COMPACT ACOUSTIC MODELING TECHNIQUES FOR SPEECH RECOGNITION

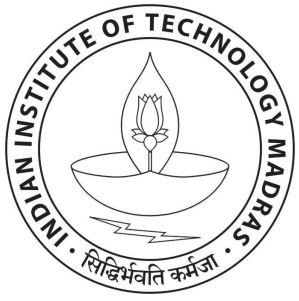
A Project Report

submitted by

VIMAL M

in partial fulfilment of the requirements
for the award of the degree of

BACHELOR OF TECHNOLOGY



DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY MADRAS.

MAY 2013

THESIS CERTIFICATE

This is to certify that the thesis titled **Compact Acoustic Modeling Techniques for Speech Recognition**, submitted by **Vimal M (EE09B041)**, to the Indian Institute of Technology, Madras for the award of the degree **Bachelor of Technology**, is a bona fide record of the research work done by him under my supervision. The contents of the thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Prof. Umesh S
Research Guide
Professor
Department of Electrical Engineering
IIT Madras, 600 036

Place: Chennai
Date: 17th May 2013

ACKNOWLEDGEMENTS

First, I would like to thank my project adviser, Dr. Umesh S. He has continuously encouraged to challenge myself and explore the domain to its depths and intricacies. He has provided me with a research atmosphere that allowed me delve into pressing questions in the field. He has been a constant source of inspiration and support for me. Thank you.

I would like to thank my lab mates, Bhargav, Vishnu, Neethu, Pavan, Raghu and others, for their valuable inputs to help me through with ideas. I would like to thank them for making the entire project a wonderful learning experience. I would like to thank all my friends for their continuous support through my four years at this institute.

I would like to thank all my professors and teachers for their continuous mentoring throughout the course of my studies. I would like to thank Dr. Hema Murthy, Department of Computer Science, for introducing me to the field of Speech Technology and opening me to the innumerable research opportunities in the field. I would like to thank the Department of Electrical Engineering for providing me a unique learning experience that enables to compete with the best in the world. I would like to thank IIT Madras for providing me exciting opportunities and making my whole undergraduate education a part of my life that is worth remembering forever.

I would like to thank my parents for their love and encouragement throughout the course of my studies and my entire life. I have no words to express my gratitude to my parents for making me what I am today.

ABSTRACT

KEYWORDS: GMM; SGMM; Phone CAT; MLLR

In this thesis, a new acoustic modeling technique, the Transform-based Phone CAT Model, for Speech Recognition is proposed. The technique is inspired from the Transform-based Cluster Adaptive Training (CAT) technique used for rapid speaker adaptation of Gaussian Mixture Models (GMMs). Analogous to the CAT, a compact canonical model is adapted through piecewise linear transformations to a set of cluster models representing the phones. The parameters of the distributions in the tied context-dependent phone states are modeled as weighted linear interpolation of the phone cluster models. Thus the tied-state GMM parameters lie in a subspace defined by the linear transformations of the canonical model, and can be conveniently generated using low-dimensional state vectors, which capture the phone context information. The model has significantly lower number of state-specific parameters than the conventional Continuous Density Hidden Markov Model (CDHMM) and outperforms the conventional model by 14.1% relative Word Error Rate (WER) improvement for Resource Management Task. This modeling technique is similar to the Subspace Gaussian Mixture Model (SGMM), and hence offers scope for similar applications and model improvements.

Contents

List of Tables

List of Figures

ABBREVIATIONS

ASR Automatic Speech Recognition

CAT Cluster Adaptive Training

CDHMM Continuous Density Hidden Markov Model

CMN Cepstral Mean Normalization

CMS Cepstral Mean Subtraction

EM Expectation Maximization

GMM Gaussian Mixture Model

HMM Hidden Markov Model

MFCC Mel-frequency Cepstral Coefficients

MLLR Maximum Likelihood Linear Regression

p.d.f. Probability Density Function

RM Resource Management

SGMM Subspace Gaussian Mixture Model

WER Word Error Rate

UBM Universal Background Model

Chapter 1

Introduction

Automatic Speech Recognition (ASR) is a prominent field that aims at conversion of spontaneous speech into machine understandable text. It is a difficult problem because of the different kinds of variability in speech due to changes in speaker and environment. Statistical parametric models like HMM are generally used to model the production of speech sounds. The performance of the speech recognition systems entirely depends on the how good the modeling is and how well the parameters of the model can be estimated using the available training data. There is a lot of focus on using compact modeling techniques that can be easily trained with limited resources. This is of particular interest in the context of Indian languages, many of which have considerably less data resources than English and other European languages.

In conventional CDHMM systems that are typically used in speech recognition applications, the p.d.f. of each HMM state is a Gaussian Mixture Model (GMM). A lot of parameters (means, variances and weights) are required to define these GMMs, thus demanding a large amount of training data. A relatively new acoustic modeling technique, known as SGMM, was introduced in Povey (2009), which takes advantage of the high correlation between the state's distributions to generate the GMM parameters indirectly using only a small number of state-specific parameters. The state GMM parameters are constrained to lie in a low dimensional subspace of the total parameter space. The parameters that are used to define this subspace are shared among all the states and thus can be estimated robustly using limited amount of data and even out-of-domain data. This has been verified through several multilingual experiments (Burget *et al.* (2010), Mohan *et al.* (2012)).

In this thesis, a new acoustic modeling technique, the transform-based Phone CAT, is developed. The model is based on the same principle as the SGMM – to use a compact subspace model that can generate the GMMs using few state-specific parameters. It is inspired from the CAT technique for rapid speaker adaptation (Gales (2000)) and models the phone models in a way analogous to the speaker models in CAT. It consists of a compact canonical model

that is adapted through piece-wise MLLR transforms to the phone models. The tied context-dependent phone state models are expressed as a linear combination of these phone models. The idea of using the phone models to define the subspace was introduced in Srinivas *et al.* (2013).

Chapter 2 describes the basic theory on which the new model is based on. Sections 2.2 and 2.3 outline the basic HMM-based speech recognition procedure and the conventional HMM-GMM system. Sections 2.4 and 2.5 describe the SGMM and the CAT techniques from which the new model is inspired from. Chapter 3 describes in depth the modeling technique and the estimation procedure of the transform-based Phone CAT models. Chapter 4 describes the results of some of the experiments conducted using this model. Chapter 5 gives the conclusions.

Chapter 2

Speech Recognition - A Review

2.1 Introduction

Information in the real world is communicated in the form of signals. Most of these signals (like speech signals) are generated continuously in time and are analog in nature (can take continuous values). But in all practical applications, we can extract only a finite number of samples of the signal and they need to be quantized to take only a finite number of values. The statistics of signals such as speech vary over time and hence are non-stationary. But they can be assumed to be stationary over a short observation window ($25ms$) and fall into a category of pseudo-stationary signals. This allows us to model the signals with efficient parametric models.

The models used to characterize signals can be broadly classified into deterministic and statistical models. Signals like speech can be modeled as the outcome of a random process and the parameters of this process can be estimated accurately. For temporal pattern recognition applications like speech recognition, stochastic models known as Hidden Markov Models (HMM) are widely used. It is called "hidden" because the underlying states are not observed; but only the output of the states are observed. The output is conventionally modeled to be generated from a Gaussian Mixture Model (GMM). This is referred to as the HMM-GMM system.

Section 2.2 gives a brief introduction to the speech recognition problem and the HMM-based speech recognition system. Section (2.3) describes the conventional HMM-GMM system. The subsequent sections reviews more complex approaches to modeling and adapting the GMM-based systems. Section 2.4 describes the Subspace Gaussian Mixture Model (SGMM) based system. Section 2.5 describes the Cluster Adaptive Training (CAT) of GMM models.

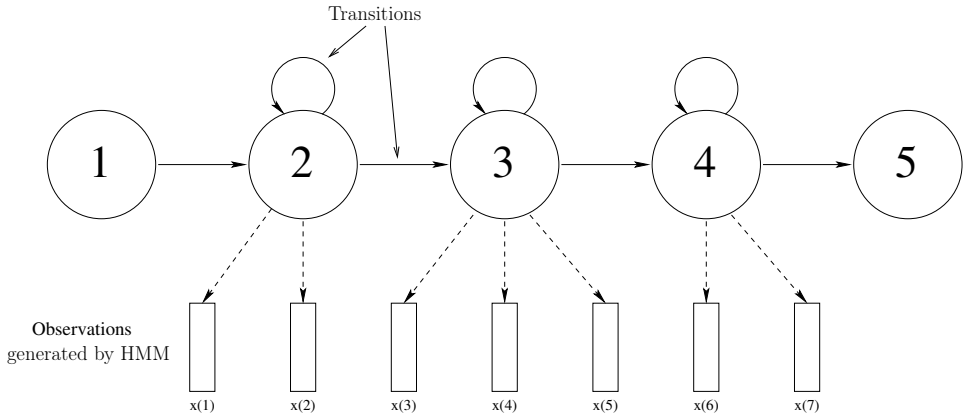


Figure 2.1: Typical left-to-right Hidden Markov model with 3 emitting states used in Speech Recognition.

2.2 HMM-based Speech Recognition

The pseudo-stationary property of speech signals allows the speech signal to be divided into $25ms$ observation windows. The statistical properties of the signal can be assumed to be constant over this window. The data in this window is converted into discrete parameter vectors. This process of conversion of continuous speech signal into a sequence of discrete vectors is known as *Feature Extraction*. These vectors are also known as feature vectors or observation vectors. One of the most widely used features is the Mel Frequency Cepstral Coefficients (MFCC). The objective of the speech recognition system is to convert this sequence of observations into a sequence of symbols (or words) that can be "understood" by a machine.

The observation sequence can be modeled as to be generated by a sequence of states as defined by a HMM. A typical acoustic modeling uses a 3-state left-to-right HMM topology (Fig. 2.1) to model the features generated by a single phonetic unit. A first order Hidden Markov process is assumed meaning that the transition into a particular state depends only on the previous state and that the observation depends only on the current state. The following characterizes the HMM:

- N, the number of states in the model. The set of states in the model is defined by $S = \{S_1, S_2..., S_N\}$. The state at the observation window or frame t is denoted as q_t. For the model of a basic phonetic unit such as a phoneme, we typically use $N = 3$.
- A, the state transition probability distribution. $A = \{a_{ij}\}$ where

$$a_{ij} = P\left[q_t = S_j | q_{t-1} = S_i\right] 1 \leq i, j \leq N. \tag{2.1}$$

For speech systems, we use a left-to-right topology, which implies that $a_{ij} = 0$ for $j < i$.

- π, the initial state distribution. $\pi = \{\pi_i\}$ where

$$\pi_i = P\left[q_0 = S_i\right], 1 \leq i \leq N. \tag{2.2}$$

The model of a basic phonetic unit such as phoneme has $\pi_i = 0$ for $i \neq 1$.

- The observation probability distribution in state j. In the case of a discrete HMM with output vectors $v_1, v_2, ..., v_k$, the probability of observing $v_k$in the state j is given by

$$b_j(k) = P\left[v_k \text{ at } t | q_t = S_j\right], 1 \leq j \leq N,\ 1 \leq k \leq M \quad . \tag{2.3}$$

The observation vector, $\vec{x}(t)$, can assumed to be generated from a continuous distribution. The probability density function (p.d.f.) can be modeled as a mixture of Gaussians or a GMM:

$$b_j(\mathbf{x}(t)) = P\left[\mathbf{x}(t) | q_t = S_j, \boldsymbol{\mu}_{ji}, \boldsymbol{\Sigma}_{ji}\right] = \sum_{i=1}^{I} w_{ji} \mathcal{N}\left(\mathbf{x}(t); \boldsymbol{\mu}_{ji}, \boldsymbol{\Sigma}_{ji}\right), 1 \leq j \leq N, \tag{2.4}$$

where I is the number of Gaussians in the GMM; $\boldsymbol{\mu}_{ji}, \boldsymbol{\Sigma}_{ji}$ are the means and the covariance matrix of the Gaussian component i of state j; and $w_{ji}$is the Gaussian prior or the Gaussian weight with the constraint $\sum_{i=1}^{I} w_{ji} = 1$.

The parameters of the HMM can be put together as a parameter set λ.

2.2.1 The HMM-based solutions

We try to solve three basic problems in HMM-based systems. These problems have been discussed in details in Rabiner Tutorial (Rabiner (1989)). A brief overview of the problems is given in this section.

Problem 1

Given the observation sequence $\mathbf{x} = \mathbf{x}(1)\mathbf{x}(2)\mathbf{x}(3)\ldots\mathbf{x}(T)$ and model parameters λ, how do we compute $P(\mathbf{x}|\lambda)$ i.e. the probability of the observation sequence given the model?

To get this probability, we need to marginalize over all state sequences. Given the state sequence and the parameters, the probability of the observation sequence $P(\mathbf{x}|Q, \lambda)$ is just the

product of the observation probabilities at every t.

$$P\left(\mathbf{x}|Q,\lambda\right)=\prod_{t=1}^{T}b_{q_t}\left(\mathbf{x}(t)\right). \tag{2.5}$$

The state sequence Q is simply a Markov chain with the probability,

$$P\left(Q|\lambda\right)=\pi_{q_0}\prod_{t=1}^{T}a_{q_{t-1}q_t} \tag{2.6}$$

$$\begin{aligned}
P\left(\mathbf{x}|\lambda\right) &= \sum_{all\,Q}P\left(\mathbf{x},Q|\lambda\right)\\
&= \sum_{all\,Q}P\left(Q|\lambda\right)P\left(\mathbf{x}|Q,\lambda\right)\\
&= \sum_{all\,Q}\pi_{q_1}b_{q_1}\left(\mathbf{x}(1)\right)a_{q_1q_2}b_{q_1}\left(\mathbf{x}(1)\right)\ldots a_{q_{T-1}q_T}b_{q_T}\left(\mathbf{x}(T)\right) &(2.7)\\
&= \max_{all\,Q}\pi_{q_1}b_{q_1}\left(\mathbf{x}(1)\right)a_{q_1q_2}b_{q_1}\left(\mathbf{x}(1)\right)\ldots a_{q_{T-1}q_T}b_{q_T}\left(\mathbf{x}(T)\right) &(2.8)
\end{aligned}$$

Actual implementation of (2.7) is through the efficient Forward Backward algorithm. The summation over all Q can be further pruned to the most likely sequences for efficiency. The limiting case (2.8) is where only the sequence of the highest probability is chosen. This is the famous Viterbi algorithm that can be implemented easily in real time.

Problem 2

Given the observation sequence $\mathbf{x}=\mathbf{x}(1)\mathbf{x}(2)\mathbf{x}(3)\ldots\mathbf{x}(T)$ and model parameters λ, how do we find the state sequence $Q=q_1q_2\ldots q_T$ that best explains the observation sequence?

We need to find $\arg\max_Q P\left(Q|\mathbf{x},\lambda\right)$. This can be computed as

$$\begin{aligned} \arg\max_{Q} P\left(Q|\mathbf{x},\lambda\right) &= \arg\max_{Q} \frac{P\left(Q|\lambda\right)P\left(\mathbf{x}|Q,\lambda\right)}{P\left(x|\lambda\right)} \\ &= \arg\max_{Q} P\left(Q|\lambda\right)P\left(\mathbf{x}|Q,\lambda\right). \end{aligned} \tag{2.9}$$

This equation (2.9) is just the same Viterbi search algorithm as in (2.8). This is the problem that is solved when attempting the recognize the word sequence that can best recognize the speech frames.

Problem 3

How do we find the model parameters λ that can best explain the observations?

If O is the sequence of all observations, we need to maximize $P\left(O|\lambda\right)$. This is done using Baum-Welch algorithm, which is a specific case of the Expectation Maximization (EM) algorithm.

2.2.2 Linguistic units

The basic linguistic unit that we model is the phoneme (also referred to as monophones or just phones). There are around 40 phones in English language. Using only these gives a very simplistic model. For large vocabulary recognition, we need to look at the left and the right context of the phone; i.e. we need to model the co-articulation in vocal tract by considering the phones uttered before and after the phone in consideration. Such a model is called a triphone model. There are as many as 40^3 triphones possible, but many of them are not used or are not observed in the training data. The GMMs used to model the triphones have many parameters to be estimated. We require a large amount of data to get a good estimate of the parameters. So, we "tie" similar triphones using a decision-tree based top-down clustering approach. The decision tree based clustering has been described in detail in Young *et al.* (1994). At the end of such a clustering process, we get a few thousand triphone models.

2.3 HMM-GMM system

HMM-GMM system, also known as CDHMM system, is the conventionally used system for speech recognition. It models each context-dependent phone (usually the triphone) with a generative model based on a left-to-right three state HMM topology. The total number of context-dependent phonetic states after tree-based clustering is of the order of a few thousands. Each state is denoted by the index j with $1 \leq j \leq J$. The observation vector is assumed to be generated within each HMM state j from a GMM:

$$P\left(\mathbf{x}|j\right) = \sum_{i=1}^{M_j} w_{ji} \mathcal{N}\left(\mathbf{x}; \boldsymbol{\mu}_{ji}, \boldsymbol{\Sigma}_{ji}\right), \tag{2.10}$$

where $\mathbf{x}$ is the observation vector, w_{ji},$\boldsymbol{\mu}_{ji}$ and $\boldsymbol{\Sigma}_{ji}$ are the prior, mean and covariance matrix of the i^{th} Gaussian component and M_j is the number of Gaussians in the j^{th} state.

2.4 Subspace Gaussian Mixture Model (SGMM)

SGMM is similar to the GMM-based system, but the model parameters for each state are specified by a single state vector $\mathbf{v}_j$. Thus $\boldsymbol{\mu}_{ji}$ lies in a state-independent subspace defined by the columns of $\mathbf{M}_i$. The covariance is shared across all states, so that we have a state-independent $\boldsymbol{\Sigma}_i$. The basic model can be expressed as:

$$\begin{aligned} P\left(\mathbf{x}|j\right) &= \sum_{i=1}^{I} w_{ji} \mathcal{N}\left(\mathbf{x}; \boldsymbol{\mu}_{ji}, \boldsymbol{\Sigma}_i\right) && (2.11) \\ \boldsymbol{\mu}_{ji} &= \mathbf{M}_i \mathbf{v}_j && (2.12) \\ w_{ji} &= \frac{\exp\left(\mathbf{w}_i^T \mathbf{v}_j\right)}{\sum_{i'=1}^{I} \exp\left(\mathbf{w}_{i'}^T \mathbf{v}_j\right)}, && (2.13) \end{aligned}$$

where $\mathbf{v}_j \in \mathbb{R}^S$ is the state projection vector, $\mathbf{x}$ is the feature vector, $\mathbf{M}_i$ and $\mathbf{w}_i$ define the subspaces in which the means and the unnormalized log weights respectively lie and $\boldsymbol{\Sigma}_i$ is

the shared covariance. j is the index of the context-dependent state $(1 \leq j \leq J)$ with J in the order of a few thousands. i is the Gaussian index in the GMM of I mixtures (usually $200 < I < 2000$). $\mathbf{v}_j$ is the only state specific parameter. $\mathbf{M}_i, \mathbf{w}_i, \mathbf{\Sigma}_i$ are "shared" parameters.

The basic strategy of the SGMM is to reduce the number of state specific parameters and increase the number of shared (global) parameters. The intuition is that the means of the tied state models span a smaller subspace of the entire acoustic space. This allows us to reduce the number of state specific parameters. Also, since the global parameters do not depend on a specific phone, there is a lot of data available to train the parameters. It is possible to train these parameters using out-of-domain data even from other languages as shown in Povey *et al.* (2011*a*).

2.4.1 Training procedure

The training of the SGMM system begins with the traditional HMM-GMM system. First, a large GMM consisting of all the gaussians in the HMM-GMM system is built. This is typically in the order of tens of thousands. The gaussians are repeatedly merged to get a desired number of gaussians with diagonal covariances. The actual procedure of doing this can be found in Povey *et al.* (2011*a*). These gaussians are trained with around 8 iterations of EM algorithm for full covariance re-estimation. The resulting model is called a Universal Background Model (UBM). The UBM can be viewed as a compact model representing all kinds of speech from all speakers. The UBM need not necessarily be built from a specific HMM-GMM system; any generic UBM can be used. This UBM is used to initialize the SGMM model. This is done in such a way that the initial p.d.f. of all states is equal to the UBM. The HMM-GMM system provides the Viterbi alignments for the initial SGMM parameter re-estimation iterations. Once the SGMM parameters are estimated by EM algorithm to a sufficient extent, the SGMM training can be continued with self-alignment (alignments from the SGMM itself).

2.5 Cluster Adaptive Training (CAT)

CAT is a very popular method for speaker adaptive training of speech models. Having one GMM model (for each context-dependent state) to model the variability across all the speakers and environments, known as Speaker Independent (SI) models, results in models having a high variance. These models have lower discriminatory capabilities for specific speakers when compared to models, known as Speaker Dependent (SD) models, that are trained for that specific speaker. SD models are practically not feasible considering the large amount of data required from the user of speech recognition system. To overcome this difficulty, speaker adaptation was introduced to adapt the models to specific speakers during testing time (Woodland (2001)). Speaker adaptive training takes a step ahead to apply the adaptation techniques during the training stage (Anastasakos *et al.* (1996)). It builds a compact canonical model to model the average variability of the training speakers. It maps the compact model to speaker dependent models using speaker adaptation techniques.

Section 2.5.1 gives an overview of the model. A detailed description of the model can be found in Gales (2000).

2.5.1 Model-based CAT

The basic model consists of a set of P speaker clusters, each having a cluster dependent mean for each Gaussian component in the HMM model. The speaker dependent model for the Gaussian component is a linear combination of the P cluster means with the interpolation weights $\boldsymbol{\lambda} = \begin{bmatrix} \lambda_1 & \dots & \lambda_P \end{bmatrix}$. Sometimes the speaker independent characteristics are represented with the P^{th} cluster as the bias cluster having the weight $\lambda_P = 1$. The interpolation weights $\boldsymbol{\lambda}$ are different for different Gaussian components. This model is shown in Fig. 2.2.

A regression class tree based clustering can be done on the Gaussian components to cluster them into R disjoint cluster weight classes, $M_w^{(1)}$ to $M_w^{(R)}$, so that a different set of interpolation weights $\boldsymbol{\lambda}^{(sr)}$ is used for each regression class r. For a particular Gaussian component $m \in$

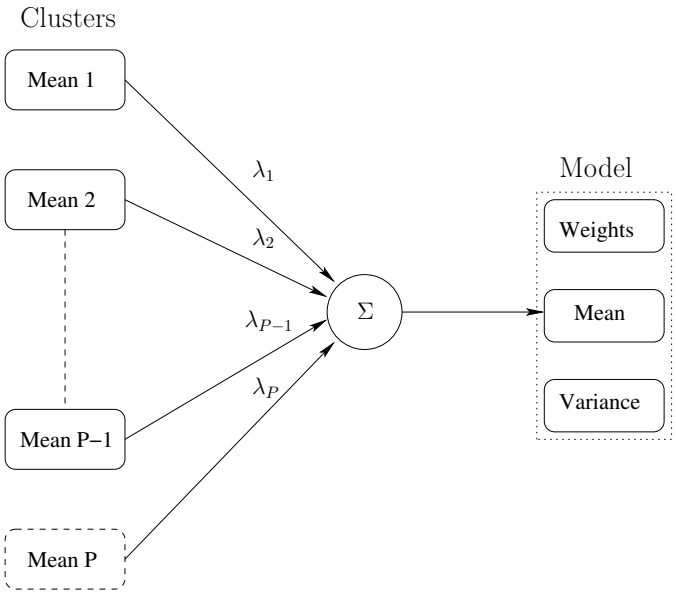


Figure 2.2: Cluster Adaptive Training

$M_w^{(r_m)}$, the model mean for speaker $1 \leq s \leq S$ is given by

$$\boldsymbol{\mu}^{(sm)} = \mathbf{M}^{(m)} \boldsymbol{\lambda}^{(sr_m)}, \tag{2.14}$$

where $\mathbf{M}^{(m)}$ is the matrix of P cluster means for component m,

$$\mathbf{M}^{(m)} = \left[\begin{array}{ccc} \boldsymbol{\mu_c}^{(m1)} & \ldots & \boldsymbol{\mu_c}^{(mP)} \end{array} \right] \tag{2.15}$$

$$\boldsymbol{\lambda}^{(sr_m)} = \left[\begin{array}{ccc} \lambda_1^{(sr_m)} & \ldots & \lambda_P^{(sr_m)} \end{array} \right] \tag{2.16}$$

where $\boldsymbol{\mu_c}^{(mp)}$ is the mean of the Gaussian component m associated with the speaker cluster p, $\boldsymbol{\lambda}^{(sr_m)}$ is the cluster weight vector for speaker s and r_m is the cluster weight class to which Gaussian component m belongs. Thus the model-based CAT parameters are the model parameters $\mathcal{M} = \left\{ \left\{ \begin{array}{ccc} \mathbf{M}^{(1)} & \ldots & \mathbf{M}^{(M)} \end{array} \right\}, \left\{ \begin{array}{ccc} \boldsymbol{\Sigma}^{(1)} & \ldots & \boldsymbol{\Sigma}^{(M)} \end{array} \right\} \right\}$, where $\boldsymbol{\Sigma}^{(m)}$ is the covariance of Gaussian component m and cluster weight vector parameters $\Lambda = \left\{ \boldsymbol{\lambda}^{(sr)} \right\}$, $1 \leq r \leq R$, $1 \leq s \leq S$.

The re-estimation of the CAT parameters can be done in the Maximum Likelihood (ML) framework. The training procedure aims at maximizing the likelihood of the training data given the model parameters. Similar to the HMM training, the EM algorithm is used to estimate the parameters. An iterative training scheme is followed:

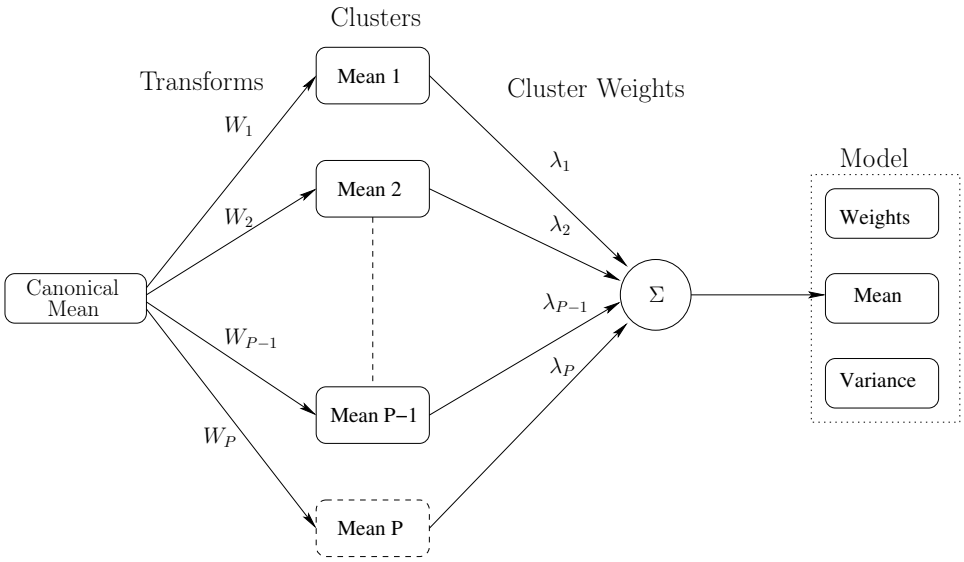


Figure 2.3: Transform-based Cluster Adaptive Training

1. Estimate the cluster weight vectors Λ fixing the canonical model parameters.
2. Estimate the canonical model parameters $\mathcal{M}$ fixing the cluster weight vectors.
3. Repeat until convergence.

2.5.2 Transform-based CAT

In the transform-based CAT, the cluster-specific means are not directly specified, but are obtained as cluster-specific linear transformations of the canonical means. The MLLR transform (Leggetter and Woodland (1995)) is generally used for this. The Gaussian components are clustered into Q transform classes, $M_t^{(1)}$ to $M_t^{(Q)}$ based on regression tree clustering. Each transform class q has a transform $\mathbf{W_c}^{(pq)}$ associated with each cluster p, where $\mathbf{W_c}^{(pq)} = \left[\begin{array}{cc} \mathbf{A_c}^{(pq)} & \mathbf{b_c}^{(pq)} \end{array} \right]$. The model is as shown in Fig. 2.3.

The cluster-specific mean of the Gaussian component $m \in M_t^{(t_m)}$ associated with cluster p is given by

$$\boldsymbol{\mu_c}^{(mp)} = \mathbf{A_c}^{(pt_m)} \boldsymbol{\mu_a}^{(m)} + \mathbf{b_c}^{(pt_m)} = \mathbf{W_c}^{(pt_m)} \boldsymbol{\xi_a}^{(m)}, \tag{2.17}$$

where $\boldsymbol{\mu_a}^{(m)}$ is the canonical mean of the Gaussian component m, $\boldsymbol{\xi_a}^{(m)} = \left[\begin{array}{cc} \boldsymbol{\mu_a}^{(m)} & 1 \end{array} \right]^T$ is the extended mean vector of the canonical mean, $1 \leq t_m \leq Q$ is the transform class of the

Gaussian component m. The mean for a particular speaker can be obtained similar to 2.14,

$$\boldsymbol{\mu}^{(sm)} = \left(\sum_{p=1}^{P} \lambda_p^{(sr_m)} \mathbf{W_c}^{(pt_m)} \right) \boldsymbol{\xi_a}^{(m)}. \tag{2.18}$$

Thus the transform-based CAT has the same cluster weight vector parameters as in model-based CAT; but the model parameters consist of the transform parameters $\mathcal{T} = \left\{ \mathbf{W_c}^{(pq)} \right\}, 1 \leq p \leq P, 1 \leq q \leq Q$ and the Canonical model parameters $\tilde{\mathcal{M}} = \left\{ \left\{ \begin{array}{cccc} \boldsymbol{\mu_a}^{(1)} & \ldots & \boldsymbol{\mu_a}^{(M)} \end{array} \right\}, \left\{ \begin{array}{cccc} \boldsymbol{\Sigma}^{(1)} & \ldots & \boldsymbol{\Sigma}^{(M)} \end{array} \right\} \right\}$.

The re-estimation procedure for the transform-based CAT is similar to the model-based CAT. But unlike the model-based version, the model parameters need to be estimated in two stages for $\mathcal{T}$ and $\tilde{\mathcal{M}}$ respectively. The training process is as follows:

1. Estimate the cluster weight vector parameters Λ given the model parameters $\left\{ \begin{array}{cc} \mathcal{T} & \tilde{M} \end{array} \right\}$.
2. Estimate the MLLR transform parameters $\mathcal{T}$ given the cluster weight vectors Λ and the Canonical model parameters $\tilde{\mathcal{M}}$.
3. Estimate the Canonical model parameters $\tilde{\mathcal{M}}$ given the cluster weight vectors Λ and the transform parameters $\mathcal{T}$.
4. Go to 2 until convergence.
5. Go to 1 until convergence.

The auxiliary function, the sufficient statistics and the update equations for both the model-based and transform-based CAT are given in detail in Gales (2000).

Chapter 3

Transform-based Phone CAT

Phone CAT (Srinivas *et al.* (2013)) is an acoustic modeling technique inspired from the Cluster Adaptive Training (CAT) (Gales (2000)) for rapid speaker adaptation, which was described in Section 2.5. While the CAT adapts a speaker independent model to different clusters of speakers, the Phone CAT adapts a Universal Background Model (UBM) to a set of clusters representing the phones (monophones). The context-dependent phone (triphones) states are modeled as linear weighted interpolations of the phone cluster models, just as in the case of CAT where the model means for a speaker are obtained as a linear weighted interpolation of the cluster means corresponding to different speakers. The context information of the phone is captured in the form of a linear interpolation weight vector. This technique has many similarities to the SGMM (Povey *et al.* (2011*a*)), described in Section 2.4.

In this thesis, a new technique inspired from the transform-based CAT is introduced. This technique exploits the correlations in the acoustic space between the distributions of the context-dependent phone states and gives a very compact representation using a UBM and several MLLR transforms.

Section 3.1 briefly describes the model-based Phone CAT technique. Section 3.2 introduces the Transform-based Phone CAT model. Sections 3.3 and 3.4 describe in detail the initialization of the model and the training procedure. Section 3.5 describes the extensions possible to the basic model.

3.1 Model-based Phone CAT

The model-based Phone CAT consists of a set of P clusters corresponding to the P monophone models. Each cluster p has a cluster-specific mean $\boldsymbol{\mu}_i^{(p)}$ for each Gaussian component $1 \leq i \leq I$. Each state j corresponding to a context-dependent HMM state is expressed as linear

combination of the P cluster means with the interpolation weights $\mathbf{v}_j$, which is called as the state vector. Thus the mean of the i^{th} Gaussian of the j^{th} state is modeled as follows:

$$\boldsymbol{\mu}_{ji} = \mathbf{M}_i \mathbf{v}_j, \tag{3.1}$$

where $\mathbf{v}_j = \begin{bmatrix} v_j^{(1)} & v_j^{(2)} & \dots & v_j^{(P)} \end{bmatrix}$ is the state vector, and $\mathbf{M}_i = \begin{bmatrix} \boldsymbol{\mu}_i^{(1)} & \boldsymbol{\mu}_i^{(2)} & \dots & \boldsymbol{\mu}_i^{(P)} \end{bmatrix}$ is the matrix obtained by stacking the i^{th} mean of all the P phone clusters, where $\boldsymbol{\mu}_i^{(p)}$ is the mean of the i^{th} Gaussian of the p^{th} cluster.

The Model-based Phone CAT has 2 distinct model sets. At the lower level, there is a set of P monophone models. The monophone models cannot model the context. So, at the higher level, there are J triphone model states. The Model-based Phone CAT assumes that each of these tied states has a strong relation to the P monophone models; that it lies in a subspace spanned by the monophone models. (3.1) represents this relation. The monophone means $\boldsymbol{\mu}_i^{(1)}$, $\boldsymbol{\mu}_i^{(2)}$, $\dots$, $\boldsymbol{\mu}_i^{(P)}$ form the basis vectors of this subspace. During the training process, both the basis vectors and the interpolation weights are re-estimated; with the model in effect learning a better subspace.

3.2 Transform-based Phone CAT

In the transform-based Phone CAT, the means of the P clusters, corresponding to the P monophones[1], are not specified directly, but as linear transformations of the means of a canonical model. In the basic model, there is an MLLR transform, $\boldsymbol{\mathcal{W}}_p$, associated with each cluster p. The cluster-specific mean $\boldsymbol{\mu}_i^{(p)}$ for Gaussian component i is specified as:

$$\boldsymbol{\mu}_i^{(p)} = \boldsymbol{\mathcal{W}}_p \boldsymbol{\xi}_i = \boldsymbol{\mathcal{W}}_p \begin{bmatrix} \boldsymbol{\mu}_i & 1 \end{bmatrix}^T, \tag{3.2}$$

where $\boldsymbol{\xi}_i$ is the extended canonical model mean $\begin{bmatrix} \boldsymbol{\mu}_i & 1 \end{bmatrix}^T$ with $\boldsymbol{\mu}_i$ being the canonical mean of the i^{th} Gaussian. The mean for the i^{th} Gaussian of the context-dependent state j is expressed

[1]This is a typical mapping. But the P clusters may not necessarily correspond to P monophones.

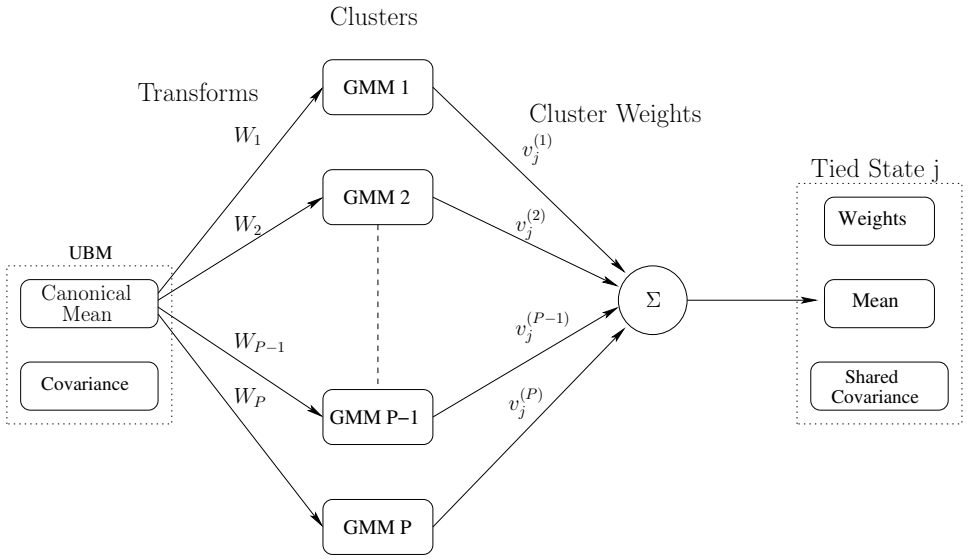


Figure 3.1: Transform-based Phone CAT

as a weighted linear interpolation of the cluster-specific means given in (3.2):

$$\begin{aligned}
\boldsymbol{\mu}_{ji} &= \left[\begin{array}{ccc} \boldsymbol{\mu}_i^{(1)} & \ldots & \boldsymbol{\mu}_i^{(P)} \end{array} \right] \mathbf{v}_j, \\
&= \left[\begin{array}{ccc} \boldsymbol{\mu}_i^{(1)} & \ldots & \boldsymbol{\mu}_i^{(P)} \end{array} \right] \left[\begin{array}{c} v_j^{(1)} \\ \vdots \\ v_j^{(P)} \end{array} \right], \\
&= \sum_{p=1}^{P} \boldsymbol{\mu}_i^{(p)} v_j^{(p)}, \\
&= \left(\sum_{p=1}^{P} v_j^{(p)} \boldsymbol{\mathcal{W}}_p \right) \boldsymbol{\xi}_i, \qquad (3.3)
\end{aligned}$$

where $\mathbf{v}_j = \left[\begin{array}{ccc} v_j^{(1)} & \ldots & v_j^{(P)} \end{array} \right]^T$ is the state vector. The model is as shown in Fig. 3.1.

The Transform-based Phone CAT model has 3 distinct model sets. At the lowest level, there is a compact canonical model representing the average variability of all the speech data. At the intermediate level, there is a set of P clusters representing the P phone models. These P models are linear transformations, represented by (3.2) , of the canonical model. At the highest level, there is a set of J tied states, whose models are obtained as linear interpolation of the P models in the clusters.

3.2.1 Model description

The transform-based Phone CAT model has a GMM as the generative model in each context-dependent state. But the means are not specified directly, but with a mapping from the the P dimensional state vector $\mathbf{v}_j$. The covariance matrix $\mathbf{\Sigma}_i$ is diagonal and shared across all the context-dependent states. The weights are expressed through a subspace model similar to the SGMM (2.13). The model can be expressed as:

$$P\left(\mathbf{x}|j\right) = \sum_{i=1}^{I} w_{ji} \mathcal{N}\left(\mathbf{x}; \boldsymbol{\mu}_{ji}, \mathbf{\Sigma}_i\right), \tag{3.4}$$

$$w_{ji} = \frac{\exp\left(\mathbf{w}_i^T \mathbf{v}_j\right)}{\sum_{i'=1}^{I} \exp\left(\mathbf{w}_{i'}^T \mathbf{v}_j\right)}, \tag{3.5}$$

where $\mathbf{x} \in \mathbb{R}^D$ is the feature vector, $1 \leq j \leq J$ is the state index of the context-dependent state, $\mathbf{w}_i$ is the weight projection vector, $\boldsymbol{\mu}_{ji}$ is obtained as in (3.3) and I is the number of Gaussian components in the GMM. The number of Gaussians I is typically 400 to 4000. In the SGMM, typically a 400 mixture full-covariance matrix is used. Here, since the number of global parameters is lower, the number of mixtures can be higher. If the weights are modeled directly as w_{ji} rather than using (3.5), the number of parameters in the model will be dominated by the weights, which is undesirable. The only state-specific parameters are the state vectors $\mathbf{v}_j$. The rest $\mathbf{w}_i, \mathbf{\Sigma}_i, \boldsymbol{\mu}_i, \boldsymbol{\mathcal{W}}_p$ are global parameters and are independent of state. Hence there is a large amount of data to estimate these parameters.

3.2.2 Overview of the Training procedure

The model training starts with a traditional HMM-GMM system. This provides the phonetic context information (the decision trees), a set of Gaussian mixtures to build a UBM as the canonical model and the Viterbi state alignments for the initial training iterations. The model is initialized using these and trained for a few iterations using the alignments obtained from the HMM-GMM system. In the next phase of training, the alignments are obtained from the transform-based Phone CAT system itself. There are three distinct parameter sets

as in the case of CAT. The state vector parameters $\Lambda = \{\mathbf{v}_j\}, 1 \leq j \leq J$, canonical model parameters $\mathcal{M} = \left\{ \left\{ \begin{array}{cccc} \boldsymbol{\mu}_1 & \ldots & \boldsymbol{\mu}_I \end{array} \right\}, \left\{ \begin{array}{cccc} \boldsymbol{\Sigma}_1 & \ldots & \boldsymbol{\Sigma}_I \end{array} \right\} \right\}$ and the subspace parameters $\mathcal{S} = \left\{ \left\{ \begin{array}{cccc} \mathbf{w}_1 & \ldots & \mathbf{w}_I \end{array} \right\}, \left\{ \begin{array}{cccc} \boldsymbol{\mathcal{W}}_1 & \ldots & \boldsymbol{\mathcal{W}}_P \end{array} \right\} \right\}$. The training scheme followed is analogous to the case of transform-based CAT:

1. Re-estimate the state vector parameters Λ using $\{\mathcal{M}, \mathcal{S}\}$ and the pre-update value of Λ.
2. Re-estimate the subspace parameters $\mathcal{S}$ given $\{*, \mathcal{M}\}$ and the pre-update value of $\mathcal{S}$.
3. Re-estimate[2] the canonical model parameters $\mathcal{M}$ given $\{\mathcal{S}, *\}$ and the pre-update value of $\mathcal{M}$.
4. Go to 2 until convergence.
5. Go to 1 until convergence.

The pre-update values are used to calculate the Gaussian posteriors. These values are usually accumulated in the form of statistics. Also practically, this scheme does not have to be followed strictly and different sets of parameters can be updated simultaneously to attain convergence in fewer iterations.

The structure of the model allows efficient pruning of the gaussians that are used for likelihood computation in each frame: only the top few gaussians in the UBM that give the highest likelihood for the frame are selected and used. The statistics accumulated and the update equations are described in Section 3.4.

3.3 Model initialization

First the UBM is trained and it is then used to initialize the transform-based Phone CAT model. The UBM is initialized by a bottom-up-clustering algorithm as in the case of SGMM (Povey *et al.* (2011*a*)). The set of diagonal Gaussians in all the states of the HMM-GMM system is clustered to create a mixture of diagonal Gaussians. This is done by repeatedly merging Gaussians that would result in the least log-likelihood reduction. This mixture of Gaussians is further trained by EM algorithm using all the speech data to get the final UBM.

[2]The canonical means $\boldsymbol{\xi}_i$ must be updated first, and the updated means are used to re-estimate the covariances.

The transform-based Phone CAT model is initialized such that the GMM in each state is identical to the UBM. The MLLR transforms are all set to identity matrices with 0 bias so that all the cluster-specific means are initially identical to the UBM means. The state vectors $\mathbf{v}_j$ is assigned a vector giving a weight 1 to only one cluster depending on a mapping function $\mathcal{C}$ and 0 to every other cluster. Therefore the initialization is:

$$\boldsymbol{\mathcal{W}}_p = \left[\begin{array}{cc} \mathbf{I}_{D\times D} & \mathbf{0}_{D\times 1} \end{array} \right], 1 \leq p \leq P \tag{3.6}$$

$$\boldsymbol{\mu}_i = \boldsymbol{\mu}_i^{(UBM)} \quad , 1 \leq i \leq I \tag{3.7}$$

$$\boldsymbol{\Sigma}_i = \boldsymbol{\Sigma}_i^{(UBM)} \quad , 1 \leq i \leq I \tag{3.8}$$

$$\mathbf{v}_j = \mathbf{e}_k \in \mathbb{R}^P \quad , 1 \leq j \leq J,\ k = \mathcal{C}(j) \tag{3.9}$$

$$\mathbf{w}_i = \mathbf{0} \in \mathbb{R}^P \quad , 1 \leq i \leq I \tag{3.10}$$

where $\mathbf{I}_{D\times D}$ is a $D \times D$ identity matrix with D being the dimension of the feature vector, $\mathbf{0}_{D\times 1}$ is a vector of D zeros, $\boldsymbol{\mu}_i^{(UBM)}$, $\boldsymbol{\Sigma}_i^{(UBM)}$ are the mean and the covariance matrix of the i^{th} Gaussian component of the UBM, $\mathbf{e}_k$ is a P dimensional unit vector with the k^{th} dimension as 1 and every other dimension 0 and $\mathcal{C} : \{1, \ldots, J\} \rightarrow \{1, \ldots, P\}$ is a mapping from the state j to cluster p.

In the simplest case, the mapping function can be defined such that $\mathcal{C}(j) = p$, where p is the index of the central phone of the context-dependent state j. Instead, it is possible to take into account the state in the HMM topology to which j belongs to. If the context-dependent phone has 3 states, the context-dependent states corresponding to each of the 3 states can a mapped to different clusters. If every context-dependent phone has 3 states, then with this mapping the model will end up having $P = 3K$ clusters, where K is the number of phones. Similarly, there can be more complex mapping functions taking into account other context information.

3.4 Training of the model

This section describes the accumulation and the update stages of the training of the model.

3.4.1 Expectation Maximization (EM) algorithm

The auxiliary function to be optimized is similar to ones used in CAT:

$$\mathcal{Q} = \sum_{j,i,t} \gamma_{ji}(t) \left[\log(w_{ji}) - \frac{1}{2}|\boldsymbol{\Sigma}_i| - \frac{1}{2}\left(\mathbf{x}(t) - \boldsymbol{\mu}_{ji}\right)^T \boldsymbol{\Sigma}_i^{-1} \left(\mathbf{x}(t) - \boldsymbol{\mu}_{ji}\right) \right], \tag{3.11}$$

where $\gamma_{ji}(t) = P(j,i|t)$ is the posterior probability of the j^{th} state, i^{th} Gaussian component at time t, $\mathbf{x}(t)$ is the feature vector at time t and w_{ji} and $\boldsymbol{\mu}_{ji}$ are expressed according to (3.3) and (3.5). The rest of the symbols are as defined in Section 3.2.1. The update equations for each of the parameters $\mathbf{v}_j, \boldsymbol{\mathcal{W}}_p, \mathbf{w}_i, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i$ are obtained by optimizing $\mathcal{Q}$ with respect to the parameter keeping the other parameters fixed. The update equations along with the required accumulations are described in the subsequent sections.

3.4.2 Estimation of Cluster Transforms

Gales (2000) gives an efficient method for re-estimation of an entire row of a cluster transform matrix $\boldsymbol{\mathcal{W}}_p$. The update equation for the k^{th} row of $\boldsymbol{\mathcal{W}}_p$ is given by

$$\boldsymbol{\mathcal{W}}_p^{(k)} = \mathbf{k}_p^{(k)} \left[\mathbf{G}_p^{(k)}\right]^{-1}, \tag{3.12}$$

where the accumulates $\mathbf{k}_p^{(k)}$ and $\mathbf{G}_p^{(k)}$ are given by

$$\mathbf{k}_p^{(k)} = \sum_{i=1}^{I} \frac{1}{\sigma_{kk}^{(i)2}} \left[\left\{ k_{pk}^{(i)} - \sum_{l \neq p}^{P} g_{lp}^{(i)} \boldsymbol{\mathcal{W}}_l^{(k)} \boldsymbol{\xi}_i \right\} \boldsymbol{\xi}_i^T \right], \tag{3.13}$$

$$\mathbf{G}_p^{(k)} = \sum_{i=1}^{I} \frac{g_{pp}^{(i)}}{\sigma_{kk}^{(i)2}} \boldsymbol{\xi}_i \boldsymbol{\xi}_i^T \tag{3.14}$$

with $g_{pq}^{(i)}$, $k_{pk}^{(i)}$ being sufficient statistics defined as

$$\mathbf{G}^{(i)} = \left[g_{pq}^{(i)}\right]_{1 \leq p,q \leq P} = \sum_{j,t} \gamma_{ji}(t)\, \mathbf{v}_j \mathbf{v}_j^T, \tag{3.15}$$

$$\mathbf{K}^{(i)} = \left[k_{pk}^{(i)}\right]_{1 \leq p \leq P,\, 1 \leq k \leq D} = \sum_{j,t} \gamma_{ji}(t)\, \mathbf{v}_j \mathbf{x}(t)^T. \tag{3.16}$$

From (3.13), we see that the accumulate for $\mathbf{k}_p^{(k)}$ depends on the set of other cluster transforms $\{\boldsymbol{\mathcal{W}}_{l\neq p}\}$. Therefore, each time a transform is to be updated, the $\mathbf{k}_p^{(k)}$ must be recomputed with the latest updated values of the other cluster transforms. The process is iterative and converges in a few iterations.

The derivation of the update equation (3.12) is given in Appendix A.1.

3.4.3 Estimation of State Vectors

The auxiliary function for state vectors $\mathbf{v}_j$ consists of two parts, one related to the mean and one to the weights. The dependency of the weights on $\mathbf{v}_j$ through (3.5) makes the auxiliary function more complex to optimize. However, by making several approximations, as in Povey (2009), it is possible to get closed-form expression for the update of $\mathbf{v}_j$.

The update equation for $\mathbf{v}_j$ is given by

$$\mathbf{v}_j = \mathbf{G}_j^{-1}\mathbf{k}_j, \tag{3.17}$$

where the accumulates $\mathbf{G}_j$ and $\mathbf{k}_j$ are given by

$$\mathbf{k}_j = \mathbf{y}_j + \sum_{i=1}^{I} \mathbf{w}_i \left(\gamma_{ji} - \gamma_j w_{ji} + \max\left(\gamma_{ji}, \gamma_j w_{ji}\right) \mathbf{w}_i^T \mathbf{v}_j\right), \tag{3.18}$$

$$\mathbf{G}_j = \sum_{i=1}^{I} \gamma_{ji}\mathbf{H}_i + \max\left(\gamma_{ji}, \gamma_j w_{ji}\right) \mathbf{w}_i^T \mathbf{v}_j, \tag{3.19}$$

where $\mathbf{H}_i = \mathbf{M}_i^T \boldsymbol{\Sigma}_i^{-1} \mathbf{M}_i$ and $\mathbf{y}_j$, γ_{ji}, γ_j are sufficient statistics defined as

$$\mathbf{y}_j = \sum_{t,i} \gamma_{ji}(t)\, \mathbf{M}_i^T \boldsymbol{\Sigma}_i^{-1} \mathbf{x}(t), \tag{3.20}$$

$$\gamma_{ji} = \sum_{t} \gamma_{ji}(t), \tag{3.21}$$

$$\gamma_j = \sum_{i} \gamma_{ji} \tag{3.22}$$

with $\mathbf{M}_i = \begin{bmatrix} \boldsymbol{\mathcal{W}}_1\boldsymbol{\xi}_i & \dots & \boldsymbol{\mathcal{W}}_P\boldsymbol{\xi}_i \end{bmatrix}$.

The derivation of the update equation (3.17) is given in Appendix A.2.

3.4.4 Estimation of Canonical model parameters

The canonical model parameter estimation is done exactly like transform-based CAT (Gales (2000)). The update equations for the mean and covariance of the i^{th} Gaussian component are:

$$\boldsymbol{\mu}_i = \left[\sum_{p=1}^{P}\sum_{q=1}^{P} g_{pq}^{(i)} \mathbf{A}_p^T \boldsymbol{\Sigma}_i \mathbf{A}_q\right]^{-1} \left[\sum_{p=1}^{P} \mathbf{A}_p^T \boldsymbol{\Sigma}_i^{-1} \left(\mathbf{k}_p^{(i)T} - \sum_{q=1}^{P} g_{pq}^{(i)} \mathbf{b}_q\right)\right], \tag{3.23}$$

$$\boldsymbol{\Sigma}_i = \text{diag}\left\{\frac{\mathbf{L}^{(i)} - 2\sum_{p=1}^{P} \mathbf{k}_p^{(i)} \left(\mathbf{M}_i^{(p)}\right)^T + \sum_{p=1}^{P}\sum_{q=1}^{P} g_{pq}^{(i)} \mathbf{M}_i^{(p)} \mathbf{M}_i^{(q)T}}{\sum_j \gamma_{ji}}\right\}, \tag{3.24}$$

where $\mathbf{A}_p$ and $\mathbf{b}_p$ are the first D columns and the $(D+1)^{th}$ column of $\boldsymbol{\mathcal{W}}_p = \left[\begin{array}{cc} \mathbf{A}_p & \mathbf{b}_p \end{array}\right]$ respectively, $\mathbf{M}_i^{(p)} = \boldsymbol{\mathcal{W}}_p \boldsymbol{\xi}_i$, $\mathbf{k}_p^{(i)}$ is the p^{th} row of the statistics (3.16), $\mathbf{L}^{(i)}$ is the sufficient statistics defined by

$$\mathbf{L}^{(i)} = \sum_{j,t} \gamma_{ji}(t)\, \mathbf{x}(t)\, \mathbf{x}(t)^T. \tag{3.25}$$

The estimation of $\boldsymbol{\mu}_i$ depends of the current value of $\boldsymbol{\Sigma}_i$ and vice-versa. First, the means are updated and the updated means are used to update $\boldsymbol{\Sigma}_i$. The derivation of the update equations (3.23) and (3.24) are given in Appendix A.3.

3.4.5 Estimation of weight projections

The weight projection used is exactly the same as in the case of SGMM (Povey *et al.* (2011*a*)). The same update procedure is used here as well. It is an iterative process with the following

being computed every iteration:

$$\mathbf{w}_i^{(n)} = \mathbf{w}_i^{(n-1)} + \mathbf{F}_i^{(n)-1}\mathbf{g}_i^{(n)}, \tag{3.26}$$

$$\mathbf{F}_i^{(n)} = \sum_j \max\left(\gamma_{ji}, \gamma_j w_{ji}^{(n-1)}\right) \mathbf{v}_j \mathbf{v}_j^T, \tag{3.27}$$

$$\mathbf{g}_i^{(n)} = \sum_j \left(\gamma_{ji} - \gamma_j w_{ji}^{(n-1)}\right) \mathbf{v}_j, \tag{3.28}$$

where $\boldsymbol{.}^{(n)}$ represents the value at the n^{th} iteration.

3.5 Extensions to the model

The model described in Section 3.2.1 can easily be extended by incorporating techniques tried out in similar models. Some of these extensions are described in this section.

3.5.1 Multiple transform classes per cluster

It is possible to use piece-wise linear transformation with multiple MLLR transforms. The I Gaussians in the UBM are clustered into Q transform classes and a different MLLR transform $\boldsymbol{\mathcal{W}}_{pq}$ is used for each class q. The equations (3.12), (3.13) and (3.14) will be similar for this case as well, but the summation of i will not be over $\{1, 2 \ldots I\}$ but over the set of Gaussians in transform class q.

3.5.2 Full Covariance MLLR

The standard CAT for speaker adaptation is done with diagonal covariance. If full covariance is used, then the update equations are quite complex and computationally very expensive, making it practically infeasible. The equation (3.12) is valid only for diagonal covariance. MLLR for full covariance models was introduced in Povey and Saon (2006). The re-estimation is done using a second order gradient descent approach. The same technique can be implemented for transform-based CAT as well. This technique is an iterative approach.

In each iteration, the gradient of the auxiliary function $\mathcal{Q}$ (3.11) w.r.t. $\boldsymbol{\mathcal{W}}_p$ is computed:

$$\begin{aligned}\boldsymbol{\mathcal{L}}_p &= \frac{\partial \mathcal{Q}}{\partial \boldsymbol{\mathcal{W}}_p} \\ &= \sum_{j,i,t} \gamma_{ji}(t)\, \boldsymbol{\Sigma}^{-1} \left(\mathbf{x}(t) - \left(\sum_p \boldsymbol{\mathcal{W}}_p v_j^{(p)} \right) \boldsymbol{\xi}_i \right) \boldsymbol{\xi}_i^T v_j^{(p)} \end{aligned} \tag{3.29}$$

The second order gradient $\boldsymbol{\mathcal{G}}_p^{(k)}$ is also computed for the all dimensions $1 \leq k \leq D$. Here it is assumed that it is equal to the case when $\boldsymbol{\Sigma}^{-1}$ is diagonal. It is obtained as:

$$\begin{aligned}\boldsymbol{\mathcal{G}}_p^{(k)} &= \frac{\partial^2 \mathcal{Q}}{\partial \boldsymbol{\mathcal{W}}_p^{(k)2}} \\ &= \sum_{j,i,t} \gamma_{ji}(t)\, \frac{v_j^{(p)2}}{\sigma_{kk}^{(i)2}} \boldsymbol{\xi}_i \boldsymbol{\xi}_i^T, \\ &= \sum_i \frac{g_{pp}^{(i)}}{\sigma_{kk}^{(i)2}} \boldsymbol{\xi}_i \boldsymbol{\xi}_i^T \end{aligned} \tag{3.30}$$

where $g_{pp}^{(i)}$ is the p^{th} diagonal element of (3.15) and $\sigma_{kk}^{(i)2}$ is the variance of the i^{th} Gaussian. Using (3.29) and (3.30), the entire k^{th} row of $\boldsymbol{\mathcal{W}}_p$ can be estimated,

$$\begin{aligned}\hat{\boldsymbol{\mathcal{W}}}_p^{(k)} &= \boldsymbol{\mathcal{W}}_p^{(k)} + \alpha \left[\frac{\partial^2 \mathcal{Q}}{\partial \boldsymbol{\mathcal{W}}_p^{(k)2}} \right]^{-1} \frac{\partial \mathcal{Q}}{\partial \boldsymbol{\mathcal{W}}_p}, \\ &= \boldsymbol{\mathcal{W}}_p^{(k)} + \alpha \boldsymbol{G}_p^{(k)-1} \boldsymbol{l}_p^{(k)}, \end{aligned} \tag{3.31}$$

where $\boldsymbol{\mathcal{W}}_p^{(k)}$ is the k^{th} row of $\boldsymbol{\mathcal{W}}_p$, $\boldsymbol{l}_p^{(k)}$ is the k^{th} row of $\boldsymbol{\mathcal{L}}_p$ and α is some learning rate.

The change in the auxiliary function $\mathcal{Q}$ (3.11) after update of all rows is computed using

$$\Delta\mathcal{Q} = \Delta \sum_{j,i,t} \left(\gamma_{ji}(t)\, \mathbf{x}(t)^T\, \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_{ji} - 0.5 \gamma_{ji}(t)\, \boldsymbol{\mu}_{ji}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_{ji} \right). \tag{3.32}$$

Appendix A.4 shows how this change can be computed efficiently from the sufficient statistics. If the auxiliary function has increased, we move on to the next iteration; otherwise, then the learning rate is reduced by a factor of $1/2$. $\boldsymbol{\mathcal{W}}_p^{(k)}$ is reset to its original value and $\hat{\boldsymbol{\mathcal{W}}}_p^{(k)}$ is computed again for all k with the new learning rate. The auxiliary function is tested again for increase and the process is repeated until an increase is observed or until a limiting learning rate

value is reached. In the next iteration, the gradients and second gradients are computed again using the new updated value of $\boldsymbol{\mathcal{W}}_p$. After completing the estimation of one MLLR transform $\boldsymbol{\mathcal{W}}_p$, the next transform $\boldsymbol{\mathcal{W}}_{p+1}$ is estimated with (3.29) and (3.30) using the update value for $\boldsymbol{\mathcal{W}}_p$.

In order to expedite the process, it might be possible to estimate all the D rows of $\boldsymbol{\mathcal{W}}_p$ in parallel, with all computations for one row in (3.31) done independent of the other rows.

Chapter 4

Results

4.1 Experimental setup

The performance of the Transform-based Phone CAT model is tested on the Resource Management (RM) (Price *et al.* (1993)) and the Aurora 4 tasks. Only the speaker independent training set (RM1) of the RM task was used. It has a total of 3,360 recorded sentence utterances from 80 different speakers. The test material consists of 5 DARPA benchmark tests, each containing 300 test utterances. A combined test set containing utterances from all these 5 tests was used for evaluating the models. The Aurora 4 database is derived from the Wall Street Journal (WSJ0) task. It has 7138 continuous utterances for training, equivalent to nearly 15hrs of training data. The test set consists of utterances with 14 different noise and channel effects. Only the 330 utterances of the clean set with no channel effect was considered for evaluating the models.

13-dimensional MFCC were used as features for parametrizing the speech waveforms. The delta and acceleration of these features were augmented to get 39-dimensional features. Cepstral Mean Normalization (CMN) and Cepstral Mean Subtraction (CMS) were done to increase the noise-robustness of features. The Kaldi toolkit (Povey *et al.* (2011*b*)) was used for training and testing the acoustic models. Standard C++ programs in the Kaldi toolkit were used to build the baseline HMM-GMM system to initialize the Phone CAT acoustic models. The SGMM system used for comparison is also implemented using the standard programs in the toolkit. Various libraries in the toolkit were used for the standard computations in the algorithms and techniques implemented for the Transform-based Phone CAT model system.

4.2 Parameters

The baseline HMM-GMM system used for RM task had a total of 1428 tied states and 9000 Gaussians. The dictionary had a set of 48 phones. The silence was modeled as a context-independent phone with a 5 state HMM, while all other phones were context-dependent with 3 state HMMs. This was used to initialize the Transform-based Phone CAT model, which had 1601 tied states. Since the feature vector used was of 39 dimension, full-MLLR matrices of dimension 39×40 was used for the cluster transforms. The UBM was initialized by a bottom-up clustering approach by merging the Gaussians from the HMM-GMM system till I mixtures were obtained. I was varied from 400 to 3200.

The baseline HMM-GMM system used for Aurora 4 task had a total of 2913 tied states and 24000 Gaussians. CMU Sphinx Dictionary with 40 phones was used. The modeling of the phones was similar to that in RM task. The Transform-based Phone CAT model initialized from this system had 4036 tied states.

The SGMM system used for comparison with the Transform-based Phone CAT model had a subspace dimension of 40.

4.3 Experiments and Discussion

Tables 4.1 and 4.2 show the results of experiments evaluating the Transform-based Phone CAT models on the RM and Aurora 4 tasks respectively. The details of the experiments, along with the motivation and the conclusions are described in the subsequent sections.

4.3.1 Baseline CDHMM system

The baseline CDHMM system in Expt. 0 is the conventional speech recognition system. It used to initialize the Transform-based Phone CAT model and SGMM as described in Sections 3.3 and 2.4.1 respectively. All the other experiments are compared with this baseline system in terms of accuracy and number of parameters. The CDHMM system has a huge number

#	Experiment	Accuracy	# Parameters		
			# State Parameters	# Global Parameters	# Total
0	**CDHMM (9000 Gaussians)**	96.53	711k	0	711k
A	**Transform-based Phone CAT (1591 tied states, 400 mix)**				
1	Basic without Canonical model update	96.07	713k	106k	819k
2	Basic with Canonical model update	96.38	713k	106k	819k
B	**+ With Gaussian weights tied to the cluster**				
1	400 mix, 48 clusters	95.62	76k	125k	202k
C	**+ With state-dependent Cluster map**				
1	400 mix, 146 clusters	96.36	232k	317k	550k
D	**+ Increased tied states (2386)**				
1	400 mix, 146 clusters	96.42	348k	317k	666k
E	**+ With Weight Projection (1591 tied states)**				
1	400 mix, 48 clusters	96.65	76k	125k	202k
2	800 mix, 48 clusters	96.84	76k	176k	252k
3	1600 mix, 48 clusters	96.9	76k	276k	353k
4	3200 mix, 48 clusters	96.88	76k	478k	554k
5	400 mix, 146 clusters	96.77	232k	317k	550k
6	800 mix, 146 clusters	96.78	232k	407k	639k
F	**+ With Multiple Transform Classes per cluster (1591 tied states)**				
1	1600 mix, 48 clusters, 2 Classes	96.92	76k	351k	428k
G	**+ Full Covariance MLLR (1591 tied states, 1 Transform class)**				
2	400 mix, 48 clusters	97.02	76k	406k	482k
S	**SGMM (1591 tied states, 40 dimensional subspace)**				
1	400 mix Diagonal Covariance	96.3	64k	656k	719k
2	400 mix Full Covariance	97.5	64k	952k	1.016M

Table 4.1: RM Experiment Results

#	Experiment	Accuracy	# Parameters		
			# State Parameters	# Global Parameters	# Total
0	**CDHMM (24000 Gaussians)**	86.34	1.9M	0	1.9M
A	**Transform-based Phone CAT (4036 tied states)**				
B	**+ With weights tied to the center phone**				
1	400 mix, 42 clusters	81.86	170k	114k	283k
2	800 mix, 42 clusters	82.74	170k	162k	331k
3	1600 mix, 42 clusters	83.04	170k	258k	428k
C	**+ With state-dependent Cluster map**				
1	400 mix, 132 clusters	84.23	533k	290k	823k
2	800 mix, 132 clusters	85.3	533k	374k	907k
3	1600 mix, 132 clusters	85.67	533k	542k	1.075M
D	**+ Increased tied states (5422 tied states)**				
1	800 mix, 42 clusters	84.12	228k	162k	390k
2	800 mix, 132 clusters	85.69	716k	374k	1.090M
E	**With Weight Projection**				
1	1600 mix, 132 clusters, 4036 tied states	88	533k	542k	1.075M
S	**SGMM (4036 tied states)**				
1	400 mix Diagonal Covariance	85.3	161k	656k	817k
2	400 mix Full Covariance	90.1	161k	1.248M	1.410M

Table 4.2: Aurora 4 Clean Case Experiment Results

of parameters (0.7M and 1.8M in RM and Aurora 4 respectively), which demands huge data resources. Our objective is to develop better models to achieve a similar or better performance using significantly less number of parameter.

4.3.2 Basic Transform-based Phone CAT model

The experiments in the Set A use a basic model with only the cluster transforms and a canonical model; but without the weight projection (3.5). The Gaussian weights w_{ji} are specified directly; hence they have a huge number of state parameters. For RM Task, Expts. A-1 and A-2 in Table 4.1 show a significant drop in performance for the baseline CDHMM even though the number of parameters are quite similar. It should be noted that 636k of the 711k state parameters are Gaussian weights. Such a model that is dominated by the weights is undesirable. The corresponding experiments were not attempted with Aurora 4 as they would have even more state parameters.

Updating the canonical model gives a significant rise in accuracy by 0.31%. This shows that updating canonical model is crucial.

4.3.3 Tying Gaussian weights to clusters

In the Set B of experiments, the weights were tied to the cluster to reduce the number of Gaussian weight parameters. As a result, each state j had w_{pi} for each state j, where p is the cluster corresponding to the state j. All the experiments again show a significant drop in accuracy from the baseline performance. Expt. B-1 for RM and Aurora 4 have accuracy nearly 1% and 5% respectively below the baseline. However, the number of parameters have been significantly reduced. This allows us to increase the number of Gaussians from 400 to even up to 3200. Expts. B-2 and B-3 for Aurora 4 show slightly better performances, but still lower than CDHMM. Doubling the number of Gaussians in Expt. B-3 compared to B-2 has only resulted in small 0.3% improvement. So merely increasing the Gaussians is not the answer.

4.3.4 State-dependent cluster map

The experiments in Set C attempt to increase both the state and global parameters by increasing the number of clusters. The HMM in each tied triphone has 3 states (or 5 in the case of silence and noise models). Each of these states in the tied triphone can be assigned to distinct clusters. This results in one cluster for every state in the HMM topology of all the phones. This gives 146 clusters in RM task (3 clusters from each of the 47 non-silence phones and 5 from the silence phone) and 132 clusters in Aurora 4 task (3 clusters from each of the 39 non-silence phones and 5 clusters each from the silence, unknown and noise phones).

For RM task, using Expt. C-1 shows nearly 0.8% improvement over Expt. B-1. For Aurora 4 task, the experiments in Set C show a 2.5% improvement over the corresponding experiments in Set C.

4.3.5 Increasing the number of tied states

The number of tied states is increased by choosing the tied states by going further down the context-dependency decision tree. This increases only the state parameters keeping the number of global parameters the same. Expt. D-1 in Aurora 4 shows a good 1.5% improvement over Expt. B-2. This is only because the number of parameters in that model is very less. Expt. D-2 in Aurora 4 shows only a 0.4% improvement over the corresponding Expt. C-2 and Expt. D-1 in RM shows only a 0.06% improvement over the corresponding Expt. C-1.

Increasing the number of tied states increases the state parameters only by a little. And there are serious limitations to increasing the number of tied states, as we may not have enough data to estimate some tied state parameters. There is not much improvement possible on this front, but optimizing the number of tied states for the model might still be required to get the best system.

4.3.6 Weight Projection

Weight projection (3.5), when used in SGMM gave significant improvement. It also gave the model a good structure, making the unnormalized logarithm of weights a linear function of $\mathbf{v}_j$. By tying the Gaussian weights to the clusters, we were severely restricting them. But using the same number of parameters as before, we can get significant improvement with weight projection.

Expt. E-1 in Aurora 4 performs nearly 2.5% better than the model with tied Gaussian weights (Expt. D-2). At 88% accuracy, this shows 1.5% absolute improvement over the baseline CDHMM, while using only half the number of parameters. Even in RM task, the weight projection gives great improvement with even the simplest 400 mixture model with a mere $1/4^{th}$ of the number of CDHMM parameters outperforming the baseline system (Expt. E-1).

When the number of clusters is low (48), we can afford to increase the number of Gaussians. Doubling the number of Gaussians to 800 increases the accuracy significantly by 0.2% (Expt. E-2). But further increasing it does not help much with Expt. E-3 showing only a further improvement of 0.06%. The accuracy even falls on increasing the Gaussians to 3200, probably due to poorer estimates of the parameters.

When the number of clusters is high (146), increasing the number of mixtures does not give much improvement (Expt. E-5 and Expt. E-6). Although at 400 mixtures, using a higher number of clusters gives a better performance (Expt. E-5 vs Expt. E-1), at 800 mixtures, the performance drops on using higher number of number of clusters (Expt. E-6 vs Expt. E-2), possibly again due to poorer estimates of the increased number of parameters.

4.3.7 Multiple MLLR Transform Classes

In Expt. F-1, the set of 1600 Gaussians in the UBM was divided into two transform classes using a bottom-up clustering algorithm. Until only two Gaussians are left, all the Gaussians in the UBM are repeatedly merged such that the decrease in the data likelihood is the least. The sets of Gaussians that have been merged into each of the two Gaussians form the two transform classes. A separate MLLR transform is used for each class. This however gave only a slight

improvement of 0.02% over using a single MLLR for all Gaussians (Expt. E-3). Increasing the number of transform classes is one way to relieve the linear transformation constraint in the transform-base Phone CAT model that is not present in the SGMM. There is a lot of scope of experimenting with this in the future.

4.3.8 Full covariance MLLR

In Expt. G-1, a full covariance matrix is used instead of the diagonal one. This gives achieves an accuracy of 97.02% for RM task, which is a 0.5% absolute improvement or a 14.1% relative improvement in WER. With 400 Gaussians, this uses far less parameters than the CDHMM model and also some of the other Transform-based Phone CAT model with higher number of Gaussians. However, the update equation for the MLLR matrices is quite complicated and computationally expensive. There is a need for parallelization of computations to estimate the model in a duration close that of the SGMM.

4.3.9 SGMM

Comparing with SGMM with 400 Gaussians and diagonal covariance (Expt. S-1), the Transform-based Phone CAT with the same number of Gaussians (Expt. E-1) performs better for the RM task, while using less number of parameters. The same trend is expected in Aurora 4 as well because the Expt. E-1 with a similar number of parameters as the diagonal SGMM performs significantly better. This verifies that the MLLR transforms defines a good structure in the speech models. However, the SGMM full covariance still outperforms the Phone Transform-based CAT model for the same number of Gaussians. However the number of global parameters used is still lower in the Transform-based Phone CAT. So there is scope for more experimentation and improvement.

4.4 Observations

The experiments in Section 4.3 show that the Transform-based Phone CAT model in general performs better than the conventional HMM-GMM system. The higher discriminatory capability of this model can be attributed to modeling the tied state parameters as vectors in a subspace of the total parameter space. This works because the tied state can be better discriminated in the subspace. The model is similar to SGMM in many aspects. But, instead of learning the subspace directly as in the case of SGMM, the structure of the subspace is defined in the form of linear transformations of a canonical mean model. Rather than using a lot of state specific parameters, introducing global parameters to define this subspace results in better performance of the system. This is verified from the experiments in the previous section.

Apart from better performance, we look to capture intuitive phone-context information using the interpolation weights v_j. From the analysis of the plots of v_j for different triphone states, we see that v_j does capture intuitive context information and in some cases it can be observed easily. These are the observations from the plots:

- In all the plots in the figures 4.1, 4.2 and 4.3, the central phone carries a high weight.
- The three states of the HMM – state 0, 1 and 2 – are all affected differently by the context of the surrounding phones. The states 0 and 1 of the triphone /dx/-/aa/+/r/ are not affected greatly by the right context of the phone /r/. But state 2 of /dx/-/aa/+/r/ is largely influenced by the phones /r/ and /ar/ (Fig. 4.1).
- In some cases, the central phone may not have the highest weight and the highest weight goes to some phone very similar to the central phone. In the state 0 of /iy/-/ax/+/sil/ (Fig. 4.2a), the phones /ax/, /eh/ and /ae/, which are all similar, have high weights. Similarly the state 2 of /dx/-/aa/+/r/ (Fig. 4.1c), which is also tied to the state 2 of /dx/-/aa/+/er/, is influenced by both /r/ and /er/ phones.
- Some consonants like the stop consonants have less context effect than other phones. The phones /dx/ in Fig. 4.1 and /p/ in Fig. 4.3 does not affect the context much.
- Some phones like fricatives, which are much different from other phones, have a very high influence on the triphone. For the triphone X-/s/+/p/ where X is tied across many phones (Fig. 4.3), the central phone /s/ has a high and dominating weight.

These observations only show that v_j is capable of capturing context information; but it should not be relied on for all context information. Only in the initial iterations, the clusters

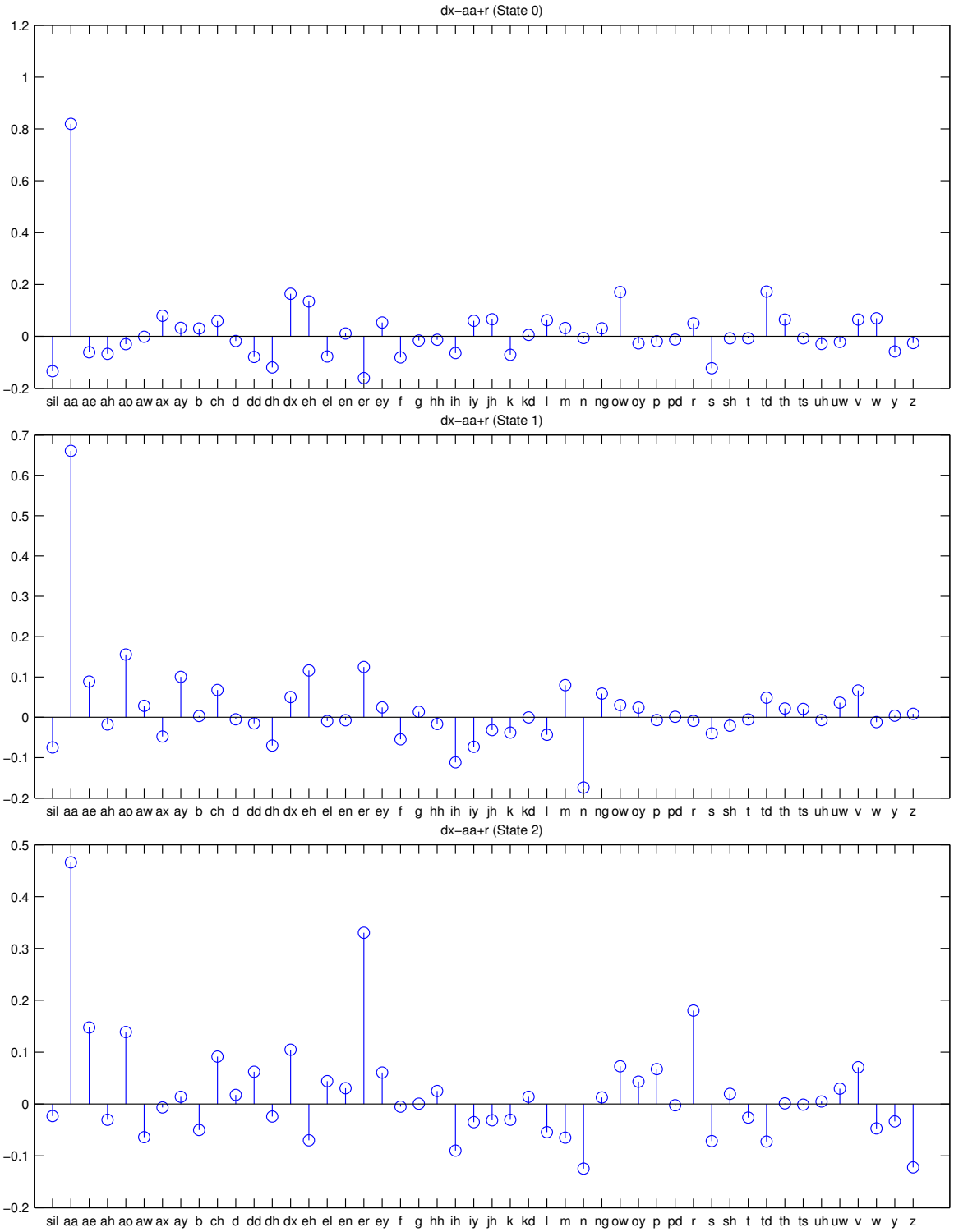


Figure 4.1: Analysis of $\mathbf{v}_j$ for /dx/-/aa/+/r/

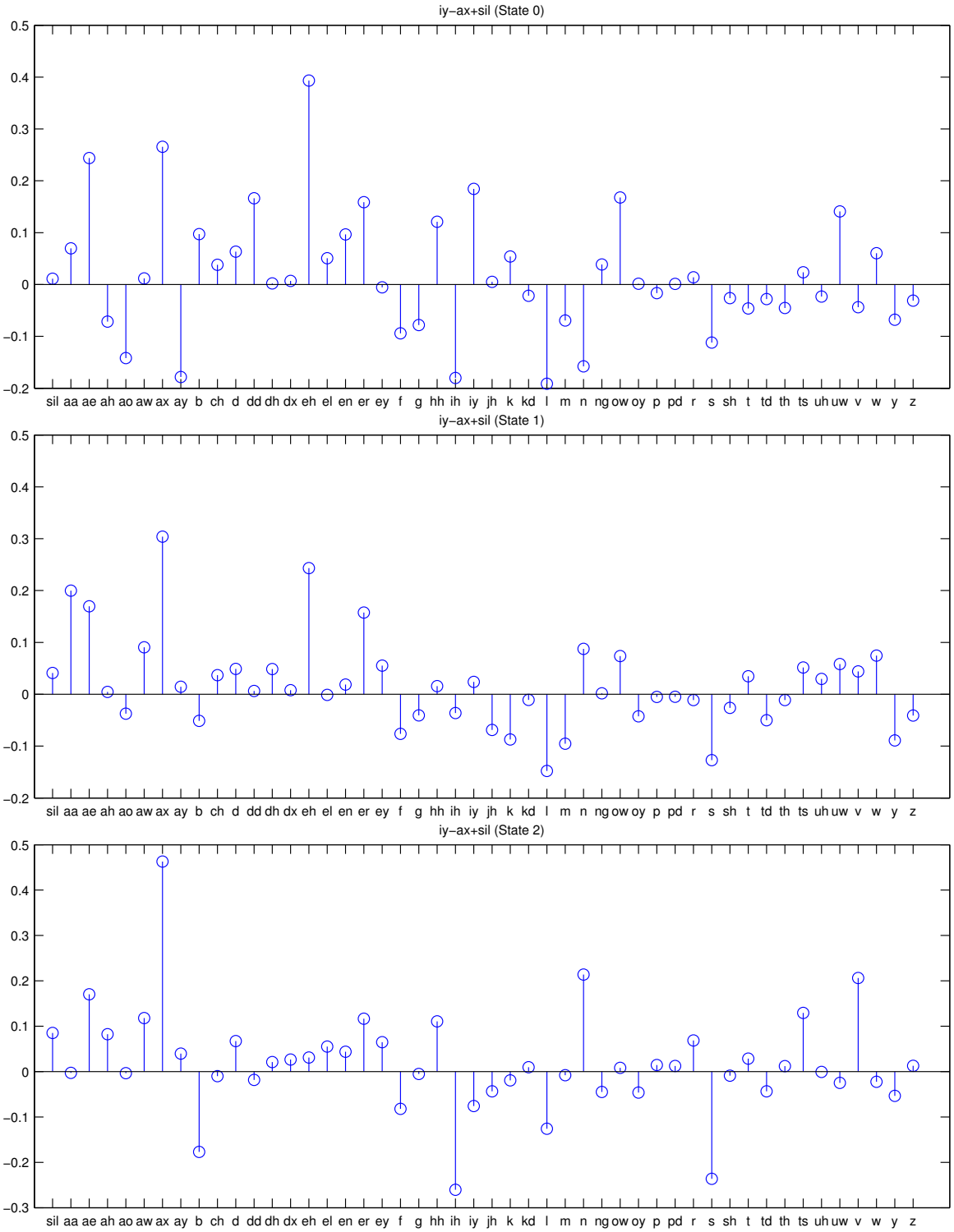


Figure 4.2: Analysis of $\mathbf{v}_j$ for /iy/-/ax/+/sil/

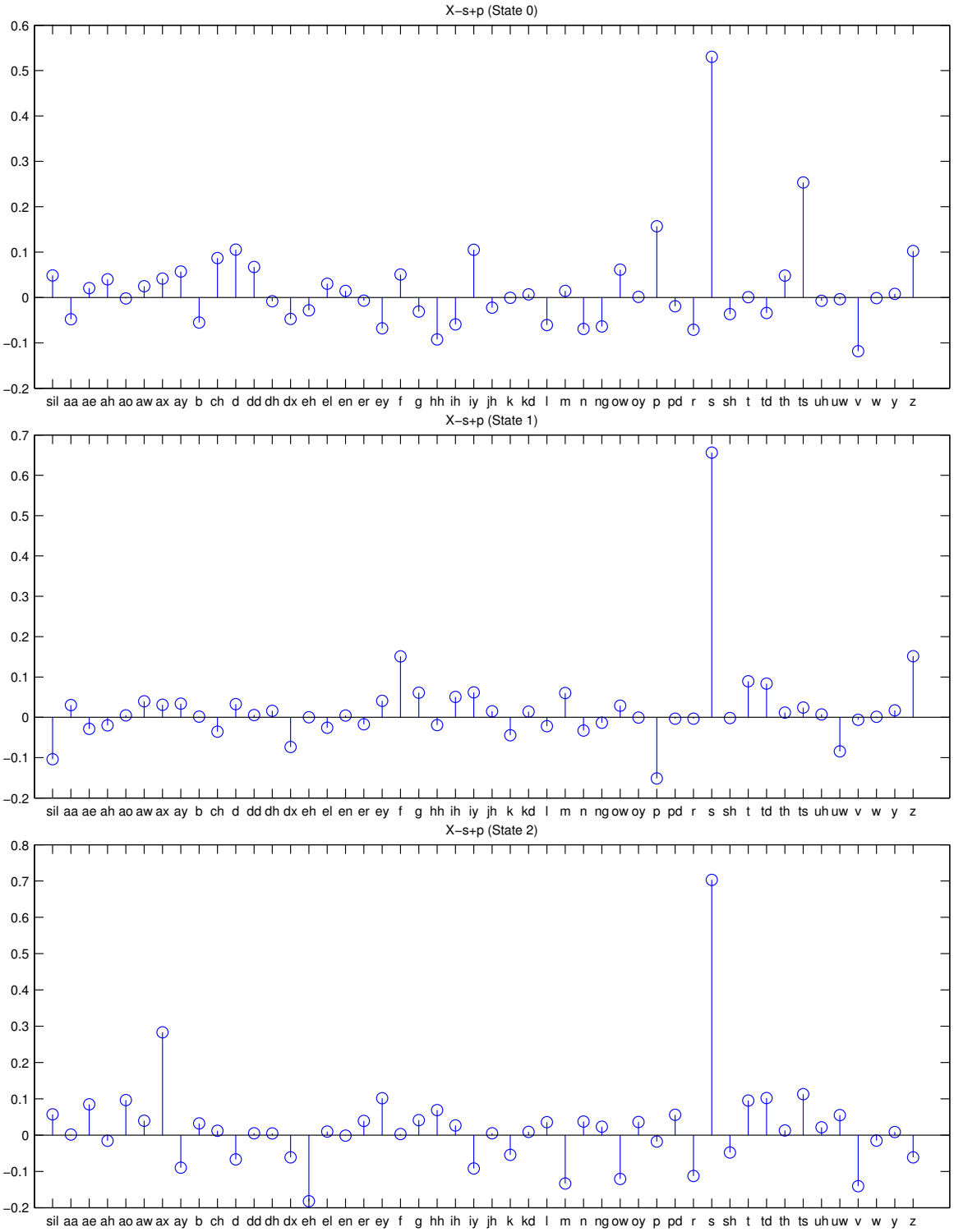


Figure 4.3: Analysis of $\mathbf{v}_j$ for X-/s/+/p/

strictly represent the monophone models. Since, many monophones themselves are similar, the training procedure may learn to use the same cluster model for the similar monophones, leaving the other cluster models open to new eigen-directions of the parameters space.

Chapter 5

Conclusions and Future Work

A new kind of acoustic model, the Transform-based Phone CAT model, is introduced. Unlike, the conventional HMM-GMM system, this model does not specify the parameters of the distribution directly, but generates the parameters of the distribution. This allows to represent complex GMM distributions in a compact way. By restricting the dimensions in the total parameter space of the distribution to a compact subspace, the discriminatory capability of the speech models is improved. The use of shared, global parameters instead of the conventional state-specific parameters, allows a better modeling of the speech sounds for similar parameter count. The global parameters also allow the possibility of using out-of-domain data and hence the model can be efficiently trained on less in-domain data than in CDHMM models. The structure of the model allows to train and evaluate the models efficiently. The compact canonical model allows efficient pruning of Gaussians evaluated in each frame.

The experiments conducted on Resource Management (RM) Task and Aurora 4 Task confirm that the model gives better results than the conventional HMM-GMM system. On the RM task, the Transform-based Phone CAT model shows an improvement of 0.5% absolute, which is a 14.1% relative improvement in Word Error Rate (WER). The observation that this model performs better than a similar SGMM system with 400 mixture diagonal covariance shows that the addition of the MLLR transformations gives the model a good structure and hence improves the performance of the system. Being very similar to the SGMM, this model offers scope for similar modeling improvements. Use of piece-wise MLLR with multiple transform classes per cluster, full covariance cluster adaptive training and multiple substates per state offers possibility for further improvement with this model. It also allows the possibility of further using speaker adaptation techniques like CMLLR and VTLN, in a similar way as in SGMM. In addition to providing improvements over the conventional system, this model also gives an intuitive way of representing phone context information. The linear interpolation weights of the clusters in the models are shown to capture this context information.

Appendix A

Estimation of Parameters

A.1 Cluster Transforms

The term $\boldsymbol{\mu}_{ji}$ in (3.11) can be expressed using (3.3) as

$$\begin{aligned}
\boldsymbol{\mu}_{ji} &= \left[\begin{array}{cccc} \boldsymbol{\mathcal{W}}_1\boldsymbol{\xi}_i & \boldsymbol{\mathcal{W}}_2\boldsymbol{\xi}_i & \ldots & \boldsymbol{\mathcal{W}}_P\boldsymbol{\xi}_i \end{array} \right] \left[\begin{array}{c} v_j^{(1)} \\ v_j^{(2)} \\ \vdots \\ v_j^{(P)} \end{array} \right], \\
&= \left[\begin{array}{cccc} \boldsymbol{\mathcal{W}}_1 & \boldsymbol{\mathcal{W}}_2 & \ldots & \boldsymbol{\mathcal{W}}_P \end{array} \right] \left[\begin{array}{cccc} \boldsymbol{\xi}_i & 0 & \cdots & 0 \\ 0 & \boldsymbol{\xi}_i & 0 & 0 \\ \vdots & 0 & \ddots & \vdots \\ 0 & 0 & \cdots & \boldsymbol{\xi}_i \end{array} \right] \left[\begin{array}{c} v_j^{(1)} \\ v_j^{(2)} \\ \vdots \\ v_j^{(P)} \end{array} \right], \\
&= \mathbf{H}\mathbf{b}_{ji}. \qquad \text{(A.1)}
\end{aligned}$$

(3.11) needs to be optimized with respect to $\mathbf{H} = \left[\begin{array}{cccc} \boldsymbol{\mathcal{W}}_1 & \boldsymbol{\mathcal{W}}_2 & \ldots & \boldsymbol{\mathcal{W}}_P \end{array} \right]$. Using (A.1) and differentiating (3.11) with respect to $\mathbf{H}$, we get:

$$\frac{\partial \mathcal{Q}}{\partial \mathbf{H}} = \sum_{j,i,t} \gamma_{ji}(t)\, \boldsymbol{\Sigma}_i^{-1} \left(\mathbf{x}(t) - \mathbf{H}\mathbf{b}_{ji} \right) \mathbf{b}_{ji}^T = 0 \qquad \text{(A.2)}$$

A closed-form expression for re-estimation of $\mathbf{H}$ can be obtained for the case of a diagonal $\boldsymbol{\Sigma}_i$ (Gales (1998)). This allows an entire row of $\mathbf{H}$ to be estimated at once. The detailed steps follow:

$$\sum_{j,i,t} \gamma_{ji}(t)\, \boldsymbol{\Sigma}_i^{-1} \mathbf{x}(t)\, \mathbf{b}_{ji}^T = \sum_{j,i,t} \gamma_{ji}(t)\, \boldsymbol{\Sigma}_i^{-1} \mathbf{H}\mathbf{b}_{ji}\mathbf{b}_{ji}^T \qquad \text{(A.3)}$$

The LHS and RHS of (A.3) can be simplified for diagonal $\mathbf{\Sigma}_i$ as

$$\begin{aligned}
\text{LHS} &= \sum_{j,i,t} \gamma_{ji}(t)\, \mathbf{\Sigma}_i^{-1} \mathbf{x}(t)\, \mathbf{b}_{ji}^T \\
&= \sum_{j,i,t} \gamma_{ji}(t) \begin{bmatrix} \frac{x_1(t)}{\sigma_{11}^{(i)2}} \\ \vdots \\ \frac{x_k(t)}{\sigma_{kk}^{(i)2}} \\ \vdots \\ \frac{x_D(t)}{\sigma_{DD}^{(i)2}} \end{bmatrix} \begin{bmatrix} v_j^{(1)}\boldsymbol{\xi}_i^T & v_j^{(2)}\boldsymbol{\xi}_i^T & \dots & v_j^{(P)}\boldsymbol{\xi}_i^T \end{bmatrix}, \qquad \text{(A.4)} \\
\text{RHS} &= \sum_{j,i,t} \gamma_{ji}(t)\, \mathbf{\Sigma}_i^{-1} \mathbf{H} \mathbf{b}_{ji} \mathbf{b}_{ji}^T \\
&= \sum_{j,i,t} \gamma_{ji}(t) \begin{bmatrix} \frac{\mathbf{H}_1}{\sigma_{11}^{(i)2}} \\ \vdots \\ \frac{\mathbf{H}_k}{\sigma_{kk}^{(i)2}} \\ \vdots \\ \frac{\mathbf{H}_D}{\sigma_{DD}^{(i)2}} \end{bmatrix} \begin{bmatrix} \boldsymbol{\xi}_i v_j^{(1)} \\ \boldsymbol{\xi}_i v_j^{(2)} \\ \vdots \\ \boldsymbol{\xi}_i v_j^{(P)} \end{bmatrix} \begin{bmatrix} v_j^{(1)}\boldsymbol{\xi}_i^T & v_j^{(2)}\boldsymbol{\xi}_i^T & \dots & v_j^{(P)}\boldsymbol{\xi}_i^T \end{bmatrix} \qquad \text{(A.5)}
\end{aligned}$$

where $x_k(t)$ is the k^{th} dimension of $\mathbf{x}(t)$, $\mathbf{H}_k$ is the k^{th} row of $\mathbf{H}$ and $\sigma_{kk}^{(i)2}$ is the k^{th} element in the diagonal covariance $\mathbf{\Sigma}_i$. Equating the k^{th} row in (A.4) and (A.5), we get

$$\mathbf{k}^{(k)} = \mathbf{H}_k \mathbf{G}^{(k)}, \qquad \text{(A.6)}$$

where

$$\begin{aligned}
\mathbf{k}^{(k)} &= \sum_{j,i,t} \gamma_{ji}(t)\, \frac{x_k(t)}{\sigma_{kk}^{(i)2}} \mathbf{v}_j^T \otimes \boldsymbol{\xi}_i^T, \\
\mathbf{G}^{(k)} &= \sum_{j,i,t} \gamma_{ji}(t)\, \frac{1}{\sigma_{kk}^{(i)2}} \mathbf{v}_j \mathbf{v}_j^T \otimes \boldsymbol{\xi}_i \boldsymbol{\xi}_i^T,
\end{aligned}$$

and $\otimes$ is the Kronecker product. (A.6) can be solved to $\mathbf{H}_k$ by inverting the $P(D+1) \times P(D+1)$ square matrix $\mathbf{G}^{(k)}$. This is a very expensive process. A more efficient way makes

an approximation that only one of the elements in $\mathbf{v}_j$ is dominating and estimates one cluster transform at a time. The RHS of (A.6) can be expressed as

$$\begin{aligned}\mathbf{H}_k\mathbf{G}^{(k)} &= \sum_{j,i,t}\frac{\gamma_{ji}(t)}{\sigma_{kk}^{(i)2}}\sum_{p=1}^{P}\boldsymbol{\mathcal{W}}_p^{(k)}v_j^{(p)}\mathbf{v}_j^T\otimes\boldsymbol{\xi}_i\boldsymbol{\xi}_i^T \\ &= \sum_{j,i,t}\frac{\gamma_{ji}(t)}{\sigma_{kk}^{(i)2}}\boldsymbol{\mathcal{W}}_p^{(k)}v_j^{(p)}\mathbf{v}_j^T\otimes\boldsymbol{\xi}_i\boldsymbol{\xi}_i^T+\sum_{j,i,t}\frac{\gamma_{ji}(t)}{\sigma_{kk}^{(i)2}}\sum_{l\neq p}^{P}\boldsymbol{\mathcal{W}}_l^{(k)}v_j^{(l)}\mathbf{v}_j^T\otimes\boldsymbol{\xi}_i\boldsymbol{\xi}_i^T. \quad \text{(A.7)}\end{aligned}$$

Using (A.7) in (A.6), we get

$$\sum_{j,i,t}\frac{\gamma_{ji}(t)}{\sigma_{kk}^{(i)2}}\boldsymbol{\mathcal{W}}_p^{(k)}v_j^{(p)}\mathbf{v}_j^T\otimes\boldsymbol{\xi}_i\boldsymbol{\xi}_i^T=\sum_{j,i,t}\gamma_{ji}(t)\frac{x_k(t)}{\sigma_{kk}^{(i)2}}\mathbf{v}_j^T\otimes\boldsymbol{\xi}_i^T-\sum_{j,i,t}\frac{\gamma_{ji}(t)}{\sigma_{kk}^{(i)2}}\sum_{l\neq p}^{P}\boldsymbol{\mathcal{W}}_l^{(k)}v_j^{(l)}\mathbf{v}_j^T\otimes\boldsymbol{\xi}_i\boldsymbol{\xi}_i^T. \quad \text{(A.8)}$$

For estimation of $\boldsymbol{\mathcal{W}}_p$, $\mathbf{v}_j^T$ can be approximated to $\begin{bmatrix}0 & \dots & 0 & v_j^{(p)} & 0 & \dots & 0\end{bmatrix}$. Using this in (A.13), we get

$$\begin{aligned}\boldsymbol{\mathcal{W}}_p^{(k)}\left[\sum_i\frac{\left(\sum_{j,t}\gamma_{ji}(t)v_j^{(p)2}\right)}{\sigma_{kk}^{(i)2}}\boldsymbol{\xi}_i\boldsymbol{\xi}_i^T\right]&=\sum_i\frac{1}{\sigma_{kk}^{(i)2}} \\ &\left[\left\{\left(\sum_{j,t}\gamma_{ji}(t)x_k(t)v_j^{(p)}\right)-\sum_{l\neq p}^{P}\left(\sum_{j,t}\gamma_{ji}(t)v_j^{(l)}v_j^{(p)}\right)\boldsymbol{\mathcal{W}}_l^{(k)}\boldsymbol{\xi}_i\right\}\boldsymbol{\xi}_i^T\right]. \quad \text{(A.9)}\end{aligned}$$

The update equation for $\boldsymbol{\mathcal{W}}_p^{(k)}$ can be written as

$$\boldsymbol{\mathcal{W}}_p^{(k)}=\mathbf{k}_p^{(k)}\left[\mathbf{G}_p^{(k)}\right]^{-1}, \quad \text{(A.10)}$$

where

$$\mathbf{k}_p^{(k)} = \sum_i \frac{1}{\sigma_{kk}^{(i)2}} \left[\left\{ k_{pk}^{(i)} - \sum_{l \neq p}^{P} g_{lp}^{(i)} \boldsymbol{\mathcal{W}}_l^{(k)} \boldsymbol{\xi}_i \right\} \boldsymbol{\xi}_i^T \right], \tag{A.11}$$

$$\mathbf{G}_p^{(k)} = \sum_i \frac{g_{pp}^{(i)}}{\sigma_{kk}^{(i)2}} \boldsymbol{\xi}_i \boldsymbol{\xi}_i^T, \tag{A.12}$$

with $k_{pk}^{(i)}$ and $g_{lp}^{(i)}$ being sufficient stats defined by (3.16) and (3.15).

A.2 State vectors

The auxiliary function (3.11) can be expressed in two parts; one, $\mathcal{Q}_1$, containing the terms from the means $\boldsymbol{\mu}_{ji}$ and the other, $\mathcal{Q}_2$, containing the terms from the weights. $\mathcal{Q}_1$ is given by

$$\mathcal{Q}_1 = -\frac{1}{2} \sum_{i,t} \gamma_{ji}(t) \left(\mathbf{x}(t) - \mathbf{M}_i \mathbf{v}_j \right)^T \boldsymbol{\Sigma}_i^{-1} \left(\mathbf{x}(t) - \mathbf{M}_i \mathbf{v}_j \right), \tag{A.13}$$

where $\mathbf{M}_i = \begin{bmatrix} \mathbf{A}_1 \boldsymbol{\xi}_i & \dots & \mathbf{A}_P \boldsymbol{\xi}_i \end{bmatrix}$ with $\boldsymbol{\xi}_i = \begin{bmatrix} \boldsymbol{\mu}_i & 1 \end{bmatrix}^T$. Neglecting the terms independent of $\mathbf{v}_j$, we get

$$\begin{aligned} \mathcal{Q}_1 &= K + \sum_{i,t} \gamma_{ji}(t) \mathbf{v}_j^T \mathbf{M}_i^T \boldsymbol{\Sigma}_i^{-1} \mathbf{x}(t) - 0.5 \sum_{i,t} \gamma_{ji}(t) \mathbf{v}_j^T \mathbf{H}_i \mathbf{v}_j, \\ &= K + \mathbf{v}_j^T \mathbf{y}_j - 0.5 \mathbf{v}_j^T \left(\sum_i \gamma_{ji} \mathbf{H}_i \right) \mathbf{v}_j, \end{aligned} \tag{A.14}$$

where K is some constant independent of $\mathbf{v}_j$ and $\mathbf{H}_i = \mathbf{M}_i^T \boldsymbol{\Sigma}_i^{-1} \mathbf{M}_i$.

$\mathcal{Q}_2$ is given by

$$\begin{aligned} \mathcal{Q}_2 &= \sum_{i,t} \gamma_{ji}(t) \log w_{ji} \\ &= \sum_{i,t} \gamma_{ji}(t) \left(\mathbf{w}_i^T \mathbf{v}_j - \log \sum_{i'=1}^{I} \exp\left(\mathbf{w}_{i'}^T \mathbf{v}_j \right) \right) \end{aligned}$$

Using the inequality $1 - (x/\bar{x}) \leq -\log(x/\bar{x})$ with x corresponding to $\sum_{i'=1}^{I} \exp\left(\mathbf{w}_{i'}^T \mathbf{v}_j\right)$ and $\bar{x}$ corresponding to its pre-update value (which is a constant with respect to the auxiliary function), we get

$$\mathcal{Q}_2' = K + \sum_{i,t} \gamma_{ji} \left(\mathbf{w}_i^T \mathbf{v}_j - \frac{\sum_{i'=1}^{I} \exp\left(\mathbf{w}_{i'}^T \mathbf{v}_j\right)}{\sum_{i'=1}^{I} \exp\left(\mathbf{w}_{i'}^T \bar{\mathbf{v}}_j\right)} \right).$$

$\exp(x)$ can be expanded using second-order Taylor series approximation about x_0 as $\exp(x) \simeq \exp(x_0)\left(1 + (x - x_0) + 0.5(x - x_0)^2\right)$. Neglecting the terms independent of x as constants, we get

$$\exp(x) \simeq K + \exp(x_0)\left(x(1 - x_0) + 0.5x^2\right) \tag{A.15}$$

. Expanding $\exp\left(\mathbf{w}_{i'}^T \mathbf{v}_j\right)$ using (A.15), we get

$$\mathcal{Q}_2'' = K' + \sum_{i,t} \gamma_{ji} \left(\mathbf{w}_i^T \mathbf{v}_j - \frac{\sum_{i'=1}^{I} \left(\mathbf{w}_{i'}^T \mathbf{v}_j \left(1 - \mathbf{w}_{i'}^T \bar{\mathbf{v}}_j\right) + 0.5\left(\mathbf{w}_{i'}^T \bar{\mathbf{v}}_j\right)^2\right) \exp\left(\mathbf{w}_{i'}^T \bar{\mathbf{v}}_j\right)}{\sum_{i'=1}^{I} \exp\left(\mathbf{w}_{i'}^T \bar{\mathbf{v}}_j\right)} \right), \tag{A.16}$$

which can be simplified using $\bar{w}_{ji} = \dfrac{\exp\left(\mathbf{w}_i^T \bar{\mathbf{v}}_j\right)}{\sum_{i'=1}^{I} \exp\left(\mathbf{w}_{i'}^T \bar{\mathbf{v}}_j\right)}$ as

$$\mathcal{Q}_2'' = K'' + \sum_{i,t} \gamma_{ji} \mathbf{w}_i^T \mathbf{v}_j - \gamma_j \sum_{i'=1}^{I} w_{ji'} \left(\mathbf{w}_{i'}^T \mathbf{v}_j \left(1 - \mathbf{w}_{i'}^T \bar{\mathbf{v}}_j\right) + 0.5\left(\mathbf{w}_{i'}^T \bar{\mathbf{v}}_j\right)^2 \right). \tag{A.17}$$

The final auxiliary function for state vector update is the sum of (A.14) and (A.17). There-

fore the final auxiliary function $\mathcal{Q}$ and the update equations are given by

$$\begin{aligned}
\mathcal{Q} &= K + \mathbf{v}_j^T \mathbf{k}_j - 0.5 \mathbf{v}_j^T \mathbf{G}_j \mathbf{v}_j, && \text{(A.18)} \\
\hat{\mathbf{v}}_j &= \mathbf{G}_j^{-1} \mathbf{k}_j, && \text{(A.19)} \\
\mathbf{k}_j &= \mathbf{y}_j + \sum_{i=1}^{I} \mathbf{w}_i \left(\gamma_{ji} - \gamma_j w_{ji} + \max\left(\gamma_{ji}, \gamma_j w_{ji}\right) \mathbf{w}_i^T \mathbf{v}_j \right), && \text{(A.20)} \\
\mathbf{G}_j &= \sum_{i=1}^{I} \gamma_{ji} \mathbf{H}_i + \max\left(\gamma_{ji}, \gamma_j w_{ji}\right) \mathbf{w}_i^T \mathbf{v}_j, && \text{(A.21)}
\end{aligned}$$

where $\mathbf{y}_j$,γ_{ji} and γ_j are sufficient stats defined by (3.20), (3.21) and (3.22) respectively, $\hat{\mathbf{v}}_j$ is the updated state vector.

A.3 Canonical model

For canonical mean estimation, the auxiliary function (3.11) can be written as,

$$\mathcal{Q} = -\frac{1}{2} \sum_{j,i,t} \gamma_{ji}(t) \left[\mathbf{x}(t) - \sum_p \left(\mathbf{A}_p \boldsymbol{\mu}_i + \mathbf{b}_p \right) v_j^{(p)} \right]^T \boldsymbol{\Sigma}^{-1} \left[\mathbf{x}(t) - \sum_p \left(\mathbf{A}_p \boldsymbol{\mu}_i + \mathbf{b}_p \right) v_j^{(p)} \right]. \tag{A.22}$$

Differentiating (A.22) w.r.t. $\boldsymbol{\mu}_i$ and equating to 0, we get

$$\frac{\partial \mathcal{Q}}{\partial \boldsymbol{\mu}_i} = \sum_{j,t} \gamma_{ji}(t) \left(\sum_p \mathbf{A}_p v_j^{(p)} \right)^T \boldsymbol{\Sigma}^{-1} \left[\mathbf{x}(t) - \sum_p \mathbf{A}_p \boldsymbol{\mu}_i v_j^{(p)} + \sum_p \mathbf{b}_p v_j^{(p)} \right] = 0.$$

This can be simplified as

$$\sum_{p=1}^{P} \mathbf{A}_p^T \boldsymbol{\Sigma}^{-1} \left[\left(\sum_{j,t} \gamma_{ji}(t) v_j^{(p)} \mathbf{x}(t) \right) - \sum_{q=1}^{P} \left(\sum_{j,t} \gamma_{ji}(t) v_j^{(p)} v_j^{(q)} \right) \mathbf{b}_q \right] = \left[\sum_{p,q=1}^{P} \left(\sum_{j,t} \gamma_{ji}(t) v_j^{(p)} v_j^{(q)} \right) \mathbf{A}_p^T \boldsymbol{\Sigma}^{-1} \right. \tag{A.23}$$

or

$$\boldsymbol{\mu}_i = \left[\sum_{p,q=1}^{P} g_{pq}^{(i)} \mathbf{A}_p^T \boldsymbol{\Sigma}^{-1} \mathbf{A}_q \right]^{-1} \left[\sum_p \mathbf{A}_p^T \boldsymbol{\Sigma}^{-1} \left(\mathbf{k}_p^{(i)T} - \sum_q g_{pq}^{(i)} \mathbf{b}_q \right) \right], \tag{A.24}$$

where $g_{pq}^{(i)}$ is the sufficient statistics defined in (3.15), $\mathbf{k}_p^{(i)}$ is the p^{th} row of the sufficient statistics

3.16 and $\left[\begin{array}{cc} \mathbf{A}_p & \mathbf{b}_p \end{array} \right] = \boldsymbol{\mathcal{W}}_p$ is the MLLR transform of the cluster p.

For covariance update, the standard covariance update equation can be modified:

$$\boldsymbol{\Sigma}_i = \frac{\sum_{j,t} \gamma_{ji}(t) \left(\mathbf{x}(t) - \boldsymbol{\mu}_{ji}\right) \left(\mathbf{x}(t) - \boldsymbol{\mu}_{ji}\right)^T}{\sum_{j,t} \gamma_{ji}(t)}, \tag{A.25}$$

$$= \frac{\sum_{j,t} \gamma_{ji}(t) \mathbf{x}(t) \mathbf{x}(t)^T - \sum_{j,t} \gamma_{ji}(t) \mathbf{x}(t) \boldsymbol{\mu}_{ji}^T - \sum_{j,t} \gamma_{ji}(t) \boldsymbol{\mu}_{ji} \mathbf{x}(t)^T + \sum_{ji} \gamma_{ji}(t) \boldsymbol{\mu}_{ji} \boldsymbol{\mu}_{ji}}{\sum_{j,t} \gamma_{ji}(t)} \tag{A.26}$$

The term $\sum_{j,t} \gamma_{ji}(t) \mathbf{x}(t) \boldsymbol{\mu}_{ji}^T$ can be computed as

$$\begin{aligned} \sum_{j,t} \gamma_{ji}(t) \mathbf{x}(t) \boldsymbol{\mu}_{ji}^T &= \sum_{j,t} \gamma_{ji}(t) \mathbf{x}(t) \left(\sum_p \boldsymbol{\mathcal{W}}_p v_j^{(p)} \boldsymbol{\xi}_i \right)^T, \\ &= \sum_p \sum_{j,t} \gamma_{ji}(t) \mathbf{x}(t) v_j^{(p)} \left(\boldsymbol{\mathcal{W}}_p \boldsymbol{\xi}_i\right)^T, \\ &= \sum_p \boldsymbol{k}_p^{(i)} \left(\boldsymbol{\mathcal{W}}_p \boldsymbol{\xi}_i\right)^T, \end{aligned} \tag{A.27}$$

where $\boldsymbol{k}_p^{(i)}$ is the i^{th} row of the sufficient statistics (3.16). The term $\sum_{j,t} \gamma_{ji}(t) \boldsymbol{\mu}_{ji} \mathbf{x}(t)^T$ is just the transpose of this.

The term $\sum_{j,t} \gamma_{ji}(t) \boldsymbol{\mu}_{ji} \boldsymbol{\mu}_{ji}^T$ can be computed as

$$\begin{aligned} \sum_{j,t} \gamma_{ji}(t) \boldsymbol{\mu}_{ji} \boldsymbol{\mu}_{ji}^T &= \sum_{j,t} \gamma_{ji} \sum_p \boldsymbol{\mathcal{W}}_p v_j^{(p)} \boldsymbol{\xi}_i \boldsymbol{\xi}_i^T \sum_q \boldsymbol{\mathcal{W}}_q^T v_j^{(q)}, \\ &= \sum_{p=1,q=1}^{P} g_{pq}^{(i)} \left(\boldsymbol{\mathcal{W}}_p \boldsymbol{\xi}_i\right) \left(\boldsymbol{\mathcal{W}}_q \boldsymbol{\xi}_i\right)^T, \end{aligned} \tag{A.28}$$

where $g_{pq}^{(i)}$ is the sufficient statistics defined in (3.15).

Therefore (A.26) can be expressed as:

$$\boldsymbol{\Sigma}_i = \frac{\mathbf{L}_i - \sum_p \boldsymbol{k}_p^{(i)} \left(\boldsymbol{\mathcal{W}}_p \boldsymbol{\xi}_i\right)^T - \left[\sum_p \boldsymbol{k}_p^{(i)} \left(\boldsymbol{\mathcal{W}}_p \boldsymbol{\xi}_i\right)^T\right]^T + \sum_{p=1,q=1}^{P} g_{pq}^{(i)} \left(\boldsymbol{\mathcal{W}}_p \boldsymbol{\xi}_i\right)\left(\boldsymbol{\mathcal{W}}_q \boldsymbol{\xi}_i\right)^T}{\sum_j \gamma_{ji}}, \tag{A.29}$$

where $\mathbf{L}_i$ and γ_{ji} are sufficient statistics defined in (3.25) and (3.21). For a diagonal covariance re-estimation, only the diagonal is retained in (A.29).

A.4 Auxiliary function change for Full covariance Transform-based Phone CAT

The change in the auxiliary function $\mathcal{Q}$ after update (3.32) can be computed as

$$\begin{aligned}
\Delta\mathcal{Q} &= \Delta \sum_{j,i,t} \left(\gamma_{ji}(t)\, \mathbf{x}(t)^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_{ji} - 0.5 \gamma_{ji}(t)\, \boldsymbol{\mu}_{ji}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_{ji} \right), \\
&= \Delta \sum_{j,i,t} \left(\gamma_{ji}(t)\, \mathbf{x}(t)^T \boldsymbol{\Sigma}^{-1} \left(\sum_p \boldsymbol{\mathcal{W}}_p v_j^{(p)} \right) \boldsymbol{\xi}_i \right. \\
&\quad \left. -0.5 \gamma_{ji}(t)\, \boldsymbol{\xi}_i^T \left(\sum_p \boldsymbol{\mathcal{W}}_p^T v_j^{(p)} \right) \boldsymbol{\Sigma}^{-1} \left(\sum_q \boldsymbol{\mathcal{W}}_q v_j^{(q)} \right) \boldsymbol{\xi}_i \right).
\end{aligned} \tag{A.30}$$

Since only the p^{th} cluster has been updated the summation over p reduces to a single term involving the p^{th} cluster. Using this (A.30) can be simplified as:

$$\begin{aligned}
\Delta\mathcal{Q} &= \sum_i \boldsymbol{k}_p^{(i)} \boldsymbol{\Sigma}^{-1} \left(\Delta \boldsymbol{\mathcal{W}}_p\right) \boldsymbol{\xi}_i - 0.5 g_{pp}^{(i)} \boldsymbol{\xi}_i^T \Delta \left(\boldsymbol{\mathcal{W}}_p^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mathcal{W}}_p \right) \boldsymbol{\xi}_i \\
&\quad -0.5 \sum_{q \neq p} g_{pq}^{(i)} \boldsymbol{\xi}_i^T \left(\Delta \boldsymbol{\mathcal{W}}_p^T\right) \boldsymbol{\Sigma}^{-1} \boldsymbol{\mathcal{W}}_q \boldsymbol{\xi}_i \\
&\quad -0.5 \sum_{q \neq p} g_{pq}^{(i)} \boldsymbol{\xi}_i^T \boldsymbol{\mathcal{W}}_q^T \boldsymbol{\Sigma}^{-1} \left(\Delta \boldsymbol{\mathcal{W}}_p\right) \boldsymbol{\xi}_i.
\end{aligned} \tag{A.31}$$

where $g_{pq}^{(i)}$ is sufficient statistics defined in (3.15), $\boldsymbol{k}_p^{(i)}$ is the p^{th} row of sufficient statistics in (3.16) and $\Delta\boldsymbol{\mathcal{W}}_p$ is the change in $\boldsymbol{\mathcal{W}}_p$ after update.

Bibliography

1. **Anastasakos, T.**, **J. McDonough**, **R. Schwartz**, and **J. Makhoul**, A compact model for speaker-adaptive training. *In Spoken Language, 1996. ICSLP 96. Proceedings., Fourth International Conference on*, volume 2. IEEE, 1996.

2. **Burget, L.**, **P. Schwarz**, **M. Agarwal**, **P. Akyazi**, **K. Feng**, **A. Ghoshal**, **O. Glembek**, **N. Goel**, **M. Karafiát**, **D. Povey**, *et al.*, Multilingual acoustic modeling for speech recognition based on subspace gaussian mixture models. *In Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*. IEEE, 2010.

3. **Gales, M. J.** (1998). Maximum likelihood linear transformations for hmm-based speech recognition. *Computer speech and language*, **12**(2).

4. **Gales, M. J.** (2000). Cluster adaptive training of hidden markov models. *Speech and Audio Processing, IEEE Transactions on*, **8**(4), 417–428.

5. **Leggetter, C.** and **P. Woodland** (1995). Maximum likelihood linear regression for speaker adaptation of continuous density hidden markov models. *Computer speech and language*, **9**(2), 171.

6. **Mohan, A.**, **S. Umesh**, and **R. Rose**, Subspace based for indian languages. *In Information Science, Signal Processing and their Applications (ISSPA), 2012 11th International Conference on*. IEEE, 2012.

7. **Povey, D.** (2009). A tutorial-style introduction to subspace gaussian mixture models for speech recognition. Technical Report MSR-TR-2009-111, Microsoft Research.

8. **Povey, D.**, **L. Burget**, **M. Agarwal**, **P. Akyazi**, **K. Feng**, **A. Ghoshal**, **O. Glembek**, **N. K. Goel**, **M. Karafiát**, **A. Rastrow**, *et al.*, Subspace gaussian mixture models for speech recognition. *In Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*. IEEE, 2010.

9. **Povey, D.**, **L. Burget**, **M. Agarwal**, **P. Akyazi**, **F. Kai**, **A. Ghoshal**, **O. Glembek**, **N. Goel**, **M. Karafiát**, **A. Rastrow**, **R. C. Rose**, **P. Schwarz**, and **S. Thomas** (2011*a*). The subspace gaussian mixture model - a structured model for speech recognition. *Computer Speech & Language*, **25**(2), 404 – 439. ISSN 0885-2308.

10. **Povey, D.**, **A. Ghoshal**, **G. Boulianne**, **L. Burget**, **O. Glembek**, **N. Goel**, **M. Hannemann**, **P. Motlicek**, **Y. Qian**, **P. Schwarz**, *et al.*, The kaldi speech recognition toolkit. *In IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. 2011*b*.

11. **Povey, D.** and **G. Saon** (2006). Feature and model space speaker adaptation with full covariance gaussians. *Interspeech*.

12. **Price, P.**, **W. M. Fischer**, **J. Bernsteina**, and **D. S. Pallett** (1993). Resource management complete set 2.0.

13. **Rabiner, L. R.** (1989). A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, **77**(2), 257–286.

14. **Srinivas, B.**, **N. M. Joy**, **R. R. Bilgi**, and **S. Umesh**, Subspace modeling technique using monophones for speech recognition. *In Communications (NCC), 2013 National Conference on*. 2013.

15. **Woodland, P. C.**, Speaker adaptation for continuous density hmms: A review. *In ISCA Tutorial and Research Workshop (ITRW) on Adaptation Methods for Speech Recognition*. 2001.

16. **Young, S. J.**, **J. Odell**, and **P. Woodland**, Tree-based state tying for high accuracy acoustic modelling. *In Proceedings of the workshop on Human Language Technology*. Association for Computational Linguistics, 1994.