IMPLEMENTATION OF PHONETXCAT TECHNIQUE FOR AUTOMATIC SPEECH RECOGNITION

A Project Report

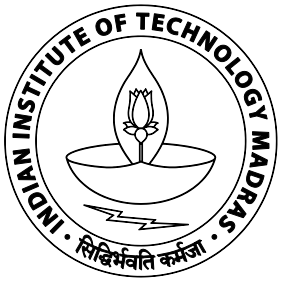
submitted by

HARIDASULA SEKHAR BABU

in partial fulfillment of the requirements

for the award of the degree of

MASTER OF TECHNOLOGY



Department of Electrical Engineering

Indian Institute of Technology Madras, India.

JUNE, 2014

THESIS CERTIFICATE

This is to certify that the thesis titled **"Implementation of PhoneTxCAT Technique for Automatic Speech Recognition"**, submitted by **H Sekhar Babu (EE12M006)**, to the Indian Institute of Technology, Madras for the award of the degree **Master 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: 13th June 2014

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 Swetha, Bhargav, Neethu, Basil, Angel 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 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 Postgraduate 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;

In this thesis, a new acoustic modelling technique, the Transform-based Phone CAT Model, for Speech Recognition Introduced by our Lab mates Bhargav Srinivas and Vimal M has been Implemented for TIMIT and some of the Indian Languages. Various simulations have been performed by varying different parameters involved in PhoneTxCAT and results have been obtained. Comparison of performance is done for various parameters and also with CDHMM and LDA+MLLT. The exact procedure to be followed and the various optimized parameters have been explained in detail. The significance of each parameter is also explained. Also, the results for various transform classes and UBMs have been given in detail.

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

TIMIT Texas Instruments MIT

SGMM Subspace Gaussian Mixture Model

WER Word Error Rate

UBM Universal Background Model

Chapter 1

Introduction

The translation of speech to text is called speech recognition. The goal is to make human machine interaction possible.

There is growing interest in the field of automatic speech recognition (ASR) because of its fascinating applications in many fields.

1.1 Automatic Speech Recognition

Automatic Speech Recognition mainly consists of three phases namely feature extraction, training and testing. Feature extraction stage extracts relevant information from the speech signal. This is series of signal processing steps which try to imitate the human perception process. The way ear responds to different frequency bands, loudness scales MFCCs (provide reference) are the conventional features used for speech recognition. Speech signal can be assumed to be stationary over a period of 25ms. Feature extraction converts the speech signal into stream of MFCC vectors.

The acoustic models which can capture the statistical behaviour and temporal information of a speech signal are built during the training phase. Generally statistical models like Hidden markov models are used for temporal pattern recognition applications like speech recognition. HMMs consists of states which represent a part of phone or word and the transition between the states capture temporal dynamics in the speech signal. The information that a particular feature vector belongs to a particular state is not known. The state information is “hidden” and HMMs assume the samples of a speech signal as the outcomes of a markov process, hence the name hidden markov model. The outcomes are assumed to be

generated by Gaussian Mixture model (GMM). HMMs in which each state is characterized by GMM is referred to as HMM-GMM system. These models are trained using Baum-Welch algorithm using MFCCs and transcriptions of a speech signal. Block diagram of standard ASR system is shown in Fig. below

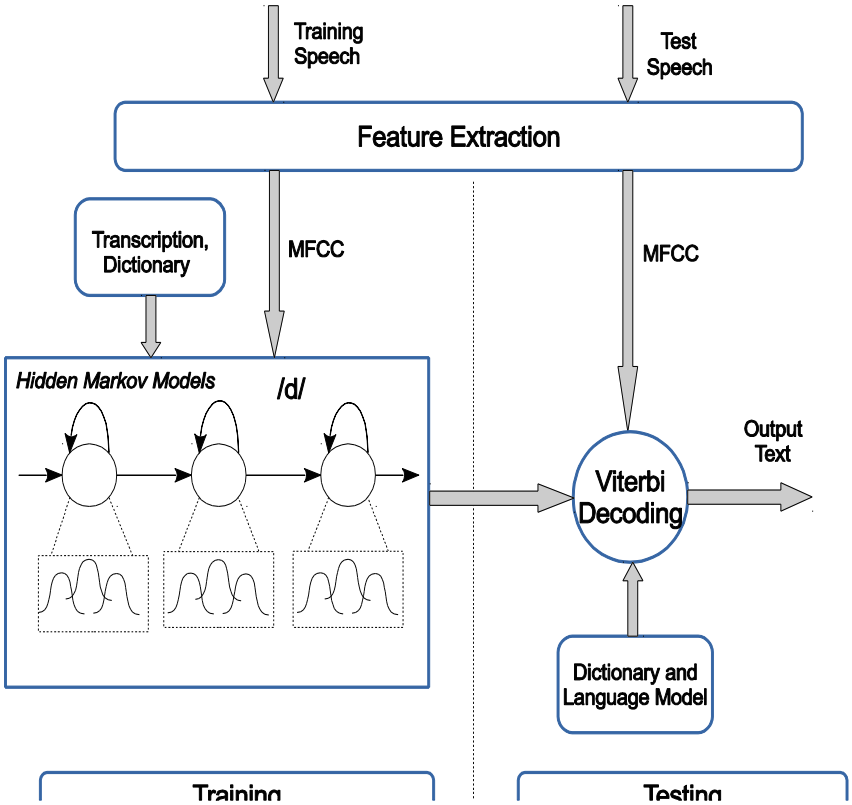


Figure 1.1: Standard ASR System

The same feature extraction procedure is performed during testing. The task is to identify the underlying phone (state) sequence for a given speech signal. Viterbi search algorithm is used for identifying the most probable state sequence. Language model defines the probability of different word sequences based on the grammar rules of the language and helps in improving the performance of recognizer.

1.2 Linguistic units

The choice of the basic linguistic unit depends on the size of vocabulary. For a small vocabulary task like digit recognition, words can be basic units. For a large vocabulary tasks

like continuous speech recognition, monophones are the linguistic units. Acoustic models for just monophones give very simple model. Also, the acoustic characteristics of a monophone is greatly affected by the preceding and succeeding monophones because of the co-articulation effect of vocal tract. Therefore, we need to build HMM considering the left and right context of the phone in consideration. Such models are referred as triphone models. There are about 40 monophones in English language which gives 40^3 different triphones. Large amount of training data is needed to estimate the parameters of all the triphones and also all these triphones are not used or not seen during the training phase. So, we "tie" the states of many similar triphones using decision tree based top-down clustering approach (reference). These are referred as tied states and these are the physical states representing many similar triphone states. We get few thousands of tied states at the end of clustering approach.

1.3 HMM-GMM system

Generally each triphone is modelled as a three state HMM with left-to-right topology. A typical HMM-GMM system has few thousands of tied states after the clustering. If each tied state is characterized by GMM, then j^{th} tied state can be expressed as follows:

$$P(O_t|j) = \sum_{i=1}^{M_j} w_{ji} N(o_t; \mu_{ji}, \Sigma_{ji})$$

where o_t is the observation or MFCC vector $w_{ji}, \mu_{ji}, \Sigma_{ji}$ are Gaussian prior, mean and covariance matrix of i^{th} Gaussian of j^{th} tied state, M_j is the total number of gaussians in j_{th} tied state. Means, Covariance and Gaussian priors constitute the parameter set of HMM-GMM system. This system is also referred as continuous density HMM (CDHMM). The

parameters of this system are estimated from the train data using expectation maximization (EM) algorithm.

1.4 Focus of the work

State tying is performed to reduce the number of parameters in the model. Nevertheless number of parameters is still high as each tied state has individual parameters. We need enough amount of training data to robustly estimate these parameters. Reliability of the estimates of parameters is not guaranteed if we have *less amount of training data*. One way to reduce the effective number of parameters is to reduce the number of Gaussians in each tied state. But this is not a solution as detailed modelling (or we need enough number of Gaussians) is necessary to capture the variability of the speakers and environments of training data. So there should be balance between detailed modelling and robust estimation of the parameters.

The main focus of the work is to effectively reduce the number of parameters of the model. The conventional parameter estimation technique of CDHMM does not exploit any relationship among tied states i.e., parameter estimates of one tied state is completely independent of estimates of the other parameters.

Chapter 2

Background

Our focus in this chapter is going to be as follows..

The parameters of CDHMM are estimated from training data using Baum-Welch algorithm. For tasks like continuous speech recognitions having large vocabulary, we need sufficient amount of training data to robustly estimate the parameters. In past, attempts have been tried to address the problem of insufficiency of the train data. We will see how techniques from speaker adaptation, verification and recognition literature are adopted for triphone state modelling.

In literature researchers often resorted to subspace modelling techniques when there is insufficiency of the data (be it either train data or test data). Let's consider the case of insufficiency of test data in speaker adaptation. MLLR is the standard technique for speaker adaptation. MLLR requires more amount (nearly 35-40 seconds) of test data to robustly estimate the adaptation parameters. Subspace modelling techniques like Eigen-voices, CAT were proposed which uses very less number of parameters which can be estimated with less amount of test data. In the same way if we consider speaker recognition, UBM is MAP adapted to every speaker. Subspace modelling technique called JFA is proposed which tries to reduce the dimensionality of a speaker. So all these subspace modelling techniques Eigen voices, CAT for speaker adaptation and JFA for speaker recognition are proposed to handle the case of insufficiency of test data. In the above paragraph, we discussed how to handle the cases where there is insufficiency of test data. Now let's consider the problem of insufficiency of train data. At this point, we have to ask a question as to how to estimate parameters of SI model when there is fewer amounts of training data. Researchers have tried

to address this problem by using subspace modelling techniques which are inspired from above mentioned speaker adaptation, recognition techniques.

For example, Eigen-triphones is inspired from Eigen voices. Eigen triphone tries to identify low resource triphone as a point in space formed by doing PCA on mean supervectors of high resource triphone. As PCA is dimensionality reduction technique, lesser number of parameters are to be estimated to get triphone parameters. But, Eigen-triphone gave very marginal improvements over conventional CDHMM.

Similarly, Subspace Gaussian Mixture Model (SGMM) is another acoustic modelling technique which is successfully applied for the problem of low resource data. This is actually inspired from JFA. The parameters of speaker in JFA and parameters of tied state in SGMM are modelled exactly in same fashion. But SGMM is not very intuitive.

Similarly, UBM adaptation to tied state and UBM adaptation to speaker. This is not very successful.

Because of all these parallels between speaker modelling and tied state modelling in literature, is there any other subspace modelling technique for speaker that we can replicate for tied state modelling. As mentioned earlier CAT

2.1 Subspace Gaussian Mixture Model (SGMM)

While JFA assumes that *GMM supervector of speaker* is coming from low dimensional factors , SGMM assumes that the *GMM supervector of tied state* is coming from low dimensional factors. This can be expressed as follows

$$\boldsymbol{\mu_j = m + Mv_j + Nv^s}$$

where μ_j ,m are supervectors of j^{th} tied state and UBM respectively. M,N are phonetic, speaker subspaces respectively (as compared to channel and speaker spaces in JFA). M is

named as phonetic space as SGMM is modelling at phone level. If we consider more basic version of SGMM neglecting speaker space (which is used only for adaptation), then

$$\boldsymbol{\mu_j = m + M v_j}$$

Here M is *state independent phonetic space* from which tied state parameters are derived using v_j as in above equation. v_j is called as *state specific vector.* The dimension of M is typically around 40 to 45, experimentally chosen, which is way less compared to dimension of μ_j . In other words, high dimensional μ_j is lying in the space of low dimension M and hence the name subspace Gaussian mixture model (SGMM). If each tied state is assumed to contain I

Gaussians i.e., $\mu_j = [\mu_{j1}^T \ldots \mu_{jI}^T]^T$ and $M = [M_1^T \ldots M_I^T]^T$ then expression for i^{th} mean of j^{th} tied state can be written as follows

$$\boldsymbol{\mu_j = m_i + M_i v_j}$$

where $\boldsymbol{M_i}$ is the i^{th} submatrix of M. In SGMM, the covariances are shared across all the tied states so that we have a state independent covariance Σ_i. The Gaussian priors are obtained using soft-max function and these can be expressed as follows

$$\boldsymbol{\Sigma_{ji} = \Sigma_i}$$

$$\boldsymbol{w_{ji} = \frac{\exp(w_i^T v_j)}{\sum_{i'=1}^{I} \exp(w_{i'}^T v_j)}}$$

where $\boldsymbol{w_i}$ is called as Weight Projection Vector and it defines the subspace in which the unnormalized log Gaussian priors lie. The above three equations define the complete modelling structure of a tied state in SGMM. $v_j, m_i, w_i, \Sigma_i, M_i$constitute the total parameter set of SGMM. We can observe that all the means and Gaussian priors of a tied state are obtained using single state specific vector v_j . The detailed of explanation of SGMM can be found in (add reference). The training of the SGMM system begins with the traditional CDHMM system. An UBM is built by repeatedly merging the Gaussians of CDHMM to get a

desired number of Gaussians with diagonal covariances. These Gaussians are trained with around 8 iterations of EM algorithm for full covariance re-estimation. The UBM need not necessarily be built from a specific CDHMM system; any generic UBM can be used. This UBM is used to initialize the SGMM. The parameters are initialized in such a way that the initial p.d.f. in every tied state is equal to UBM. CDHMM system provides the Viterbi alignments for the initial SGMM parameter re-estimation. Once the SGMM parameters are estimated by EM algorithm to a sufficient extent, SGMM training can be continued with self-alignment (alignments from the SGMM itself).

2.2 Outline to Cluster Adaptive Training (CAT)

In CAT, all the training speakers are initially divided into "P" groups or clusters and a CDHMM is built, known as *cluster model*, for each cluster using the data of speakers belonging to that particular cluster. The parameters of each speaker model are obtained as linear interpolation of the parameters of all the P clusters. If p^{th} cluster parameter is represented as μ_{ji}^{p},then mean of j^{th} state, i^{th} Gaussian for speaker"s" in CAT model is obtained as follows

$$\mu_{ji}^{s} = \lambda_s^1 \mu_{ji}^1 + \lambda_s^2 \mu_{ji}^2 + \cdots + \lambda_s^p \mu_{ji}^p = \left[\mu_{ji}^1 \, \mu_{ji}^2 \ldots \mu_{ji}^p\right][\lambda_s^1 \, \lambda_s^2 \ldots \lambda_s^p]^T = M_{ji}\lambda^s$$

where M_{ji} is the matrix of means from P clusters and λ^s is referred as the *speaker weight vector*. Thus the model-based CAT parameters are the model parameters $\mathcal{M}$=$\{\{M_1 \ldots M_{JI}\}, \{\Sigma_1 \ldots \Sigma_{JI}\}\}$, where Σ_{ji} is the covariance of i^{th} Gaussian component of j^{th} tied state and cluster weight vector parameters $\Lambda = \{\lambda^s\}$, $1 \leq s \leq S$. During the training phase the cluster model parameters $\mathcal{M}$ and weight vectors for training speakers Λ are iteratively estimated so that clusters become compact. During the test phase, adapted model

for the test speaker is obtained by estimating speaker weight vector λ^s. As the number of parameters to be estimated is less (only P parameters), CAT supports model adaptation when the amount of adaptation data available is less (i.e., rapid adaptation).

Chapter 3

Subspace Model and CAT

Gaussian Mixture Model (GMM) characterizing the distribution of each CDHMM state is the standard acoustic model in speech recognition. The requirement of sufficiently large amount of training data for the robust estimation of these GMM parameters (i.e., context dependent phone models) is an important issue for building these models. On-going research in the field of acoustic modelling is gaining momentum in the present scenario where robust models need to be built on fewer amounts of training data. Efficient and robust estimation of acoustic model parameters utilizing limited availability of training data is steering the research focus to techniques where the number of parameters to be estimated are few. SGMM and Canonical State Models (CSM) are two acoustic modelling techniques which exploits the relationship between the context dependent phone models (or triphones). Both these methods have similarities to the Cluster Adaptive Training (CAT), a speaker adaptation technique. While SGMM exploits the correlation among GMM parameters of tied state, CSM strives to transform a canonical model to a context dependent state. Both these methods achieve considerable parameter reduction and hence can be used in cases where the amount of training data available is limited. Following similar lines of arguments as the afore mentioned techniques, we propose a new acoustic modelling technique called Monophone Subspace Model (MSM) which takes a more intuitive approach to the estimation of tied state parameters. Here, we assume that tied state models are formed by the liner combination of Monophone models. It is imperative that we attribute the inspiration of our proposed technique to CAT.

In CAT, the means of the new speaker HMM is formed by the linear interpolation of several

cluster model means. A cluster is formed by grouping together a set of speakers and during CAT training phase a HMM is built for each of these clusters. Similarly in the proposed Subspace Model, the means of the triphone model is formed by the linear combination of monophone model means.

On an implementation level, our technique is more similar to SGMM where the means and mixture weights of the tied states vary in the full GMM parameter space and the mapping for a particular tied state is done via a state projection vector. SGMM also constraints the covariance's of all the HMM states to be the same as that of the canonical model. In our proposed method, the covariance's of the monophone models are tied together, thereby constraining the covariance's of the tied state model to be same as that of the monophone models.

3.1 CAT & MSM

3.1.1 Cluster Adaptive Training (CAT)

CAT is a rapid adaptation technique which compactly represents a speaker with lesser number of parameters. It gains an upper hand over SI-CDHMM and gives comparable performance to that of SD Model, but without the added hassle of requiring more amount of training data for a particular speaker.

In CAT, all the training speakers are initially divided into "P" groups or clusters and an HMM model known as *cluster model* is built for each cluster using the data of the speakers belonging to that particular cluster. The parameters of each speaker model is obtained by the linear interpolation of the parameters of all the P clusters. The mean of j^{th} state, m^{th} Gaussian for speaker "s" in CAT model is as follows:

$$\mu_{ji}^{s} = \left[\mu_{ji}^{1}\, \mu_{ji}^{2} \ldots \mu_{ji}^{p}\right]\left[\lambda_{s}^{1}\, \lambda_{s}^{2} \ldots\, \lambda_{s}^{p}\right]^{T} = M_{ji}\lambda^{s}$$

Where M_{ji} is the matrix of means from P clusters and λ^s is referred to as the *speaker weight vector*.

During training phase the cluster model parameters and cluster weight vectors for training speakers are iteratively estimated so that clusters become compact. During the test phase, adapted model for the test speaker is obtained by estimating speaker weight vector λ^s. As the number of parameters to be estimated is less (only P parameters), CAT supports model adaptation when the amount of adaptation data available is less (i.e., rapid adaptation).

The driving force behind MSM is the need to build a robust acoustic model with limited amount of training data. The proposed technique accomplishes this by sharing a large number of parameters among the tied states, thereby reducing the total number of parameters to be estimated for the model.

3.1.2 MSM

MSM technique is closely related to CAT as the former adopts the parameter estimation framework of the latter. However, CAT is a model adaptation technique and MSM operates on the acoustic modelling level. In MSM, the parameters of each tied state are obtained by linear interpolation of parameters of the monophone models. Mean μ_{ji} of the j^{th} state, i^{th} Gaussian is as follows:

$$\boldsymbol{\mu_{ji} = M_i v_j}$$

where, M_i is the matrix formed by stacking the i^{th} mean of all the monophone models and v_j is the *tied state weight vector* for the j^{th} tied state as shown in Eq.above

$$\boldsymbol{M_i = \left[\mu_i^1\ \mu_i^2 \dots \mu_i^p\right];\ v_j = [v_i^1\ v_i^2 \dots v_i^K]^T}$$

and $\boldsymbol{\mu}_i^k$ is i^{th} mean of k^{th} monophone and P is the total number of monophones. Hence to estimate the means of each triphone model, only v_j needs to be estimated.

3.1.3 Analogy between MSM & CAT

In this section, we try to draw an analogy between the proposed Subspace Model and CAT. As already mentioned in the previous section, CAT is a speaker adaptation technique and Subspace Model is an acoustic modelling technique. Consequently, the term "cluster" in CAT refers to a HMM for a group or cluster of speakers and that in MSM is thought of as monophone models. M_{ji} is formed from the cluster means and M_i from monophone GMM means in CAT and Subspace Model respectively. In CAT, each speaker is associated with weight vector λ^s which weights the cluster means. Similarly, in MSM the tied state specific weight vector v_j, which weights monophone means, identifies each of the triphone models.

3.1.4 Overall Training Procedure

A. Initialization

1. Build monophone GMMs of required size (96 or 128 mix) from the train data. Form M_i by stacking together the i^{th} mean of all the monophones as shown in Eq. 3.3.

2. Build a standard CDHMM baseline model (6 mix). From this model, both the initial alignment information to bootstrap the MSM and phonetic context information (i.e, decision tree) are taken.

3. Copy the monophone GMMs built in step 1 to all the tied states (for eg. /aa/ is copied to all tied states having /aa/ as middle context) in CDHMM. Now each tied state contains GMM of the required size. This serves as an initialization for our MSM.

B. Training the MSM consists of two phases.

• Phase1: Update the model parameters (shown in Eqs. 3.6, 3.7, 3.8) using *alignment* $\gamma_{ji}(t)$ *from the baseline model.*

• Phase2: Update the model parameters using self-alignment i.e., *alignment from MSM*.

Both these phases has to undergo through at least 7-8 iterations of EM as mentioned in section 3.1. The whole training procedure can be summarized as shown in Fig 3.1

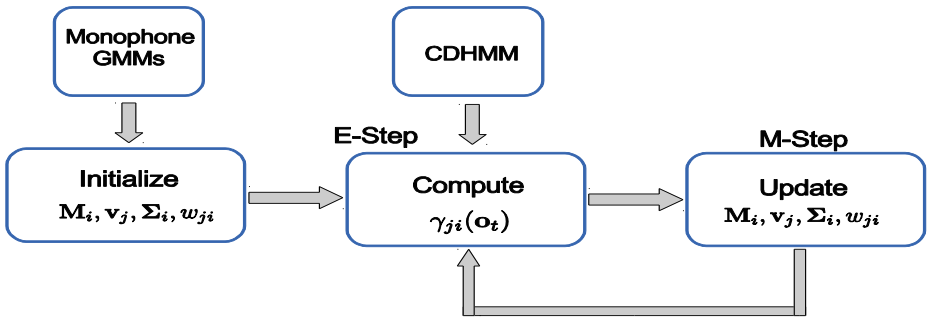


Figure 3.1: MSM training procedure

Chapter 4

PhoneCAT

Phone CAT (Srinivas et al. (2013)) is an acoustic modelling technique inspired from the Cluster Adaptive Training (CAT) (Gales (2000)) for rapid speaker adaptation, which was described in Section 3.1.1. 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 modelled 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. (2011a)), 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.

4.1 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 $\boldsymbol{\mu}_i^{\boldsymbol{P}}$ 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 v_j, which is called as the state vector. Thus the mean of the i^{th} Gaussian of the j^{th} state is modelled as follows:

$$\boldsymbol{\mu_{ji} = M_i v_j,}$$

Where ; $\boldsymbol{v_j = \left[v_i^{(1)}\ v_i^{(2)} \ldots v_i^{(P)}\right]^T}$ is the state vector , and $\boldsymbol{M_i = \left[\mu_i^{(1)}\ \mu_i^{(2)} \ldots \mu_i^{(P)}\right]}$ 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 $\boldsymbol{\mu_i^{(1)}\ \mu_i^{(2)} \ldots \mu_i^{(P)}}$ 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.

4.2 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, W_p, associated with each cluster p. The cluster-specific mean $\boldsymbol{\mu_i^{(P)}}$ for Gaussian component i is specified as:

$$\boldsymbol{\mu_i^{(P)}} = W_p \xi_i = W_p [\mu_i \; 1]^T,$$

where ξ_i is the extended canonical model mean $[\mu_i \; 1]^T$ with μ_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

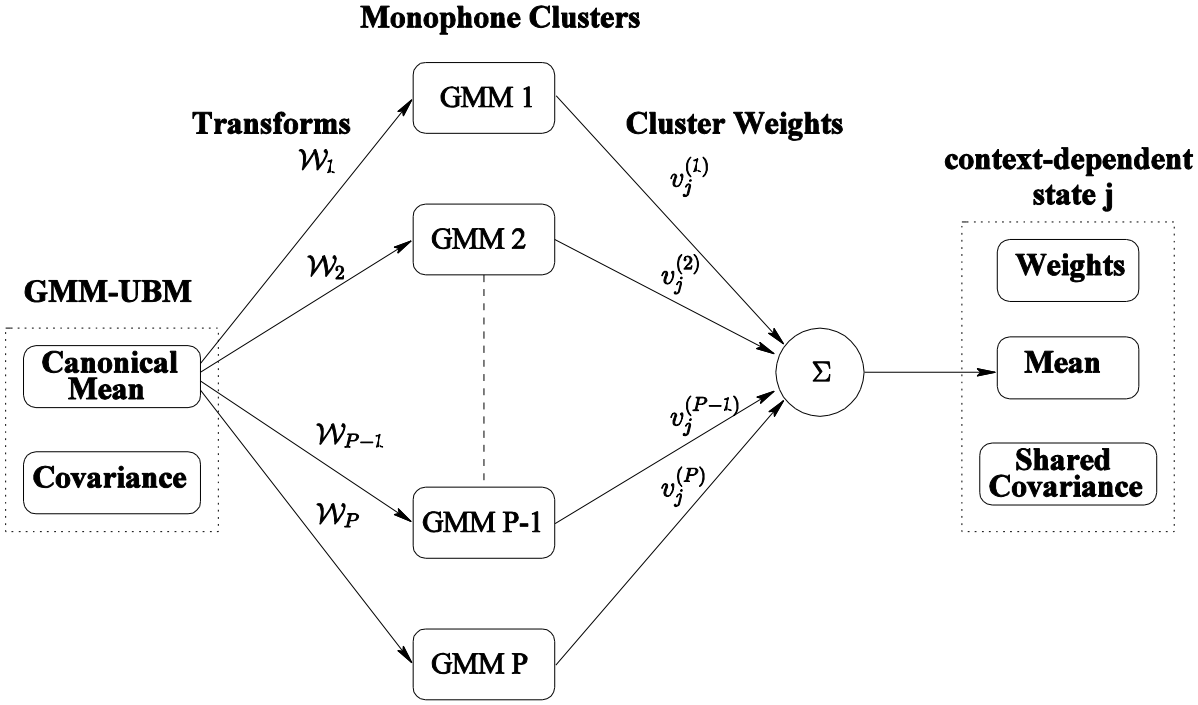


Figure 4.1: Transform based PhoneTxCAT

as a weighted linear interpolation of the cluster-specific means given in

$$\boldsymbol{\mu_{ji}} = \left[\boldsymbol{\mu_i^{(1)}} \; \boldsymbol{\mu_i^{(2)}} \ldots \boldsymbol{\mu_i^{(P)}}\right] \boldsymbol{v_j} = \left[\boldsymbol{\mu_i^1} \; \boldsymbol{\mu_i^2} \ldots \boldsymbol{\mu_i^p}\right] \begin{bmatrix} \boldsymbol{v_j^{(1)}} \\ \cdot \\ \cdot \\ \boldsymbol{v_j^{(P)}} \end{bmatrix}, = \sum_{\boldsymbol{p=1}}^{\boldsymbol{P}} \boldsymbol{\mu_i^p v_j^{(p)}},$$

$$= \left(\sum_{\boldsymbol{p=1}}^{\boldsymbol{P}} \boldsymbol{v_j^{(p)} W_p} \right) \boldsymbol{\xi_i}$$

where $\boldsymbol{v_j} = \left[\boldsymbol{v_i^{(1)}} \; \boldsymbol{v_i^{(2)}} \ldots \boldsymbol{v_i^{(P)}}\right]^{\boldsymbol{T}}$ is the state vector. The model is as shown in above Fig. .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.

4.2.1 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 $\boldsymbol{v_j}$. The covariance matrix Σ_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:

$$\boldsymbol{P(x|j) = \sum_{i=1}^{I} w_{ji}\, N\left(x; \mu_{ji}, \Sigma_i\right)},$$

$$\boldsymbol{w_{ji} = \frac{\exp(w_i^T v_j)}{\sum_{i'=1}^{I} \exp(w_{i'}^T v_j)}},$$

where $x \in \mathbb{R}^D$ is the feature vector, $1 \leq j \leq J$ is the state index of the context-dependent state, w_i is the weight projection vector,μ_{ji} is obtained as in last equation 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 modelled

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 v_j . The rest $w_{i,}\Sigma_{i,}\mu_{i,}W_p$ are global parameters and are independent of state. Hence there is a large amount of data to estimate these parameters.

4.2.2 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 $\Lambda = \{\boldsymbol{v_j}\}$, $1 \leq j \leq J$ canonical model parameters $\mathcal{M}$=$\{\{M_1 \ldots M_{JI}\}, \{\Sigma_1 \ldots \Sigma_{JI}\}\}$, and the subspace parameters S= $\{\{w_1 \ldots w_I\}$, $\{W_1 \ldots W_P\}\}$.The training scheme followed is analogous to the case of transform-based CAT:

1. Re-estimate the state vector parameters Λ using $\{\mathcal{M}, S\}$ and the pre-update value of Λ.
2. Re-estimate the subspace parameters S given $\{*, \mathcal{M}\}$ and the pre-update value of S.
3. Re-estimate2 the canonical model parameters $\mathcal{M}$ given $\{S,*\}$ 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.

4.3 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. (2011a)). 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.

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 v_j is assigned a vector giving a weight 1 to only one cluster depending on a mapping function C and 0 to every other cluster. Therefore the initialization is:

$$W_p = [\boldsymbol{I_{D\times D}}\ \boldsymbol{0_{D\times 1}}], 1 \le p \le P \boldsymbol{\mu_i} = \boldsymbol{\mu_i^{(UBM)}}, \boldsymbol{1 \le i \le I} \boldsymbol{\Sigma_i} = \boldsymbol{\Sigma_i^{(UBM)}}, \boldsymbol{1 \le i \le I} \boldsymbol{v_j} = \boldsymbol{e_k} \in \mathbb{R}^{\boldsymbol{P}}, \quad \boldsymbol{1 \le j \le J, k = C(j) w_i = 0} \in \mathbb{R}^{\boldsymbol{P}}, \quad \boldsymbol{1 \le i \le I}$$

where $I_{D\times D}$ is a $D \times D$ identity matrix with D being the dimension of the feature vector, $0_{D\times 1}$ is a vector of D zeros, $\mu_i^{(UBM)}$, $\Sigma_i^{(UBM)}$ are the mean and the covariance matrix of the i^{th} Gaussian component of the UBM, e_k is a P dimensional unit vector with the k^{th} dimension as 1 and every other dimension 0 and C : $\{1,..,J\} \rightarrow \{1,\dots,P\}$ is a mapping from the state j to cluster p.

In the simplest case, the mapping function can be defined such that 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.

4.4 Training of the model

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

4.4.1 Expectation Maximization (EM) algorithm

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

$$\boldsymbol{Q} = \sum_{\boldsymbol{j,i,t}} \boldsymbol{\gamma_{ji}(t)} \left[\boldsymbol{log(w_{ji})} - \frac{\mathbf{1}}{\mathbf{2}} \boldsymbol{\left(x(t) - \mu_{ji}\right)^T \sum_i^{-1} \left(x(t) - \mu_{ji}\right)} \right],$$

Where $\gamma_{ji}(t)$=P(j,i|t) is the posterior probability of the j^{th} state, i^{th} Gaussian component at time t, x (t) is the feature vector at time t and w_{ji} and μ_{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 $v_j, w_{i,}\Sigma_{i,}\mu_{i,}W_p$ are obtained by optimizing 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.

4.4.2 Estimation of Cluster Transforms

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

$$W_p{}^{(k)} = k_p{}^{(k)}\left[G_p{}^{(k)}\right]^{-1}$$

$$k_p{}^{(k)} = \sum_{i=1}^{I} \frac{1}{\sigma_{kk}^{(i)2}} \left[\left\{ k_{pk}{}^{(i)} - \sum_{l \neq p}^{P} g_{lp}^{(i)} W_l{}^{(k)} \xi_i \right\} \xi_i^T \right]$$

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

4.4.3 Estimation of State Vectors

The auxiliary function for state vectors v_j consists of two parts, one related to the mean and one to the weights. The dependency of the weights on 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 v_j .

The update equation for v_j is given by

$$v_j = {G_j}^{-1} k_j$$

4.4.4 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:

$$\boldsymbol{\mu_i} = \left[\sum_{p=1}^{P}\sum_{q=1}^{P} \boldsymbol{g_{pq}^{(i)} A_p^T \Sigma_i A^q}\right]^{-1}\left[\sum_{p=1}^{P} \boldsymbol{A_p^T {\Sigma_i}^{-1}}\left(\boldsymbol{k_p^{(i)T}} - \sum_{q=1}^{P} \boldsymbol{g_{pq}^{(i)} b_q}\right)\right]\boldsymbol{\Sigma_i}$$

$$= \boldsymbol{diag}\left\{\frac{\boldsymbol{L^{(i)} - 2\sum_{p=1}^{P} k_p^{(i)}\left(M_i^{(p)}\right)^T + \sum_{p=1}^{P}\sum_{q=1}^{P} g_{pq}^{(i)} M_i^{(p)} M_i^{(q)T}}}{\boldsymbol{\sum_j \gamma_{ji}}}\right\}$$

where A_p and b_p are the first D columns and the $(D+1)^{th}$ column of $W_p = [A_p\ b_p]$ respectively, $\boldsymbol{M_i^{(p)}} = W_p \xi_i, {k_p}^{(i)}$ is the p^{th} row of the statistics (3.16), $L^{(i)}$ is the sufficient statistics defined by

$$L^{(i)} = \sum_{j,t} \boldsymbol{\gamma_{ji}(t) x(t)\, x(t)^T}$$

The estimation of μ_i depends of the current value of Σ_i and vice-versa. First, the means are updated and the updated means are used to update Σ_i.

4.4.5 Estimation of weight projections

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

$$\boldsymbol{w_i^{(n)} = w_i^{(n-1)} + F_i^{(n)-1} g_i^{(n)}},$$

$$\boldsymbol{F_i^{(n)} = \sum_j max\left(\gamma_{ji}, \gamma_j w_{ji}^{(n-1)}\right) \mathrm{v}_j \, v_j^T},$$

$$\boldsymbol{g_i^{(n)} = \sum_j \left(\gamma_{ji} - \gamma_j w_{ji}^{(n-1)}\right) \mathrm{v}_j},$$

Where $.^{n}$ represents the value at n^{th} iteration

4.5 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.

4.5.1 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 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.

4.5.2 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.

Chapter 5

Experiments & Results

5.1 Experimental Setup

The performance of the Transform-based Phone CAT model is tested on the TIMIT database along with HINDI and TAMIL languages of MANDI database. TIMIT has a total of 3,396 utterances for training and 192 utterances for testing. The HINDI database consists of different hours of data namely 1hr, 3hr, 5hr, and 22hrs of data for training along with 5974 utterances for testing. Similarly, TAMIL database also has 1hr, 3hr, 5hr and 22hrs of data for training along with 3564 utterances for testing. .

13-dimensional MFCC were used as features for parameterizing 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. (2011b)) 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 and also LDA+MLLT to initialize the Phone CAT acoustic models.

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.

5.2 Parameters

The LDA+MLLT system used for TIMIT task had a total of 1040 tied states and 22047 Gaussians. The dictionary had a set of 38 phones. The silence was modelled 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. Since the feature vector used was of 39 dimension, full-MLLR matrices of dimension 39 x 40 was used for the cluster transforms. The UBM was initialized by a bottomup clustering approach by merging the Gaussians from the LDA+MLLT system till I mixtures were obtained. I value was kept as 400 .

The baseline LDA+MLLT system used for TAMIL and HINDI task had different number of tied states and Gaussians for different hours of data which is mentioned in the next table Dictionary with 39 phones was used for TAMIL and 41 phones for HINDI. The modelling of the phones was similar to that in TIMIT task. The Transform-based Phone CAT model initialized from this system had the following tied states mentioned in the next table

5.3 Experiments and Discussion

Tables 5.1 to 5.14 show the results of experiments evaluating the Transform-based Phone CAT models on the TIMIT, HINDI and TAMIL tasks respectively. The details of the experiments, along with the motivation and the conclusions are described in the subsequent sections.

5.3.1 Baseline system

At first, basic CDHMM system is built and then LDA+MLLT is done on top of it which is used to initialize PhoneTxCAT . All the other experiments are compared with this baseline system in terms of accuracy.

For TIMIT, the baseline used is only CDHMM and PhoneTxCAT is built on it

TIMIT:

Name of Expt	Transform Classes	Tied States formed	% WER
CDHMM	-	1049	28.38
LDA+MLLT	-	1040	25.45
PhoneTxCAT	4	841	**23.99**

Table 5.1 TIMIT Results

TAMIL:

Baseline Results

The best CDHMM and LDA+MLLT results have been tabulated below and all the remaining iterations have been given in APPENDIX B

Hours of Data	Pdfs,Gaussians	CDHMM(%WER)	LDA+MLLT(%WER)
1	213,1504	42.00	37.93
3	267,1803	31.61	27.34
5	668,5414	25.91	23.17
22	1114,23262	22.11	19.37

Table 5.2 TAMIL Baseline Results

PhoneTxCAT Results

The best PhoneTxCAT results have been given below and all the other iterations are given in APPENDIX B

Expt	Hours of Data	UBM Mixtures	Transform class	%WER
PhoneTxCAT	1	128	1	37.19
	3	256	2	28.06
	5	500	1	22.51
	22	750	2	20.04

Table 5.3 TAMIL PhoneTxCAT Results

HINDI:

For 1hr data, the various iterations used in baseline system building are shown in table5.10 and the best is shown in bold

(Tied States, Gaussians)	CDHMM	LDA+MLLT
262,1203	16.50	15.10
262,1803	16.26	14.73
305,1405	**16.18**	**14.12**
305,2105	16.05	15.32
344,1606	15.29	14.75
344,2408	15.28	15.10
382,1804	15.71	15.53
382,2707	16.46	14.60
422,2002	15.62	14.65
422,3006	15.93	15.66

Table 5.4 HINDI 1hr Baseline Results

The best baseline is taken and PhoneTxCAT is built on top of it for different transform classes and result is tabulated in table 5.11 and the best is given in bold

Expt Name	UBM Mixtures	Transform Classes	% WER
CDHMM	-	-	16.18
LDA+MLLT	-	-	14.12
PhoneTxCAT	64	1	14.08
		2	14.40
		3	**13.71**
		4	14.14

Table 5.5 HINDI 1hr PhoneTxCAT Results

For 3hr, 5hr and 22hrs of data, the best baseline is directly taken and PhoneTxCAT is done on it and the corresponding results are tabulated in tables 5.12, 5.13, 5.14.

Expt Name	UBM Mixtures	Transform Classes	% WER
CDHMM	-	-	11.59
LDA+MLLT	-	-	10.77
PhoneTxCAT	200	1	**10.30**
		2	10.32
		3	10.47
		4	10.70

Table 5.6 HINDI 3hr Results

Expt Name	UBM Mixtures	Transform Classes	% Accuracy
CDHMM	-	-	9.06
LDA+MLLT	-	-	8.53
PhoneTxCAT	256	1	**7.67**
		2	7.90
		3	7.67
		4	7.80

Table 5.7 HINDI 5hr Results

Expt Name	UBM Mixtures	Transform Classes	% WER
CDHMM	-	-	5.75
LDA+MLLT	-	-	5.68
PhoneTxCAT	300	1	5.64
		2	5.70
		3	5.70
		4	5.65
	400	1	5.44
		2	**5.39**
		3	5.40
		4	5.47

Table 5.8 HINDI 22hr Results

5.3.2 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. 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.

5.3.3 Multiple Transform Classes

The number of transform classes have been varied and results have been tabulated above. The Number of transform class which gives best performance for a particular data set can only be determined through a set of experiments.

5.4 Observations

The experiments in Section 5.3 show that the Transform-based Phone CAT model in general performs better than the conventional HMM-GMM and LDA+MLLT systems. The higher discriminatory capability of this model can be attributed to modelling 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.

Chapter 6

Conclusions

A new kind of acoustic model, the Transform-based Phone CAT model, is implemented. 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 modelling 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 TIMIT, HINDI and TAMIL tasks confirm that the model gives better results than the conventional HMM-GMM and LDA+MLLT systems. On the TIMIT task, the Transform-based Phone CAT model shows an improvement of 1.46% absolute, which is a 5.73% relative improvement in Word Error Rate (WER). Similar improvements can laso be observed with HINDI and TAMIL databases. Also, in general PhoneTxCAT gives better %WER for transform classes 1 and 2. Being very similar to the SGMM, this model offers scope for similar modelling 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

Things To Be Noted While Performing The Experiments:

- All the experiments should be performed under IITM Libra Cluster because change in environment is giving different result.
- Experiments should be done by using 20 cores as change in splits gives different performance.
- Also, all the features should be sorted out because unsorted features are giving different results.
- All the scripts that are to be used should be latest standard KALDI scripts because they have slight modifications compared to old scripts and hence performance might be different.
- Optimisation should be done for both LDA+MLLT and CDHMM but not just alone CDHMM because the input to PhoneTxCAT is LDA+MLLT in our current experiments.
- The value of the TIED States formed in PhoneTxCAT will be reaching a saturation limit at certain point beyond which they will not be any increase in TIED states and performance.
- Also , one should be careful at giving the number of TIED states to CDHMM and LDA+MLLT as giving too many to lesser amount of data will hinder performance.
- The best result in PhoneTxCAT might be obtained for any transform class which is unknown and can be determined only by experiments.
- Similarly, the number of Gaussian Mixtures in UBM can also be determined only experimenting many different UBMs**.**

Appendix B

Detailed list of all the iterations done for TAMIL

Each of the different hours of data like 1hr, 3hr,5hr,22hr have been optimised after testing for many pdf, Gaussian combinations and the best is given in bold.

(# pdfs, # Gaussians)	CDHMM(%WER)	LDA+MLLT(%WER)
174, 802	43.78	39.86
174,1202	43.19	40.67
213,1005	41.51	39.83
213,1504	**42.00**	**37.93**
249,1202	42.10	41.07
249,1808	42.13	42.30
260,1402	42.62	39.29
260,2107	43.95	39.44
260,1606	41.07	38.57
260,2405	43.26	39.17
260,1804	43.16	39.24
260,2710	44.25	41.34
260,2007	42.69	38.97
260,3010	44.64	40.03
260,2207	42.60	39.12
260,3312	43.53	41.54
260,2408	42.77	41.86
260,3612	45.04	42.77
260,2605	43.46	41.51
260,3915	43.83	41.21

Table B.1 TAMIL 1hr Baseline Results

Using the best value of LDA+MLLT, PhoneTxCAT is built on it and the best result is given in bold.

Expt Name	UBM Mixtures	Transform Classes	%WER
CDHMM	-	-	42.00
LDA+MLLT	-	-	37.93
PhoneTxCAT	32	1	39.17
		2	39.34
		3	39.34
		4	38.77
	64	1	37.64
		2	38.33
		3	38.30
		4	38.35
	100	1	37.24
		2	37.76
		3	40.15
		4	37.78
	128	1	**37.19**
		2	38.82
		3	38.47
		4	38.77
	192	1	37.73
		2	37.69
		3	39.36
		4	39.56
	256	1	39.88
		2	38.18
		3	41.46
		4	42.69

Table B.2 TAMIL 1hr PhoneTxCAT Results

(Tied States, Gaussians)	CDHMM	LDA+MLLT
179,803	33.46	30.80
179,1207	32.31	28.65
220,1004	32.92	28.73
220,1502	31.17	28.78
267,1204	32.72	28.48
267,1803	**31.61**	**27.34**
306,1408	32.21	28.16
306,2107	32.31	29.47
338,1603	32.21	27.74
338,2409	32.58	27.89
372,1803	32.26	27.71
372,2705	32.11	29.47
405,2010	31.74	29.54
405,3011	32.72	28.48
443,2206	32.08	28.21
443,3311	31.71	28.36
459,2405	32.01	29.91
459,3610	32.75	30.40
459,2608	32.21	29.42
459,3911	33.07	29.89
459,2807	32.63	29.94
459,4215	33.19	30.50
459,3009	32.31	30.38
459,4512	33.49	30.85

Table B.3 TAMIL 3hr Baseline Results

Expt Name	UBM Mixtures	Transform Classes	%WER
CDHMM	-	-	31.61
LDA+MLLT	-	-	27.34
PhoneTxCAT	128	1	29.32
		2	29.69
		3	30.55
		4	29.99
	200	1	29.00
		2	30.31
		3	29.84
		4	29.54
	256	1	28.33
		2	**28.06**
		3	28.60
		4	31.17
	300	1	29.96
		2	30.73
		3	31.02
		4	32.70
	350	1	29.22
		2	29.71
		3	30.48
		4	30.70
	400	1	29.34
		2	29.64
		3	31.05
		4	32.23
	450	1	29.74
		2	30.08
		3	31.59
		4	32.65

Table B.4 TAMIL 3hr PhoneTxCAT Results

(Tied States, Gaussians)	CDHMM	LDA+MLLT
410,2008	29.81	25.35
410,3012	28.83	24.16
442,2208	29.94	26.78
442,3310	28.92	23.96
475,2408	29.39	25.69
475,3608	29.15	23.69
515,2609	28.21	24.31
515, 3914	28.11	23.91
554,2808	28.92	25.30
554,4214	27.00	24.04
583,3012	28.60	24.83
583,4518	27.89	23.45
624,3210	27.84	23.52
624,4817	27.49	24.04
650,3408	28.11	24.58
650,5112	27.00	23.35
668,3612	28.01	24.19
668,5414	**25.91**	**23.17**
668,3813	28.26	23.49
668,5716	27.22	23.27
668,4010	27.42	24.16
668,6020	27.54	24.90

Table B.5 TAMIL 5hr Baseline Results

Expt Name	UBM Mixtures	Transform Classes	% WER
CDHMM	-	-	25.91
LDA+MLLT	-	-	23.17
PhoneTxCAT	300	1	24.36
		2	23.67
		3	24.61
		4	25.49
	350	1	23.91
		2	23.49
		3	23.49
		4	24.04
	400	1	24.98
		2	25.52
		3	24.61
		4	26.21
	450	1	24.33
		2	24.31
		3	23.49
		4	24.78
	500	1	**22.51**
		2	23.37
		3	24.21
		4	23.30
	550	1	23.72
		2	25.05
		3	24.98
		4	25.17

Table B.6 TAMIL 5hr PhoneTxCAT Results

(Tied States, Gaussians)	CDHMM	LDA+MLLT
887,16145	22.43	19.84
887,18451	22.01	19.79
921,16846	21.45	19.45
921,19248	22.31	19.74
952,17556	21.50	19.87
952,20060	21.32	19.69
993,18246	21.69	19.47
993,20850	21.96	19.60
1030,18948	22.14	20.04
1030,21652	22.16	19.37
1071,22458	21.92	19.45
1114,23262	**22.11**	**19.37**
1139,24052	22.24	20.06
1172,21767	22.93	19.40
1172,24861	22.36	20.09
1203,22465	21.89	19.45
1203,25662	22.43	19.89
1245,23154	22.09	20.16
1245,26481	22.38	19.97

Table B.7 TAMIL 22hr Baseline Results

Expt Name	UBM Mixtures	Transform Classes	% WER
CDHMM	-	-	22.11
LDA+MLLT	-	-	19.37
PhoneTxCAT	400	1	21.05
		2	20.41
		3	20.63
		4	21.42
	450	1	20.61
		2	20.34
		3	20.78
		4	20.78
	500	1	20.80
		2	20.43
		3	20.93
		4	20.53
	550	1	20.83
		2	20.46
		3	21.30
		4	20.80
	600	1	20.95
		2	20.43
		3	20.06
		4	21.69
	650	1	20.80
		2	20.19
		3	21.35
		4	20.34
	700	1	20.68
		2	20.43
		3	20.38
		4	22.29
	750	1	20.63
		2	**20.04**
		3	21.15
		4	20.16
	800	1	20.85
		2	20.36
		3	20.56
		4	21.64

Table B.8 TAMIL 22hr PhoneTxCAT Results

Bibliography

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

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

3. 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.

4. 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.

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

6. 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. 2011b.

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

8. 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.