech data are generated by artificially adding the noise signals at a variety of SNR levels (clean, 20 dB, 15 dB, 10 dB, 5 dB, 0 dB and -5 dB). For each SNR level and each type of noise in Sets A and B, there are 1001 utterances from 52 male and 52 female adult speakers. Thus, in total, each set consists of 24024 utterances.

The Aurora 3 database is a multi-lingual speech corpus which includes utterances of connected digit strings from German, Spanish, Finnish and Danish. For the German part of the Aurora 3 database, the utterances are recorded in a real car environment that includes four different conditions: stopped with motor running (SMR), town traffic (TT), low speed rough road (LSRR), and high speed good road (HSGR). The data are recorded from different scenarios such as left front window open or closed, sunroof open or closed, etc. All the data are recorded by a close-talking microphone and a hands-free microphone. There are 2929 utterances in the database from which utterances are selected for three training and test conditions: well-matched (WM), medium-mismatched (MM) and high-mismatched (HM). The WM experiment utilizes speech from both microphone types and all driving conditions for both training and testing

conditions. It includes 2032 utterances in training and 897 utterances for testing. The MM experiment utilizes 997 utterances as training data from a hands-free microphone using all driving conditions except for the HSGR driving condition, and 241 utterances as test data from a hands-free microphone with HSGR driving condition. The HM experiment utilizes 1007 utterances as training data from the close-talking microphone and all driving conditions, and 394 utterances as test data from the hands-free microphone for all driving conditions except the SMR driving condition. In all cases, there are 36 male and 43 female speakers in the training set, and 15 male and 18 female talkers in the test set.

1.7 Organization of Dissertation

This dissertation is composed of two parts: environmental robustness and speaker robustness.

In part I, environmental robustness is addressed which consists of two chapters. Chapter 2 introduces the weighted Viterbi decoding algorithm and Chapter 3 presents a feature compensation algorithm using polynomial regression of utterance SNR.

In part II, speaker adaptation is discussed which consists of three chapters. Chapter 4 is concerned with adaptation text design based on the Kullback-Leibler measure. Chapter 5 proposes a rapid adaptation scheme by formant-like peak alignment. Chapter 6 investigates structured MLLR transformations and model averaging on the basis of minimum description length criterion.

Finally, Chapter 7 summarizes the dissertation and discusses future work.

Part I

Environmental Robust Speech Recognition

This part of the dissertation is focused on the environmental robust issues in automatic speech recognition with an emphasis on additive background noise. This type of noise is frequently encountered and most harmful in real-world deployment of speech recognition systems.

The performance degradation of speech recognition systems is due to the mismatch between training and test data which are not collected in the same environments. Hidden Markov modeling of acoustic patterns is based on statistical inference which is quite sensitive to this mismatch. Therefore, compensation is required to remedy the discrepancy between front-end features and back-end acoustic models.

Two approaches are investigated in Part I to combat background noise. Both of them keep the clean acoustic models unchanged. In Chapter 2, a computationally inexpensive approach - weighted Viterbi decoding (WVD) - is discussed. WVD utilizes the frame SNR information in the Viterbi decoding stage of recognition. The tradeoff between its computational complexity and recognition performance makes WVD attractive in distributed speech recognition (DSR) scenarios. In Chapter 3, an approach of feature compensation by polynomial regression of utterance SNR is investigated. Regression polynomials are used to describe the environmental non-stationarity in terms of utterance SNR and also to predict unobserved environments. As will be shown, this feature compensation algorithm yields significant performance improvements under a variety of SNR conditions.

CHAPTER 2

Weighted Viterbi Decoding

In adverse environments, speech signals are corrupted by noise. The mismatch between the noise-corrupted features and the HMMs trained using clean speech significantly deteriorates the system performance. For a noisy utterance, SNRs vary in a relatively wide range from frame to frame. Intuitively, those frames of higher SNR match the clean models better than those of lower SNR. In [BA02], a weighted Viterbi decoding (WVD) algorithm was introduced to deal with channel impairments, frame erasures and network congestion for distributed speech recognition (DSR). In this chapter, we use the WVD algorithm to cope with background acoustic noise without changing the acoustic speech models, where weights are assigned as a function of frame SNR to the feature observations in the original Viterbi decoding stage.

The remainder of this chapter is organized as follows. In Section 2.1, the system implementation scheme of the WVD algorithm and its formulation are provided. In Section 2.2, the frame SNR estimation based on minimum statistics tracking is described. Experimental results are given in Section 2.3, and Section 2.4 concludes the chapter with a summary and discussion.

2.1 System Implementation

Fig. 2.1 shows the implementation of WVD where the acoustic HMMs are trained using clean speech signals and the front-end feature extraction uses common fea-

tures such as MFCC, LPCC and PLP. The SNR is estimated for each speech frame and the estimate is provided to the Viterbi decoder where a final decision is made based on the clean acoustic models and the confidence/quality of each speech frame. WVD modifies the recursive step of the Viterbi algorithm

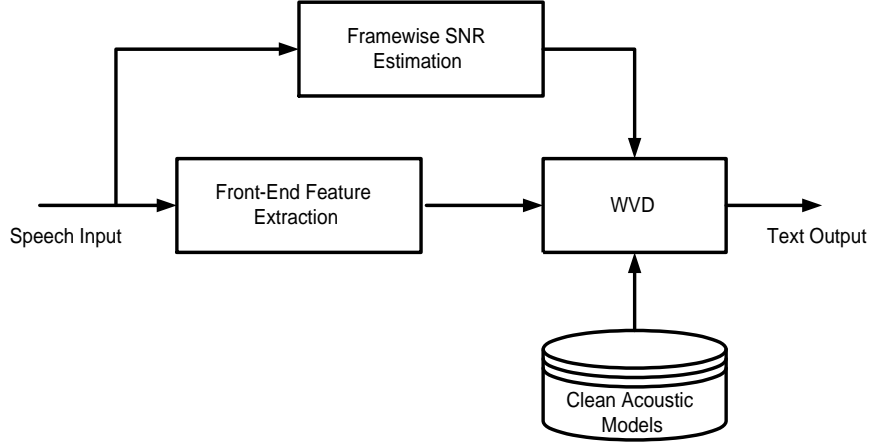


Figure 2.1: Diagram of weighted Viterbi decoding (WVD).

(Eq. 1.20) to take into account the effect of SNR by weighting the probability of observing features given the HMM state j , $b_j(\mathbf{o}_t)$, with the confidence factor of the current feature observation \mathbf{o}_t . The time-varying confidence factor γ_t is inserted into the Viterbi algorithm by raising the probability $b_j(\mathbf{o}_t)$ to the power γ_t to obtain the following state update equation:

$$\phi_j(t) = \max_i \{ \phi_i(t-1) \cdot a_{ij} \} [b_j(\mathbf{o}_t)]^{\gamma_t} \quad (2.1)$$

where $\phi_j(t)$ represents the maximum likelihood of observing speech feature \mathbf{o}_1 to \mathbf{o}_t and being in state j at time t , a_{ij} stands for the transition probability from state i to state j and $\gamma_t \in [0, 1]$ is a time-varying frame confidence factor that maps the frame SNR into the interval $[0, 1]$. The values of γ_t are determined empirically.

In extreme cases, when $\gamma_t = 0$, $\phi_j(t)$ is updated only by state transition probability a_{ij} and the probability $b_j(\mathbf{o}_t)$ for the current frame is discarded; when $\gamma_t = 1$, the current frame is decoded by the regular Viterbi decoding scheme.

2.2 Frame SNR Estimation

SNR plays an important role in noise robust speech recognition since it pertains to the degree at which a clean speech signal is corrupted by noise. Therefore, it is used as an informative index in the noise robust algorithms discussed in Part I of this dissertation. This necessitates a good SNR estimate. Depending on the algorithm, frame SNR or utterance SNR is used. To be specific, the weighted Viterbi decoding algorithm investigated in this chapter uses frame SNR and the feature compensation algorithm investigated in the next chapter employs utterance SNR.

To estimate SNRs, a minimum statistics tracking method proposed in [Mar01] is adopted. Assume noise and speech signals are statistically independent, the power of a noisy speech signal is a summation of the powers of clean speech signal and background noise. The power of the noisy speech signal decays to the power of the background noise in the silence part of the signal. Hence, tracking power spectral minima provides fairly accurate estimation of the background noise power, hence good estimation of SNR. Also, by tracking minimum statistics, this algorithm can deal with nonstationary background noise with slowly changing statistical characteristics, as shown in Figs. 2.2. One disadvantage of this approach is the bias between the mean and minimum value of the background noise signal. A simple yet effective way to alleviate this bias is to apply a well-chosen factor to the estimate as shown in [Mar01]. Power spectral minimum statistics are searched within a 0.5 second interval preceding each speech frame.

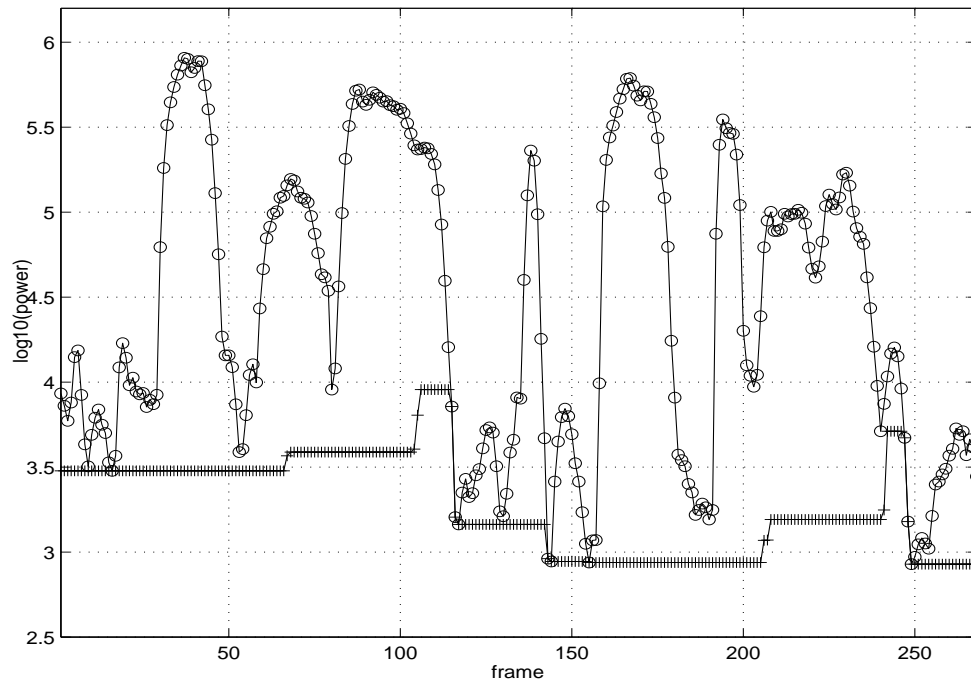


Figure 2.2: Noise power(+) estimated by minimum statistics tracking from the noisy speech power spectrum (o) for the utterance “43o6571”. The utterance is labeled 15 dB SNR in the Aurora 2 database.

In real-world applications, this will introduce an extra memory requirement and time delay. However, compared to other complicated front-end processing algorithms, this overhead is quite tolerable.

After the noise power is estimated, the clean speech power is computed by subtracting the noise power from the noisy speech power. In case negative values occur, a small positive floor is set. Estimates of clean and noise power being available, the frame-wise SNR can be readily calculated as

$$\text{SNR} = 10 \cdot \log_{10} \frac{\text{clean speech power}}{\text{noise power}} \quad (2.2)$$

Figs. 2.3 and 2.4 show estimated frame-based SNRs and the confidence factor

(γ_t) for two speech utterances from the Aurora 2 database. It is obvious from the figures that frame-based SNR values vary in a wide range compared to the overall utterance SNR. Also note that there is an SNR floor set at 0 dB for all frames because we assume that SNR estimates below 0 dB are not reliable.

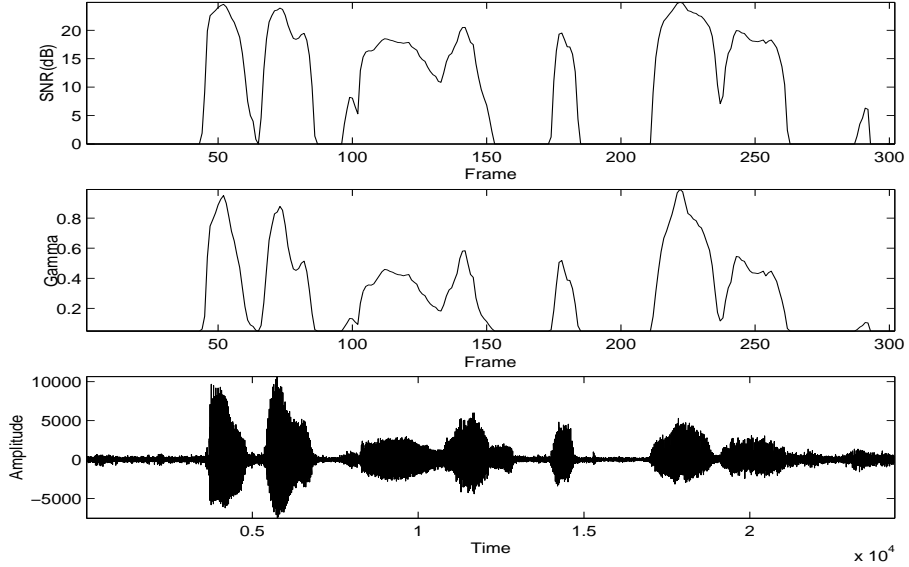


Figure 2.3: Estimated SNR (top panel), confidence factor (γ_t) (middle panel) and waveform (bottom panel) of the utterance “0021641” labeled in the Aurora 2 database as having a signal-to-noise ratio of 15 dB.

Two sets of γ_t are chosen depending on the utterance SNR as shown in Fig. 2.5. When the utterance SNR is above 10 dB, then

$$\gamma_t = \begin{cases} 1 & \text{for SNR} \geq 25 \text{ dB} \\ e^{0.12 \cdot (\text{SNR} - 25)} & \text{for SNR} < 25 \text{ dB.} \end{cases} \quad (2.3)$$

If the utterance SNR is less than 10 dB, a simple normalization by the maximum SNR value of the utterance is adopted:

$$\gamma_t = \frac{\text{SNR}}{\text{SNR}_{max}} \quad (2.4)$$

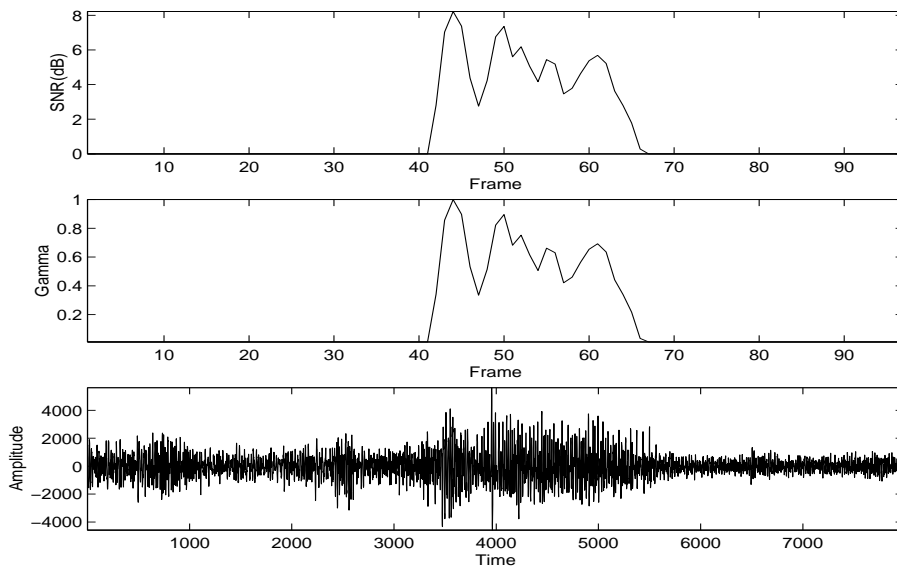


Figure 2.4: Estimated SNR (top panel), confidence factor (γ_t) (middle panel) and waveform (bottom panel) of the utterance “4” labeled in the Aurora 2 database as having a signal-to-noise ratio of 0 dB.

In the above γ_t definitions, the SNRs refer to the frame SNR if it is not mentioned as utterance SNR.

2.3 Experimental Results

Experiments are conducted on the Aurora 2 and Japan Electronic Industry Development Association (JEITA) databases. WVD is performed with a variety of common front-end features (e.g. MFCC, LPCC and PLP) and also compared to the widely-used back-end model adaptation technique – MLLR.

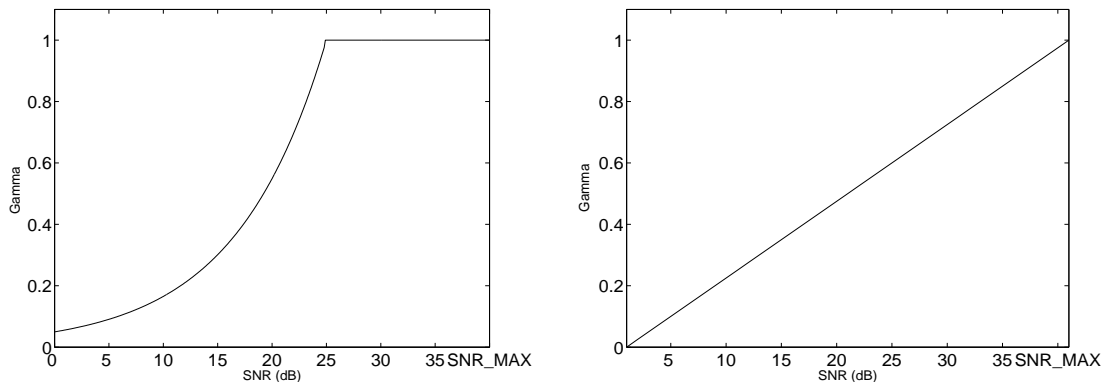


Figure 2.5: γ_t (gamma) for utterance SNRs higher than 10 dB (left) and γ_t (gamma) for utterance SNRs lower than 10 dB (right).

2.3.1 Aurora 2 database

As mentioned in Chapter 1, the Aurora 2 database has clean speech data in its training corpus while contains noisy speech at various SNR levels in its test corpus. Sets A and B which are only corrupted by additive noise are chosen for test. A left-to-right topology is adopted for the acoustic HMMs. Each digit has 16 states with 3 Gaussian mixtures per state. There are one 3-state silence model with 6 mixtures and one short pause model that shares the middle state of the silence model. The above HMM structure is the standard setup provided by the Aurora 2 database [HP00].

2.3.1.1 WVD with MFCC, LPCC and PLP

Tables 2.1, 2.2 and 2.3 show the improvements of WVD over the baseline performance for an MFCC, LPCC and PLP front-end, respectively. The test Set A has four types of background noise: subway, babble, car and exhibition noise and Set B has restaurant, street, airport and station noise. For each type of noise, six noise levels are tested ranging from clean to 0 dB as shown in the tables. For

each noise level, word accuracy averaged over all the four noise types are presented. From the tables, we observe that WVD yields consistent improvements for all noise conditions and noise levels. On average, the algorithm reduces the word error rate by 38%, 45% and 47% for MFCC, LPCC and PLP features, respectively compared with their baselines in Set A and 50%, 53% and 53% for Set B. For both sets of data, the average improvements did not include the clean condition. The highest error reduction is achieved at 10 dB SNR. Furthermore, WVD performs better for data from Set B compared to Set A. MFCC features give the best baseline results. With WVD, PLP and MFCC have comparable performance.

	Set A			Set B		
	Baseline	WVD	Imprv.	Baseline	WVD	Imprv.
Clean	98.9	99.0	1.9	98.9	99.0	1.9
20 dB	95.0	96.5	30.1	92.4	96.4	53.2
15 dB	86.9	92.9	45.6	80.8	91.8	57.5
10 dB	67.3	84.7	53.2	58.1	86.5	67.7
5 dB	39.4	63.0	39.0	32.0	63.5	46.2
0 dB	17.1	34.9	21.5	14.6	35.3	24.2

Table 2.1: WVD performance (%) with MFCC features on the Aurora 2 database.

	Set A			Set B		
	Baseline	WVD	Imprv.	Baseline	WVD	Imprv.
Clean	98.7	98.7	1.6	98.7	98.7	1.6
20 dB	91.1	94.9	42.7	86.8	93.9	53.7
15 dB	76.7	90.7	60.3	69.4	92.5	75.4
10 dB	52.9	79.4	56.4	45.9	81.6	66.1
5 dB	27.2	57.7	42.0	24.3	59.1	45.9
0 dB	12.1	31.6	22.2	9.7	32.9	25.7

Table 2.2: WVD performance (%) with LPCC features on the Aurora 2 database.

	Set A			Set B		
	Baseline	WVD	Imprv.	Baseline	WVD	Imprv.
Clean	98.9	99.0	9.1	98.9	99.0	9.1
20 dB	93.8	96.5	43.3	91.7	96.2	54.0
15 dB	82.1	92.4	57.2	78.0	91.0	59.2
10 dB	59.9	84.8	62.2	54.0	86.9	71.6
5 dB	33.6	63.7	45.4	27.6	65.4	52.3
0 dB	13.4	35.9	26.1	10.4	37.2	29.9

Table 2.3: WVD performance (%) with PLP features on the Aurora 2 database.

2.3.1.2 WVD vs. MLLR

Since WVD constitutes a very simple back-end technique for noise robustness, it is interesting to compare it with MLLR which is a widely-applied back-end model adaptation technique with good performance. In experiments, 40 utterances are randomly selected from each type of noise for MLLR adaptation. Therefore, there are 320 adaptation utterances in total.

Table 2.4 shows the performance of WVD and MLLR compared with the baseline for the Aurora 2 database with 8 types of noise and using MFCC features. For each type of noise, an average over all the SNR levels (clean, 20 dB, 15 dB, 10 dB, 5 dB and 0 dB) are calculated. Both WVD and MLLR result in improvements over the baseline. However, given the 320 adaptation utterances, WVD outperforms MLLR by 2.4% on average without the need for a priori knowledge of noise statistics nor the need for off-line training while MLLR has these requirements.

	Baseline	MLLR	WVD	Imprv.
Subway	74.7	77.0	76.6	-1.7
Babble	58.4	73.0	80.1	26.3
Car	66.4	78.0	79.9	8.6
Exhibition	70.3	74.5	77.4	11.4
Restaurant	59.4	74.8	75.9	4.4
Street	67.8	77.6	78.7	4.9
Airport	60.9	78.1	81.2	14.2
Station	63.1	76.8	79.2	10.3

Table 2.4: Performance (%) of WVD vs. MLLR for eight types of noise from Set A (first four types) and Set B (second four types) of the Aurora 2 database.

2.3.2 JEITA Database

In JEITA database¹, the training corpus contains 40 male and 40 female speakers with 3 repetitions of 35 connected digit strings for each speaker. The test corpus contains 6 SNR levels (clean, 20 dB, 15 dB, 10 dB, 5 dB and 0 dB) of car noise which is added to the clean data manually. For each SNR level, 35 male and 35 female speakers with only 1 repetition of 35 connected digit strings are included. The utterances are the same across all the SNR subsets. Thus, totally there are 8400 utterances for training and 14700 utterances for test with 2450 utterances for each SNR subset. The selection the corpus makes sure that speakers in the training and test set are exclusive. There are 10 Japanese digits in the training and test corpus which are listed in Table. 2.5.

number	0	1	2	3	4	5	6	7	8	9
Japanese	zero	ichi	ni	san	yon	go	roku	nana	hachi	kyu

Table 2.5: 10 Japanese digits used in the experiments

Word models with 16 states are used for each Japanese digit. The HMM topology is from-left-to-right. There are 3 Gaussian mixtures assigned to each state. Besides the HMMs for the digits, there are one 3-state silence model (sil) and one single-state short-pause model (sp) where the single state has a "T" topology allowing skipping. All of these share the same setup as the Aurora 2 database.

WVD is tested versus the baseline under all SNR levels. Its performance with MFCC, LPCC and PLP as front-end features are shown in Table 2.6, 2.7 and 2.8, respectively. On average, WVD achieves 20% word error rate reduction compared to the baseline.

¹This part of work was performed when the author was a summer intern student at Texas Instruments.

	Baseline	WVD	Imprv.
Clean	98.09	97.94	-7.9
20 dB	79.96	84.78	24.1
15 dB	68.15	76.21	25.3
10 dB	47.24	61.15	26.4
5 dB	24.78	38.72	18.5
0 dB	14.25	18.30	4.7

Table 2.6: WVD performance (%) with MFCC features on the JEITA database.

	Baseline	WVD	Imprv.
Clean	98.09	97.91	-9.4
20 dB	80.21	85.39	26.2
15 dB	66.84	77.16	31.1
10 dB	46.13	61.38	28.3
5 dB	24.89	39.37	19.3
0 dB	14.48	19.09	5.4

Table 2.7: WVD performance (%) with PLP features on the JEITA database.

	Baseline	WVD	Imprv.
Clean	97.51	97.33	-7.2
20 dB	78.27	83.40	23.6
15 dB	64.49	74.51	28.2
10 dB	46.01	60.26	26.4
5 dB	23.65	38.12	19.0
0 dB	13.99	18.14	4.8

Table 2.8: WVD performance (%) with LPCC features on the JEITA database.

2.4 Summary

In this chapter, a weighted Viterbi decoding algorithm is investigated to deal with background noise. A confidence factor is assigned to each speech frame based on its SNR estimate. The clean acoustic models and noisy features are not changed or updated in this approach. Experimental results show consistent improvements of WVD with MFCC, LPCC and PLP features with different types of background noise and SNR levels (except a slight performance drop under clean condition on the JEITA database).

Weighted Viterbi decoding has low computational complexity and is simple to implement. However, it does not take into account the noise characteristics and make use of information provided by clean acoustic models. Therefore, the performance improvements are not significant. In the next chapter, a feature compensation algorithm using polynomial regression of utterance SNR is proposed to cope with the background noise. The feature compensation reduces the mismatch between the clean acoustic models and noisy features and thus yields better recognition accuracy than weighted Viterbi decoding.

CHAPTER 3

Feature Compensation Based on Polynomial Regression of SNR

Intuitively, the degree of mismatch between noisy speech features and clean acoustic models depends on SNRs. To recover a clean speech feature in the presence of noise, the SNR information should be taken into account.

In [CG03], significant improvements are achieved in noisy speech recognition by using variable parameter Gaussian mixture hidden Markov models (VPGMHMM) whose Gaussian mean vectors under different environments are described by polynomial functions of an environment-dependent continuous variable. In recognition, one set of HMMs is instantiated according to the environment. By modeling the trend of the Gaussian means, VPHMM has smaller Gaussian variances which indicates higher model discriminative abilities. Typically, the estimation of the state emission parameter polynomials requires relatively a large amount of data under the target environments.

In this chapter, feature compensation based on polynomial regression of the utterance SNR is investigated where the bias between the clean and noisy speech features in the cepstral domain is approximated by a set of polynomials with respect to utterance SNR. During the recognition stage, utterance SNR is first estimated and compensation bias is then computed and removed from the noisy speech cepstral feature. The compensated feature is fed into the decoding network created using clean acoustic HMMs. The maximum likelihood estimate of the

feature compensation polynomials are obtained by the Expectation-Maximization (EM) algorithm. Depending on the amount of adaptation data available, the polynomials could be flexibly tied at different levels of granularity. By learning the trend of the bias as a function of SNR, the algorithm is able to predict the bias at an unobserved SNR condition in the training data. The biases, not the means, are approximated by polynomials in this chapter because the biases allow more flexible tying schemes with limited amounts of adaptation data, e.g. global tying or a few tying classes. This is not easy to accomplish directly on the means. Furthermore, with the knowledge of clean acoustic models, biases can achieve more robust estimation compared with the means.

The remainder of this chapter is organized as follows. In Section 3.1, the motivation and formulation of feature compensation utilizing polynomial regression of utterance SNRs is given. The utterance SNR estimation based on minimum statistics tracking is described in Section 3.2. The training and recognition scheme of the algorithm and comparative experimental results with MLLR are shown in Section 3.3. Finally, Section 3.4 concludes the chapter with a summary.

3.1 Polynomial Regression of Utterance SNR

In this section, the motivation of using SNR-based regression polynomials for feature compensation is introduced. The ML estimation of the regression polynomials in an EM framework is also described.

3.1.1 Bias Approximation by SNR Polynomials

For additive noise, assuming that clean speech signals and noise are statistically independent in each filter bin, the power of a noisy speech signal in the k th filter bin of each frame is the summation of the powers of clean speech and noise of

the filter bin:

$$Y_k^{\text{lin}} = X_k^{\text{lin}} + N_k^{\text{lin}} \quad (3.1)$$

where Y_k^{lin} , X_k^{lin} and N_k^{lin} denote noisy speech, clean speech and noise in the linear power domain of the k th filter bin, respectively. For the k th filter bin in the log-power domain, Eq. 3.1 could be rewritten as:

$$\begin{aligned} Y_k^{\text{log}} &= X_k^{\text{log}} + \log\left(1 + \frac{N_k^{\text{lin}}}{X_k^{\text{lin}}}\right) \\ &= X_k^{\text{log}} + \log\left(1 + \frac{1}{\text{SNR}_k}\right) \end{aligned} \quad (3.2)$$

$$= X_k^{\text{log}} + g_k \quad (3.3)$$

where Y_k^{log} and X_k^{log} represent noisy and clean speech in the log-power domain, SNR_k is the signal-to-noise ratio and

$$g_k \triangleq \log\left(1 + \frac{1}{\text{SNR}_k}\right) \quad (3.4)$$

Applying Discrete Cosine Transform (DCT) on both sides of Eq. 3.3, we get the n th cepstral coefficient as:

$$\begin{aligned} Y_n^{\text{cep}} &= \sum_k d_{nk} Y_k^{\text{log}} \\ &= \sum_k d_{nk} (X_k^{\text{log}} + g_k) \\ &= \sum_k d_{nk} X_k^{\text{log}} + \sum_k d_{nk} g_k \end{aligned} \quad (3.5)$$

Since g_k is a function of utterance SNR, Eq. 3.5 could be written as

$$Y_n^{\text{cep}} = X_n^{\text{cep}} + f_n(\text{SNR}) \quad (3.6)$$

where Y_n^{cep} and X_n^{cep} are the n th cepstral component of noisy and clean speech, d_{nk} 's are the DCT coefficients and $f_n(\text{SNR})$ denotes a function of utterance SNR of the n th cepstral coefficient.

From Eq. 3.6, it is clear that the bias between the clean and noisy features is a nonlinear function of utterance SNR. In this work, this nonlinear function is approximated by a polynomial of order P regressing on utterance SNR, that is:

$$Y_n^{\text{cep}} \approx X_n^{\text{cep}} + \sum_{j=0}^P \tilde{c}_{jn}(\text{SNR})^j \quad (3.7)$$

where \tilde{c}_{jn} 's are the coefficients for the j th order terms of the n th cepstrum. The above relation provides a way to recover the clean speech feature (X_n^{cep}) by compensating the noisy feature (Y_n^{cep}) with the polynomial approximated bias if one has the SNR for the utterance:

$$X_n^{\text{cep}} \approx Y_n^{\text{cep}} - \sum_{j=0}^P \tilde{c}_{jn}(\text{SNR})^j \quad (3.8)$$

3.1.2 Feature Compensation

Assuming that the clean acoustic models are Gaussian mixture HMMs, the probability density function of observing feature \mathbf{o}_t from state i is computed as:

$$p(\mathbf{o}_t | s_t = i) = \sum_k w_{ik} b_{ik}(\mathbf{o}_t) \quad (3.9)$$

where $b_{ik}(\mathbf{o}_t) \sim \mathcal{N}(\mathbf{o}_t; \boldsymbol{\mu}_{ik}, \boldsymbol{\Sigma}_{ik})$ is the k th multivariate Gaussian mixture in state i with weight w_{ik} , $\boldsymbol{\mu}_{ik}$ and $\boldsymbol{\Sigma}_{ik}$ are the mean vector and covariance matrix associated with it, respectively.

The feature compensation algorithm removes the polynomial approximated bias from the noisy speech feature using the estimated SNR during the mixture Gaussian probability calculation, which is shown in Eq. 3.10:

$$p(\mathbf{o}_t | s_t = i) = \sum_k w_{ik} \mathcal{N}(\mathbf{o}_t - \sum_{j=0}^P \mathbf{c}_{ikj} \eta^j; \boldsymbol{\mu}_{ik}, \boldsymbol{\Sigma}_{ik}) \quad (3.10)$$

In Eq. 3.10, \mathbf{o}_t is the noisy speech feature, $\boldsymbol{\mu}_{ik}$ and $\boldsymbol{\Sigma}_{ik}$ are the mean and covariance of Gaussian mixtures in clean acoustic HMMs. η is the utterance

SNR. \mathbf{c}_{ikj} 's are the coefficients of the regression polynomials of state i , mixture k and polynomial order j . \mathbf{c}_{ikj} is a vector with the same dimension as the feature vector which means each component in the feature vector has its own regression polynomial with coefficients \tilde{c}_{ikjn} .

Depending on the adaptation data available from the new environment, the regression polynomials could be tied flexibly at different levels of granularity - mixtures, states, phonetic classes or globally shared for all HMMs.

3.1.3 Maximum Likelihood Estimation of Regression Polynomials

The ML estimation of the regression polynomials from the environmental adaptation data is performed in an EM [DLR77] framework.

We assume that the incorporation of the feature compensation into the Gaussian mixture does not affect the initial state probabilities, state transition probabilities and Gaussian mixture weights. Therefore, define the EM auxiliary function we are interested in as:

$$Q_b(\lambda; \bar{\lambda}) = \sum_{r=1}^R \sum_{i \in \Omega_s} \sum_{k \in \Omega_m} \sum_{t=1}^{T^r} \gamma_t^r(i, k) \cdot \log b_{ik}(\mathbf{o}_t^r) \quad (3.11)$$

where R is the utterance number of adaptation data and T^r is the frame number of the r th utterance. $\Omega_s = \{1, 2, \dots, N\}$ and $\Omega_m = \{1, 2, \dots, M\}$ are the state and mixture sets, respectively. $\gamma_t^r(i, k) = p(s_t^r = i, \xi_t^r = k | O^r, \bar{\lambda})$ is the posterior probability of staying at state i mixture k at time t given the r th observation sequence.

Without loss of generality, we assume that each Gaussian mixture has one set of distinct regression polynomials. For other tying strategies, the derivations follow accordingly utilizing the collection of the corresponding statistics within each tying set. The extension to other strategies is straightforward and will be discussed later.

Optimizing $Q_b(\lambda; \bar{\lambda})$ with respect to c_{ikl} , one obtains:

$$\begin{aligned}
\frac{\partial Q_b(\lambda; \bar{\lambda})}{\partial c_{ikl}} &= \frac{\partial}{\partial c_{ikl}} \sum_{r=1}^R \sum_{i=1}^N \sum_{k=1}^M \sum_{t=1}^{T^r} \gamma_t^r(i, k) \cdot \log N(\mathbf{o}_t^r - \sum_{j=0}^P \mathbf{c}_{ikj}(\eta^r)^j; \boldsymbol{\mu}_{ik}, \boldsymbol{\Sigma}_{ik}) \\
&= \frac{\partial}{\partial c_{ikl}} \sum_{r=1}^R \sum_{t=1}^{T^r} \gamma_t^r(i, k) \cdot \\
&\quad \left[-\frac{1}{2} (\mathbf{o}_t^r - \sum_{j=0}^P \mathbf{c}_{ikj}(\eta^r)^j - \boldsymbol{\mu}_{ik})^T \boldsymbol{\Sigma}_{ik}^{-1} (\mathbf{o}_t^r - \sum_{j=0}^P \mathbf{c}_{ikj}(\eta^r)^j - \boldsymbol{\mu}_{ik}) \right] \\
&= \sum_{r=1}^R \sum_{t=1}^{T^r} \gamma_t^r(i, k) \cdot \boldsymbol{\Sigma}_{ik}^{-1} \cdot (\mathbf{o}_t^r - \sum_{j=0}^P \mathbf{c}_{ikj}(\eta^r)^j - \boldsymbol{\mu}_{ik}) \cdot (\eta^r)^l = 0 \\
&\hspace{20em} l = 0, 1, \dots, P \tag{3.12}
\end{aligned}$$

By regrouping terms, Eq. 3.12 can be rewritten as:

$$\sum_{j=0}^P \left[\sum_{r=1}^R \sum_{t=1}^{T^r} \gamma_t^r(i, k) \boldsymbol{\Sigma}_{ik}^{-1} (\eta^r)^{j+l} \right] \mathbf{c}_{ikj} = \sum_{r=1}^R \sum_{t=1}^{T^r} \gamma_t^r(i, k) \boldsymbol{\Sigma}_{ik}^{-1} (\mathbf{o}_t^r - \boldsymbol{\mu}_{ik}) (\eta^r)^l \\
l = 0, 1, \dots, P \tag{3.13}$$

In a similar way as [CG03], define:

$$\psi(\zeta, \rho, \alpha, \beta) = \sum_{r=1}^R \sum_{t=1}^{T^r} \gamma_t^r(i, k) \cdot \boldsymbol{\Sigma}_{ik}^{-1} \cdot \zeta^\alpha \rho^\beta \tag{3.14}$$

Eq. 3.13 is simplified into:

$$\sum_{j=0}^P \psi(\eta^r, \eta^r, l, j) \cdot \mathbf{c}_{ikj} = \psi(\eta^r, \mathbf{o}_t^r - \boldsymbol{\mu}_{ik}, l, 1) \\
l = 0, 1, \dots, P \tag{3.15}$$

The $P + 1$ equations in Eq. 3.15 can be expressed in a matrix form:

$$\mathbf{U}_{ik} \cdot \mathbf{c}_{ik} = \mathbf{v}_{ik} \tag{3.16}$$

In Eq. 3.16, \mathbf{U}_{ik} is a $(P + 1) \times (P + 1)$ dimensional block matrix:

$$\mathbf{U}_{ik} = \begin{bmatrix} u_{ik}(0, 0) & \cdots & u_{ik}(0, P) \\ \vdots & u_{ik}(l, j) & \vdots \\ u_{ik}(P, 0) & \cdots & u_{ik}(P, P) \end{bmatrix} \tag{3.17}$$

with elements $u_{ik}(l, j)$ being a $D \times D$ matrix where D denotes the feature dimensionality:

$$u_{ik}(l, j) = \psi_{ik}(\boldsymbol{\eta}^r, \boldsymbol{\eta}^r, l, j) \quad (3.18)$$

and

$$\mathbf{c}_{ik} = [c_{ik0}^T, \dots, c_{ikl}^T, \dots, c_{ikP}^T]^T \quad (3.19)$$

is composed of $P + 1$ coefficient vectors c_{ikl} ($l = 0, \dots, P$), each of which is D dimensional. On the right side of Eq. 3.16, \mathbf{v}_{ik} is a $P + 1$ dimensional block vector:

$$\mathbf{v}_{ik} = [v_{ik}(0), \dots, v_{ik}(l), \dots, v_{ik}(P)]^T \quad (3.20)$$

where $v_{ik}(l)$ is a D dimensional vector:

$$v_{ik}(l) = \psi_{ik}(\boldsymbol{\eta}^r, \boldsymbol{\sigma}_t^r - \boldsymbol{\mu}_{ik}, l, 1) \quad (3.21)$$

From Eq. 3.16, the polynomial coefficients c_{ikl} ($l = 0, \dots, P$) can be computed by inverting the matrix \mathbf{U}_{ik} . This operation is computationally expensive if the covariance matrices in ψ are full matrices. However, when the covariance matrices $\boldsymbol{\Sigma}_{ik}$ are diagonal (which is usually the case), the computational load could be significantly reduced as discussed in [Gon02].

The above describes the formulation for estimating the polynomials that are distinct for each Gaussian mixture. For other tying schemes, the extension of the above derivation is straightforward. Suppose there are K classes $\{\omega_1, \omega_2, \dots, \omega_K\}$ within which the regression polynomials of different Gaussian mixtures are shared.

The optimization of $Q_b(\lambda; \bar{\lambda})$ with respect to $\mathbf{c}_{\omega_q l}$ ($q = 1, \dots, K$) changes to:

$$\begin{aligned}
\frac{\partial Q_b(\lambda; \bar{\lambda})}{\partial \mathbf{c}_{\omega_q l}} &= \frac{\partial}{\partial \mathbf{c}_{\omega_q l}} \sum_{r=1}^R \sum_{i=1}^N \sum_{k=1}^M \sum_{t=1}^{T^r} \gamma_t^r(i, k) \cdot \log N(\mathbf{o}_t^r - \sum_{j=0}^P \mathbf{c}_{\omega_q j} (\eta^r)^j; \boldsymbol{\mu}_{ik}, \boldsymbol{\Sigma}_{ik}) \\
&= \frac{\partial}{\partial \mathbf{c}_{\omega_q l}} \sum_{r=1}^R \sum_{(i,k) \in \omega_q} \sum_{t=1}^{T^r} \gamma_t^r(i, k) \cdot \\
&\quad \left[-\frac{1}{2} \left(\mathbf{o}_t^r - \sum_{j=0}^P \mathbf{c}_{\omega_q j} (\eta^r)^j - \boldsymbol{\mu}_{ik} \right)^T \boldsymbol{\Sigma}_{ik}^{-1} \left(\mathbf{o}_t^r - \sum_{j=0}^P \mathbf{c}_{\omega_q j} (\eta^r)^j - \boldsymbol{\mu}_{ik} \right) \right] \\
&= \sum_{r=1}^R \sum_{(i,k) \in \omega_q} \sum_{t=1}^{T^r} \gamma_t^r(i, k) \cdot \boldsymbol{\Sigma}_{ik}^{-1} \cdot \left(\mathbf{o}_t^r - \sum_{j=0}^P \mathbf{c}_{\omega_q j} (\eta^r)^j - \boldsymbol{\mu}_{ik} \right) \cdot (\eta^r)^l = 0 \\
&\qquad\qquad\qquad l = 0, 1, \dots, P \tag{3.22}
\end{aligned}$$

where $(i, k) \in \omega_q$ denotes the k th Gaussian mixture in state i that belongs to the tying class ω_q .

Similarly, the shared polynomial coefficients satisfy:

$$\begin{aligned}
&\sum_{j=0}^P \left[\sum_{r=1}^R \sum_{(i,k) \in \omega_q} \sum_{t=1}^{T^r} \gamma_t^r(i, k) \cdot \boldsymbol{\Sigma}_{ik}^{-1} \cdot (\eta^r)^{j+l} \right] \mathbf{c}_{\omega_q j} \\
&= \sum_{r=1}^R \sum_{(i,k) \in \omega_q} \sum_{t=1}^{T^r} \gamma_t^r(i, k) \cdot \boldsymbol{\Sigma}_{ik}^{-1} \cdot \left(\mathbf{o}_t^r - \boldsymbol{\mu}_{ik} \right) \cdot (\eta^r)^l \\
&\qquad\qquad\qquad l = 0, 1, \dots, P \tag{3.23}
\end{aligned}$$

and $\mathbf{c}_{\omega_q l}$'s can be solved accordingly.

In [DDA04a] and [ZA02], the phase relationship between the clean and noisy speech is investigated. In particular, the phase information is incorporated into the minimum mean square error estimation of clean speech features in the log-Mel power domain in [DDA04a] which achieves impressive performance improvements. The consequent non-Gaussian probability density function in the estimator is approximated by single-point, second-order Taylor series expansion. Unlike the phase-sensitive model in [DDA04a], phase is not explicitly taken into account Eq. 3.1. However, phase, which is also a function of utterance SNR, is implicitly

represented in the nonlinear bias which is, in turn, approximated by regression polynomials. Compared with the phase-sensitive model which uses a single expansion point for all clean speech mixture components, the regression polynomials can be considered as expansion at a particular Gaussian mixture mean (mixture specific polynomials) or averaged Gaussian mixture means (tied polynomials) which are also optimized iteratively under ML criterion. In addition, the proposed feature compensation algorithm is less computationally expensive than the phase-sensitive model.

3.2 Utterance SNR Estimation

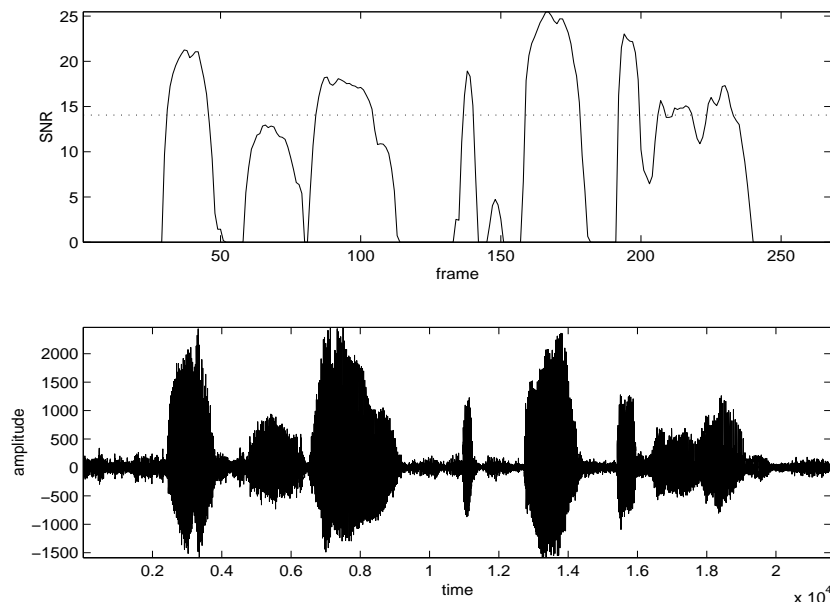


Figure 3.1: Estimated frame-wise SNR (solid line in top panel), estimated utterance SNR (dotted line in top panel) and waveform (bottom panel) of the utterance “43o6571” labeled 15 dB in the Aurora 2 database.

The estimation of the utterance SNR is based on the minimum statistics tracking algorithm from [Mar01] which was described in Chapter 2. It is computed

as the average of the non-zero frame SNRs of the utterance, as the dotted line shown in Fig. 3.1. The noisy speech signal in the figure is generated as a 15 dB signal by the Aurora 2 database. The estimate by minimum statistics tracking is very close to the original labeling of the database.

The reason for employing utterance SNR instead of frame SNR in the proposed feature compensation algorithm is that the polynomial approximated bias is only meaningful with respect to the clean speech. Frame SNR reflects the power variation of different portions within the utterance. For example, for a clean speech signal, no compensation of the feature is needed but its frame SNRs still can vary in a wide range. On the other hand, the averaged utterance SNR reflects how the overall signal is corrupted by noise.

3.3 Experimental Results

3.3.1 Experimental Conditions

The proposed feature compensation algorithm by polynomial regression of utterance SNR is trained and tested on the connected digits from the Aurora 2 [HP00] and the German part of the Aurora 3 databases.

For the Aurora 2 database experiments, there are 8440 clean utterances from 55 male and 55 female adult speakers in the clean training set from which the acoustic HMMs are trained. Speech data in testing sets A and B are used for evaluation. There are eight types of background noise in the Aurora 2 database, which are subway, babble, car and exhibition noise in Set A and restaurant, street, airport and station noise in Set B. Noisy speech data are generated by artificially adding the noise signals at a variety of SNR levels. Six SNR conditions are evaluated in the test which are clean, 20 dB, 15 dB, 10 dB, 5 dB and 0 dB. For the German part of the Aurora 3 database, the acoustic HMMs are trained and

tested for all three conditions, namely WM, MM and HM.

Fig. 3.2 shows the training and recognition scheme employed in the experiments. For each utterance, Mel-Frequency Cepstral Coefficients (MFCC) features are extracted and the utterance SNR is estimated from the speech signal. The frame length is 25 ms and the frame shift is 10 ms. The speech feature for each frame contains 12 static MFCCs (excluding C0) plus log energy (E) and their first and second order derivatives. Therefore, there are 39 components in each feature vector. In the training stage, the regression polynomials are estimated from the adaptation data. In the recognition stage, for each frame in the utterance, the polynomial approximated bias is computed based on the utterance SNR and removed from the noisy MFCC features. Cepstral features are then decoded using the Viterbi decoding network with original acoustic HMMs. The HMMs adopt a

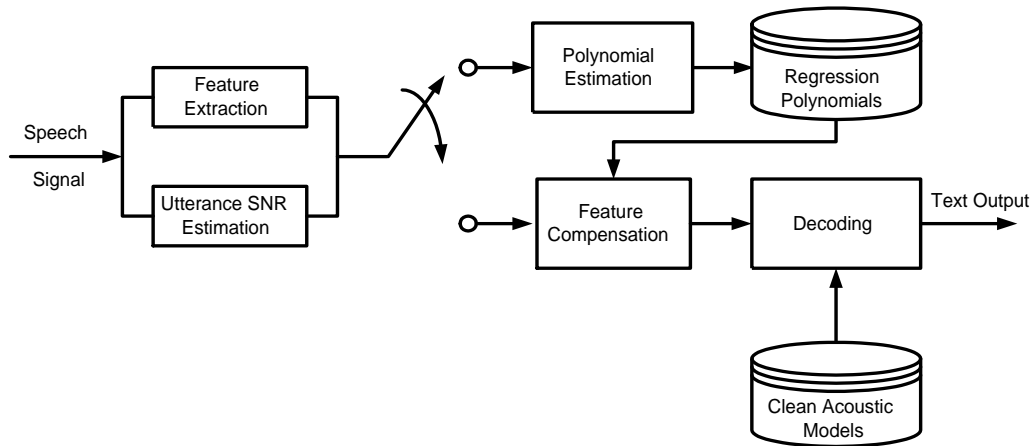


Figure 3.2: Training and recognition scheme

left-to-right topology and are word-based models with 16 emission states for each digit, 3 states for the silence model and 1 state for the short pause model. There are 3 mixtures in each state of digit models and 6 mixtures for silence and short pause models. All the Gaussian mixtures have diagonal covariance matrices. The above setup follows the Aurora 2 and 3 specifications.

3.3.2 Distribution of Utterance SNRs

Since estimated utterance SNRs are used in the algorithm, it is interesting to compare them with the original labeled SNR provided by the database. Figures 3.3, 3.4 and 3.5 show histograms of utterance SNRs in Set A estimated by the minimum statistics tracking algorithm at six SNR conditions in the Aurora 2 database. Small variances (2-3 dB) are observed in the estimated utterance SNRs for each condition in those figures. Since frames with SNR below 0 dB are not included in the utterance SNR calculation and bias exists in the minimum statistics tracking algorithm, the utterance SNRs estimated are not exactly the same as the labeled SNRs. The calibration with the database labeling shows a good utterance SNR estimation by the minimum statistics tracking algorithm. Figures 3.6 to 3.8 indicate the utterance SNR distributions of the training and testing sets under three experimental conditions of the German part of the Aurora 3 database. For the proposed feature compensation algorithm, the absolute utterance SNR accuracy is not critical while the consistency of utterance SNR estimation between training and recognition is.

3.3.3 Regression Polynomial Orders

As stated before, the nonlinear bias can be approximated by polynomials of different orders. The higher the order of the polynomials, the smaller the approximation errors. However, higher order polynomials can also result in overfitting and more parameters to estimate. With limited learning data, non-reliable parameter estimation may occur. Table 3.1 shows the FC performance with respect to different regression polynomial orders. The state-tied regression polynomials are estimated from 300 utterances from Sets A and B (each) in the Aurora 2 database. For the special case when the polynomial order is 0, it is equal to a state-based SNR independent bias removal method. From the table, performance

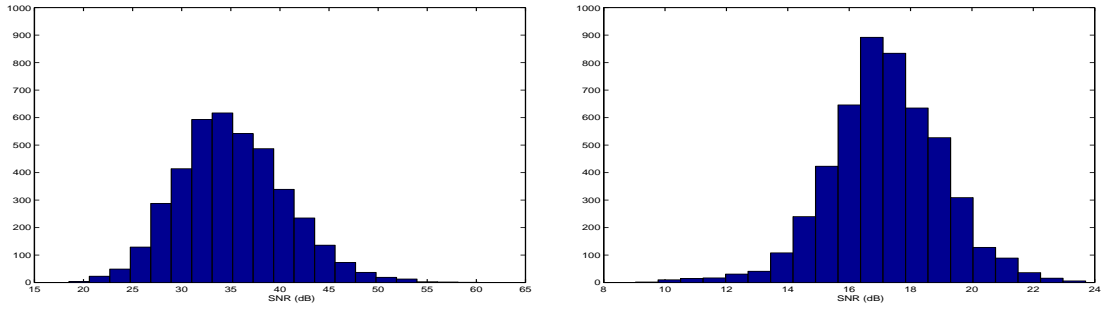


Figure 3.3: Histograms of estimated utterance SNRs labeled as clean (left) and 20 dB SNR (right) data in Set A of Aurora 2 database.

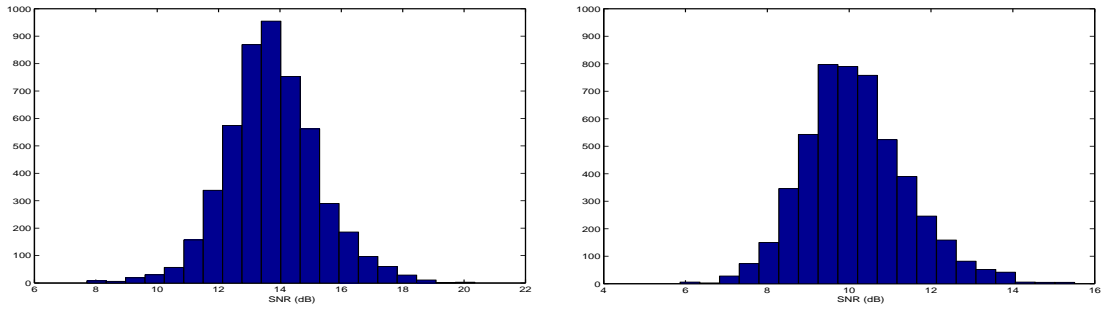


Figure 3.4: Histograms of estimated utterance SNRs labeled as 15 dB (left) and 10 dB SNR (right) data in Set A of Aurora 2 database.

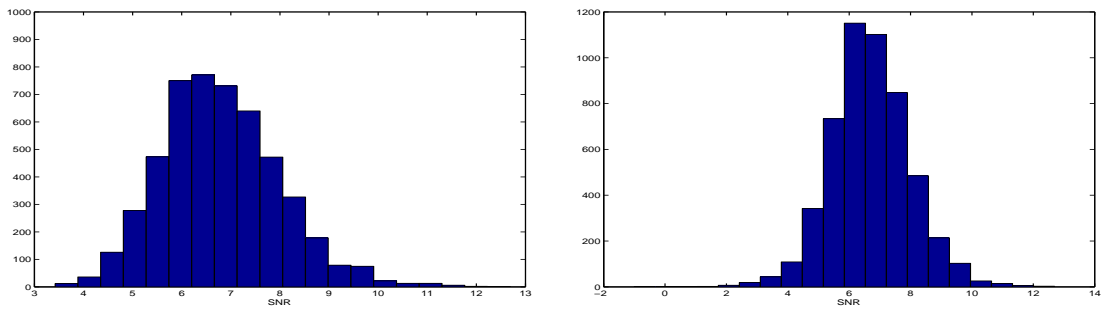


Figure 3.5: Histograms of estimated utterance SNRs labeled as 5 dB (left) and 0 dB SNR (right) data in Set A of Aurora 2 database.

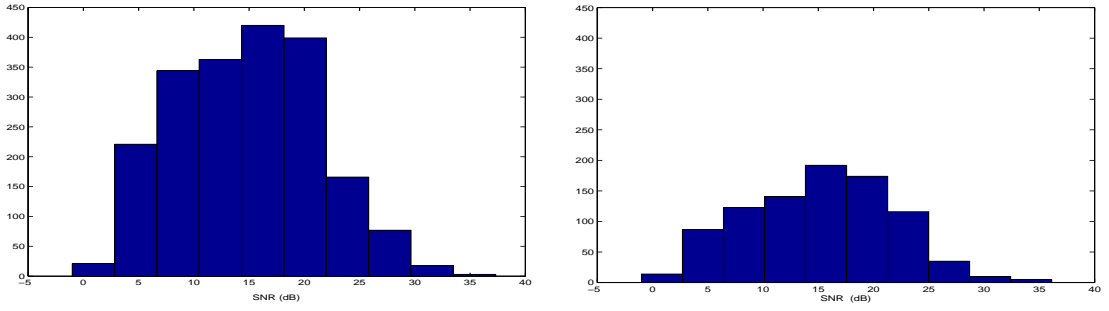


Figure 3.6: Histograms of estimated utterance SNRs in training (left) and testing (right) of the well-matched condition of the Aurora 3 German database.

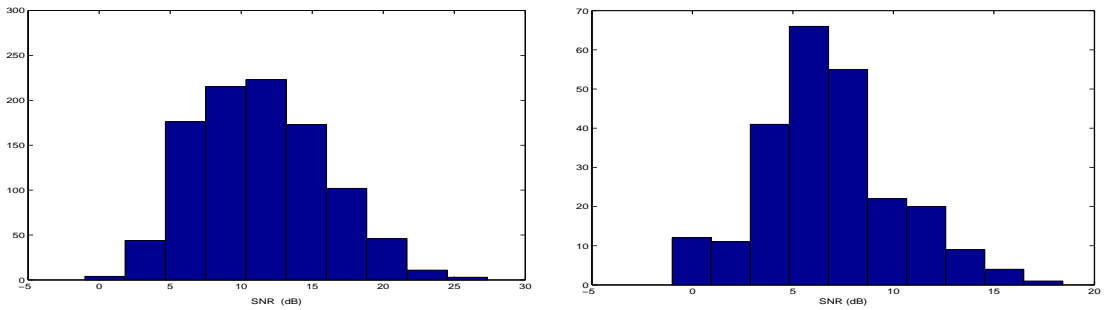


Figure 3.7: Histograms of estimated utterance SNRs in training (left) and testing (right) of the medium-mismatched condition of the Aurora 3 German database.

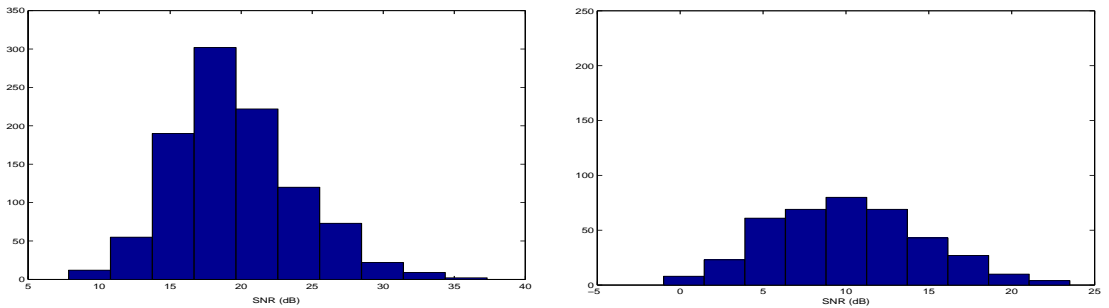


Figure 3.8: Histograms of estimated utterance SNRs in training (left) and testing (right) of the high-mismatched condition of the Aurora 3 German database.

improves when the polynomial order increases from 0 to 2. Third order polynomials give slightly worse performance due to a larger number of parameters to be estimated. Considering the goodness of approximation and the number of parameters, 1st and 2nd order polynomials are chosen for the experiments.

Data Sets	Polynomial Orders			
	0	1	2	3
Set A	83.0	83.5	83.8	83.7
Set B	83.1	83.5	83.9	83.4

Table 3.1: Average performances of Sets A and B in Aurora 2 with respect to regression polynomial orders. The polynomials are state tied and estimated from 300 utterances.

3.3.4 Estimated Regression Polynomials

The effectiveness of regression polynomial fitting to noisy speech data can be observed from the change of the average likelihood of the adaptation utterances. The effect of the number of EM iterations is illustrated in Figures 3.9 and 3.10, the figures show average likelihood of 50 utterances using 1 to 10 EM iterations in the right panels for airport and station noise, respectively. The corresponding polynomials of the energy component of the features are illustrated on the left. The polynomials are shared globally and zero polynomials are chosen as the initial conditions under which no feature compensation is performed. From the figures, a significant increase of the average likelihood can be observed after the 1st iteration which is attributed to the feature compensation beginning to take effect and the regression polynomials change from zero polynomials to non-zero polynomials. Afterwards, the average likelihood increases monotonically until it converges to

a stationary point. The monotonicity and convergence is guaranteed by the EM algorithm. Fast convergence of the FC algorithm can be observed in the figures. Typically, the algorithm converges after 3 or 4 iterations. The increase of the likelihood indicates a better fit of the features to the original models after the compensation.

The effect of the number of utterances is illustrated in Figures 3.11 and 3.12 where the estimated global polynomials after 6 EM iterations are demonstrated for a variety of feature components with different numbers of adaptation utterances with car and subway background noise. The shape of the polynomials varies dramatically when the number of adaptation utterances is small. As the adaptation data size grows, the polynomials become stable since the statistics collected for polynomial estimation become more robust.

As observed from the estimated regression polynomials, biases exist under clean conditions (e.g. SNR > 20dB). This will degrade the performance for clean speech to a certain degree. Table 3.2 shows the baseline (0 utterances) and FC performance averaged over all the clean conditions in the Aurora 2 database. 10, 100 and 200 adaptation utterances are utilized which tie the polynomials at the global, state and mixture levels, respectively. Compared with the baseline, the FC algorithm degrades recognition performance. Therefore, in the decoding stage of the following experiments on the Aurora 2 database, no compensation is performed on the speech features with SNRs higher than 20 dB.

Number of utterances	0	10	100	200
Accuracy(%)	99.0	98.8	97.4	97.6

Table 3.2: Word recognition accuracy averaged over all clean conditions in the Aurora 2 database. Feature compensation is performed with 10, 100 and 200 utterances.

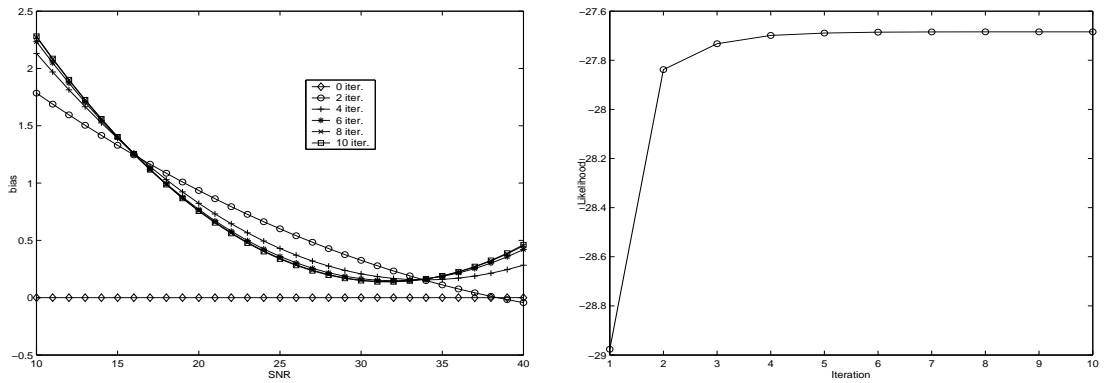


Figure 3.9: The left panel shows estimated global polynomials as a function of SNR and iteration number. The right panel shows average likelihood as a function of the number of EM iterations. Both panels use the energy feature component (E) for the airport noise data.

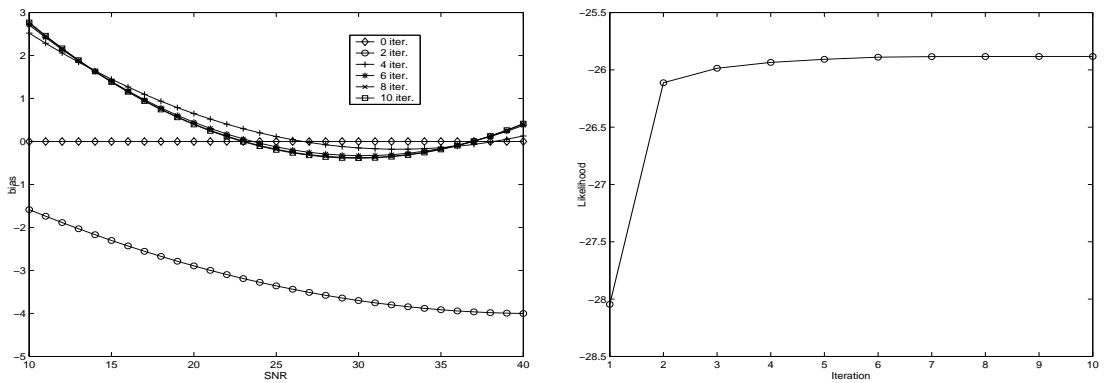


Figure 3.10: The left panel shows estimated global polynomials as a function of SNR and iteration number. The right panel shows average likelihood as a function of the number of EM iterations. Both panels use the energy feature component (E) for the station noise data.

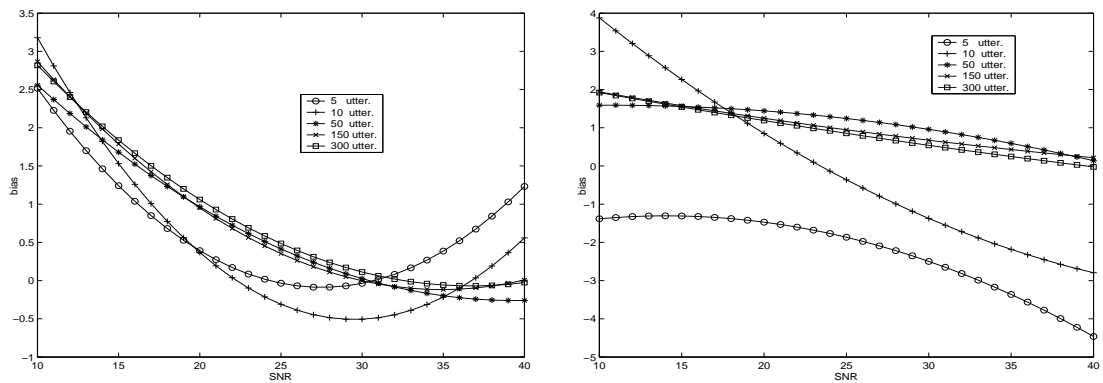


Figure 3.11: The left and right panels show estimated global polynomials as a function of SNR and number of utterances for energy (E) and the first cepstral coefficient (C1), respectively, under car noise. The number of EM iterations is fixed at 6.

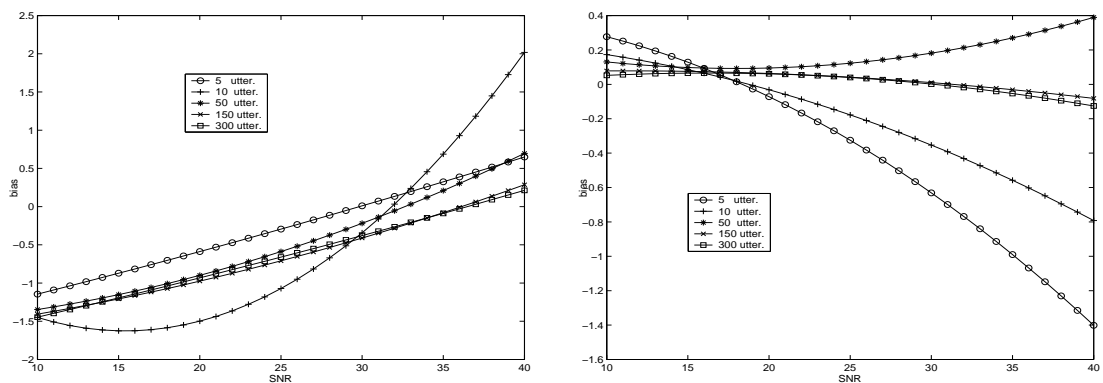


Figure 3.12: The left and right panels show estimated global polynomials as a function of SNR and number of utterances for the 6th (C6) and 10th (C10) cepstral coefficients, respectively, under subway noise. The number of EM iterations is fixed at 6.

3.3.5 Recognition Results

3.3.5.1 The Aurora 2 database

Comparative experiments are performed using the proposed feature compensation (FC) algorithm and the maximum likelihood linear regression (MLLR) algorithm. Different amounts of adaptation data are used ranging from 5 to 300 utterances for each type of noise. Adaptation utterances are randomly chosen from Sets A and B for each type of noise and are excluded from the testing. The average length for each utterance is 4.6 digits. Regression polynomials are noise dependent and depending on the adaptation data size, they are tied at different levels of granularity: for adaptation sizes of 5, 10 and 20 utterances, polynomials are tied globally; for adaptation sizes of 50, 100 and 150 utterances, polynomials are tied within states; for adaptation sizes of 200, 250 and 300 utterances, polynomials are Gaussian mixture specific. The transformations of MLLR are also tied in a similar manner. In the experiments, two MLLR schemes are tested – the transformation matrices are estimated by pooling all the adaptation data across SNR levels, and by distinct SNR levels, respectively. For the second case, considering the SNR distinction and amount of data for robust estimation, three SNR clusters are employed: clean and 20 dB, 15 dB and 10dB, and 5 dB and 0 dB. Different MLLR transformation matrices are estimated for three different SNR clusters and used to evaluate the corresponding SNR level utterances. For both MLLR implementations, the transformation matrices are 3-block diagonal. The regression polynomial order is set to 2.

Performance of FC and MLLR for each type of noise is presented in Table 3.3 for Set A, and in Table 3.4 for Set B. In the tables, MLLR1 denotes the case where MLLR transformation matrices are estimated by all of the adaptation data and MLLR2 denotes the SNR-cluster specific case. The results are averaged

across the 6 SNR levels (clean, 20 dB, 15 dB, 10 dB, 5 dB and 0 dB) tested.

In most cases, MLLR2 outperforms MLLR1 when the number of adaptation utterances is small, e.g. 5 to 150 utterances, and not as good as MLLR1 when the adaptation data set has more than 200 utterances. This is due to the number of parameters to be estimated for the transformation matrices. When the adaptation data are limited, transformation matrices are tied in a larger scale, e.g. global or within states. On the other hand, mixture specific matrices are employed when the number of utterances is higher than 200. In this case, the number of parameters to be estimated becomes larger and the transformation matrices are estimated less robustly.

Both MLLR and FC obtain improved performance with the growth of the adaptation data size. Compared with MLLR1, FC results in 17.3%, 26.9%, 8.4% and 14.2% WER reduction for subway, babble, car and exhibition noise. On average, FC outperforms MLLR1 by 16.7% for Set A. Compared with MLLR2, FC results in 17.2%, 30.1%, 2.9% and 16.5% WER reduction for the four types of noise in Set A, and overall improvement is 16.5%. Considering the restaurant, street, airport and station noise in Set B, FC obtains 25.8%, 20.1%, 23.4% and 20.6% WER reduction over MLLR1, and 21.6%, 16.1%, 18.3% and 12.3% over MLLR2. On average, FC outperforms MLLR1 by 20.5% and MLLR2 by 14.6% for Set B over all conditions. Among the eight types of noise in Sets A and B, FC achieves the highest improvement over MLLR in babble noise and least improvement with car noise.

Tables 3.5 and 3.6 summarize the performance of FC and MLLR under each SNR condition for Sets A and B, respectively. For each set, the results are averaged over the 4 types of noise in the set. On average, FC achieves a WER reduction of 33.1% for MLLR1, 23.6% for MLLR2 for Set A, and 34.5% for MLLR1, 21.4% for MLLR2 for Set B. Among the six SNR levels, FC results

in best performance for SNRs arranging from 10 to 20 dB, compared to both MLLR1 and MLLR2. In Tables 3.5 and 3.6, no compensation is performed on speech features for utterances with SNRs higher than 20 dB.

3.3.5.2 The Aurora 3 database

For the German part of the Aurora 3 database, 10, 40 and 70 utterances are randomly chosen for adaptation and excluded from each testing condition. Table 3.7 shows the performance of MLLR performance and the proposed FC algorithm. As before, the FC algorithm outperforms MLLR. To investigate whether dynamically grouping the Gaussians in the model set is beneficial, a regression tree is constructed for each condition (WM, MM and HM) with each mixture being a leaf utilizing the originally trained acoustic HMMs. Instead of specifying static tying classes explicitly in advance as static tying strategies, a regression tree is created based on the centroid splitting algorithm using the Euclidean distance between the Gaussian mixture means as described in [YEK01]. Depending on the amount of adaptation available, a regression tree can tie parameters (transformations in MLLR or polynomials in FC) at different levels dynamically. To tie the parameters, MLLR needs node occupations as well as a minimum number of mixtures (larger than the feature vector size) within one class for matrix inversion while FC needs node occupations only. In the experiments, the node occupation threshold is set to 700 and 3-block transformation matrices are used. Table 3.8 shows the performance of MLLR (MLLR(r)) and FC (FC(r)) using the dynamic tying strategy. From the table, FC(r) gives better results than MLLR(r) and static tying FC under most cases. On average, FC(r) obtains performance improvements over the baseline (no adaptation) by 12.4%, 12.1% and 46.1% for WM, MM and HM conditions, respectively. The most significant improvement is for the HM condition where the original acoustic models are trained by rela-

tively clean speech and a large mismatch exists between the training and testing environments. Although the FC algorithm is originally used to approximate the bias between clean and noisy speech, improvements still observed under MM and WM conditions where the original HMMs are trained with noisy speech. The improvements of FC(r) over MLLR(r) are 15.9%, 3.0% and 14.6% for the three testing conditions, respectively.

3.4 Summary

A feature compensation algorithm on the basis of polynomial regression of utterance SNRs for noise robust speech recognition is investigated in this chapter. In this algorithm, a set of polynomials regressed on utterance SNRs are utilized to approximate the bias between the clean and noisy speech features. The bias is used to compensate noisy features with respect to clean acoustic models. The maximum likelihood estimation of the regression polynomials is provided within an EM algorithm framework. Since the compensated bias depends on utterance SNRs, the proposed algorithm can deal with the slowly-changing environments from utterance to utterance. Besides handling non-stationarity, the regression polynomials also can predict unseen environments in the training stage.

ASR experiments are performed on the Aurora 2 and the German part of the Aurora 3 databases. Using the Aurora 2 database, 8 types of noise in Sets A and B are tested with different adaptation data sizes. Experiments are designed to compare the performance of the feature compensation algorithm with two MLLR implementations: transformation matrices estimated by pooling all the adaptation data, and by distinct SNR clusters. Significant improvements over the two MLLR schemes are observed for Sets A and B. The evaluation of the algorithm on the German part of the Aurora 3 database shows improvements

under the well-matched, medium-mismatched and high-mismatched testing conditions. The most significant improvement is observed for the high-mismatched case.

Noise	Algorithm	Number of Utterances									
		0	5	10	20	50	100	150	200	250	300
subway	MLLR1	74.5	73.2	76.0	75.5	75.8	81.2	82.8	83.6	83.5	84.0
	MLLR2	74.5	70.4	77.5	77.5	78.3	81.8	81.9	82.7	83.5	82.4
	FC	74.5	79.0	78.9	79.7	81.5	84.0	84.8	85.4	87.2	87.4
babble	MLLR1	58.1	67.0	68.9	73.0	75.5	74.1	76.6	76.0	76.5	78.3
	MLLR2	58.1	70.7	64.5	69.4	74.1	73.6	74.8	75.4	76.9	75.8
	FC	58.1	71.4	71.8	74.9	81.6	83.7	84.7	85.2	85.9	86.5
car	MLLR1	70.0	70.9	70.0	73.5	75.9	77.8	78.9	80.4	79.8	80.5
	MLLR2	70.0	69.5	70.6	75.3	81.7	79.9	80.6	79.7	79.3	81.3
	FC	70.0	71.7	74.5	74.4	77.8	79.6	80.2	81.0	82.9	83.2
exhibition	MLLR1	71.0	73.3	73.9	72.2	72.9	76.9	78.5	79.3	79.5	81.0
	MLLR2	71.0	69.5	75.2	74.7	79.7	76.5	77.1	76.0	74.8	75.4
	FC	71.0	73.7	76.6	75.6	76.8	80.2	81.5	82.1	84.5	85.4

Table 3.3: Word recognition accuracy for FC and MLLR for 4 types of noise in Set A of the Aurora 2 database. MLLR1 refers to the case where the MLLR transformation matrices are estimated across all SNR levels, while MLLR2 refers to MLLR transformation matrices being SNR-cluster specific. Baseline MFCC results are presented as adaptation with 0 utterances.

Noise	Algorithm	Number of Utterances									
		0	5	10	20	50	100	150	200	250	300
restaurant	MLLR1	60.3	70.6	70.3	70.5	76.9	78.8	79.1	79.9	80.6	81.1
	MLLR2	60.3	66.3	78.2	75.2	78.7	80.1	80.6	79.9	77.3	79.6
	FC	60.3	72.0	74.1	75.6	82.9	83.5	86.6	87.1	88.2	88.4
street	MLLR1	67.8	68.7	77.1	74.4	78.8	78.3	80.1	80.2	80.7	82.3
	MLLR2	67.8	70.4	69.2	75.7	81.3	82.7	83.4	83.8	78.6	84.2
	FC	67.8	74.8	75.3	74.6	80.4	82.5	83.2	83.9	84.2	85.1
airport	MLLR1	60.9	73.8	75.3	74.2	76.1	78.7	80.5	81.1	81.9	83.1
	MLLR2	60.9	68.5	75.3	75.7	79.5	83.4	84.0	83.8	80.1	84.0
	FC	60.9	76.1	77.1	78.3	83.4	85.0	86.0	86.9	87.4	88.1
station	MLLR1	62.9	68.3	67.5	71.6	74.6	76.9	77.3	77.3	77.5	79.1
	MLLR2	62.9	71.7	75.2	74.7	69.4	80.4	81.1	80.9	75.3	80.7
	FC	62.9	71.7	76.0	76.0	79.0	81.2	82.0	82.8	83.5	84.4

Table 3.4: Word recognition accuracy for FC and MLLR on 4 types of noise in Set B of the Aurora 2 database. See Table 3.3 caption for the definition of MLLR1 and MLLR2. Baseline MFCC results are presented as adaptation with 0 utterances.

SNR	Algorithm	Number of Utterances									
		0	5	10	20	50	100	150	200	250	300
clean	MLLR1	99.0	95.9	96.5	96.5	97.0	97.1	97.2	97.6	97.7	97.2
	MLLR2	99.0	97.8	98.7	98.8	98.7	98.9	98.9	99.0	98.6	98.9
	FC	99.0	99.0	99.0	99.0	99.0	99.0	99.0	99.0	99.0	99.0
20 dB	MLLR1	95.3	92.8	94.1	93.5	94.3	95.4	95.7	95.8	95.9	96.2
	MLLR2	95.3	92.9	94.2	95.9	94.6	95.0	95.3	95.2	94.5	94.9
	FC	95.3	96.3	96.4	96.8	96.9	96.6	97.0	97.2	97.0	97.3
15 dB	MLLR1	87.5	86.6	89.0	89.4	89.5	91.8	92.8	92.8	93.1	93.6
	MLLR2	87.5	87.9	90.4	91.0	93.1	93.6	94.3	94.5	92.6	94.5
	FC	87.5	92.4	93.2	93.3	94.4	95.1	95.5	95.8	95.6	95.9
10 dB	MLLR1	67.7	72.5	76.4	78.7	78.9	82.7	84.9	85.1	85.6	87.0
	MLLR2	67.7	76.7	80.4	82.4	85.8	86.2	87.4	88.1	83.9	87.8
	FC	67.7	79.7	80.9	79.5	88.5	90.2	90.8	91.0	91.7	91.8
5 dB	MLLR1	39.5	51.9	52.6	57.7	60.4	63.7	68.8	69.8	70.3	72.8
	MLLR2	39.5	41.8	44.2	53.9	62.9	61.6	62.8	61.6	67.2	63.4
	FC	39.5	51.6	55.6	58.1	66.4	72.6	74.4	75.5	80.0	80.4
0 dB	MLLR1	17.0	26.8	24.7	25.4	30.0	34.2	36.0	37.9	36.4	39.0
	MLLR2	17.0	23.1	24.1	27.8	35.5	32.3	32.8	32.5	34.8	33.4
	FC	17.0	24.7	27.9	30.1	31.3	37.2	40.1	42.1	47.4	49.5

Table 3.5: Word recognition accuracy for FC and MLLR for each SNR level in Set A of the Aurora 2 database. See Table 3.3 caption for the definition of MLLR1 and MLLR2. Baseline MFCC results are presented as adaptation with 0 utterances.

SNR	Algorithm	Number of Utterances									
		0	5	10	20	50	100	150	200	250	300
clean	MLLR1	99.0	95.4	96.9	94.9	97.3	97.5	97.5	97.7	97.9	97.5
	MLLR2	99.0	97.6	98.6	98.8	98.7	99.0	99.0	99.1	98.2	99.0
	FC	99.0	99.0	99.0	99.0	99.0	99.0	99.0	99.0	99.0	99.0
20 dB	MLLR1	92.8	93.7	94.5	92.9	95.3	96.2	96.4	96.4	96.6	96.8
	MLLR2	92.8	93.0	94.9	93.8	95.3	96.0	95.8	95.9	95.9	95.9
	FC	92.8	96.7	97.1	97.2	96.8	97.1	97.3	97.6	97.4	97.7
15 dB	MLLR1	81.3	87.9	90.9	88.1	90.2	92.7	93.2	93.3	93.5	94.3
	MLLR2	81.3	89.3	92.6	93.4	93.4	94.9	95.1	94.9	91.6	95.2
	FC	81.3	93.0	93.0	93.6	95.1	96.0	96.0	96.2	95.9	96.4
10 dB	MLLR1	59.0	74.9	81.9	76.1	78.4	84.2	85.6	85.8	86.4	88.0
	MLLR2	59.0	78.7	85.1	86.4	86.1	89.9	90.5	90.5	82.1	90.6
	FC	59.0	80.0	80.7	82.0	90.8	91.8	92.3	92.5	92.7	92.9
5 dB	MLLR1	31.9	52.7	59.9	54.1	57.9	65.4	67.7	68.4	69.3	72.2
	MLLR2	31.9	39.1	50.5	53.9	57.8	69.1	70.8	70.0	64.1	69.4
	FC	31.9	49.2	54.5	56.7	71.8	76.7	78.0	79.2	80.9	82.0
0 dB	MLLR1	13.7	25.8	26.7	25.2	30.1	33.0	35.1	36.1	37.2	39.6
	MLLR2	13.7	17.5	27.1	26.7	32.1	41.2	42.4	42.1	35.2	42.5
	FC	13.7	20.5	24.6	28.2	34.9	37.7	44.1	46.5	49.0	51.1

Table 3.6: Word recognition accuracy for FC and MLLR for each SNR level in Set B of the Aurora 2 database. See Table 3.3 caption for the definition of MLLR1 and MLLR2. Baseline MFCC results are presented as adaptation with 0 utterances.

Conditions	Algorithm	Number of Utterances			
		0	10	40	70
HM	MLLR	74.3	81.6	83.6	85.8
	FC	74.3	78.8	85.7	86.6
MM	MLLR	79.1	80.2	79.7	81.1
	FC	79.1	78.3	81.5	84.4
WM	MLLR	90.6	90.6	88.8	90.9
	FC	90.6	91.2	91.0	91.4

Table 3.7: Word recognition accuracy for FC and MLLR of static tying schemes under high-mismatched (HM), medium-mismatched (MM) and well-matched (WM) conditions of the German part of the Aurora 3 database. Baseline MFCC results are presented as adaptation with 0 utterances.

Conditions	Algorithm	Number of Utterances			
		0	10	40	70
HM	MLLR(r)	74.3	80.7	84.8	86.0
	FC(r)	74.3	82.8	87.2	88.4
MM	MLLR(r)	79.1	80.2	80.8	81.3
	FC(r)	79.1	78.1	81.7	84.1
WM	MLLR(r)	90.6	89.1	90.4	91.0
	FC(r)	90.6	91.4	91.9	92.0

Table 3.8: Word recognition accuracy for FC and MLLR of dynamic tying schemes under high-mismatched (HM), medium-mismatched (MM) and well-matched (WM) conditions of the German part of the Aurora 3 database. Baseline MFCC results are presented as adaptation with 0 utterances.

Part II

Speaker Robust Speech Recognition

This part of the dissertation is concerned with speaker robust issues in automatic speech recognition. Similar to noisy environments, the variation among speakers also leads to mismatch between training and testing acoustic models. As discussed in Chapter 1, the mismatch is due to the diversity of speech patterns which originate from speakers' vocal apparatus variations. In this situation, speaker adaptation techniques play an important role in mitigating the mismatch or normalizing the variation. Typically, a certain amount of adaptation data from the speaker is required by the system to learn his/her acoustic characteristics before recognition. There are two ways to perform speaker adaptation: supervised or unsupervised. In supervised adaptation, the speaker is asked to read some predefined texts and the original acoustic models are adapted accordingly. Therefore, transcription is known for the adaptation speech signals. Unsupervised adaptation, on the other hand, has no such requirements and hence is more error-prone. The work in this part is primarily focused on supervised adaptation.

Firstly, in Chapter 4, an adaptation text design algorithm based on Kullback-Leibler (KL) measure is introduced. It allows a designer to predefine a target distribution of speech units and selects texts whose speech unit distribution most resembles the target in the sense of minimized KL measure. Secondly, in Chapter 5, a fast speaker adaptation approach via formant-peak alignment is investigated. The algorithm investigates, in the discrete frequency domain, the relationship between frequency warping in the front-end feature domain and linearity of the corresponding transformation in the back-end model domain. The adaptation is conducted by performing the transformation of means deterministically based on the linear relationship investigated and estimating biases and variances statistically based on the Expectation-Maximization algorithm. Finally, in Chapter 6, a robust maximum likelihood linear regression technique via weighted model averaging is discussed. A variety of transformation structures are studied and a

general form of maximum likelihood estimation of the structures is given. The minimum description length (MDL) principle is applied to account for the compromise between transformation granularity and descriptive ability regarding the tying patterns of structured transformations with a regression tree. Weighted model averaging across the candidate structures is then performed based on the normalized MDL scores.

CHAPTER 4

Adaptation Text Design Based on the Kullback-Leibler Measure

Adaptation text design is the first step of speaker adaptation. Obviously, the less text the speaker has to read, the better. In some cases, speech unit (phoneme or sub-word unit) models may not be adapted properly due to limited adaptation data. For example, some phonemes occur more frequently than others. This unbalanced phoneme distribution can be problematic for system adaptation. Therefore, for supervised speaker adaptation, it is important to design adaptation texts that possess desired phoneme (or other units) distribution. There are several methods to tackle this problem such as those reported in [SB97], [SWL99] and [RV99]. Almost all the proposed methods are add-on procedures. Namely, given a score to each sentence to be selected, choose the sentence with the best score and add it to the list. The procedure is repeated until a certain criterion is met. Different approaches define different scores which reflect the degree of satisfaction to one's problem. In [RV99], the score is defined as the frequency difference between the entire corpus and the current sentence set while in [SWL99], a normalized inner product is used. Considering the text selection problem, phoneme frequency is certainly the most useful information which is exploited in the above two deterministic methods. A more complicated model-based strategy in [SB97] makes strong assumptions on the model structure and ties the model to a greedy algorithm. In this case, the rank of the design matrix and its cardinality

are chosen as the score for selection.

In this chapter, we will discuss an approach to design adaptation text efficiently and flexibly. The algorithm enables the designer to predefine the desired phoneme distribution as well as the size of the text. During the design process, the algorithm selects from a large text pool the sentences according to the criterion of minimum Kullback-Leibler (KL) measure [FRT97].

4.1 Kullback-Leibler (KL) Measure

The KL measure is a well-know measure in statistics which describes similarity or "closeness" between two probability distributions [Kul59].

In the discrete case, it is defined as

$$I(\mathbf{p} \parallel \mathbf{p}^0) = \sum_{k=1}^n p_k \log \frac{p_k}{p_k^0} \quad (4.1)$$

where \mathbf{p}^0 and \mathbf{p} are two probability distributions. From the statistical point of view, the KL measure is the expected logarithm of the likelihood ratio. (By convention, we let $0 \log \frac{0}{p} = 0$.) \mathbf{p}^0 can be considered the true distribution while \mathbf{p} is the one used to approximate it. In Eq. 4.1, n is the number of element events in the discrete probability space which, in our application, is the number of phonemes.

It is not difficult to prove that the KL measure has the following properties:

- 1) $I(\mathbf{p} \parallel \mathbf{p}^0) \geq 0$
- 2) $I(\mathbf{p} \parallel \mathbf{p}^0) = 0 \iff \mathbf{p} = \mathbf{p}^0$

The measure we use here is a one-way deviation measure which is also known as the Kullback-Leibler divergence or measure of cross-entropy. In optimization problems, the objective function $I(\mathbf{p} \parallel \mathbf{p}^0)$ is minimized with respect to \mathbf{p} to get

the best approximation to the true distribution. In special cases, when \mathbf{p}^0 is uniformly distributed, the minimization of cross-entropy leads to the maximum entropy.

\mathbf{p}^0 is a predefined ideal phoneme probability distribution which usually assumes uniform or other task-specific distributions. \mathbf{p} is the practical phoneme distribution one obtains during the design process. The algorithm is to choose a phoneme probability distribution that is “closest” to the desired target in the discrete probability distribution space under the KL measure.

4.2 Adaptation Text Design Algorithm

In order to design satisfactory text for speaker adaptation, we want to choose sentences from a large corpus. The goal is to use the text to cover as many phonemes as possible in a desired way. The first goal is not difficult to achieve since in most cases the entire phoneme set can be covered by a reasonable number of sentences. The second goal is more challenging. For some adaptation text, phonemes are distributed unevenly which results in unbalanced data for different phonemes. The algorithm described here solves the following problem: given the size of the text, we choose sentences that cover all the phonemes in the phoneme set while maintaining a desired phoneme distribution.

Let $S = \{s_1, s_2, \dots, s_M\}$ denote the text corpus from which we select adaptation sentences where s_i stands for the i th sentence and there are a total of M sentences. Suppose we need to select N sentences from the corpus as adaptation texts which are denoted by $A_N = \{s'_1, s'_2, \dots, s'_N\}$ where $s'_i \in S$ and $N \leq M$.

We want to select those N sentences whose phoneme distribution is the “clos-

est” to the prior one. That is,

$$\begin{aligned}
 A_{N,opt} &= \operatorname{argmin}_{A_N \subseteq S} I(\mathbf{p}(A_N) \parallel \mathbf{p}^0) \\
 &= \operatorname{argmin}_{A_N \subseteq S} \sum_{k=1}^n p_k(A_N) \log \frac{p_k(A_N)}{p_k^0}
 \end{aligned} \tag{4.2}$$

where $\mathbf{p}(A_N)$ denotes that the phoneme distribution is a function of the N sentences selected from the text pool.

A straightforward way to solve the above optimization problem is to traverse all the N -sentence selection cases in the whole M -sentence text pool. But this process is computationally expensive. If we define the calculation of phoneme statistics as one computation unit, the above process requires $\binom{M}{N}$ such computation units. Finally, we have to make a comparison and get the best choice. Even for relatively small numbers of N and M , the above computation requirement is very high (e.g. when $N=20$, $M=500$, $\binom{M}{N} \approx 2.7 \times 10^{35}$).

Here we propose a quasi-optimal heuristic design method which gives good results while keeping the computational load reasonable.

Fig. 4.1 provides the details of the algorithm. Starting from the empty set, we add one sentence at a time until the N -sentence requirement is reached. Each time, we select a sentence so that the newly formed sentence set has the minimum KL measure between its phoneme distribution and the prior one.

The computational complexity of the algorithm is in the order of $O(NM)$ (for $N=20$, $M=500$, $O(NM) = O(10^4)$) in terms of the computation unit mentioned above. This is affordable even for relatively large N and M . On a Pentium IV 933MHz processor with 256M of memory, it took about 11 minutes for the selection of 40 sentences from the 450-sentence TIMIT corpus and about 25 minutes for the 100 sentences case. Furthermore, it is easy to observe that the resulting

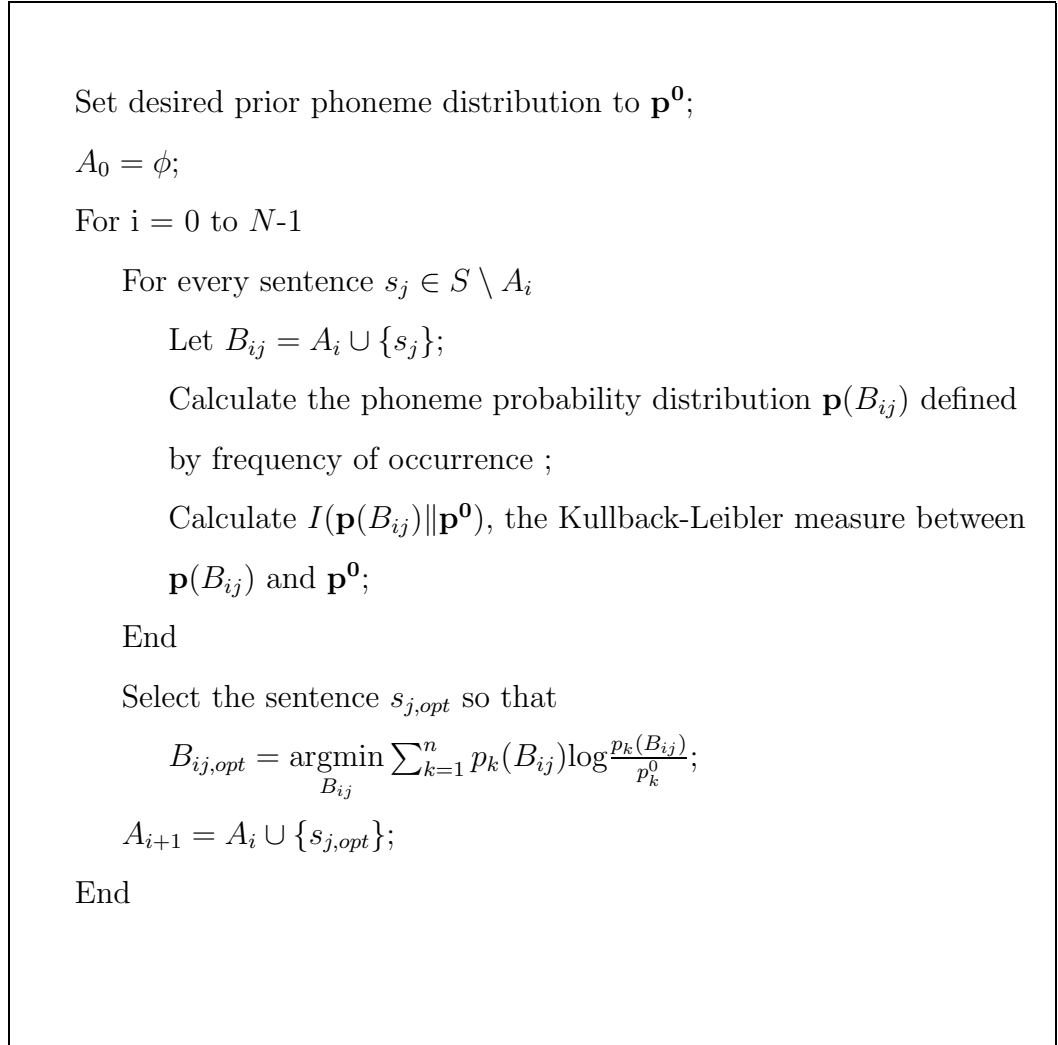


Figure 4.1: Heuristic text selection algorithm based on the minimum KL measure

adaptation text sets have the following property:

$$A_0 \subset A_1 \subset A_2 \subset \dots \subset A_N \quad (4.3)$$

This property implies a convenient way to get a new set of text from an existing one. In fact, for $L \leq N$, A_L can be constructed by choosing the first L sentences from A_N following the sequential generation order described by the algorithm.

4.3 Experimental Results

We choose the phoneme set in the BEEP dictionary provided by HTK. There are 45 phonemes in this set (including a short pause symbol). In the design experiment, we only use the first 44 phonemes, as shown in Table. 4.1, for computing the statistics and ignore the short pause symbol which is not meaningful in this design process.

EY	AA	AE	OW	IY	OH	EH	AO	AY
AH	AW	IA	UW	ER	UH	OY	UA	EA
AX	IH	Z	K	N	D	V	B	S
NG	M	T	SH	R	L	HH	G	JH
ZH	P	Y	TH	CH	F	W	DH	

Table 4.1: 44 phonemes used in the adaptation text design.

Our text pool comes from TIMIT’s 450 SA sentences which includes phonetically-compact sentences designed to provide a good coverage of phonemes and phonetic contexts.

Figs. 4.2 and 4.3 illustrate the phoneme distributions if we randomly select 20 or 40 sentences from the corpus. The horizontal axis indices from left to right are phonemes in the row order of Table. 4.1 (same order for following figures). Gener-

ally speaking, phoneme distributions from these sentences are highly unbalanced. Some of the phonemes appear much more frequently than others.

Figs. 4.4 and 4.5 show the experimental results for the distributions of the 44 phonemes from 10 and 40 sentences selected from the 450 TIMIT SA sentences by minimizing the KL measure with the prior phoneme distribution set to uniform. As mentioned before, under this condition, it amounts to maximum entropy optimization. The figures show a relatively flatter distribution in those sentences chosen by the minimum KL measure. As shown, some phonemes have higher probability than others because they occur more frequently in words. The distributions of vowels and consonants within words may be considered as constraints in this optimization problem. As the number of sentences grows, the advantage of using this algorithm is not as pronounced. This is illustrated in the 100-sentence case in Fig. 4.6.

In the above experiments, the prior phoneme probability distributions are assumed to be uniform. However, the choice of target distribution is not necessarily limited to the uniform distribution. In Fig. 4.7 we define the prior phoneme distribution \mathbf{p}^0 to the form that the five phonemes (/AX/, /IH/, /N/, /T/ and /L/) with the highest probabilities in the uniform case have one third the probabilities of the rest. By examining the result, one can observe that the distribution is flatter. If the size of text pool is larger, the result might be better since we have more sentences to choose from. In fact, the choice of \mathbf{p}^0 can allow for the control of each phoneme distribution by giving it a proper probability or weight.

In a specific speech recognition task, the text corpus defining the task also has certain specific phoneme distribution as discussed in [SWL99]. In order to design matched adaptation text, we have to select those sentences whose phoneme distribution resembles the one of the task. Our algorithm has clear advantages in this case where \mathbf{p}^0 is defined to be specific phoneme distribution from the task.

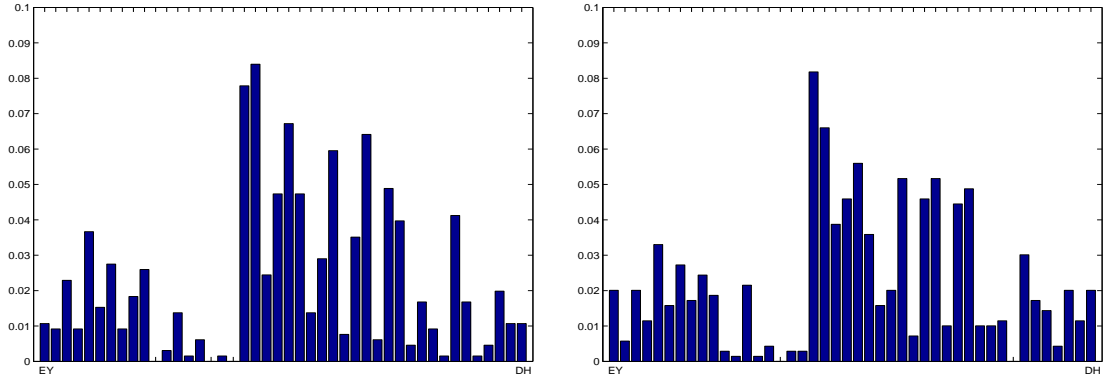


Figure 4.2: Two instances of phoneme distributions of 20 randomly selected sentences from TIMIT.

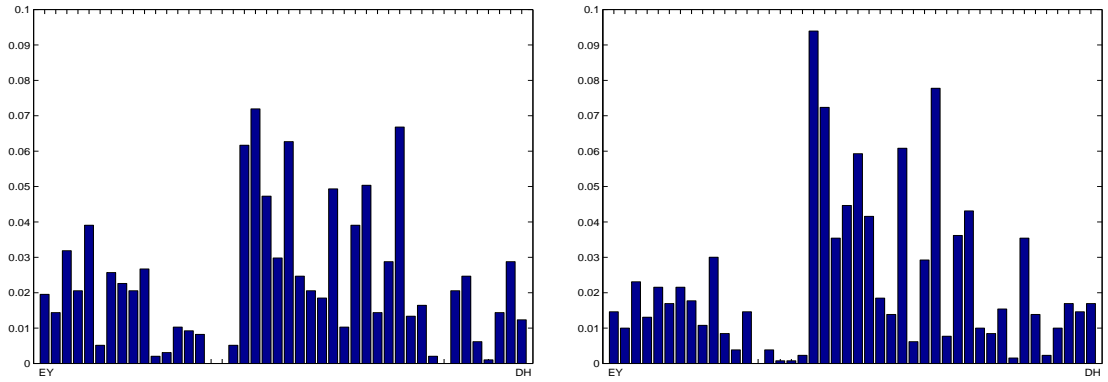


Figure 4.3: Two instances of phoneme distributions of 40 randomly selected sentences from TIMIT.

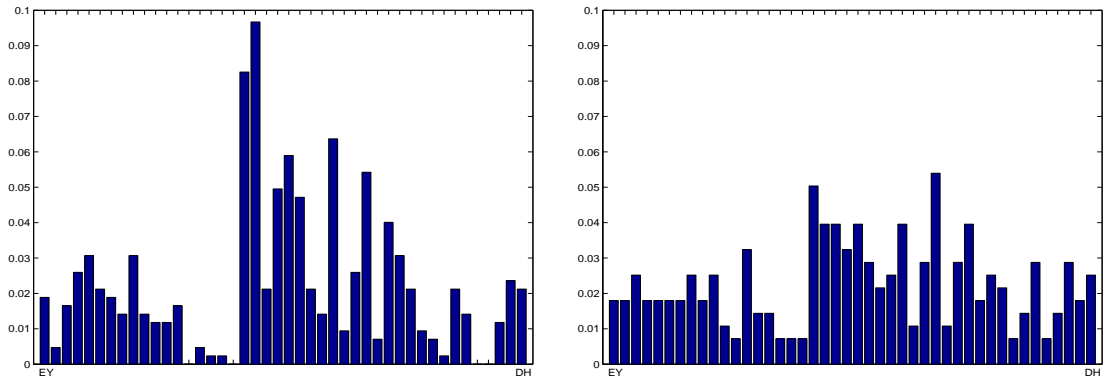


Figure 4.4: Comparison of the phoneme distributions of 10 sentences chosen randomly(left) and using the minimum KL measure(right) from TIMIT.

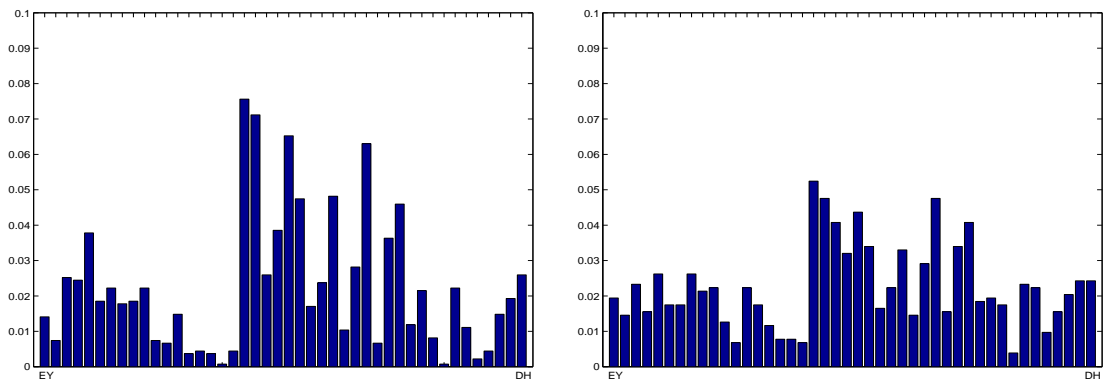


Figure 4.5: Comparison of the phoneme distributions of 40 sentences chosen randomly(left) and using the minimum KL measure(right) from TIMIT.

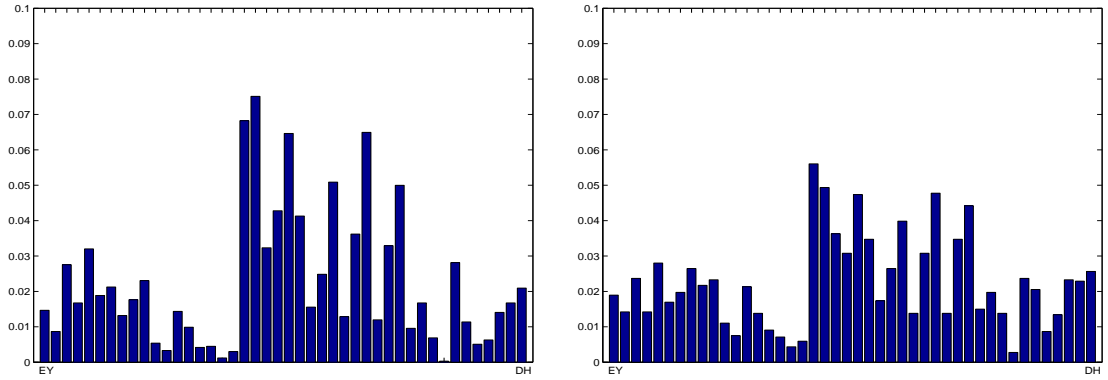


Figure 4.6: Comparison of the phoneme distributions of 100 sentences chosen randomly(left) and using the minimum KL measure(right) from TIMIT.

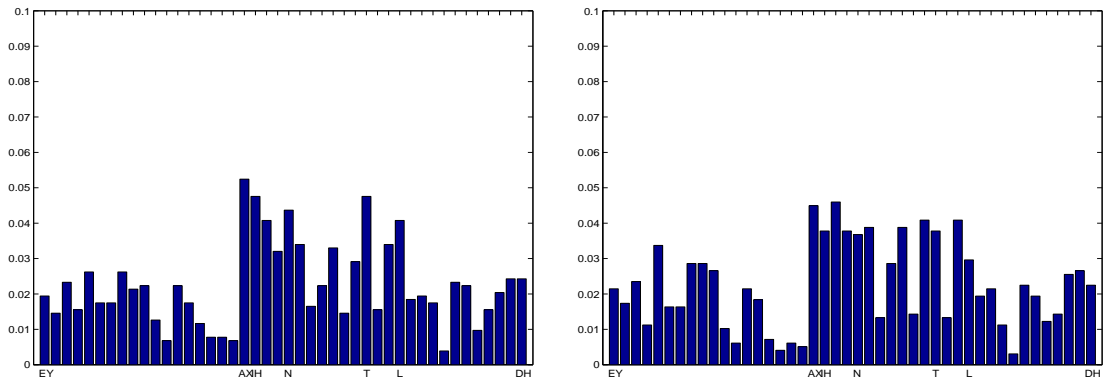


Figure 4.7: Comparison of the phoneme distributions of 40 sentences chosen with uniform prior(left) and non-uniform prior(right) from TIMIT.

CHAPTER 5

Rapid Speaker Adaptation by Formant-like Peak Alignment

Gender and age are two major factors that affect automatic speech recognition in terms of speech patterns. Speech features and acoustic models from different genders and/ or age groups (e.g. males vs. females or adults vs. children) are not similar from the statistical point of view. Since adults and children have the most dramatic acoustic difference, the mismatch of speech patterns between these two populations is the major concern in this chapter.

It is well known that speech characteristics of adults and children differ due to differences in vocal apparatus. Children have higher formant and fundamental frequencies in their spectra than adults because of their shorter vocal tracts and smaller vocal cords. These could be clearly observed from Fig. 5.1 where spectrograms of the digit “9” spoken by a male and a boy.

Since most of the current automatic speech recognition systems are trained on adult speech, such systems suffer from dramatically degraded performance for child speakers [LR01, WJ96]. To reduce spectral mismatch between adult and children’s speech, various vocal tract length normalization (VTLN) and speaker adaptation techniques have been used [BF96, DNP98, PN03b].

In MLLR, the mean of Gaussian mixtures of acoustic HMMs and that of a

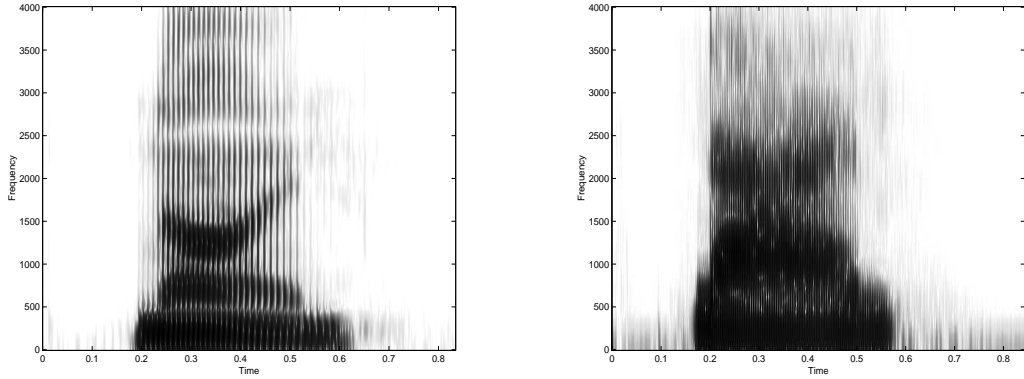


Figure 5.1: Spectrograms of the digit “9” spoken by a male (left) and a boy (right).

new speaker is described by linear regression:

$$\hat{\boldsymbol{\mu}} = \mathbf{A} \boldsymbol{\mu} + \mathbf{b} \quad (5.1)$$

The transformation matrix \mathbf{A} and bias vector \mathbf{b} are estimated using the EM algorithm with the maximum likelihood criterion. In real-world applications, one often encounters the situation where only a limited amount of adaptation data is available for the new speaker. This may be because adaptation data are difficult to obtain and/or time limitations do not permit collecting enough data. Under these conditions, MLLR performance is unsatisfactory due to unreliable parameter estimation, especially for the transformation matrix \mathbf{A} since it has more parameters to estimate than the bias vector \mathbf{b} . To reduce the number of parameters, a 3-block diagonal matrix is usually employed where the static, delta and delta-delta (first and second derivatives, respectively) parts of the features have their own full sub-matrices and independence is assumed among the three parts [GPW96, YEK01]. A diagonal form of the transformation matrix \mathbf{A} is also studied in [LW95] and is shown to have limited performance improvements. In [DBB99], several rapid speaker adaptation methods are summarized for large vocabulary speech recognizers. These methods explore the dependencies between

speech units and efficiently make use of small amounts of data by only utilizing the biases in MLLR transforms.

In recent years, the relationship between frequency warping in the linear spectral domain and the corresponding transformation in the cepstral domain has drawn increasing attention in the speaker adaptation area [CDB98, DZL02, MSW04, PMS01, PN03a]. Conclusions are made in [MSW04] and [PN03a] that VTLN equals a linear transform in the cepstral space. Perceptual linear prediction (PLP) cepstral coefficients features and MFCCs with Mel-frequency warping, instead of Mel-warped filter banks, are used in [MSW04] and [PN03a], respectively. For both features, the frequency warping is invertible and the derivation is performed in continuous frequency ω space or in the \mathcal{Z} space.

In this chapter, we first discuss, in the discrete frequency domain, the relationship between frequency warping in the front-end domain and the corresponding transformation linearity in the back-end domain for a variety of feature extraction schemes. In particular, we show that under certain approximations, the frequency warping of MFCC features with Mel-warped triangular filter banks equals a linear transformation in the model domain. The linear transformation can be considered as a special case of the traditional MLLR and serve as a basis to cope with the sparse adaptation data problem. Utilizing the linear transformation, a fast adaptation approach based on formant-like peak alignment is proposed. In this proposed approach, the transformation matrix \mathbf{A} is computed deterministically after which the bias vector \mathbf{b} is estimated statistically within the EM framework. As mentioned earlier, MLLR needs more data to reliably estimate \mathbf{A} than that needed to estimate \mathbf{b} . By generating \mathbf{A} based on re-mapping of the formant-like peaks, the proposed approach can ameliorate the spectral mismatch between adults and children’s speech while reducing the number of parameters to be estimated; this makes robust estimation of the bias \mathbf{b} possible.

The remainder of the chapter is organized as follows. In Section 5.1, the relationship between frequency warping in the front-end and corresponding transformations in the back-end is investigated for three common features and the conditions under which this relationship is equivalent to a linear transformation are discussed. The focus is on the discrete frequency domain. In Section 5.2, the estimation of formant-like peaks in speech spectra using Gaussian mixtures is described. Recognition results are presented in Section 5.3 and conclusions are made in Section 5.4.

5.1 Relationship between frequency warping and linear transformations

5.1.1 Feature Schemes

We study three kinds of speech features in this chapter: cepstra without Mel-scale warping (CEP), cepstra with Mel-scale warping (MFCC1) and Cepstra computed using Mel-warped triangular filter banks (MFCC2). While MFCC2 is the most widely used front-end feature in state-of-the-art automatic speech recognition systems, we include CEP and MFCC1 for comparison with the work in [PMS01] and [PN03a]. In the discrete frequency domain, we will show that for the first two feature extraction strategies (CEP and MFCC1), frequency warping is indeed equivalent to a linear transform in the cepstral space as stated in [PMS01] and [PN03a], and the corresponding discrete frequency transformation matrices are given. However, this conclusion is not true for MFCC2 features computed using Mel-warped filter banks unless certain approximations are made.

Fig. 5.2 illustrates the three feature extraction schemes. The input speech signal is first pre-emphasized and framed by Hamming windows. For each speech frame, the magnitude of its Discrete Fourier Transform (DFT) is obtained and

then converted into the Mel-frequency domain by certain mappings. The logarithm is then computed on the Mel-spectra to compress the dynamic range, and the output is further decorrelated by the Discrete Cosine Transform (DCT) to obtain the final cepstral coefficients.

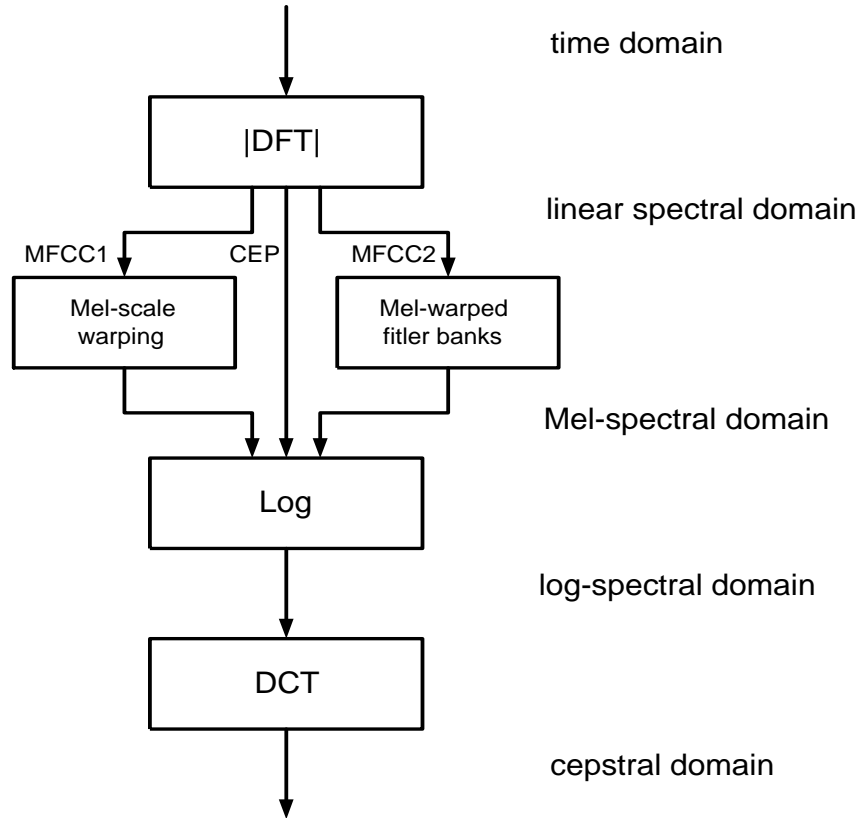


Figure 5.2: Diagram of the three feature extraction schemes discussed. The feature CEP is computed with no Mel-scale warping. MFCC1 is computed with a Mel-scale warping function, and MFCC2 is computed with Mel-warped triangular filter banks.

Let \mathbf{S}^l denote the linear spectrum and \mathbf{S}^c the cepstrum, according to Fig. 1.2, we have

$$\mathbf{S}^c = \mathbf{C} \cdot \log(\mathbf{M} \cdot \mathbf{S}^l) \quad (5.2)$$

where \mathbf{M} is the Mel-mapping matrix, \mathbf{C} the DCT matrix and \log the component-

wise logarithm function applied to the matrix. The three different features have different Mel-mapping matrices \mathbf{M} :

- For CEP, since there is no Mel-scale mapping, \mathbf{M} simply equals the identity matrix. That is,

$$\mathbf{M} = \mathbf{I} \quad (5.3)$$

- For MFCC1, the Mel-scale warping from the linear frequency f to the Mel-frequency $\varphi(f)$ is as defined by [YEK01]

$$\varphi(f) = 2595 \cdot \log_{10} \left(1 + \frac{f}{700} \right) \quad (5.4)$$

In the discrete frequency domain, the relationship between the linear frequency index l and the Mel-frequency index k is

$$k = \text{round} \left[\varphi \left(\frac{F_{max}l}{L} \right) \cdot \frac{N}{F_{max}} \right] \quad (5.5)$$

where F_{max} is the maximum frequency in the spectrum, L and N are the sample numbers in the linear and Mel-frequency domains, respectively. Define

$$\psi(l) = \varphi \left(\frac{F_{max}l}{L} \right) \cdot \frac{N}{F_{max}} \quad (5.6)$$

The Mel-mapping matrix can be expressed as

$$\mathbf{M} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 & 0 & \cdots & 0 \end{bmatrix}_{N \times L} \quad (5.7)$$

where \mathbf{M} 's components are defined as:

$$m_{ij} = \begin{cases} 1, & \text{if } i = \text{round}(\psi(j)) \\ 0, & \text{otherwise.} \end{cases} \quad (5.8)$$

Typically, in order to perform the Mel-scale mapping in the discrete frequency domain, an “oversampling” strategy is used in the linear spectral domain [DPH87] and a smaller number of samples in the Mel-spectral domain are generated by selecting appropriate frequency components from the linear spectral domain. Therefore, L is larger than N in Eq. 5.7.

- For MFCC2, triangular filter banks are employed in Mel-mapping whose central frequencies are equally spaced in the Mel-frequency axis [YEK01] as shown in Fig. 5.3.

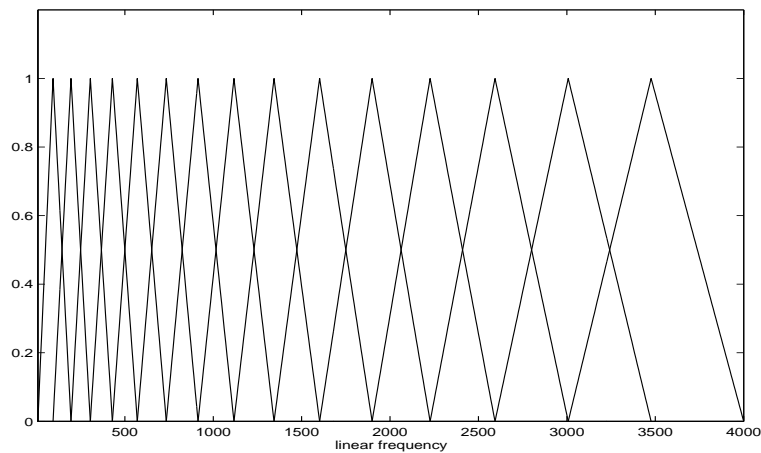


Figure 5.3: Mel-scaled triangular filter bank.

The corresponding Mel-mapping matrix \mathbf{M} can be written as:

$$\mathbf{M} = \begin{bmatrix} \theta_{1,1} & \theta_{1,2} & \cdots & \theta_{1,K_1} & 0 & 0 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & \theta_{2,1} & \cdots & \theta_{2,K_2} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & \cdots & \cdots & \cdots & 0 & \theta_{N,1} & \cdots & \theta_{N,K_N} \end{bmatrix}_{N \times L} \quad (5.9)$$

where L and N are the sample numbers in the linear and Mel-frequency domains, respectively, $\theta_{i,j}$ are weights of the triangular filters and K_1, \dots, K_N

are the numbers of non-zero weights of each triangular filter. Typically, N is much smaller than L .

5.1.2 Derivation of the transformation matrix \mathbf{A}

Suppose there exists a warping function in the discrete linear frequency domain $l = g(k)$, where k and l are the discrete frequency sample indices. This can be presented as a warping matrix \mathbf{R} whose components are defined as:

$$r_{ij} = \begin{cases} 1, & \text{if } i = \text{round}(g(j)) \\ 0, & \text{otherwise.} \end{cases} \quad (5.10)$$

in case the index j , computed using the warping function, is located outside the sample number interval, e.g. $j < 0$ or $j > L - 1$ where L is the total discrete sample number, 0 or $L - 1$ is set for the index. Let \mathbf{X} be one speech feature vector and \mathbf{Y} be the feature vector after applying the linear frequency warping, then \mathbf{X} and \mathbf{Y} have a relationship described by Eq. 5.11.

$$\mathbf{Y} = \mathbf{C} \cdot \mathbf{log}(\mathbf{M} \cdot \mathbf{R} \cdot \mathbf{M}^* \cdot \mathbf{exp}(\mathbf{C}^{-1} \cdot \mathbf{X})) \quad (5.11)$$

where \mathbf{C} and \mathbf{C}^{-1} are the DCT and inverse DCT matrices, respectively. \mathbf{R} is the linear frequency warping matrix. \mathbf{M} is the Mel-mapping matrix, \mathbf{M}^* is the matrix that transforms features from the Mel-frequency domain to the linear frequency domain and $\mathbf{log}(\cdot)$ and $\mathbf{exp}(\cdot)$ are component-wise logarithm and exponential functions of the matrix.

Let $\mathbf{T} = \mathbf{M} \cdot \mathbf{R} \cdot \mathbf{M}^*$, then Eq. 5.11 can be written as:

$$\mathbf{Y} = \mathbf{C} \cdot \mathbf{log}(\mathbf{T} \cdot \mathbf{exp}(\mathbf{C}^{-1} \cdot \mathbf{X})) \quad (5.12)$$

This equation is equivalent to the one presented in [CDB98].

Before we discuss the properties of the transform in Eq. 5.12, let us first define an index mapping (IM) matrix. A matrix is called an index mapping

matrix if there is one and only one “1” in each row and all the other components are zeros. There is no requirement on the dimension of an IM matrix. It is not necessarily a square matrix. For example, the Mel-mapping matrix \mathbf{M} in Eq. 5.7 and the warping matrix \mathbf{R} mentioned above are all IM matrices. Furthermore, it is obvious that the product of IM matrices is still an IM matrix.

Next, we show that if the matrix \mathbf{T} in Eq. 5.12 is an IM matrix, then \mathbf{X} and \mathbf{Y} are related by a linear transformation. Since \mathbf{T} is an IM matrix, it only re-maps the order of vector component indices and does not alter the value of them. Therefore, we can exchange the order of \mathbf{T} and $\log(\cdot)$:

$$\begin{aligned}
\mathbf{Y} &= \mathbf{C} \cdot \log(\mathbf{T} \cdot \exp(\mathbf{C}^{-1} \cdot \mathbf{X})) \\
&= \mathbf{C} \cdot \mathbf{T} \cdot (\log \cdot \exp(\mathbf{C}^{-1} \cdot \mathbf{X})) \\
&= \mathbf{C} \cdot \mathbf{T} \cdot \mathbf{C}^{-1} \cdot \mathbf{X} \\
&= \mathbf{A} \cdot \mathbf{X}
\end{aligned} \tag{5.13}$$

where

$$\mathbf{A} = \mathbf{C} \cdot \mathbf{T} \cdot \mathbf{C}^{-1} \tag{5.14}$$

Or, by substituting $\mathbf{T} = \mathbf{M} \cdot \mathbf{R} \cdot \mathbf{M}^*$ into Eq. 5.14, we obtain

$$\mathbf{A} = \mathbf{C} \cdot \mathbf{M} \cdot \mathbf{R} \cdot \mathbf{M}^* \cdot \mathbf{C}^{-1} \tag{5.15}$$

Consequently, the expectations of \mathbf{X} and \mathbf{Y} also satisfy the same linear relation:

$$E\{\mathbf{Y}\} = E\{\mathbf{A} \cdot \mathbf{X}\} = \mathbf{A} \cdot E\{\mathbf{X}\} \tag{5.16}$$

In other words,

$$\boldsymbol{\mu}_Y = \mathbf{A} \cdot \boldsymbol{\mu}_X \tag{5.17}$$

In most cases, speech features employed in automatic speech recognizers are a concatenation of static MFCCs, their first (delta) and second (delta-delta) order

derivatives. In this dissertation, the derivatives are computed using first order difference:

$$\Delta \mathbf{X}_t = \mathbf{X}_t - \mathbf{X}_{t-1} \quad (5.18)$$

$$\Delta^2 \mathbf{X}_t = \Delta \mathbf{X}_t - \Delta \mathbf{X}_{t-1} \quad (5.19)$$

It is straightforward that if Eq. 5.13 holds, then we have

$$\Delta \mathbf{Y} = \mathbf{A} \cdot \Delta \mathbf{X} \quad (5.20)$$

$$\Delta^2 \mathbf{Y} = \mathbf{A} \cdot \Delta^2 \mathbf{X} \quad (5.21)$$

Thus,

$$\boldsymbol{\mu}_{\Delta Y} = \mathbf{A} \cdot \boldsymbol{\mu}_{\Delta X} \quad (5.22)$$

$$\boldsymbol{\mu}_{\Delta^2 Y} = \mathbf{A} \cdot \boldsymbol{\mu}_{\Delta^2 X} \quad (5.23)$$

Now, we know that as long as $\mathbf{T} = \mathbf{M} \cdot \mathbf{R} \cdot \mathbf{M}^*$ is an IM matrix, the expectations of the original feature \mathbf{X} and warped feature \mathbf{Y} are linearly related. Next, we will investigate the properties of the mean transformation of the three feature extraction schemes discussed in Section 5.1.1.

- For CEP,

$$\mathbf{M} = \mathbf{M}^* = \mathbf{I} \quad (5.24)$$

both of which are IM matrices and the warping matrix \mathbf{R} is also an IM matrix. Hence, \mathbf{T} , which is the product of three IM matrices, is also an IM matrix. According to the discussion above,

$$\boldsymbol{\mu}_Y = \mathbf{A} \cdot \boldsymbol{\mu}_X \quad (5.25)$$

where

$$\mathbf{A} = \mathbf{C} \cdot \mathbf{R} \cdot \mathbf{C}^{-1} \quad (5.26)$$

- For MFCC1, \mathbf{M} and \mathbf{R} are both IM matrices. Since the Mel-mapping in Eq. 5.7 is performed by first “oversampling” in the linear frequency domain and then selecting the desired frequency components in the Mel-frequency, the number of rows N is smaller than the number of columns L . Therefore, in order to recover the linear frequency samples from Mel-frequency, interpolation is needed in matrix \mathbf{M}^* :

$$\mathbf{M}^* = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}_{L \times N} \quad (5.27)$$

where

$$m_{ij}^* = \begin{cases} 1, & \text{if } i = \text{round}(\psi^{-1}(j)) \\ 0, & \text{otherwise.} \end{cases} \quad (5.28)$$

The interpolation calculated according to Eq. 5.28 generates the unseen samples by repeating existing neighboring samples. In this way, \mathbf{M}^* is an IM matrix. Hence, \mathbf{T} is also an IM matrix and we have

$$\boldsymbol{\mu}_Y = \mathbf{A} \cdot \boldsymbol{\mu}_X \quad (5.29)$$

where

$$\mathbf{A} = \mathbf{C} \cdot \mathbf{M} \cdot \mathbf{R} \cdot \mathbf{M}^* \cdot \mathbf{C}^{-1} \quad (5.30)$$

- For MFCC2, the Mel-mapping involves the summation of spectra samples within each triangular filter frequency range. Therefore, \mathbf{M} is not an IM matrix. So \mathbf{T} is generally not an IM matrix either. Eq. 5.12 can not be

expressed as a linear transformation. However, suppose we substitute the output of each triangular filter in the filter banks with the value of the center frequency sample (peak) of that filter, we are able to approximate \mathbf{M} with an IM matrix $\tilde{\mathbf{M}}$:

$$\tilde{\mathbf{M}} = \begin{bmatrix} \tilde{\theta}_{1,1} & \tilde{\theta}_{1,2} & \cdots & \tilde{\theta}_{1,K_1} & 0 & 0 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & \tilde{\theta}_{2,1} & \cdots & \tilde{\theta}_{2,K_2} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & \cdots & \cdots & \cdots & 0 & \tilde{\theta}_{N,1} & \cdots & \tilde{\theta}_{N,K_N} \end{bmatrix}_{N \times L} \quad (5.31)$$

where

$$\tilde{\theta}_{i,j} = \begin{cases} 1, & \text{if } \theta_{i,j} \text{ is the central frequency of filter } i \\ 0, & \text{otherwise.} \end{cases} \quad (5.32)$$

Similarly, \mathbf{M}^* , which maps samples from the Mel-frequency domain to the linear frequency domain can be created by setting the output of each triangular filter on the Mel-frequency axis as the sample value at the corresponding center frequency on the linear frequency axis. The other frequency samples in the linear frequency domain are interpolated by repeating neighboring center frequencies that have already been generated. Thus, \mathbf{M}^* is an IM matrix. Since $\tilde{\mathbf{M}}$, \mathbf{M}^* and \mathbf{R} are all IM matrices, linear transformation between $\boldsymbol{\mu}_Y$ and $\boldsymbol{\mu}_X$ is guaranteed. That is,

$$\boldsymbol{\mu}_Y \approx \mathbf{A} \cdot \boldsymbol{\mu}_X \quad (5.33)$$

where

$$\mathbf{A} = \mathbf{C} \cdot \tilde{\mathbf{M}} \cdot \mathbf{R} \cdot \mathbf{M}^* \cdot \mathbf{C}^{-1} \quad (5.34)$$

5.1.3 Discussion

The derivation in Section 5.1.2 shows the relationship between $\boldsymbol{\mu}_Y$ and $\boldsymbol{\mu}_X$ in the discrete frequency domain for the three features. In [PMS01] and [PN03a], the

components of linear transformation matrix for Cepstral and MFCC features are computed in the continuous frequency domain as follows:

$$A_{nk}(\alpha) = \frac{2s_k}{\pi} \int_0^\pi d\tilde{\omega} \cos(\tilde{\omega}n) \cos(g_\alpha^{(-1)}(\tilde{\omega})k) \quad (5.35)$$

and

$$A_{nk}^{mel}(\alpha) = \frac{2s_k}{\pi} \int_0^\pi d\tilde{\omega}_{mel} \cos(\tilde{\omega}_{mel}n) \cos(g_{mel} \circ g_\alpha^{(-1)} \circ g_{mel}^{(-1)}(\tilde{\omega}_{mel})k) \quad (5.36)$$

where g_α and g_{mel} are linear frequency and Mel-scale warping functions, respectively, and

$$s_k = \begin{cases} 0.5, & k = 0 \\ 1, & \text{else.} \end{cases} \quad (5.37)$$

Note that Eq. 5.26 for CEP and Eq. 5.30 for MFCC1 are the discrete forms of Eq. 5.35 and Eq. 5.36, respectively, where \mathbf{R} is the matrix form of g_α . In Eq. 5.36, g_{mel} and g_{mel}^{-1} are represented by \mathbf{M} and \mathbf{M}^* in Eq. 5.30. One advantage of using matrices in the discrete frequency domain is that it can avoid complicated calculus to calculate matrix components in the continuous frequency domain (Eq. 5.35 and Eq. 5.36). Generally, the analytical expression of the transformation matrices in Eq. 5.35 and Eq. 5.36 for an invertible warping function g_α is not always available. Even for some relatively simple warping functions, e.g. piecewise linear, bilinear or quadratic functions, the computational load of Eq. 5.35 and Eq. 5.36 is high. Matrix expression in the discrete frequency domain can simplify the above calculation into simple index mapping matrices and greatly reduce the computational complexity.

Note that the studies in [MSW04] and [PN03a] use either LPC-based features or MFCC features with only the Mel-scale warping, and the calculations are in the continuous frequency domain. We showed that since Mel-warped filter bank mapping is not invertible, it will not lead to a linear transformation in the cepstral domain unless a certain approximation is made. Moreover, the approximation

made in Section 5.1.2 with triangular Mel-warped filter bank matrices is not easy to implement in the continuous frequency domain.

5.1.4 Estimation of the bias vector \mathbf{b}

Suppose we adapt the means of Gaussian mixtures of HMMs as:

$$\hat{\boldsymbol{\mu}} = \mathbf{A} \boldsymbol{\mu} + \mathbf{b} \quad (5.38)$$

where the transformation matrix \mathbf{A} is generated using the method described in Section 5.1.2. We want to estimate the bias vector \mathbf{b} based on the adaptation data under the maximum likelihood criterion. This can be performed using the EM algorithm [DLR77].

Define the EM auxiliary function we are interested in as:

$$Q_b(\lambda; \bar{\lambda}) = \sum_{u=1}^U \sum_{i=1}^N \sum_{k=1}^M \sum_{t=1}^{T^u} \gamma_t^u(i, k) \cdot \log \mathcal{N}(\mathbf{o}_t; \mathbf{A}\boldsymbol{\mu}_{ik} + \mathbf{b}, \boldsymbol{\Sigma}_{ik}) \quad (5.39)$$

where U is the number of adaptation utterances and T^u is the number frames in the u th utterance. $i \in \{1, 2, \dots, N\}$ and $k \in \{1, 2, \dots, M\}$ are the indices of state and mixture sets, respectively. $\gamma_t^u(i, k) = p(s_t^u = i, \xi_t^u = k | O^u, \bar{\lambda})$ is the posterior probability of staying at state i mixture k at time t given the u th observation sequence. $\mathcal{N}(\mathbf{o}_t; \mathbf{A}\boldsymbol{\mu}_{ik} + \mathbf{b}, \boldsymbol{\Sigma}_{ik})$ is the k th multivariate Gaussian mixture in state i with weight α_{ik} while $\boldsymbol{\mu}_{ik}$ and $\boldsymbol{\Sigma}_{ik}$ are the mean vector and covariance matrix associated with it.

Suppose the biases are tied into Q classes: $\{\omega_1, \dots, \omega_q, \dots, \omega_Q\}$. For a specific class ω_q , the bias \mathbf{b}_q is shared across all the Gaussian mixtures $\mathcal{N}(\mathbf{o}_t; \boldsymbol{\mu}_{i,k}, \boldsymbol{\Sigma}_{i,k})$ with $(i, k) \in \omega_q$. The maximum likelihood estimation of \mathbf{b}_q could be obtained by setting the differentiation of $Q_b(\lambda; \bar{\lambda})$ with respect to \mathbf{b}_q to zero:

$$\begin{aligned}
\frac{\partial Q_b(\lambda; \bar{\lambda})}{\partial \mathbf{b}_q} &= \frac{\partial}{\partial \mathbf{b}_q} \sum_{u=1}^U \sum_{(i,k) \in \omega_q} \sum_{t=1}^{T^u} \gamma_t^u(i, k) \cdot \log \mathcal{N}(\mathbf{o}_t^u; \mathbf{A}\boldsymbol{\mu}_{ik} + \mathbf{b}, \boldsymbol{\Sigma}_{ik}) \\
&= \frac{\partial}{\partial \mathbf{b}_q} \sum_{u=1}^U \sum_{(i,k) \in \omega_q} \sum_{t=1}^{T^u} \gamma_t^u(i, k) \cdot \\
&\quad \left[-\frac{1}{2} (\mathbf{o}_t^u - \mathbf{A}\boldsymbol{\mu}_{ik} - \mathbf{b}_q)^T \cdot \boldsymbol{\Sigma}_{ik}^{-1} \cdot (\mathbf{o}_t^u - \mathbf{A}\boldsymbol{\mu}_{ik} - \mathbf{b}_q) \right] \\
&= \sum_{u=1}^U \sum_{(i,k) \in \omega_q} \sum_{t=1}^{T^u} \gamma_t^u(i, k) \cdot \boldsymbol{\Sigma}_{ik}^{-1} \cdot (\mathbf{o}_t^u - \mathbf{A}\boldsymbol{\mu}_{ik} - \mathbf{b}_q) = 0 \quad (5.40)
\end{aligned}$$

By regrouping terms, Eq. 5.40 can be rewritten as:

$$\sum_{u=1}^U \sum_{(i,k) \in \omega_q} \sum_{t=1}^{T^u} \gamma_t^u(i, k) \boldsymbol{\Sigma}_{ik}^{-1} (\mathbf{o}_t^u - \mathbf{A}\boldsymbol{\mu}_{ik}) = \sum_{u=1}^U \sum_{(i,k) \in \omega_q} \sum_{t=1}^{T^u} \gamma_t^u(i, k) \boldsymbol{\Sigma}_{ik}^{-1} \mathbf{b}_q \quad (5.41)$$

Therefore, the bias vector in the class ω_q can be obtained as:

$$\mathbf{b}_q = \left[\sum_{u=1}^U \sum_{(i,k) \in \omega_q} \sum_{t=1}^{T^u} \gamma_t^u(i, k) \boldsymbol{\Sigma}_{ik}^{-1} \right]^{-1} \left[\sum_{u=1}^U \sum_{(i,k) \in \omega_q} \sum_{t=1}^{T^u} \gamma_t^u(i, k) \boldsymbol{\Sigma}_{ik}^{-1} (\mathbf{o}_t^u - \mathbf{A}\boldsymbol{\mu}_{ik}) \right] \quad (5.42)$$

Typically, $\boldsymbol{\Sigma}_{ik}$ are diagonal covariance matrices so that Eq. 5.42 can be solved one dimension at a time and there is no need for the matrix inverse operation.

5.1.5 Variance Adaptation

Given the adapted Gaussian mixture means, the diagonal covariance matrices are adapted in a non-constrained manner as described in [Gal96]:

$$\hat{\boldsymbol{\Sigma}}_{ik} = \mathbf{B}_{ik}^T \mathbf{H}_q \mathbf{B}_{ik} \quad (5.43)$$

where \mathbf{H}_q is the linear covariance transformation shared by all Gaussian mixtures in the class ω_q , namely, $(i, k) \in \omega_q$. \mathbf{B}_{ik} is the inverse of the Cholesky factor of $\boldsymbol{\Sigma}_{ik}^{-1}$. That is,

$$\boldsymbol{\Sigma}_{ik}^{-1} = \mathbf{C}_{ik} \mathbf{C}_{ik}^{-1} \quad (5.44)$$

and

$$\mathbf{B}_{ik} = \mathbf{C}_{ik}^{-1} \quad (5.45)$$

The maximum likelihood estimation of the covariance linear transformation \mathbf{H}_q is given by

$$\mathbf{H}_q = \frac{\sum_{(i,k) \in \omega_q} \mathbf{C}_{ik}^T [\sum_{u=1}^U \sum_{t=1}^{T^u} \gamma_t^u(i, k) (\mathbf{o}_t^u - \boldsymbol{\mu}_{ik})(\mathbf{o}_t^u - \boldsymbol{\mu}_{ik})^T] \mathbf{C}_{ik}}{\sum_{(i,k) \in \omega_q} \sum_{u=1}^U \sum_{t=1}^{T^u} \gamma_t^u(i, k)} \quad (5.46)$$

By forcing the \mathbf{H}_q 's off-diagonal terms to zeros, a diagonal covariance matrix $\hat{\boldsymbol{\Sigma}}_{ik}$ is obtained after adaptation.

5.2 Formant-like Peak Alignment

As mentioned earlier, speech spectra of the same sound spoken by children and adults are spectrally mismatched primarily due to vocal apparatus differences. This mismatch is the major reason for performance degradation when acoustic models trained on adult speech are used to recognize children's speech. Fig. 5.4 shows another two spectra for one speech frame (25ms) from a 30-year male and a 10-year old boy for the /uw/ sound in the digit "two". Obvious pitch and formant differences can be observed from the two figures. If the spectrum can be re-shaped by aligning the corresponding formants, then the spectral mismatch would be reduced.

In this chapter, peaks are estimated using one set of Gaussian mixtures under the EM algorithm. This technique was proposed and applied in vocoder design and feature extraction in [SG01, ZR96, ZR97]. In this algorithm, the normalized magnitude of the speech spectrum for each frame is considered as a multi-mode probability density function and a Gaussian mixture model is used to fit it. The estimation is performed in an iterative manner. The estimated means, variances and mixture weights of the Gaussians correspond to the locations, bandwidths

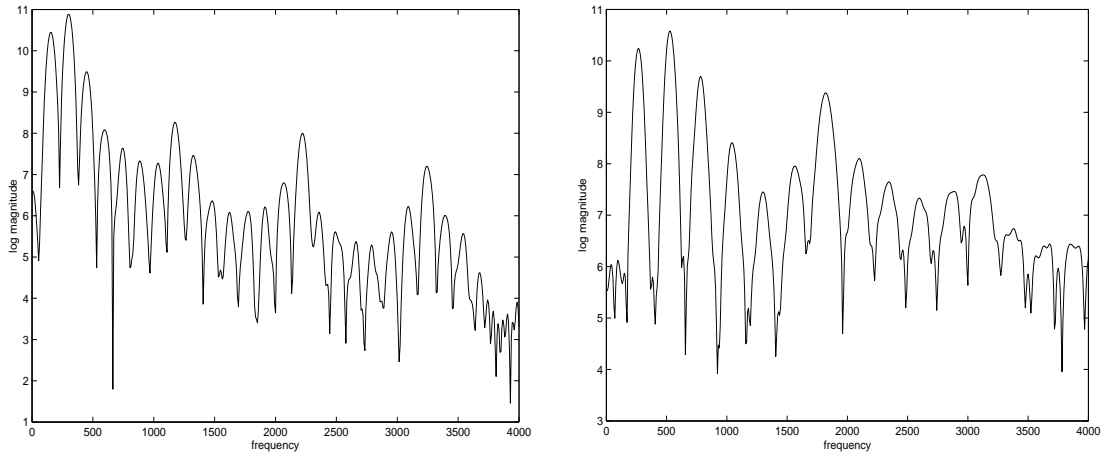


Figure 5.4: Spectra for the steady part of the sound /uw/ in the digit “two” from an adult male (left) and a boy (right).

and amplitudes of the formants. Since the peaks fitted this way are not necessarily the formants, they are called “formant-like” peaks.

Fig. 5.5 illustrates the spectrograms with peaks estimated using Gaussian mixtures. The speakers are the same as in Fig. 5.4 and the utterances are the /uw/ sound in the digit “two” from which the speech frames in Fig. 5.4 are chosen. In the estimation, four Gaussian mixtures are used for the adult male and three for the child. From the figure, one can see that the estimated peaks fit the formants quite well.

To reduce the spectra mismatch, the estimated peaks are aligned by a piece-wise linear function. Suppose we have $M-1$ peaks to align, they are $\{\omega_1^c, \dots, \omega_{M-1}^c\}$ for the child speaker and $\{\omega_1^a, \dots, \omega_{M-1}^a\}$ for the adult speaker. Also, we define $\omega_0^c = \omega_0^a = 1$. Since $\{\omega_1^c, \dots, \omega_{M-1}^c\}$ and $\{\omega_1^a, \dots, \omega_{M-1}^a\}$ are estimated Gaussian mixture means, they are real numbers, not necessarily integers. The piece-wise linear function is defined as Eq. 5.47.

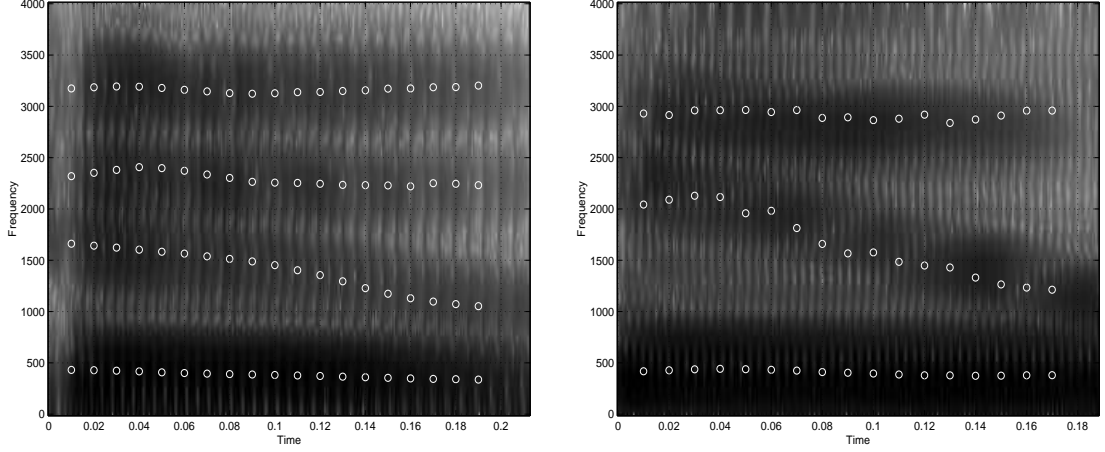


Figure 5.5: Formants estimated (white circles) using Gaussian mixtures for the sound /uw/ in digit “two” from an adult male speaker (left) and a child speaker (right).

$$\phi(l) = \begin{cases} \omega_i^c + \frac{\omega_{i+1}^c - \omega_i^c}{\omega_{i+1}^a - \omega_i^a} \cdot (l - \omega_i^a) & \text{for } l \in (\omega_i^a, \omega_{i+1}^a) \text{ and } i = 0, \dots, M-2. \\ \omega_{M-2}^c + \frac{\omega_{M-1}^c - \omega_{M-2}^c}{\omega_{M-1}^a - \omega_{M-2}^a} \cdot (l - \omega_{M-2}^a) & \text{for } l \in (\omega_{M-1}^a, \omega_M^a). \end{cases} \quad (5.47)$$

Note that we require $\omega_0^c = \omega_0^a$ but there is no requirement that $\omega_M^c = \omega_M^a$. This is because children usually have much higher formants than adults. Therefore, in the same frequency range, they may have fewer formants than adults, as shown in Fig. 5.5. By not requiring $\omega_M^c = \omega_M^a$, it is possible for the extra formants in adult spectra to disappear after alignment. Finally, we can generate the peak alignment matrix \mathbf{R} in Eq. 5.11 based on Eq. 5.47 as:

$$r_{ij} = \begin{cases} 1, & \text{if } i = \text{round}(\phi(j)) \\ 0, & \text{otherwise.} \end{cases} \quad (5.48)$$

Fig. 5.6 shows the piece-wise linear function computed according to Eq. 5.47 aligning the first (F_1) and third (F_3) formant-like peaks in Fig. 5.5. Since formants

gradually change from frame to frame, the median value for each peak is used. The two aligned peaks are marked out in the figure. In Fig. 5.7, the original spectrum of the child’s speech (solid line) and the re-shaped spectrum (dotted line) of the adult’s speech from Fig. 5.4 are illustrated. Compared with the spectra in Fig. 5.4, the mismatch between the two spectra is significantly reduced.

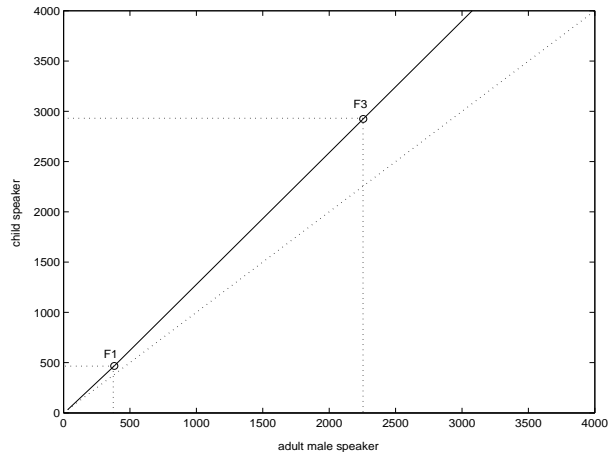


Figure 5.6: Piece-wise linear function (solid line) which aligns the first and third formant-like peaks of the adult and child’s speech in Fig. 5.5. The dotted line is the reference line for $y = x$.

5.3 Experimental Results

Pseudo-codes which describe the implementation of training, adaptation and recognition stages of the proposed approach are shown in Fig.5.8.

Experiments are performed on connected digit strings from the TIDIGITS database. Acoustic models are trained on adult males and tested on children. Utterances from 55 male speakers are used in training. There are 77 utterances from each speaker with strings consisting of either 1, 2, 3, 4, 5 or 7 digits (there

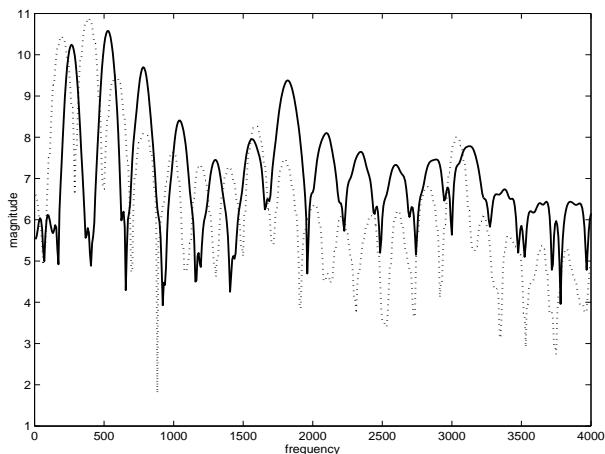


Figure 5.7: Original boy’s spectrum (solid line) and the re-shaped adult male’s spectrum (dotted line) in Fig. 5.5.

are no 6-digit strings in the database). Data from 5 boys and 5 girls are used in the test with 77 utterances from each speaker. In total, there are about 2500 digits in the test set. For each child, the adaptation utterances, which consist of 1, 4, 7, 10, 20 or 30 digits, are randomly chosen from the test set and not used in the test. The speech signals are downsampled from 20 kHz to 8 kHz. Each speech frame is 25ms in length and a 10ms frame overlap is used in the analysis. Feature vectors are of 39 dimensions: 13 static features plus their first- and second-order derivatives. The features are computed using the CEP, MFCC1 and MFCC2 schemes and the derivatives are computed according to Eq. 5.18 and Eq. 5.19. In CEP and MFCC2, a 256-point FFT is used to obtain the magnitude spectrum, and the Mel-warped filter banks are composed of 23 triangular filters. In MFCC1, the linear frequency axis is first “oversampled” by a 1024-point FFT and then warped into 128 points on the Mel-frequency axis.

Acoustic HMMs are phoneme-based with a left-to-right topology. There are 18 monophones plus silence and inter-word short pause models. Each monophone has 2 to 4 states, depending on whether it is a vowel or consonant, with 6 Gaussian

A. TRAINING

Train acoustic models $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ using adult speech data;

Select the standard speaker with the highest likelihood;

 Locate voiced segments;

 Estimate formant-like peaks of the spectrum and store as a reference;

B. ADAPTATION

Obtain adaptation data from the test child speaker;

 Locate voiced segments;

 Estimate the formant-like peaks of the spectrum;

 Align the peaks between the test and standard speakers;

 Generate warping matrix \mathbf{R} (Eq. 5.47 and Eq. 5.48);

 Generate transformation matrix \mathbf{A} (Eq. 5.26, Eq. 5.30, or Eq. 5.34);

 Estimate bias \mathbf{b} using the tree structure (Eq. 5.42);

 Adapt variance $\boldsymbol{\Sigma}$ using the tree structure (Eq. 5.46);

C. RECOGNITION

Obtain input speech signal for the test child speaker;

Perform recognition using the adapted acoustic models $(\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\Sigma}})$.

Figure 5.8: Adaptation Algorithm.

mixtures in each state.

The adaptation of the children’s speech is carried out in an unsupervised manner. For each child, voiced segments are detected from the adaptation utterance via the traditional cepstrum peak analysis technique [RS78]. Formant-like peaks are then estimated from the voiced segments by Gaussian mixtures. For a specific speaker, the median of peaks in each voiced segment is first obtained and the average over all the medians serves as the estimate of the peaks and is used in the alignment. The adult male who yields the highest likelihood in the training set is selected as the “standard” adult speaker and used to represent the acoustic characteristics of the entire adult training set.

It is observed that typically in the 4 kHz frequency range, adult speakers have four formants while child speakers have only three. Hence, four Gaussian mixtures are used for the adult males and three for the children in the peak estimation procedure. The Gaussian mixtures are initialized with means uniformly located on the frequency axis with equal mixture weights. For each frame, 20 EM iterations are performed.

The three features (CEP, MFCC1, MFCC2) are evaluated with the following peak alignment strategies:

- align F_1 and F_3 , denoted as $R(F_1, F_3)$
- align F_3 only, denoted as $R(F_3)$
- align average F_3 which is estimated from the speech of all the children and males in the database, denoted as $R(\bar{F}_3)$

Note that F_1 and F_3 refer to the formant-like peaks in the spectrum. They are not necessarily equal to the formant frequencies. We place an emphasis on the F_3 region since F_3 has been shown to correlate with the vocal tract length [Fan73].

These strategies result in different alignment matrices \mathbf{R} in Eq. 5.10, and hence different transformation matrices \mathbf{A} in Eq. 5.15.

In Tables 5.1, 5.2, and 5.3, the performance of the formant-like peak alignment algorithm with the three alignment strategies ($R(F_1, F_3)$, $R(F_3)$ and $R(\bar{F}_3)$) is compared to the traditional VTLN with CEP, MFCC1 and MFCC2 features, respectively. VTLN is implemented in two ways, namely, speaker-dependent VTLN (VTLN1) and utterance-dependent VTLN (VTLN2). In both cases, warping factors are chosen from $[0.7, 1.1]$ with a stepsize of 0.05. For VTLN1, an average likelihood is first computed with the candidate warping factors across the adaptation utterances by forced alignment. The warping factor yielding the highest average likelihood is chosen as the optimal factor to scale the frequency axis in the feature extraction stage. For VTLN2, each test utterance is first recognized to obtain an initial transcription (hypothesis) and the warping factor with the highest likelihood for the utterance by forced alignment is applied to scale the frequency axis in feature extraction. The warped features are then re-recognized. In this way, adaptation data are ignored. VTLN by pooling adaptation and test data together to estimate the warping factor was also investigated but the results didn't improve over VTLN2.

Peak alignment with and without a bias are both presented in the tables. The results in parentheses are without a bias. In both cases, variance adaptation is performed. Depending on the amount of adaptation data available, the mean bias and variance transformation matrices are dynamically tied through a tree [YEK01] with 20 base classes. The threshold for node occupation is set to 50. For $R(\bar{F}_3)$, average F_3 peaks estimated from all the adult males and children's speech in the database are used in the alignment. The average F_3 is around 2500 Hz for adult males and 3200 Hz for children.

From the tables, significant improvements over the baseline (no adaptation)

are observed for all three features when peak alignment is used. The transformation matrix \mathbf{A} , generated by aligning the formant-like peaks, contributes the most to the improved performance, and the bias \mathbf{b} gives further improvements. Among the three alignment strategies $R(F_1, F_3)$, $R(F_3)$ and $R(\bar{F}_3)$, $R(F_3)$ yields the best results which achieves, on average, 86.8%, 91.0% and 88.8% WER reduction over the baseline for CEP, MFCC1 and MFCC2, respectively. It is also very interesting to note that, since F_3 is closely related to the speaker’s vocal tract length [Fan73, CDB98], aligning F_3 peak is related to vocal tract length normalization. Both $R(F_3)$ and $R(\bar{F}_3)$ outperform the traditional VTLN when the number of adaptation digits is larger than 4. In particular, $R(F_3)$ obtains 78.8%, 32.0% and 32.7% WER reduction over VTLN for CEP, MFCC1 and MFCC2, respectively.

Since the peak alignment algorithm bridges the front-end feature domain and back-end model domain by a linear transformation in terms of a linear frequency warping function, it can be considered as a special form of traditional MLLR. Therefore, it is interesting to compare the performance of the two. Fig. 5.9 shows the recognition results of MLLR, VTLN and peak alignment with varying numbers of adaptation digits. MFCC2 features are used in the experiments and the peak alignment is performed using $R(F_3)$. The MLLR transformation matrices have a block diagonal form and are estimated based on the regression tree with 5 base classes. The threshold for node occupation is set to 500. From the figure, MLLR has poor performance when the adaptation data is limited, due to the unreliable estimation of model parameters. Peak alignment and VTLN significantly outperform MLLR under this condition because they utilize spectral information to reduce the mismatch in adaptation. As the amount of adaptation data increases, MLLR performance improves. Therefore, MLLR has an advantage when large amounts of data are available while VTLN is advantageous for limited amounts of data. In the proposed peak alignment algorithm, we primarily

generate the linear mean transformation by aligning the formant-like peaks (similar to VTLN), on the basis of which, statistical approaches such as tree-based tied variance and bias adaptation are performed. In this way, the algorithm performs well for both large and limited amounts of adaptation data.

algorithm	Number of adaptation digits					
	1	4	7	10	20	30
baseline	51.7	51.7	51.7	51.7	51.7	51.7
VTLN1	61.4	64.8	64.2	72.2	71.5	69.4
VTLN2	69.9	69.9	69.9	69.9	69.9	69.9
R(F_1, F_3)	90.3 (88.7)	89.1 (86.8)	91.5 (91.5)	92.6 (90.3)	95.4 (93.1)	95.9 (93.0)
R(F_3)	90.7 (89.6)	92.9 (91.9)	92.3 (91.9)	93.6 (92.2)	95.8 (93.9)	96.4 (93.7)
R(\bar{F}_3)	87.9 (88.6)	89.0 (87.9)	89.4 (88.5)	92.0 (91.2)	95.1 (92.1)	95.2 (92.4)

Table 5.1: Recognition accuracy of children’s speech with CEP features for (1) baseline, or no adaptation, (2) VTLN1 (speaker-dependent), (3) VTLN2 (utterance-dependent), and (4) three peak alignment schemes (R(F_1, F_3), R(F_3) and R(\bar{F}_3)) with and without a bias vector. The results in the parentheses are without a bias. The acoustic models are trained on adult male data and tested on children’s.

5.4 Summary and Conclusions

In this chapter, the relationship between linear frequency warping in the front-end feature extraction and model transformation in the back-end is investigated in the discrete frequency domain with three feature extraction schemes: cepstra without Mel-scale warping, cepstra with Mel-scale warping and cepstra with

algorithm	Number of adaptation digits					
	1	4	7	10	20	30
baseline	36.1	36.1	36.1	36.1	36.1	36.1
VTLN1	87.0	89.6	92.2	92.8	92.6	92.9
VTLN2	91.5	91.5	91.5	91.5	91.5	91.5
R(F ₁ ,F ₃)	89.4 (89.5)	89.0 (87.5)	91.9 (91.8)	92.8 (90.8)	94.9 (92.2)	95.9 (92.9)
R(F ₃)	90.0 (89.8)	93.8 (93.1)	93.9 (93.8)	94.7 (93.0)	96.2 (94.3)	96.7 (94.7)
R(\bar{F}_3)	89.1 (89.2)	91.9 (91.6)	92.4 (90.7)	93.2 (92.0)	95.6 (93.4)	96.6 (93.2)

Table 5.2: Recognition accuracy of children’s speech with MFCC1 features. See Table 5.1 caption for explanation of the test conditions.

algorithm	Number of adaptation digits					
	1	4	7	10	20	30
baseline	38.9	38.9	38.9	38.9	38.9	38.9
VTLN1	82.0	87.9	88.4	88.2	90.0	89.4
VTLN2	89.8	89.8	89.8	89.8	89.8	89.8
R(F ₁ ,F ₃)	86.1 (85.3)	90.1 (88.8)	91.1 (90.6)	91.0 (90.5)	93.5 (93.3)	95.1 (93.6)
R(F ₃)	89.5 (87.7)	93.3 (92.8)	92.6 (92.8)	92.8 (91.9)	95.0 (94.3)	95.6 (94.3)
R(\bar{F}_3)	88.2 (87.1)	91.7 (91.4)	91.5 (91.7)	92.5 (91.3)	94.6 (93.4)	96.0 (93.2)

Table 5.3: Recognition accuracy of children’s speech with MFCC2 features. See Table 5.1 caption for explanation of the test conditions.

Mel-warped triangular filter banks. A linear transformation is shown for the first two schemes. The transformation is linear for the third scheme only if certain approximations are made. The linear transformation is based on a discrete frequency mapping function (R) which can be considered as the discretized form of a general mapping function in the continuous frequency domain. Therefore, the

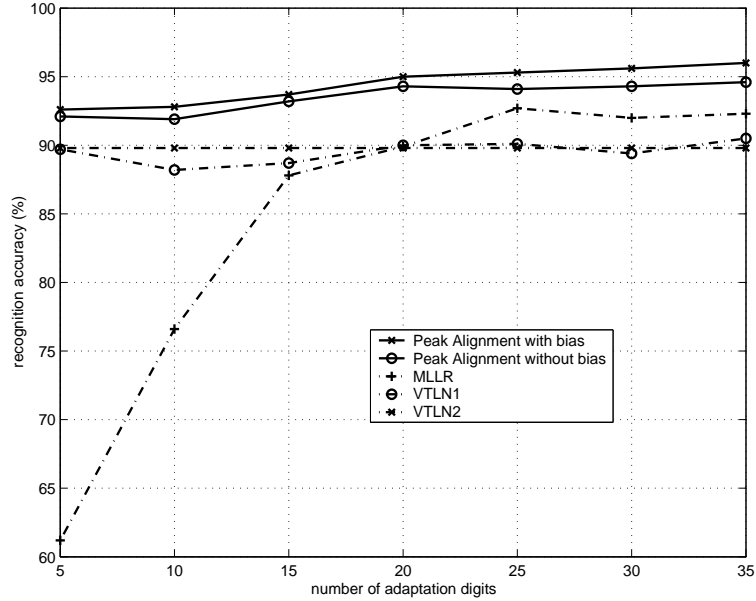


Figure 5.9: Performance of MLLR, VTLN and the peak alignment algorithm using $R(F_3)$ with different numbers of adaptation digits for MFCC2 features.

linear transformation could cover a wide range of frequency warping functions.

The linear transformation can be considered as a special case of standard MLLR and serves as a basis to deal with the sparse adaptation data problem. Utilizing the linear transformation, a fast adaptation approach based on formant-like peak alignment is proposed. In this proposed approach, the transformation matrix \mathbf{A} of Gaussian mixture means is computed deterministically after which the bias vector \mathbf{b} is estimated statistically within the EM framework. Non-constrained Gaussian covariance adaptation is also conducted statistically. Both the estimation of biases and transformations of covariances are dynamically tied via a tree structure.

The proposed algorithm is utilized to adapt children’s speech using acoustic models trained on adult data when the adaptation data is limited. Compared to traditional VTLN and MLLR with various amounts of adaptation data, sig-

nificant improvements are observed. Best results are obtained when the peak alignment scheme uses speaker-specific F_3 information for the alignment. On average, with the widely-used MFCC feature with Mel-warped triangular filter banks, speaker-specific F_3 alignment outperforms VTLN by 33%, and MLLR by 54% with limited adaptation data.

CHAPTER 6

Speaker Adaptation by Weighted Model Averaging Based on MDL

Among the speaker adaptation techniques, the maximum likelihood linear regression (MLLR) [LW95] is one of the most well-known and widely-used approaches due to its effectiveness and computational advantages. In this chapter, we investigate a robust speaker adaptation scheme using MLLR with a structured transformation matrix. The scheme yields consistent performance across various amounts of adaptation data - sparse or adequate. Structured MLLR transformations are clustered through a regression tree [LW95] and their ML estimation is provided. Given a certain amount of adaptation data, a variety of transformation structures are chosen and their tying patterns with the regression tree are described by the minimum description length (MDL) [Ris78] to account for the tradeoff between transformation granularity and descriptive ability. Based on the normalized MDL scores, the final transformation is obtained by a weighted average across the candidate structures.

Since it was first proposed in 1978 [Ris78], the MDL has been extensively studied and applied in model selection problems. There are many excellent papers reviewing MDL, e.g. [DPH87], [BRY98], [Sti], etc. In speech recognition, the MDL was also used as a means to cluster acoustic units or optimize acoustic models [SW97][WZ01]. Rooted in information theory, the MDL principle renders a view to model selection from a coding perspective. It treats a statistical model

S with parameter θ as a coding algorithm to compress data X for the estimation. The total length ($L(S)$) to describe the coding of the data via the model includes the length of the compressed data ($-\log p(X|\theta)$) plus the length describing the model itself ($L(\theta)$):

$$L(S) = -\log p(X|\theta) + L(\theta) \quad (6.1)$$

In Eq.6.1, the first term on the right-hand side accounts for how well the model fits the data and the second term describes the complexity of the model. It is desirable to describe complicated phenomena by a simple model just as the famous Occam’s razor states – “One should not increase, beyond what is necessary, the number of entities required to explain anything”. Thus, given M competing models, the one with the shortest code length is favored which results in a simple model (or short $L(\theta)$) with a good fit of the data (or short $-\log p(X|\theta)$). In this chapter, the MDL is employed to describe the structured MLLR adaptation using a regression tree.

The remainder of the chapter is organized as follows: In Section 6.1, formulation of estimation of structured MLLR transformations is provided. Structure description via the MDL principle and weighted model averaging based on normalized MDL scores are given in Sections 6.2 and 6.3. Section 6.4 discusses the choice of proper structures. Experimental results are presented in Section 6.5 which is followed by a discussion in Section 6.6. A summary is presented in Section 6.7.

6.1 Structured Transformations

As in [LW95], the MLLR transformation can be written as:

$$\hat{\boldsymbol{\mu}} = \mathbf{A} \boldsymbol{\xi} \quad (6.2)$$

where $\boldsymbol{\xi} = [1, \mu_1, \dots, \mu_N]^T$ is the augmented mean vector with $\boldsymbol{\mu} = [\mu_1, \dots, \mu_N]^T$ denoting the N -dimensional mean vector of a Gaussian mixture in speaker-independent acoustic models. The adapted Gaussian mixture mean $\hat{\boldsymbol{\mu}}$ is computed from the original augmented mean $\boldsymbol{\xi}$ via a linear transformation matrix \mathbf{A} with an $N \times (N + 1)$ dimension.

When the adaptation data are adequate to perform reliable estimation, a full matrix form of \mathbf{A} is preferred. However, most often, in practical situations, only limited adaptation data are available. Under this condition, it is interesting to investigate different structures of \mathbf{A} which may render fewer free parameters to estimate while still providing a good descriptive ability of the transformation. For a particular structure, only the elements of interest in the transformation matrix are taken into account while the rest are set to zeros. For instance, Eq.6.3 illustrates a structure of the transformation matrix \mathbf{A} with elements of interest located in the first column and along the 3 principal diagonals in the remaining sub-matrix.

$$\mathbf{A} = \begin{bmatrix} \times & \times & \times & & & \\ \times & \times & \times & \times & & \\ \times & & \times & \times & \ddots & \\ \vdots & & & \ddots & \ddots & \times \\ \times & & & & \times & \times \end{bmatrix}_{N \times (N+1)} \quad (6.3)$$

Before we derive the ML estimate of the structured transformation, let us first review the derivation of transformation matrix \mathbf{A} with no assumption of its structure (see for example [LW95]). It will become manifest that the ML estimate of the structured transformation is an extension of the estimate of a full transformation matrix in an EM framework.

Suppose there are R adaptation utterances and U^r is the number of frames in

the r th utterance. $\gamma_t^r(i, k) = p(s_t^r = i, \kappa_t^r = k | \mathcal{O}^r, \bar{\lambda})$ is the posterior probability of being at state i and Gaussian mixture k at time t given the r th observation sequence $\mathcal{O}^r = \{o_1^r, \dots, o_{U^r}^r\}$. $\boldsymbol{\xi}_{ik}$ and $\boldsymbol{\Sigma}_{ik}$ are the augmented mean vector of $\boldsymbol{\mu}_{ik}$ and covariance matrix associated with state i and Gaussian mixture k . Transformations are tied into Q classes: $\{\omega_1, \dots, \omega_q, \dots, \omega_Q\}$. For a specific class ω_q , the transformation matrix \mathbf{A}_q is shared across all the Gaussian mixtures $\mathcal{N}(\mathbf{o}_t^r; \boldsymbol{\mu}_{ik}, \boldsymbol{\Sigma}_{ik})$ with $(i, k) \in \omega_q$. The ML estimation of \mathbf{A}_q can be obtained from:

$$\sum_{r=1}^R \sum_{t=1}^{U^r} \sum_{(i,k) \in \omega_q} \gamma_t^r(i, k) \boldsymbol{\Sigma}_{ik}^{-1} \mathbf{o}_t^r \boldsymbol{\xi}_{ik}^T = \sum_{r=1}^R \sum_{t=1}^{U^r} \sum_{(i,k) \in \omega_q} \gamma_t^r(i, k) \boldsymbol{\Sigma}_{ik}^{-1} \mathbf{A}_q \boldsymbol{\xi}_{ik} \boldsymbol{\xi}_{ik}^T \quad (6.4)$$

Define the terms the same way as in [LW95]:

$$\mathbf{V}_{ik}^r = \sum_{t=1}^{U^r} \gamma_t^r(i, k) \boldsymbol{\Sigma}_{ik}^{-1} \quad (6.5)$$

$$\mathbf{D}_{ik} = \boldsymbol{\xi}_{ik} \boldsymbol{\xi}_{ik}^T \quad (6.6)$$

$$\mathbf{Z}_q = \sum_{r=1}^R \sum_{t=1}^{U^r} \sum_{(i,k) \in \omega_q} \gamma_t^r(i, k) \boldsymbol{\Sigma}_{ik}^{-1} \mathbf{o}_t^r \boldsymbol{\xi}_{ik}^T \quad (6.7)$$

Hence:

$$\text{vec}(\mathbf{Z}_q) = \left(\sum_{r=1}^R \sum_{(i,k) \in \omega_q} (\mathbf{V}_{ik}^r \otimes \mathbf{D}_{ik}) \right) \cdot \text{vec}(\mathbf{A}_q) \quad (6.8)$$

where $\text{vec}(\cdot)$ converts a matrix into a vector in terms of the rows and \otimes is the Kronecker product.

When the covariance matrix $\boldsymbol{\Sigma}_{ik}$ is diagonal, it is easy to solve Eq.6.8. In this case, $\sum_{r=1}^R \sum_{(i,k) \in \omega_q} (\mathbf{V}_{ik}^r \otimes \mathbf{D}_{ik})$ is simplified to

$$\begin{bmatrix} v_{ik(11)}^r \mathbf{D}_{ik}^r & & & & \\ & v_{ik(22)}^r \mathbf{D}_{ik}^r & & & \\ & & \ddots & & \\ & & & \ddots & \\ & & & & v_{ik(NN)}^r \mathbf{D}_{ik}^r \end{bmatrix} \quad (6.9)$$

and \mathbf{A}_q could be computed row by row from the following linear system:

$$z_{qm}^T = \mathbf{G}_{qm} \cdot a_{qm}^T \quad (6.10)$$

where z_{qm} and a_{qm} are the m th row of Z_q and \mathbf{A}_q .

$$\mathbf{G}_{qm} = \sum_{r=1}^R \sum_{(i,k) \in \omega_q} v_{ik(mm)}^r \mathbf{D}_{ik}^r \quad (6.11)$$

where $v_{ik(mm)}^r$ is the m th element on the diagonal of matrix V_{ik}^r .

In this chapter, we are interested in the structure of \mathbf{A} and ways of exploiting the structure in robust speaker adaptation. For a **structured transformation**, suppose the m th row of \mathbf{A}_q has P_m elements of interest, namely,

$$a_{qm} = [0, \dots, 0, a_{qm,l_1}, 0, \dots, 0, a_{qm,l_{P_m}}, 0, \dots, 0] \quad (6.12)$$

Define

$$\tilde{a}_{qm} = [a_{qm,l_1}, a_{qm,l_2}, \dots, a_{qm,l_{P_m}}]$$

and

$$\tilde{z}_{qm} = [z_{qm,l_1}, z_{qm,l_2}, \dots, z_{qm,l_{P_m}}]$$

as being the sub-vectors consisting of only those elements of interest. Then, \tilde{a}_{qm} can be solved using the following relationship:

$$\tilde{z}_{qm}^T = \tilde{\mathbf{G}}_{qm} \cdot \tilde{a}_{qm}^T \quad (6.13)$$

where

$$\tilde{\mathbf{G}}_{qm} = \begin{bmatrix} g_{l_1 l_1}^{(qm)} & g_{l_1 l_2}^{(qm)} & \dots & g_{l_1 l_{P_m}}^{(qm)} \\ g_{l_2 l_1}^{(qm)} & g_{l_2 l_2}^{(qm)} & \dots & g_{l_2 l_{P_m}}^{(qm)} \\ \vdots & \vdots & \ddots & \vdots \\ g_{l_{P_m} l_1}^{(qm)} & g_{l_{P_m} l_2}^{(qm)} & \dots & g_{l_{P_m} l_{P_m}}^{(qm)} \end{bmatrix} \quad (6.14)$$

In other words, the matrix $\tilde{\mathbf{G}}_{qm}$ is generated by eliminating the rows and columns of \mathbf{G}_{qm} which correspond to the zero elements in the structure, and keeping those

of interest. The ML estimation of the structured transformation obtained in Eq.6.13 is a general form for all possible structures.

6.2 Description of Structured Transformation Based on MDL

Given an amount of adaptation data and a transformation structure, a regression class tree [LW95] is a good choice to obtain robust performance by dynamically tying Gaussian mixtures in the acoustic HMMs in terms of spatial similarity. The regression tree is created based on the centroid splitting algorithm using the Euclidean distance between the Gaussian mixture means as described in [YEK01]. During adaptation, the Gaussian mixtures are pooled within their base class leaves or their parent nodes until the occupation counts are satisfactory for reliable estimation.

While different transformation structures have different numbers of parameters, they provide different transformation descriptive ability and require different amounts of data to conduct reliable estimation. For illustration purposes, Fig. 6.1 compares the tying patterns of a 1-diagonal and 3-diagonal transformations with a 6-class regression tree. The six base classes are denoted as the leaves at the bottom of the tree. In the figure, the tying of 1-diagonal structure is represented by grey nodes with solid arrows and 3-diagonal structure by black nodes with dashed arrows. For instance, in the 1-diagonal structure case, Gaussian mixtures from base class 1 share the transformation estimated from their own class while Gaussian mixtures from base class 2 are applied with the transformation estimated from both base classes 1 and 2. On the other hand, base classes 1, 2, 3 and 4 share the same transformation estimated from those classes in the 3-diagonal structure case. There are totally 4 transformations for the 1-diagonal

structure and 2 transformations for the 3-diagonal structure.

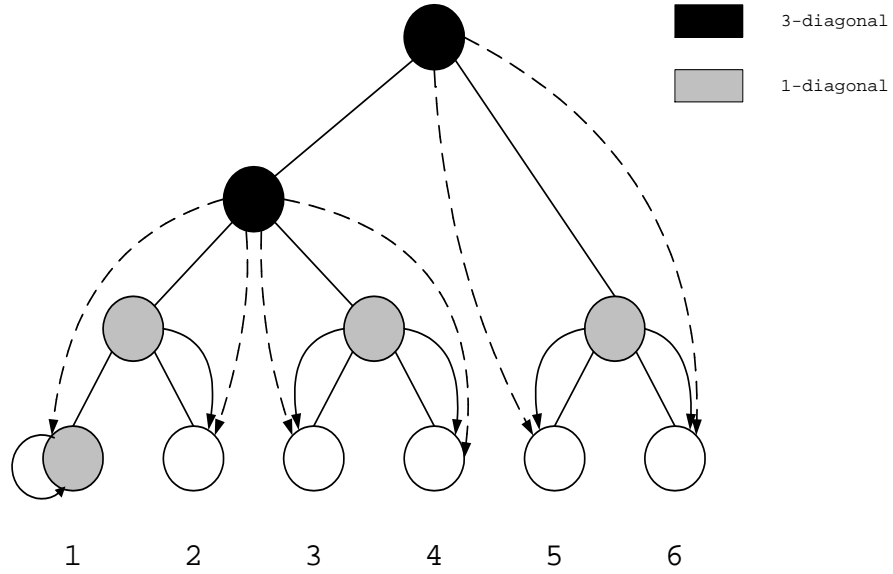


Figure 6.1: A comparison of the transformation tying patterns with a regression tree of six base classes using one-diagonal (grey node), and three-diagonal (black node) structures.

From the figure, since the 1-diagonal structure has fewer parameters than the 3-diagonal case, transformations have been tied at a lower level in the tree which indicates a better granularity. On the other hand, the 3-diagonal structure has more parameters to describe the transformation; this indicates a better descriptive ability. Therefore, a tradeoff has to be made between transformation granularity and descriptive ability.

Suppose there are M competing structures $\{S_1, \dots, S_M\}$ which result in different regression-tree tying schemes. Typically, complicated structures have transformations tied across more Gaussian mixtures (higher level in the tree toward the root node) and simple structures across less Gaussian mixtures (lower level in the tree toward the leaves). To explore the compromise between transformation granularity and descriptive ability for each transformation structure,

the minimum description length (MDL) principle is a good criterion.

In particular, suppose the Gaussian mixtures of the original acoustic HMMs are clustered into L base classes with K_l ($l = 1, \dots, L$) mixtures in the l th class. For the dynamical tying of the structure S_m ($m = 1, \dots, M$) resulting in Q_m transformations with a regression tree over R adaptation utterances $\{\mathcal{O}^1, \mathcal{O}^2, \dots, \mathcal{O}^R\}$, the description length is composed of three parts:

$$L(S_m) = L_1(S_m) + L_2(S_m) + L_3(S_m) \quad (6.15)$$

where

$$L_1(S_m) = -\log p(\mathcal{O}^1, \mathcal{O}^2, \dots, \mathcal{O}^R | \mathbf{A}_1, \dots, \mathbf{A}_{Q_m}; \bar{\lambda}) \quad (6.16)$$

$$L_2(S_m) = \sum_{q=1}^{Q_m} \frac{|S_m|}{2} \log \Gamma_{mq} \quad (6.17)$$

$$L_3(S_m) = \sum_{l=1}^L K_l I_{ml} \quad (6.18)$$

In Eq.6.15, $L_1(S_m)$ is the code length of the compressed data $\{\mathcal{O}^1, \mathcal{O}^2, \dots, \mathcal{O}^R\}$ using Q_m distinct transformations with structure S_m ; $L_2(S_m)$ is the code length of the Q_m transformations; $L_3(S_m)$ is the code length identifying one of the Q_m transformations for each Gaussian mixture. In the following, we will provide details on calculation of the three lengths.

Suppose the introduction of the transformation does not alter (a) the initial state probabilities, (b) the state transition probabilities, and (c) the frame/state alignment. Then, the first term in Eq. 6.15, $L_1(S_m)$, could be computed based on the forward-backward procedure [RJ93] using transformed Gaussian mixtures by the transformations $\{\mathbf{A}_1, \dots, \mathbf{A}_{Q_m}\}$. That is, the forward variable $\alpha_t^r(i)$ or backward variable $\beta_t^r(i)$ is computed using the state Gaussian mixture $\mathcal{N}(\mathbf{o}_t^r; \mathbf{A}_q \boldsymbol{\xi}_{ik}, \boldsymbol{\Sigma}_{ik})$.

In the second term, $L_2(S_m)$, $|S_m|$ is the number of free parameters in the transformation with the structure S_m . Γ_{mq} is the occupation counts of transformation \mathbf{A}_q with structure S_m and can be computed as:

$$\Gamma_{mq} = \sum_{r=1}^R \sum_{t=1}^{U^r} \sum_{(i,k) \in \omega_q} \gamma_t^r(i, k) \quad \text{given } S_m \quad (6.19)$$

which denotes the adaptation data's total contribution to the transformation \mathbf{A}_q with structure S_m .

The third item, $L_3(S_m)$, is the length of the code to locate a particular transformation in the tree. Each of the Q_m transformations with structure S_m is labeled by an integer from $\{1, \dots, Q_m\}$ for identification. To specify a transformation with structure S_m for each Gaussian mixture from base class l , the labelling integer $j(m, l)$ of the transformations, which is a function of structure S_m and base class l , is identified and coded. In light of the coding literature such as [Eli75] and [Ris83], the approximate universal code length for a non-zero integer $j(m, l)$ is:

$$I_{ml} = 2 + \log_2^+ |j(m, l)| + 2 \log_2^+ \log_2^+ |j(m, l)| \quad (6.20)$$

where $\log_2^+ |\cdot|$ is the positive part of the logarithm function. Substituting Eq.6.20 into Eq.6.18, we can compute the total bits needed to locate the transformations for all the Gaussian mixtures in the acoustic models.

Together, the three coding lengths, $L_1(S_m)$, $L_2(S_m)$ and $L_3(S_m)$, give the description length of a transformation with structure S_m . Note that $L_1(S_m)$ and $L_2(S_m)$ are computed in nats while $L_3(S_m)$ in bits, therefore, scaling is needed to change the different logarithm bases in the summation.

6.3 Weighted Model Averaging

Given the MDL scores for all the competing transformation structures with a regression tree, the structure with the shortest coding length is preferred and may be considered as the best candidate among all the competing structures. However, problems may occur if only the “best” structure is adopted. First, the MDL is asymptotically accurate when applied to a large amount of data. In case of limited data, the MDL choice may vary from one data set to another and give unsatisfactory results. Moreover, when the MDL scores are close, there is no one structure that is clearly superior to the others. In this situation, weighted model averaging could provide a more stable and robust performance than a single structure.

Suppose the MDL scores for the M competing structures $\{S_1, \dots, S_M\}$ are $\{\zeta_1, \dots, \zeta_M\}$ with ζ_{min} and ζ_{max} being the minimum and maximum scores, respectively. A normalized score of the m th candidate structure S_m is defined as

$$\Delta_m = \eta \cdot \frac{\zeta_m - \zeta_{min}}{\zeta_{max} - \zeta_{min}} \quad (6.21)$$

where η is empirically determined, and the weight for the structure S_m is computed as

$$\pi_m = \frac{e^{-\Delta_m}}{\sum_{m=1}^M e^{-\Delta_m}} \quad (6.22)$$

Assume the transformation applied to base class l ($l = 1, \dots, L$) with structure S_m is $\mathbf{A}_{q(m,l)}$, the final transformation for this base class is calculated as:

$$\mathbf{A}_l = \sum_{m=1}^M \pi_m \mathbf{A}_{q(m,l)} \quad (6.23)$$

Eq. 6.23 represents a model by the weighted average of different structured transformations.

6.4 Choice of Structure Form

Ideally, for the given amount of adaptation data, all possible transformation structures should be considered and their corresponding MDL scores be calculated. Suppose the transformation matrix \mathbf{A} is $N \times (N + 1)$ in dimension, then there are $2^{N \times (N+1)}$ possible structures to investigate, which is computationally prohibitive in practical situations. However, earlier research could shed light on the appropriate choices of transformation forms. For instance, [PN03a], [DZL02] and [CA05a] show that vocal tract length normalization (VTLN) in the linear spectral domain can translate into a linear transformation in the cepstral domain, which could be considered as a special case of linear regression. The transformation obtained this way has a special structure: dominant components are located along the several principal diagonals of the matrix. Figs. 6.2 and 6.3 visualize two transformation matrices associated with two scaling factors using the approach investigated in [CA05a] for the Mel-frequency Cepstral Coefficients (MFCC) feature. Similar structures could also be found in [PN03a].

The above-mentioned VTLN results provide an interesting acoustic motivation on the choice for the transformation structure. Taking into account a reasonable coverage of structures and computational considerations, we choose four structures for our experiments: 3-diagonal (3D), 7-diagonal (7D), 3-block (3B) and full matrix (full).¹ The 3-block structure with sub-full-matrix for the static, first and second order derivative of the transformation is widely used in MLLR speaker adaptation [YEK01]. Table 6.1 shows the number of free parameters for the four structured transformation matrices.

¹The structures discussed here refer to the sub-matrix after the first column in \mathbf{A} in Eq.6.2. For simplicity, we refer to them as the structure of \mathbf{A} .

matrix structure	3-diag	7-diag	3-block	full
number of parameters	154	300	546	1560

Table 6.1: Number of parameters of MLLR transformation matrix under different structures.

6.5 Experimental Results

Fig. 6.4 elaborates the implementation of the proposed weighted model averaging approach with structured transformations. Experiments are performed on the TIDIGITS database which consists of connected digit string composed of 1 to 7 digits. The speech data are sampled at 16k Hz. MFCC features are computed with a 25 ms frame length and a 10 ms frame shift. The feature is 39 in dimension consisting of 13 static MFCCs (including C0) and their first and second order derivatives. The phoneme-specific HMMs adopt a left-to-right topology with 3 to 5 states for each phoneme. A 3-state silence model and 1-state short pause model are also used. There are 6 mixtures in each state. All the Gaussian mixtures have diagonal covariance matrices.

Four sets of experiments are designed for testing: male-trained-female-tested, female-trained-male-tested, adult-trained-adult-tested and adult-trained-child-tested. The male speaker independent acoustic models are trained with 55 males and the female models with 55 females. The adult models are trained by pooling together the 55 males and 55 females. In the testing set, there are 10 males, 10 females and 10 children. In both training and testing sets, each speaker provides 77 utterances. Before recognition, data from each speaker are extracted to adapt the speaker-independent models in terms of MLLR. The adaptation is performed with 2, 5, 10, 15, 20, 25, 30 and 35 digits.

An MLLR regression tree with 128 base classes is created for each speaker-

independent acoustic model. To ensure matrix invertibility during the transformation tree-tying, a minimum number of Gaussian mixtures is required at the tying nodes which is 3, 7, 13 and 39 for 3D, 7D, 3B and full matrix, respectively. Furthermore, for reliable estimation, a threshold has to be set for each transformation structure depending on its number of free parameters. In this chapter, we choose the threshold to be approximately equal to the number of parameters for each structure. That is, 150, 300, 550 and 1500 are the occupation counts for a valid transformation estimation with 3D, 7D, 3B and full matrix, respectively. The scaling factor η in Eq. 6.21 is set to 2.0.

Tables 6.2 - 6.5 show experimental results with four transformation structures using different amounts of adaptation data. Baseline results are without adaptation. Adaptation results using the “single” best structure based on MDL, and using the averaged transformation across all four structures weighted using the MDL scores are denoted as “MDL” and “MDL-Ave”, respectively. The 3-block and full matrix structure results with very limited data (e.g. 2 digits for 3-block matrix structure and 2 and 5 digits for full matrix structure) are not shown in the tables since even the global tying for the transformation can not meet the occupation threshold requirement, and the results are thus not meaningful.

From the tables, structures with less parameters (3D or 7D) tend to give better performance than those with more parameters (3B or full) when the amount of adaptation data is small. When the amount of data increases, however, the situation is reversed. This is mainly due to the tradeoff between transformation granularity and descriptive ability. By choosing the single “best” model with the minimum score, MDL gives a better balanced performance with respect to the amount of adaptation data. Very often MDL is able to obtain the best performance among the four candidate structures given a certain amount of data. Weighted model averaging across all structures based on the normalized MDL

scores gives more consistent and robust performance than MDL alone.

6.6 Discussion

The tying of the MLLR transformation with competing structures with a regression tree is different from the model selection problems in nested linear regression coefficient selection [DPH87] or the optimal tree cut approaches [SW97][WZ01] in that the tied transformations could utilize data from overlapped Gaussian mixture sets, which is neither nested nor a partition of the Gaussian mixture space. This makes it a more interesting problem. Furthermore, to locate the transformations, the regression tree has to be traversed to get to certain nodes. In this situation, MDL seems to be a superior choice to Akaike information criterion (AIC)[Aka74] or Bayesian information criterion (BIC)[Sch78] because MDL could be interpreted from the coding point of view, and the traverse of the tree to locate the transformations can be taken into account as a part of the model itself. This can not be dealt with by AIC or BIC.

The coding of the model parameters in MDL is related to the Fisher information matrix which could be directly employed to calculate the MDL score as in [WZ01]. However, the asymptotic form used in this chapter may be a good approximation even without a large amount of data under certain Bayesian assumptions [DPH87] and this form also has its computational advantages. As we know, in speech recognition, while the training stage is performed under the ML criterion, the Viterbi decoding algorithm is applied in the recognition stage. This means that a good fit of a model based on the ML criterion may or may not result in good recognition performance. The afore-mentioned MDL's asymptotic approximation in combination with the "mismatch" schemes in training and recognition could explain why MDL, in a few cases, does not give the best ASR

results.

Compared with MDL alone, the weighted model averaging strategy renders more robust performance in most cases. This is because although MDL can not always choose the best structure, it does give a good “guess” on the goodness-of-fit of the structures. Therefore, a reasonable weight of the structure can produce better results. The tying pattern of the transformation with the regression tree is decided by (a) the structure and (b) the threshold of reliable estimation for the structure, both can be handled by the MDL weighted model averaging. Different structures other than those investigated in this chapter are also possible and the weighted model averaging algorithm can be carried out accordingly.

6.7 Summary

In this chapter, we investigate a robust maximum likelihood linear regression speaker adaptation approach with weighted model averaging across a variety of transformation structures. A general form of the maximum likelihood estimation of the structured transformation is given. The minimum description length (MDL) is adopted to describe the balance between transformation granularity and descriptive ability of the structured transformations tied using a regression tree. Based on the normalized MDL scores, transformations are averaged across all structures. Experimental results show that the proposed approach obtains robust performance with respect to the amount of adaptation data.

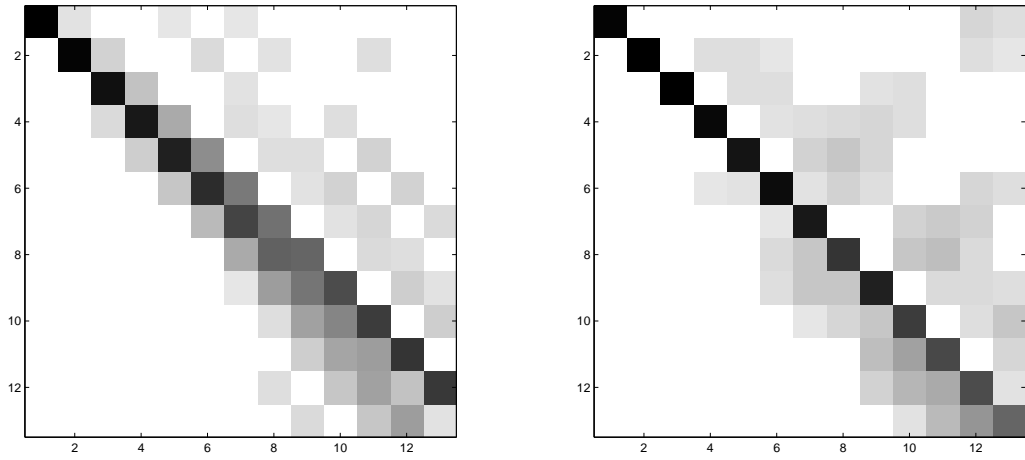


Figure 6.2: Transformation matrices generated based on vocal tract length normalization with scaling factor equal to 1.1 (left) and 0.9 (right). The darker the color, the more significant the element is.

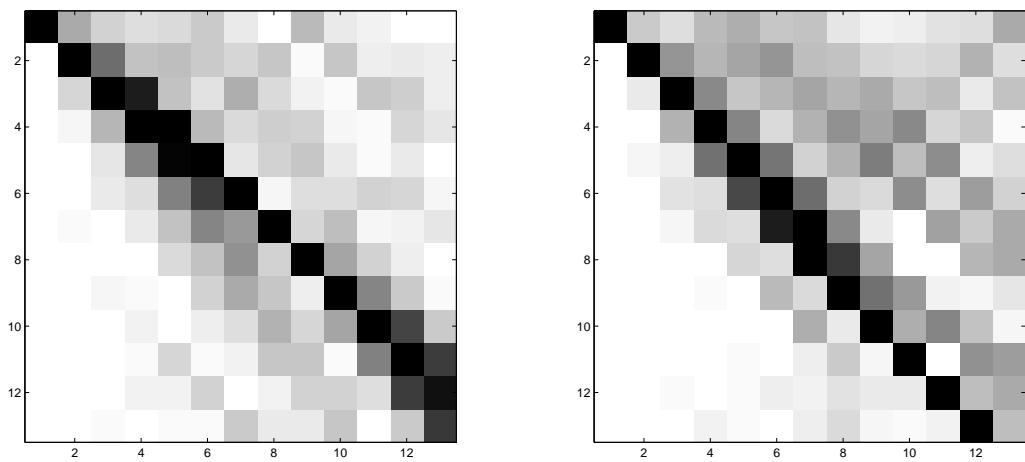


Figure 6.3: Transformation matrices generated based on vocal tract length normalization with scaling factor equal to 1.2 (left) and 0.8 (right). The darker the color, the more significant the element is.

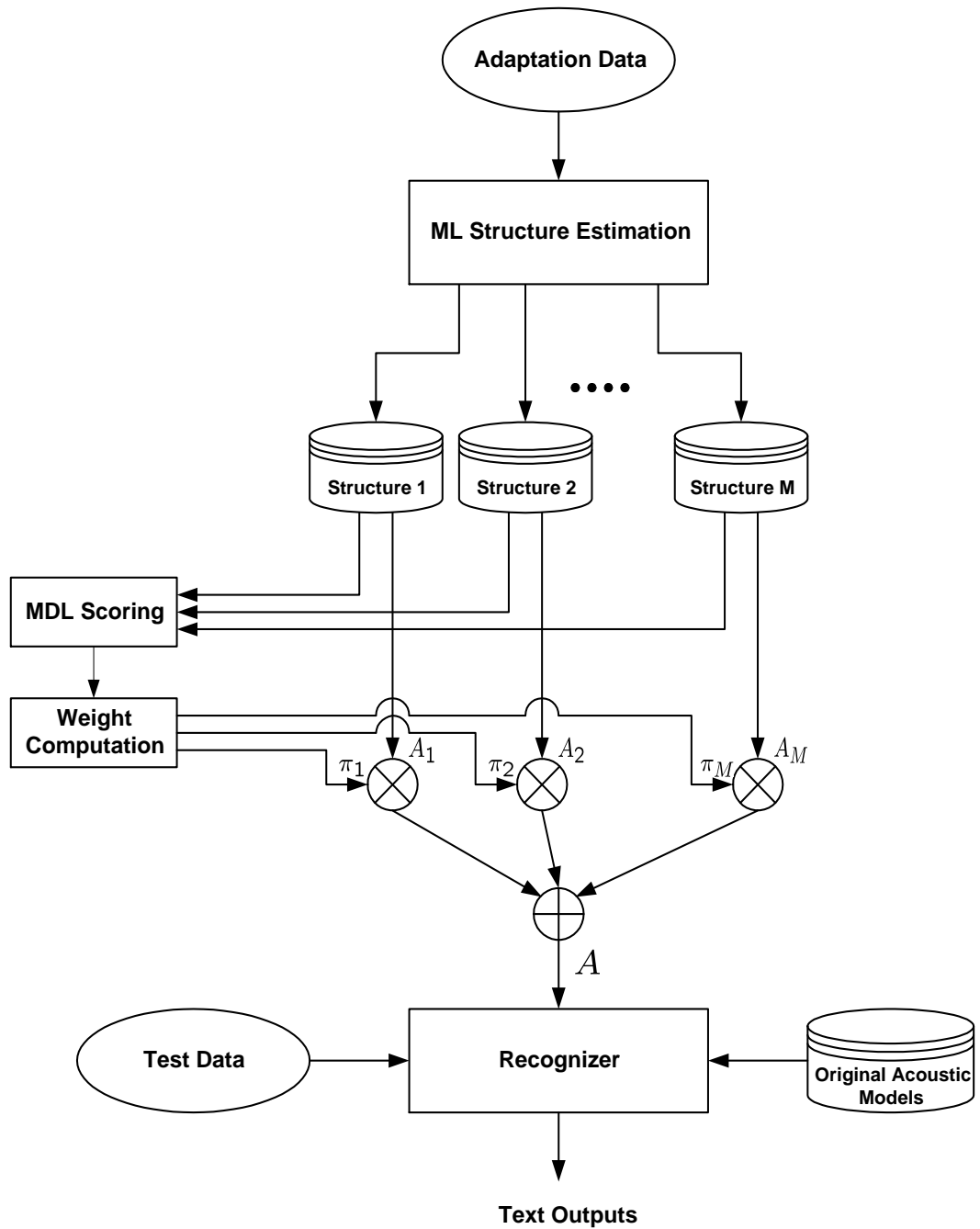


Figure 6.4: Flow chart of implementation of weighted model averaging with structured MLLR transformations. The structures are appropriately tied with a regression tree.

	Number of adaptation digits							
	2	5	10	15	20	25	30	35
No adaptation	13.0	13.0	13.0	13.0	13.0	13.0	13.0	13.0
3D	2.7	1.4	1.3	1.1	0.8	0.8	0.8	0.6
7D	7.0	1.7	1.3	0.9	0.7	0.7	0.6	0.6
3B	-	2.9	1.2	0.7	0.7	0.6	0.6	0.5
full	-	-	4.8	1.8	0.8	0.6	0.6	0.5
MDL	3.0	1.7	1.2	0.8	0.6	0.7	0.6	0.6
MDL-Ave	2.1	1.3	0.9	0.7	0.6	0.6	0.6	0.5

Table 6.2: Word error rate (%) of MLLR with different structured transformations on TIDIGITS database. Acoustic models are trained with male speech and tested on female speech. The performance is the average over the 10 female speakers in the test set. 3D, 7D, 3B and full denote 3-diagonal, 7-diagonal, 3-block and full transformation matrices, respectively.

	Number of adaptation digits							
	2	5	10	15	20	25	30	35
No adaptation	4.4	4.4	4.4	4.4	4.4	4.4	4.4	4.4
3D	1.7	0.9	0.9	0.9	0.6	0.6	0.6	0.6
7D	3.6	1.9	1.2	1.0	0.8	0.7	0.6	0.6
3B	-	5.6	1.1	0.8	0.8	0.8	0.7	0.6
full	-	-	4.0	1.5	0.9	0.8	0.8	0.6
MDL	1.7	1.3	1.2	0.9	0.7	0.7	0.7	0.6
MDL-Ave	1.0	1.0	0.8	0.8	0.6	0.6	0.6	0.6

Table 6.3: Word error rate (%) of MLLR with different structured transformations on TIDIGITS database. Acoustic models are trained with female speech and tested on male speech. The performance is the average over the 10 male speakers in the test set. 3D, 7D, 3B and full denote 3-diagonal, 7-diagonal, 3-block and full transformation matrices, respectively.

	Number of adaptation digits							
	2	5	10	15	20	25	30	35
No adaptation	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9
3D	1.1	0.7	0.9	0.8	0.7	0.6	0.7	0.7
7D	2.0	0.8	0.9	0.8	0.8	0.7	0.7	0.7
3B	-	2.4	0.8	0.7	0.8	0.7	0.8	0.6
full	-	-	2.6	1.0	0.7	0.7	0.7	0.7
MDL	1.1	0.8	0.8	0.8	0.7	0.7	0.7	0.7
MDL-Ave	1.0	0.7	0.7	0.7	0.7	0.6	0.7	0.6

Table 6.4: Word error rate (%) of MLLR with different structured transformations on TIDIGITS database. Acoustic models are trained and tested on adult speech (both male and female). The performance is the average over the 20 adult speakers in the test set. 3D, 7D, 3B and full denote 3-diagonal, 7-diagonal, 3-block and full transformation matrices, respectively.

	Number of adaptation digits							
	2	5	10	15	20	25	30	35
No adaptation	2.9	2.9	2.9	2.9	2.9	2.9	2.9	2.9
3D	2.4	1.4	1.2	1.3	1.2	1.1	1.1	1.0
7D	5.0	1.6	1.2	0.9	1.0	1.1	1.0	0.8
3B	-	9.3	1.3	0.9	0.8	1.0	0.9	0.8
full	-	-	8.2	3.0	1.3	1.1	1.1	1.0
MDL	2.4	1.4	1.2	0.9	0.9	1.0	0.9	0.8
MDL-Ave	1.9	1.4	1.1	0.9	0.9	0.9	0.9	0.6

Table 6.5: Word error rate (%) of MLLR with different structured transformations on TIDIGITS database. Acoustic models are trained on adult speech and tested on child speech. The performance is the average over the 10 kid speakers in the test set. 3D, 7D, 3B and full denote 3-diagonal, 7-diagonal, 3-block and full transformation matrices, respectively.

CHAPTER 7

Summary and Future Work

7.1 Summary

This dissertation investigates environmental and speaker robustness in ASR. In particular, Chapters 2 - 3 are focused on noise robust issues and Chapters 4 - 6 are concerned with speaker adaptation.

Chapter 2 introduces a weighted Viterbi decoding (WVD) algorithm to deal with background acoustic noise. The motivation of WVD is quite straightforward - frames with high SNRs match the clean acoustic models better than those with low SNRs. Accordingly, feature observation probabilities are weighted in the Viterbi decoding stage by a confidence factor which is a function of frame SNR. WVD finds its application in distributed speech recognition because of its low computational complexity and fairly good performance.

Chapter 3 presents a feature compensation algorithm based on polynomial regression of utterance SNR. The algorithm compensates noisy speech features to match clean acoustic models. First, it is shown that the bias between noisy and clean speech features is a nonlinear function of utterance SNR. Second, the bias is approximated by a set of polynomials. The regression polynomials are estimated under the maximum likelihood criterion in an EM framework. Finally, in the recognition stage, utterance SNR is evaluated from the speech signals and features are compensated in accordance with the regression polynomials which

could be tied at various granularity levels. Using polynomials to regress utterance SNRs can deal with time-varying environments and also is able to predict unseen environments in the training. The ML estimation of the regression polynomials discussed in the chapter is noise dependent. In situations where mixed types of noise occur, the polynomials can be adapted in terms of the new noise type. It is also interesting to investigate the combination of polynomial regression feature compensation and weighted Viterbi decoding. In [CA05b], some preliminary work has been attempted. In future work, the effect of the feature compensation (compensation error) can be taken into account in the Viterbi decoding stage through a rigorous probabilistic framework.

Chapter 4 is devoted to adaptation-text design on the basis of the Kullback-Leibler measure. The design approach enables a designer to predefine a target distribution for the speech units and selects from a large text pool, via a greedy algorithm, the texts whose speech unit distribution minimizes the Kullback-Leibler measure. Although this topic is not covered in the literature as much as other adaptation issues are, it is helpful in achieving good performance.

Chapter 5 presents a rapid speaker adaptation algorithm by formant-peak alignment using limited data. In this chapter, the relationship between frequency warping in the front-end domain and the linearity of the corresponding transformation in the back-end domain is first discussed, in the discrete frequency domain, with a variety of common feature extraction schemes. In particular, it is shown that under certain approximations, the frequency warping of MFCC features with Mel-warped triangular filter banks equals a linear transformation in the model domain. The linear transform can be considered as a special case of the traditional MLLR and serves as a basis to cope with the sparse adaptation data problem. Based on the linear transformation, a rapid adaptation approach by aligning formant-like peaks is investigated. In this proposed approach, the

transformation matrix is deterministically generated on the basis of which the bias vector is statistically estimated. By generating the transformation matrix to re-map the formant-like peaks, this approach can reduce the spectral mismatch between speakers while decreasing the number of parameters to be estimated. The peak alignment in this dissertation is applied globally. Since the alignment-based transform is constructed in the back-end model domain, it could be readily extended in future work to be phoneme, state or even mixture dependent.

Chapter 6 investigates structured maximum likelihood linear regression for speaker adaptation. The aim of using structured MLLR is to reduce the number of parameters to be estimated while maintaining good transformation resolution. The maximum likelihood estimation of general structures is derived in terms of the EM algorithm. The structured transformations are dynamically tied with a regression tree. Given a certain amount of adaptation data and the number of parameters of a specific transformation structure, different tying patterns are obtained in the tree, which implies a tradeoff between the transformation granularity and descriptive ability. The final transformation is obtained by a weighted average based on the normalized scores of minimum description length which yields robust performance with respect to various adaptation data sizes.

7.2 Discussion and Future Work

Dealing with limited data is one of the major concerns in the work presented in this dissertation. This is not only an interesting research topic in its own right but also a natural demand of the deployment of ASR systems in the real world. Therefore, feature compensation by polynomial regression, formant-like peak alignment and structured MLLR transformation are all directed to combat sparse data problem, although they would perform better if a massive amount

of adaptation data is available. The polynomial regression on SNR, which is a scalar, discussed in Chapter 3 can be generalized to regression over multiple variables, e.g. SNR, age, dialect, etc. to take into account environmental and speaker variations. The tying of polynomials in Chapter 3 as well as the tying of bias and variance transformations in Chapter 5 through a tree structure can also be considered in the same MDL model-selection framework as Chapter 6.

Looking forward to the future, incorporating acoustic knowledge effectively into current statistics-based ASR systems is worth exploring. Acoustic phonetics knowledge may be helpful in cases of under-estimation or over-estimation when traditional statistical techniques are used. For instance, limited data result in under-estimation in speaker adaptation in the sense that parameters could not be reliably estimated; too much data result in over-estimation in acoustic modeling where heavy speech diversity confuses the models and reduces the model's discriminative ability. In these situations, acoustic or perceptual knowledge may provide information to guide ASR systems to a smart training. Recently, speech recognizers utilizing discriminative acoustic clues have drawn increasing research interests. Interesting works and discussions like [HBB05] and [JE04] have been reported. The combination of statistics and acoustics seems quite promising to achieve good and robust performance.

The acoustic models investigated in this dissertation are estimated under ML criterion. Although the asymptotical optimality of the ML estimation is well justified statistically, the connection between high training likelihood and good recognition performance is not strong. In some occasions, acoustic models with high likelihood in training can not deliver satisfactory results in recognition. Discriminative optimization towards minimum classification errors [JCL97] [JK92] [WW04] [PKM05] is an alternative training approach which has been proven effective and achieves better performance under many circumstances. Since dis-

criminative training tried to separate apart the most confusable models which are strongly connected to the decoding process, it is not a surprise that it results in better performance. However, discriminative training is more computationally demanding than ML training and usually more than one pass or alignment are needed in the training process. Optimizing the acoustic models in terms of minimum classification error is also worth exploring in the future.

REFERENCES

- [Ace92] A. Acero. *Acoustical and Environmental Robustness in Automatic Speech Recognition*. Kluwer Academic Publishers, 1992.
- [Aka74] H. Akaike. “A new look at the statistical model identification.” *IEEE Trans. on Automatic Control*, **19**(6):716–723, 1974.
- [Ata74] B. Atal. “Effectiveness of linear prediction characteristics of the speech wave for automatic speaker identification and verification.” *Journal of the Acoustical Society of America*, **55**(6):1304–1312, 1974.
- [BA02] A. Bernard and A. Alwan. “Low-bitrate distributed speech recognition for packet-based and wireless communication.” *IEEE Trans. on Speech and Audio Processing*, **10**(8):570–580, 2002.
- [Bau72] L. Baum. “An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes.” *Inequalities*, **3**:1–8, 1972.
- [BDC99] E. Bocchieri, V. Digalakis, A. Corduneanu, and C. Boulis. “Correlation modeling of MLLR transform biases for rapid HMM adaptation to new speakers.” *Proc. of IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, pp. 773–776, 1999.
- [BF96] D. Burnett and M. Fanty. “Rapid unsupervised adaptation to children’s speech on a connected-digit task.” *Proc. of Int. Conf. on Spoken Language Processing*, pp. 1145–1148, 1996.
- [Bol79] S. Boll. “Suppression of acoustic noise in speech using spectral subtraction.” *IEEE Trans. on Acoustics, Speech and Signal Processing*, **ASSP-27**(2):113–120, 1979.
- [BRY98] A. Barron, J. Rissanen, and B. Yu. “The minimum description length principle in coding and modeling.” *IEEE Trans. on Information Theory*, **44**(6):2743–2760, 1998.
- [CA05a] X. Cui and A. Alwan. “Adaptation of children’s speech with limited data based on formant-like peak alignment.” *Computer Speech and Language*, p. to appear, 2005.
- [CA05b] X. Cui and A. Alwan. “Combining feature compensation and weighted Viterbi decoding for noise robust speech recognition with limited adaptation data.” *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, **1**:969–972, 2005.

- [CDB98] T. Claes, I. Dologlou, L. Bosch, and D. Compernelle. “A novel feature transformation for vocal tract length normalization in automatic speech recognition.” *IEEE Trans. on Speech and Audio Processing*, **11**(6):603–616, 1998.
- [CG03] X. Cui and Y. Gong. “Variable parameter Gaussian mixture hidden Markov modeling for speech recognition.” *Proc. of IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, **1**:12–15, 2003.
- [CIZ02] X. Cui, M. Iseli, Q. Zhu, and A. Alwan. “Evaluation of noise robust features on the Aurora databases.” *Proc. of Int. Conf. on Spoken Language Processing*, pp. 481–484, 2002.
- [DBB52] K. Davis, R. Biddulph, and S. Balashek. “Automatic recognition of spoken digits.” *Journal of Acoustical Society of America*, **24**(6):637–642, 1952.
- [DBB99] V. Digalakis, S. Berkowitz, E. Bocchieri, C. Boulis, W. Byrne, H. Collier, A. Corduneanu, A. Kannan, S. Khudanpur, and A. Sankar. “Rapid speech recognizer adaptation to new speakers.” *Proc. of IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, 1999.
- [DDA04a] 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. on Speech and Audio Processing*, **12**(3):133–143, 2004.
- [DDA04b] L. Deng, J. Droppo, and A. Acero. “Estimating cepstrum of speech under the presence of noise using a joint prior of static and dynamic features.” *IEEE Trans. on Speech and Audio Processing*, **12**(3):218–233, 2004.
- [DLR77] A. Dempster, N. Laird, and D. Rubin. “Maximum likelihood from incomplete data via the EM algorithm.” *Journal of the Royal Statistical Society*, **39**(1):1–38, 1977.
- [DM80] S. Davis and P. Mermelstein. “Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences.” *IEEE Trans. on Acoustics, Speech, Signal Proc.*, **28**(4):357–366, 1980.
- [DNP98] S. Das, D. Nix, and M. Picheny. “Improvements in children’s speech recognition performance.” *Proc. of IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, pp. 433–436, 1998.

- [DPH87] J. Deller, J. Proakis, and J. Hansen. *Discrete-Time Processing of Speech Signals*. Prentice Hall, 1987.
- [DZL02] G. Ding, Y. Zhu, C. Li, and B. Xu. “Implementing vocal tract length normalization in the MLLR framework.” *Proc. of Int. Conf. on Spoken Language Processing*, pp. 1389–1392, 2002.
- [EG96] E. Eide and H. Gish. “A parameteric approach to vocal tract length normalization.” *Proc. of IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, pp. 346–349, 1996.
- [Eli75] P. Elias. “Universal codeword sets and representation of the intergers.” *IEEE Trans. on Information Theory*, **21**:194–203, 1975.
- [Fan73] G. Fant. *Speech Sounds and Features*. The MIT Press, 1973.
- [FF59] J. Forgie and C. Forgie. “Results obtained from a vowel recognition computer.” *Journal of Acoustical Society of America*, **31**(11):1480–1489, 1959.
- [FRT97] S. Fang, J. Rajasekera, and H. Tsao. *Entropy optimization and mathematical programming*. Kluwer Academic Publishers, 1997.
- [Gal96] M. Gales. “Mean and variance adaptation within the MLLR framework.” *Computer Speech and Language*, **10**:249–264, 1996.
- [GL94] J.-L. Gauvain and C.-H. Lee. “Maximum A posteriori estimation for multivariate Gaussian mixture observations of Markov chains.” *IEEE Trans. on Speech and Audio Processing*, **2**(2):291–298, 1994.
- [Gon95] Y. Gong. “Speech recognition in noisy environments: a survey.” *Speech Communication*, **16**:261–291, 1995.
- [Gon02] Y. Gong. “Noise-dependent Gaussian mixture classifiers for robust rejection decision.” *IEEE Trans. on Speech and Audio Processing*, **10**(2):57–64, 2002.
- [GPW96] M. Gales, D. Pye, and P. Woodland. “Variance compensation with the MLLR framework for robust speech recognition and speaker adaptation.” *Proc. of IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, pp. 1832–1835, 1996.
- [GS97] E. Gouvea and R. Stern. “Speaker normalization through formant-based warping of the frequency scale.” *Proc. of European Conf. on Speech Communication and Technology*, pp. 1139–1142, 1997.

- [GY96] M. Gales and S. Young. “Robust continuous speech recognition using parallel model combination.” *IEEE Trans. on Speech and Audio Processing*, **4**:352–359, 1996.
- [HBB05] M. Hasegawa-Johnson, J. Baker, S. Borys, K. Chen, E. Coogan, S. Greenberg, A. Juneja, K. Kirchho, K. Livescu, S. Mohan, J. Muller, K. Sonmez, and T. Wang. “landmark-based speech recognition: report of the 2004 Johns Hopkins Summer Workshop.” *Proc. of IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, pp. 214–216, 2005.
- [Her90] H. Hermansky. “Perceptual Linear Prediction (PLP) Analysis of Speech.” *Journal of Acoustical Society of America*, **87(4)**:1738–1752, 1990.
- [HM94] H. Hermansky and N. Morgan. “RASTA processing of speech.” *IEEE Trans. on Speech and Audio Processing*, **2**:578–589, 1994.
- [HP00] H. Hirsch and D. Pearce. “The Aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions.” *ASR2000 - International Workshop on Automatic Speech Recognition*, pp. 181–188, 2000.
- [JCL97] B. Juang, W. Chou, and C. Lee. “Minimum classification error rate methods for speech recognition.” *IEEE Trans. on Speech and Audio Processing*, **5(3)**:257–265, 1997.
- [JE04] A. Juneja and C. Espy-Wilson. “Significance of invariant acoustic cues in a probabilistic framework for landmark-based speech recognition.” *From sound to sense: 50+ years of discoveries in speech communication*, pp. 151–156, 2004.
- [JK92] B. Juang and S. Katagiri. “Disriminative learning for minimum error classification.” *IEEE Trans. on Signal Processing*, **40(12)**:3043–3054, 1992.
- [KJN00] R. Kuhn, J. Junqua, P. Nguyen, and N. Niedzielski. “Rapid speaker adaptaion in Eigenvoice space.” *IEEE Trans. on Speech and Audio Processing*, **8(6)**:695–707, 2000.
- [KL97] K.shinoda and C. Lee. “Structural MAP speaker adaptation using hierarchical priors.” *Proc. IEEE-SP Workshop on Automatic Speech Recognition and Understanding*, 1997.
- [Kul59] S. Kullback. *Information theory and statistics*. John Wiley & Sons, 1959.

- [LR98] L. Lee and R. Rose. “A frequency warping approach to speaker normalization.” *IEEE Trans. on Speech and Audio Processing*, **6**(1):49–60, 1998.
- [LR01] Q. Li and M. Russell. “Why is automatic recognition of Children’s speech difficult.” *Proc. of European Conf. on Speech Communication and Technology*, pp. 2671–2674, 2001.
- [LW95] C. Leggetter and P. Woodland. “Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models.” *Computer Speech and Language*, **9**:171–185, 1995.
- [Mar01] R. Martin. “Noise power spectral density estimation based on optimal smoothing and minimum statistics.” *IEEE Trans. on Speech and Audio Processing*, **9**(5):504–512, 2001.
- [MG76] J. Markel and A. Gary. *Linear Prediction of Speech*. Springer-Verlag, 1976.
- [MKH04] B. Mak, J. Kwok, and S. Ho. “A study of various composite kernels for kernel eigenvoice speaker adaptation.” *Proc. of IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, pp. 325–328, 2004.
- [MMN02] D. Macho, L. Mauuary, B. Noe, Y.M. Cheng, D. Ealey, D. Juvet, H. Kelleher, D. Perace, and F. Saadoun. “Evaluation of a noise-robust DSR Front-End on Aurora databases.” *Proc. of Int. Conf. on Spoken Language Processing*, pp. 17–20, 2002.
- [Mor96] P. Moreno. *Speech recognition in noisy environments*. PhD thesis, Carnegie Mellon Univerisity, 1996.
- [MSW04] J. McDonough, T. Schaaf, and A. Waibel. “Speaker adaptation with all-pass transforms.” *speech communication*, **42**:75–91, 2004.
- [PD04] M. Padmanabhan and S. Dharanipragada. “Maximum-Likelihood nonlinear transformation for acoustic adaptation.” *IEEE Trans. on Speech and Audio Processing*, **12**(6):572–578, 2004.
- [Pet01] S. Douglas Peters. “Hypothesis-driven adaptation(HYDRA: a flexible eigenvoice architecture).” *Proc. of IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, pp. 349–352, 2001.
- [PKM05] D. Povey, B. Kingsbury, L. Mangu, G. Saon, H. Soltau, and G. Zweig. “FMPE: Discriminatively trained features for speech recognition.” *Proc. of IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, pp. 961–964, 2005.

- [PMS01] M. Pitz, S. Molau, R. Schluter, and H. Ney. “Vocal tract normalization equals linear transformation in cepstral space.” *Proc. of European Conf. on Speech Communication and Technology*, pp. 2653–2656, 2001.
- [PN03a] M. Pitz and H. Ney. “Vocal tract normalization as linear transformation of MFCC.” *Proc. of European Conf. on Speech Communication and Technology*, pp. 1445–1448, 2003.
- [PN03b] A. Potomianos and S. Narayanan. “Robust recognition of children’s speech.” *IEEE Trans. on Speech and Audio Processing*, **11**(6):603–616, 2003.
- [RGM96] B. Raj, E. Gouvea, P. Moreno, and R. Stern. “Cepstral compensation by polynomial approximation for environment-independent speech recognition.” *Proc. of Int. Conf. on Spoken Language Processing*, pp. 2340–2343, 1996.
- [Ris78] J. Rissanen. “Modeling by shortest data description.” *Automatica*, **14**:465–471, 1978.
- [Ris83] J. Rissanen. “A universal prior for integers and estimation by minimum description length.” *Annals of Statistics*, **11**:416–431, 1983.
- [RJ93] L. Rabiner and B.-H. Juang. *Fundamentals of Speech Recognition*. Prentice Hall, 1993.
- [RS78] L. Rabiner and R. Schafer. *Digital Processing of Speech Recognition*. Prentice Hall, 1978.
- [RV99] Vlasta Radova and Petr Vopalka. “Methods of sentences selection for read-speech corpus design.” *Proceedings of second International Workshop on Text, Speech and Dialogue*, pp. 165–170, 1999.
- [SA97] B. Strobe and A. Alwan. “A model of dynamic auditory perception and its application to robust word recognition.” *IEEE Trans. on Speech and Audio Processing*, **5**:451–464, 1997.
- [SB97] Jan P.H. van Santen and Adam L. Buchsbaum. “Methods for optimal text selection.” *Proc. of European Conf. on Speech Communication and Technology*, pp. 553–556, 1997.
- [Sch78] G. Schwartz. “Estimating the Dimension of a Model.” *Annals of Statistics*, **6**:461–464, 1978.

- [SG01] M. Stuttle and M. Gales. “A mixture of Gaussians front end for speech recognition.” *Proc. of European Conf. on Speech Communication and Technology*, pp. 675–678, 2001.
- [SL01] K. Shinoda and C.-H. Lee. “A structural Bayes approach to speaker adaptation.” *IEEE Trans. on Speech and Audio Processing*, **9**(3):276–287, 2001.
- [Sti] R. Stine. “Model selectin using information theory and the MDL principle.”.
- [SW97] K. Shinoda and T. Watanabe. “Acoustic modeling based on the MDL principle for speech recognition.” *Proc. of European Conf. on Speech Communication and Technology*, pp. 99–102, 1997.
- [SWL99] J. Shen, H. Wang, R. Lyu, and L. Lee. “Automatic selection of phonetically distributed sentence sets for speaker adaptation with application to large vocabulary Mandarin speech recognition.” *Computer Speech and Language*, **13**:79–97, 1999.
- [SYT97] S. Sagayama, Y. Yamaguchi, S. Takahashi, and J. Takahashi. “Jacobian approach to fast acoustic model adaptation.” *Proc. of IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, pp. 835–838, 1997.
- [WJ96] J. Wilpon and C. Jacobsen. “A study of speech recognition for children and the elderly.” *Proc. of IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, pp. 349–352, 1996.
- [WMO96] S. Wegmann, D. McAllaster, J. Orlok, and B. Peskin. “Speaker normalization on conversational telephone speech.” *Proc. of IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, pp. 339–341, 1996.
- [Woo99] P. Woodland. “Speaker adaptation: techniques and challenges.” *Automatic Speech Recognition and Understanding Workshop*, **1**:85–90, 1999.
- [WW04] L. Wang and P. Woodland. “MPE-based discriminative linear transform for speaker adaptation.” *Proc. of IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, pp. 321–324, 2004.
- [WZ01] S. Wang and Y. Zhao. “Online Bayesian tree-structured transformation of HMMs with optimal model selection for speaker adaptation.” *IEEE Trans. on Speech and Audio Processing*, **9**(6):663–677, 2001.
- [YEK01] S. Young, G. Evermann, D. Kershaw, G. Moore, J. Odell, D. Ollason, V. Valtchev, and P. Woodland. *The HTK Book (version 3.1)*. Cambridge University Engineering Department, 2001.

- [ZA00a] Q. Zhu and A. Alwan. “Amplitude demodulation of speech and its application to noise robust speech recognition.” *Proc. of Int. Conf. on Spoken Language Processing*, pp. 341–344, 2000.
- [ZA00b] Q. Zhu and A. Alwan. “On the use of variable frame rate analysis in speech recognition.” *Proc. of IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, pp. 1783–1786, 2000.
- [ZA02] Q. Zhu and A. Alwan. “The effect of additive noise on speech amplitude spectra: A quantitative analysis.” *IEEE Signal Proc. Letters*, **9**(9):275–277, 2002.
- [ZF04] Z. Zhang and S. Furui. “Piecewise-linear transformation-based HMM adaptation for noisy speech.” *Speech Communication*, **42**:43–58, 2004.
- [ZR96] P. Zolfaghari and T. Robinson. “Formant analysis using mixtures of Gaussians.” *Proc. of Int. Conf. on Spoken Language Processing*, pp. 1229–1232, 1996.
- [ZR97] P. Zolfaghari and T. Robinson. “A segmental formant vocoder based on linearly varying mixture of Gaussians.” *Proc. of European Conf. on Speech Communication and Technology*, pp. 425–428, 1997.