Discriminative Adaptive Training and Bayesian Inference for Speech Recognition

Chandra Kant Raut

University of Cambridge
Emmanuel College

December 2009

This dissertation is submitted to the University of Cambridge
for the degree of Doctor of Philosophy.

Declaration

This dissertation is the result of my own work carried out at the Cambridge University and includes nothing which is the outcome of any work done in collaboration except where explicitly stated. It has not been submitted in whole or in part for a degree at any other university. Some of the work has been previously published in international conference proceedings [146, 147]. The length of this thesis including footnotes, appendices and references is approximately 54000 words. This thesis contains 34 figures and 15 tables.

Summary

State-of-the-art speech recognition systems are based on statistical techniques and use hidden Markov models (HMMs) as acoustic models. These acoustic models are trained from a large amount of speech data usually collected from a large number of speakers and in different acoustic environments. The training data contains both the desired variabilities for speech recognition as well as unwanted variabilities from speakers and the environment. Adaptive training plays an important role in building acoustic models from such non-homogeneous data. In adaptive training, speech and non-speech variabilities are separately modelled through canonical HMMs and adaptation transforms. The transforms are applied to HMM parameters to obtain an adapted acoustic model for a particular speaker or acoustic environment. In state-of-the-art systems, though HMMs are usually trained using discriminative criteria such as minimum phone error, the transforms for unsupervised adaptation are still obtained through maximum-likelihood (ML). This is because discriminative transforms are highly sensitive towards errors in the supervision hypothesis. In this thesis, adaptive training based on discriminative mapping transforms (DMTs) has been proposed. A DMT is a speaker-independent discriminative transform which maps speaker-specific ML transforms to discriminative ones. As DMTs are estimated during training, they are not affected by errors in the supervision hypothesis. Therefore, the proposed scheme can be used in unsupervised adaptation tasks when the supervision hypothesis is not known. The DMT-based discriminative speaker adaptive training (DSAT) was found to significantly outperform the standard MLLR-based DSAT scheme.

The trained acoustic models are adapted for the test speaker or environment to reduce the mismatch in training and testing and improve the performance of the system. Linear transform-based speaker adaptation is a standard part of many speech recognition systems. This process requires some adaptation data from the test speaker or environment. However, for online adaptation in many real-time applications, there may be only a small amount of adaptation data available. This may not yield robust estimates of the transforms. A Bayesian framework can be used to deal with this problem, which treats transforms as random variables and uses prior distributions for them. However, it leads to intractable integrals for the marginal likelihood and some forms of approximations are required. An expectation propagation based Bayesian inference scheme has been proposed in this thesis to approximate the marginal likelihood. It was found to give very accurate estimates of the marginal likelihood, compared to other lower-bound approaches. However, the method was found to be too computationally expensive to be applied to large vocabulary continuous speech recognition. The Bayesian framework is further extended for discriminative criteria in this dissertation. This reduces the hypothesis bias problem of discriminative transforms and gives robust estimates for them even with a limited amount of data. Various forms of approximations required for discriminative adaptive inference are investigated in this work, including maximum-a-posteriori (MAP) estimation. The MAP estimation of discriminative transforms requires optimisation of a discriminative MAP objective function. The use of the reverse-Jensen inequality, weak-sense auxiliary functions and other gradient-based optimisations is investigated for the discriminative MAP estimation. In an alternative approach, DMTs are integrated into the Bayesian framework as well which improved the performance compared to other commonly used techniques for online adaptation.

The proposed methods were evaluated on an English conversational telephone speech (CTS) task.

Keywords: speech recognition, HMMs, adaptive training and adaptation, discriminative criteria, Bayesian inference, maximum-a-posteriori estimation, expectation propagation

Acknowledgement

I would like to extend my utmost gratitude to my supervisor, Dr. Mark Gales, for his valuable suggestions, thorough guidance and constant support throughout the PhD. He has always inspired me through his enthusiasm, energy and expertise. He helped me in conceiving exciting ideas, and guided me to conduct research with a great mathematical and logical rigour. I am much indebted to him, indeed, for his contributions in making this work possible.

I am also much grateful to my advisor, Prof. Phil Woodland, for his feedback and constructive suggestions throughout these years. I also owe thanks to Andrew Liu and Kai Yu for many fruitful and interesting discussions, and proof reading of this thesis. I am thankful to my colleagues Sarah Airey, Graeme Blackwood, Catherine Breslin, James Brunning, Rogier van Dalen, Frank Diehl, Tao Li, Hank Liao, Chris Longworth, Junho Park, Zoi Roupakia, Khe Chai Sim, Rohit Sinha, Blaise Thomson, Marcus Tomalin and Lan Wang for making the stay at machine intelligence laboratory (MIL) interesting. I am also indebted to our system administrators Patrick Gosling and Anna Langley, and other members of MIL for their kind cooperation and support.

This work is funded by Defense Advanced Research Projects Agency (DARPA) GALE (Global Autonomous Language Exploitation) programme (Contract No. `HR0011-06-C-0022`). I would like to thank the sponsors of the project, specially Prof. Phil Woodland, the principal investigator of the GALE AGILE (Autonomous Global Integrated Language Exploitation) project at Cambridge University, for providing the financial support.

And finally, I am also much thankful to my family for their love, encouragement and support. I am specially grateful to my father whose values to education and knowledge have been crucial in shaping my life.

Acronyms

ASR	Automatic Speech Recognition
BN	Broadcast News
CDHMM	Continuous Density Hidden Markov Model
CML	Conditional Maximum Likelihood
CMLLR	Constrained Maximum Likelihood Linear Regression
CMN	Cepstral Mean Normalisation
CTS	Conversational Telephone Speech
CVN	Cepstral Variance Normalisation
DBN	Dynamic Bayesian Network
DCT	Discrete Cosine Transform
DMT	Discriminative Mapping Transforms
DSAT	Discriminative Speaker Adaptive Training
EBW	Extended Baum-Welch
EM	Expectation Maximisation
EP	Expectation Propagation
FFT	Fast Fourier Transform
GD	Gender Dependent
GI	Gender Independent
GMM	Gaussian Mixture Model
HLDA	Heteroscedastic Linear Discriminant Analysis
HMM	Hidden Markov Model
HTK	Hidden Markov Model Toolkit
LVCSR	Large Vocabulary Continuous Speech Recognition
MAP	Maximum-A-Posteriori
MBR	Minimum Bayesian Risk
MCE	Minimum Classification Error
MFCC	Mel-Frequency Cepstral Coefficients
MLLR	Maximum Likelihood Linear Regression
ML	Maximum Likelihood

MMI	Maximum Mutual Information
MPE	Minimum Phone Error
MWE	Minimum Word Error
NIST	National Institution of Standard Technology
PLP	Perceptual Linear Prediction
RHS	Right Hand Side
SAT	Speaker Adaptive Training
SD	Speaker Dependent
SI	Speaker Independent
VBEM	Variational Bayesian Expectation Maximisation
VB	Variational Bayes
VTLN	Vocal Tract Length Normalisation
WER	Word Error Rate

Notation

General Notation

$\approx$	approximately equal to
$\propto$	proportional to
s	scalar quantity (lowercase plain letter)
$\mathbf{v}$	vector quantity (lowercase bold letter)
$\mathbf{M}$	matrix (uppercase bold letter)
$\mathbf{M}^{\mathsf{T}}$	transpose of matrix $\mathbf{M}$
$\lvert \cdot \rvert$	determinant of a square matrix
$\{\cdot\}^{-1}$	inverse of a square matrix
$\mathrm{diag}(\cdot)$	diagonal vector of a square matrix
$\mathrm{tr}(\cdot)$	trace of a square matrix
$\mathrm{vec}(\cdot)$	vectorised form of a matrix

Functions

$\mathcal{F}(\cdot)$	objective function or training criterion
$\mathcal{Q}(\cdot;\cdot)$	auxiliary function at the current estimates of parameters
$\nabla\{\cdot\}$	gradient of a function
$\nabla^2\{\cdot\}$	Hessian of a function
$\mathcal{L}(\cdot)$	lower bound of a function
$f(x)\big\rvert_{x=\hat{x}}$	value of function $f(x)$ at $x = \hat{x}$
$\arg\max\limits_{x} f(x)$	value of x that maximises $f(x)$
$\arg\min\limits_{x} f(x)$	value of x that minimises $f(x)$

Probability Distributions

$p(\cdot)$	probability density function
$p(\cdot\vert\cdot)$	conditional probability density
$P(\cdot)$	probability mass distribution
$P(\cdot\vert\cdot)$	conditional probability mass distribution
$\mathtt{KL}(\cdot\Vert\cdot)$	Kullback Leibler (KL) divergence between two distributions
$\mathtt{H}(\cdot)$	entropy of a distribution
$< f(x) >_{g(x)}$	expectation of $f(x)$ with respect to $g(x)$
$\mathcal{N}(\mathbf{x};\boldsymbol{\mu},\boldsymbol{\Sigma})$	Gaussian multivariate distributions of $\mathbf{x}$
$\delta(x)$	Dirac-delta function which is zero at $x \neq 0$

HMM Parameters

$\mathcal{M}$	HMM parameters set
$\mathcal{H}$	hypothesis, or word sequence $\{\mathcal{W}_1, \ldots, \mathcal{W}_K\}$
$\mathbf{o}_t$	observation vector at time t
D	dimension of feature vector $\mathbf{o}_t$
$\mathbf{O}$	observation sequence $\mathbf{O} = \{\mathbf{o}_1, \ldots, \mathbf{o}_T\}$
a_{ij}	discrete state transition probability from state i to state j
$b_j(\mathbf{o})$	output probability distribution at state j
ψ_t	state at time t
$\boldsymbol{\psi}$	state sequence $\boldsymbol{\psi} = \{\psi_1, \ldots, \psi_T\}$
θ_t	Gaussian component at time t
$\boldsymbol{\theta}$	Gaussian component sequence $\boldsymbol{\theta} = \{\theta_1, \ldots, \theta_T\}$
m	Gaussian component index
$\boldsymbol{\mu}_m$	mean vector of the mth Gaussian component
$\boldsymbol{\Sigma}_m$	covariance matrix of the mth Gaussian component
$\gamma_m(t)$	posterior probability of component m at time t

Adaptation and Adaptive Training

s	index for a speaker or a homogeneous data block
$\mathbf{O}^{(s)}$	observation sequence for homogeneous data block s
$\mathcal{H}^{(s)}$	hypothesis for homogeneous data block s
$\mathbf{W}^{(s)}$	transform for homogeneous data block s
$\mathbb{O}$	a set of observation sequences $\mathbb{O} = \{\mathbf{O}^{(1)}, \ldots, \mathbf{O}^{(S)}\}$
$\mathbb{H}$	a set of hypothesis sequences $\mathbb{H} = \{\mathcal{H}^{(1)}, \ldots, \mathcal{H}^{(S)}\}$
$\mathbb{W}$	a set of transforms $\mathbb{W} = \{\mathbf{W}^{(1)}, \ldots, \mathbf{W}^{(S)}\}$
$\mathbf{A}$	linear transform matrix
$\mathbf{b}$	bias vector
$\mathbf{W}$	affine transform, $\mathbf{W} = [\mathbf{A}\ \mathbf{b}]$
r_m	regression base class to which component m belongs
$\boldsymbol{\xi}_m$	extended mean vector of component m, $\boldsymbol{\xi}_m = [\boldsymbol{\mu}_m^{\mathtt{T}}\ \ 1]^{\mathtt{T}}$
$\boldsymbol{\zeta}_t$	extended observation vector at time t, $\boldsymbol{\zeta}_t = [\mathbf{o}_t^{\mathtt{T}}\ \ 1]^{\mathtt{T}}$
Φ	hyperparameters for the HMM parameter prior distribution
ϕ	hyperparameters for the transform prior distribution

Table of Contents

List of Tables

List of Figures

Chapter 1
Introduction

The aim of automatic speech recognition (ASR) systems is to transcribe speech into words. As speech is a natural mode of communication for human-beings, automatic speech recognition is finding numerous uses in building human-machine interfaces. It has a wide range of applications in several domains including command-and-control, dictation, medical transcription, information retrieval, dialogue systems, audio indexing and speech-to-speech translation [73].

State-of-the-art speech recognition systems are based on statistical approaches to learning acoustic and linguistic characteristics from the training data. Two forms of statistical model are involved in the recognition process: the acoustic model and the language model. Hidden Markov models (HMMs) are widely used as acoustic models in these systems which are trained from a corpus of speech data [145]. The acoustic models along with the language models are then used to recognise the word sequence in the test speech signal. However, when the test speech is from different speakers or acoustic environments than that of the training corpus, the performance of speech recognition systems degrades severely [72]. This is due to the mismatch between the training and the testing acoustic conditions. One approach to deal with this mismatch is to adapt the acoustic models to the target speaker or acoustic environment. This is usually referred as *speaker adaptation* [40, 73]. Linear transforms are widely used to adapt HMM parameters [152, 189]. They are estimated from the sample speech from the target speaker and the corresponding supervision transcripts.

Moreover, rather than using speech data recorded in a well-controlled environment for training, there has been an increasing interest to build speech recognition systems with *found* data, such as broadcast news and telephone speech recordings. Such *found* data often has varying acoustic conditions and comes from a large number of different speakers. One of the techniques to deal with the training of speech recognition systems with such non-homogeneous data is *adaptive training* [5, 44], in which speech and non-speech variabilities are separately modelled. This allows the underlying speech models to be extracted from such data.

The training of acoustic models in state-of-the-art speech recognition systems is commonly based on discriminative criteria such as minimum phone error (MPE) [140]. The use of discriminative criteria in training has been found to improve the performance of ASR systems significantly compared to using the conventional maximum likelihood criterion [140]. Hence, the use of discriminative criteria has also been investigated for transform estimation for model adaptation [63, 118, 171, 175, 182]. Though discriminative transforms can give performance gains for supervised adaptation, they are seldom used for unsupervised adaptation for which the correct transcript is not known. This is because the generated hypothesis used as supervision may contain several errors, and the discriminative transforms are highly sensitive to errors in the supervision hypothesis as they are biased towards it. Though the confidence score and lattice based approaches [182, 202] have been investigated to deal with these problems, only limited, if any, gains are obtained. Recently, discriminative mapping transforms (DMTs) [202] have been also successfully applied in these situations giving improved performance.

In many cases in real-life applications of speech recognition, the models are required to be adapted as soon as data becomes available. The adaptation and decoding process cannot be delayed, as the application requires responding in real or near real time. In this case, there may be only a small amount of data for transform estimation. This may not yield a robust estimate of transforms for adapting the models. A maximum-a-posteriori (MAP) estimation method has been used in [21, 22, 57, 77] to deal with this problem, and to robustly estimate maximum likelihood transforms even with a small amount of adaptation data. Similarly, an N-best list based instantaneous unsupervised adaptation scheme has been used in [114, 129] that uses MAP estimates of mean bias. The N-best list based scheme can also deal with the errors in the supervision hypothesis. An N-best list based Bayesian framework for maximum likelihood affine transforms has been investigated in [201] for the unsupervised instantaneous adaptation. A number of other techniques including cluster-based methods have been also developed to deal with this problem as reviewed in [152, 189]

This thesis focuses on these issues of adaptation and adaptive training of acoustic models in large vocabulary continuous speech recognition (LVCSR) systems. The goal is to build a robust acoustic model from the found data and to improve the performance of the system by adapting it to the test speaker and/or acoustic environment. The discriminative and Bayesian approaches will be investigated to achieve these goals.

1.1 Organisation of the Thesis

This thesis is organised as follows. Chapter 2 describes a standard HMM-based automatic speech recognition system along with the training and decoding algorithms as well as some of the widely employed techniques in the state-of-the-art systems. Commonly used techniques for adaptation and adaptive training of acoustic models are reviewed in chapter 3. In chapter 4, discriminative adaptation and adaptive training techniques are investigated and a new approach for adaptive training based on discriminative mapping transforms is proposed. Thereafter, chapter 5 first reviews earlier work on Bayesian adaptive training and inference for maximum-likelihood systems, and then proposes an expectation-propagation based inference scheme. In chapter 6, a Bayesian framework for discriminative adaptive training and inference is investigated. The inference schemes in discriminative adaptive systems are described, along with several approximations for the Bayesian inference in discriminative systems. Bayesian discriminative adaptation and inference based on MAP as well as DMT are also proposed. Subsequently, chapters 7, 8 and 9 evaluate the proposed methods and present results from speech recognition experiments on a conversational telephone speech (CTS) task. Finally, the thesis is concluded in chapter 10, with a summary and a discussion of future work.

Chapter 2

HMM-based Speech Recognition Systems

This chapter describes a statistical speech recognition system based on hidden Markov models (HMMs). The architecture of a large vocabulary speech recognition system is first described showing its basic building blocks. This is followed by a detailed description of each unit of the speech recognition system including frontend processing, the acoustic and language models, and the decoder. Several techniques that are used for training of acoustic and language models in the state-of-the-art speech recognition systems are also described.

2.1 Automatic Speech Recognition Systems

The task of a speech recognition system is to recognise the word sequence present in a speech waveform. A block diagram of a statistical speech recognition system is shown in figure 2.1. The speech signal captured from a microphone is first converted into a stream of acoustic features by a frontend processing module. This is then decoded by using the knowledge obtained from acoustic and language models, and a dictionary or lexicon to produce hypotheses for the recognised words. The output hypotheses are also commonly used to adapt the models to the test environment and domain, and redecode the given speech.

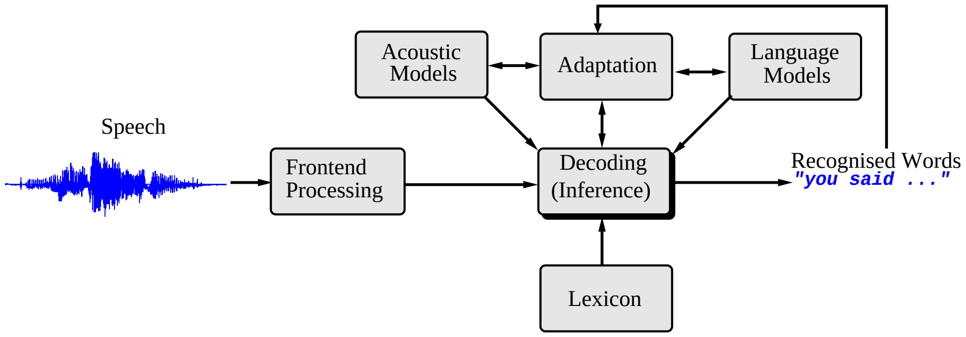


Figure 2.1: A block diagram of an automatic speech recognition (ASR) system

A statistical speech recognition system finds the most probable word sequence or hypothesis $\hat{\mathcal{H}}$ for a given speech observation sequence $\mathbf{O}$. This can be expressed as

$$\hat{\mathcal{H}} = \arg\max_{\mathcal{H}} \left\{ P(\mathcal{H}|\mathbf{O}) \right\} \tag{2.1}$$

The posterior probability of the hypothesis in the above equation can be expressed in terms of the class conditional probability and the prior, by applying Bayes' rule

$$\begin{aligned} \hat{\mathcal{H}} &= \arg\max_{\mathcal{H}} \left\{ \frac{p(\mathbf{O}|\mathcal{H})P(\mathcal{H})}{p(\mathbf{O})} \right\} \\ &= \arg\max_{\mathcal{H}} \left\{ p(\mathbf{O}|\mathcal{H})P(\mathcal{H}) \right\} \end{aligned} \tag{2.2}$$

The normalisation constant in the denominator, $p(\mathbf{O})$, has been dropped as it does not depend upon a particular hypothesis $\mathcal{H}$, and thus does not alter the search for the best hypothesis. In the above equation, $p(\mathbf{O}|\mathcal{H})$ is the likelihood of the observation sequence $\mathbf{O}$ for the given hypothesis $\mathcal{H}$, and $P(\mathcal{H})$ is the prior probability of the hypothesis. The likelihood $p(\mathbf{O}|\mathcal{H})$ is computed by using the acoustic model and $P(\mathcal{H})$ is obtained from the language model. The decoding process in a statistical speech recognition system thus involves finding a hypothesis using the acoustic and language model scores that maximises the posterior probability for the given observation sequence.

The focus of this thesis is on adaptation and adaptive training of acoustic models as well as the decoding or inference process in the speech recognition system. The next sections describe each block of the speech recognition system shown in figure 2.1 in detail. The adaptation and adaptive training of acoustic models is separately described in the next chapter.

2.2 Frontend Processing of Speech

The first process involved in a speech recognition system is to convert the speech signal captured by a microphone to an appropriate form that is compact and effective for the recognition process. The analog speech signal is first digitised and relevant segments of speech excluding unwanted music or silence are isolated; and then a set of salient features are extracted. The frontend processing of speech may involve preprocessing, feature extraction and postprocessing stages, as described in the following sections.

2.2.1 Speech Preprocessing

In an automatic speech recognition system, the speech signal from the microphone is first digitised by an analog-to-digital converter (ADC). A sampling frequency of 8kHz or 16kHz is commonly used for speech recognition. This yields a stream of samples of the speech signal. In many cases, resampling and format or encoding conversion may be also required for further processing, specially for prerecorded speech. All of the samples may not be relevant for the speech recognition purpose, as some of the segments may be just long silence, noise, music interludes or commercials [163]. Therefore, only the relevant speech segments are isolated from the stream, through a process called *segmentation*, which are then passed for feature extraction.

2.2.2 Feature Extraction

Mel-frequency cepstral coefficients (MFCC) [26] and perceptual linear prediction (PLP) coefficients [66] are commonly used speech features in state-of-the-art speech recognition systems. In both types of feature extraction, the stream of samples from the speech signal is divided

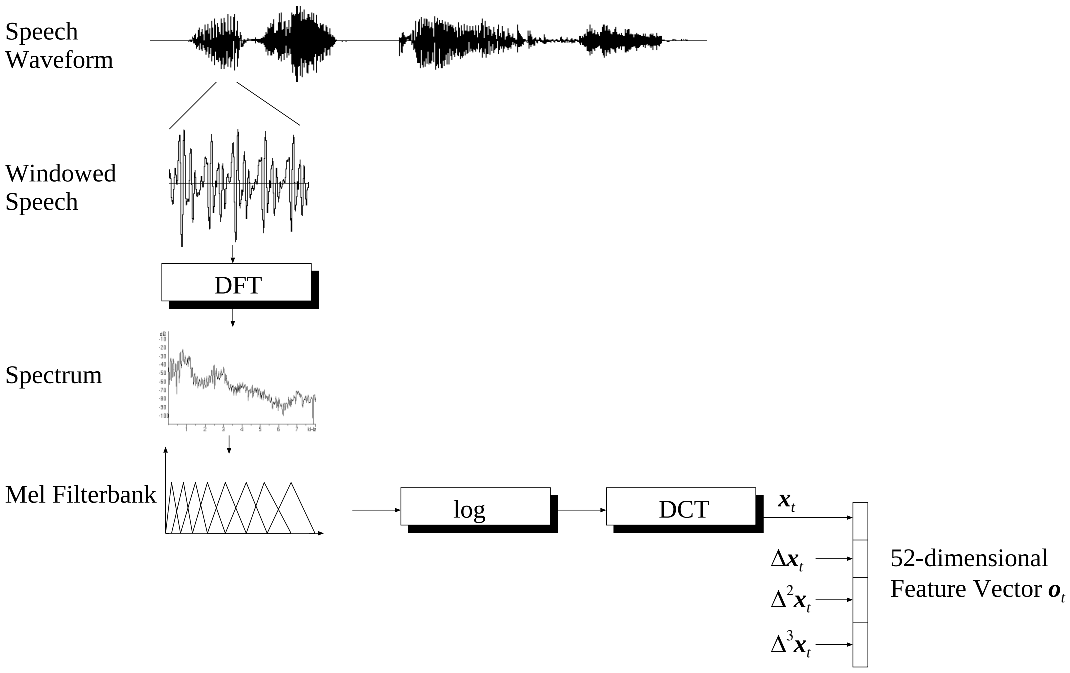


Figure 2.2: The MFCC feature extraction of speech signal

into a series of quasi-stationary segments usually referred as *frames*. These frames are obtained by applying an overlapping windowing function to the stream of speech samples. A typical window of 25ms to 30ms width, usually shifting in 10ms step, is used for speech recognition [73]. A Hamming or Hanning window function is commonly used to attenuate the discontinuities at the edge of the window to reduce the Gibbs effect [73]. A Fourier transform is then applied to the samples of each frame window, and the frequency domain power spectrum for the frame is obtained.

For computing MFCCs [26], the linear frequency scale is warped into a Mel-frequency scale, using

$$f_{\texttt{mel}} = 1125 \log\left(1 + \frac{f_{\texttt{Hz}}}{625}\right) \tag{2.3}$$

where $f_{\texttt{mel}}$ is the warped Mel-scale frequency of the linear scale frequency $f_{\texttt{Hz}}$. The power spectrum is then down-sampled using a bank of triangular filters, with typically 24 to 40 channels [73]. The calculation of Mel filterbank coefficients involves multiplying each FFT magnitude or power with a corresponding gain of the triangular filter and accumulating them. Therefore, a Mel filterbank coefficient represents the weighted sum of spectral magnitude or power in that filterbank channel. These Mel filterbank coefficients are transformed to the natural log domain and a discrete Cosine transform (DCT) is finally applied giving Mel frequency cepstral features

$$x_{td} = \sqrt{\frac{2}{B}} \sum_{b=1}^{B} \log(m_{tb}) \cos\left(\frac{\pi d}{B}(b - 0.5)\right) \tag{2.4}$$

where x_{td} is the dth cepstral coefficient for tth frame. m_{tb} represents the Mel coefficient for band b for the tth frame, and B is the total number of filterbank channels. The DCT allows cepstral coefficients to be decorrelated and diagonal covariances to be used in HMMs. The number of cepstral coefficients is usually limited to 13 [73].

In PLP [66] feature extraction, the linear frequency of the power spectrum is warped into a Bark frequency scale as

$$f_{\texttt{bark}} = 6 \log\left(\left[\left(\frac{f_{\texttt{Hz}}}{600}\right)^2 + 1\right]^{\frac{1}{2}} + \frac{f_{\texttt{Hz}}}{600}\right) \tag{2.5}$$

where $f_{\texttt{bark}}$ is a Bark scale frequency. Critical band filters are applied to the power spectra to downsample them, which are then scaled by using equal-loudness and intensity-loudness power law. Finally, linear prediction (LP) analysis is applied and resulting LP coefficients are converted to cepstral coefficients. In [195], a modified form of PLP features based on MFCC filterbank analysis is used, which was found to be more effective than the standard

PLP analysis. The Mel filterbank coefficients are scaled by an equal-loudness curve and then compressed by taking a cubic root [196]. The resulting spectrum is used to compute LP coefficients, which are then converted to cepstral coefficients. This type of PLP feature is also referred as MF-PLP [195], and is used in this work.

Generally, energy and/or zeroth order cepstral coefficients are also used to augment the feature vector. The coefficients obtained for each frame as described above are also called *static coefficients*, and they do not account the temporal dynamics of the speech signal between frames. One of the popular techniques for accounting temporal dynamics of speech is to include delta coefficients [38]. The delta coefficients are computed as

$$\Delta\mathbf{x}_t = \frac{\sum_{k=1}^{K} k(\mathbf{x}_{t+k} - \mathbf{x}_{t-k})}{2\sum_{k=1}^{K} k^2} \tag{2.6}$$

where $2K + 1$ is the size of the regression window operating on the speech feature vector $\mathbf{x}_t$. In the above equation, choosing $K = 1$ gives delta coefficients that are simple differences of cepstral coefficients between two consecutive frames. However, higher values of K can give more robust estimates of dynamic coefficients. The delta coefficients can be regarded as the approximation to time-derivatives of the static parameters [48]. The second and the third order delta coefficients can be also computed in a similar way, and appended to the speech features. These coefficients are also called *dynamic coefficients*.

Therefore, a typical 13-dimensional MFCC or LPC cepstra including the energy or the zeroth order coefficient may be augmented by first, second and third order derivatives leading to a 52-dimensional observation vector

$$\mathbf{o}_t = \begin{pmatrix} \mathbf{x}_t \\ \Delta\mathbf{x}_t \\ \Delta^2\mathbf{x}_t \\ \Delta^3\mathbf{x}_t \end{pmatrix} \tag{2.7}$$

2.2.3 Feature Postprocessing

The ideal speech features should only capture the phonetic variabilities in speech, but not the non-speech variabilities coming from speaker or acoustic environment variations. This allows the underlying words in the speech to be recognised independent of speaker or environmental variations. A number of techniques are used in the state-of-the-art speech recognition systems to make speech features robust to speaker or environmental variations. The commonly used feature normalisation techniques are described in the following sections.

2.2.3.1 Cepstral Mean and Variance Normalisation

In cepstral mean normalisation (CMN) or cepstral mean subtraction [6], the observed cepstral features are transformed to have a zero mean by subtracting the mean of the observation features. This can be expressed as

$$\hat{\mathbf{o}}_t = \mathbf{o}_t - \frac{1}{T}\sum_{\tau=1}^{T}\mathbf{o}_\tau \tag{2.8}$$

where $\mathbf{o}_\tau$ is the observation vector for frame τ, $\hat{\mathbf{o}}_t$ is the normalised observation vector for the tth frame after CMN and the mean is computed over T frames. CMN removes the bias in the cepstrum that arises due to multiplicative noise coming from channel distortion or microphone characteristics, and thus makes the features robust to slowly varying multiplicative noise. This normalisation involves computing the average value of observation vector. In offline recognition of speech, it can be computed easily over longer segments of speech. However, for online recognition, only an utterance or a window of few frames is used. Similarly, cepstral variance normalisation (CVN) normalises the variance of each dimension of the observations to have unity variance. The normalised observations after CVN is given as

$$\hat{o}_{td} = \frac{o_{td}}{\sqrt{\sigma_d^2}} \tag{2.9}$$

where o_{td} and $\hat{o}_{td}$ are the dth dimension of the observation vector for the tth frame before and after CVN, respectively, and σ_d^2 is the variance of the dth dimension of observations given as

$$\sigma_d^2 = \frac{1}{T}\sum_{\tau=1}^{T} o_{\tau d}^2 \tag{2.10}$$

Both CMN and CVN are inexpensive to apply yet very effective in reducing the mismatch between training and testing conditions by removing the environmental dependent variations. They are widely used in state-of-the-art speech recognition systems.

2.2.3.2 Gaussianisation

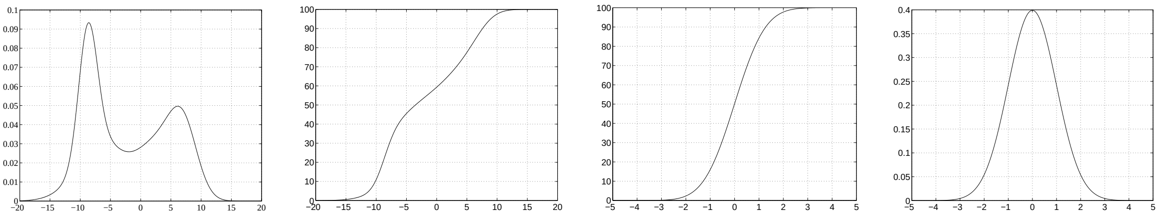
CMN and CVN normalise the mean and variance of observations which are first and second order moments. The higher order moments of the observation can be also normalised to give it the desired distribution. The distribution of observations may not always be Gaussian. Gaussianisation [153] normalises higher order moments by transforming $\mathbf{o}_t$ using a nonlinear function f_{g}

$$\hat{\mathbf{o}}_t = f_{\text{g}}(\mathbf{o}_t) \tag{2.11}$$

such that the transformed observation has normal distribution with zero mean and identity variance

$$\hat{\mathbf{o}}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \tag{2.12}$$

The process of Gaussianisation is illustrated in figure 2.3. The source PDF is normalised to have a Gaussian PDF by normalising the cumulative density function (CDF) of the observations to a CDF of a standard Gaussian. Gaussianisation can be obtained through an iterative scheme based on histogram matching [153] or a GMM based approach [53, 105]. The later approach provides a more compact and smooth representation of the distribution of the original observations by using GMMs. In this method, each dimension of the original observation is represented by GMM. The source CDF is then mapped using the inverse Gaussian CDF such that the source distribution is transformed to a Gaussian PDF. This can be written as



Source PDF Source CDF Target CDF Target PDF

Figure 2.3: The Gaussianisation of features (from [39])

$$\hat{o}_{td} = \Phi^{-1}\left(\int_{-\infty}^{o_{td}} \sum_{m=1}^{M_d} c_{md}\, \mathcal{N}\left(o;\ \mu_{md}, \sigma_{md}^2\right)\, do\right) \tag{2.13}$$

where $\Phi^{-1}(\cdot)$ is the Gaussian inverse CDF, and μ_{md}, σ_{md}^2 and c_{md} are the means, variances and weights of the GMM component m for dimension d. Gaussianisation can be applied at the utterance, speaker or global level. At each level, a total of D single dimension M_d-component GMMs need to be trained using the ML criterion. When there is only one component in GMMs, it is equivalent to applying CMN and CVN. As with CMN and CVN, it should be applied both to training and test features.

2.2.3.3 Vocal Tract Length Normalisation

Differences between speakers is one of the main source of undesired variabilities in speech recognition. One reason for this is the anatomical difference between speakers, such as vocal tract length and shape. The vocal tract length is related to the formant centre frequencies. Females and children have shorter vocal tract and thus produce higher formant frequencies, whereas male speakers have longer vocal tracts and produce lower formant frequencies. This makes the same word differ in spectral content when spoken by male, female or just different speakers, and adds unwanted variabilities to speech. Vocal tract length normalisation (VTLN) [101] is a commonly used technique to remove this variability coming from varying vocal tract length. In VTLN, the specific speaker's formant frequency range is compressed or expanded to the normalised frequency, using a warping factor α. This is usually achieved by warping the frequency axis in the filterbank analysis before the features are extracted. VTLN is thus a non-linear feature transform. The frequency warping in VTLN is illustrated in figure 2.4, using a piece-wise linear warping function. It warps the original frequency f into the scaled frequency $\tilde{f}$ using warping factors from $\alpha_{\mathtt{min}}$ to $\alpha_{\mathtt{max}}$. As warping may result in some filters being outside the analysis frequency range, different warping factors are used at the upper and lower boundaries such that the end frequencies are mapped to themselves. The regions for piecewise linear mapping are defined with lower (f_{L}) and upper ($f_{\mathtt{U}}$) cutoff frequencies.

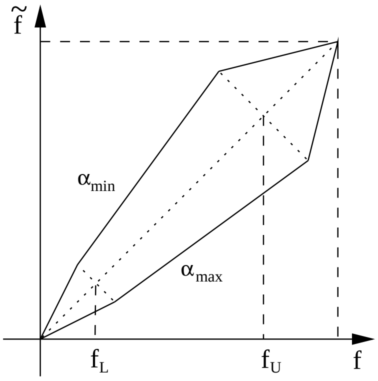


Figure 2.4: Frequency warping by VTLN

The optimal warping factor is selected by maximising the likelihood of the warped observations. This is done through a grid search scheme [101, 144] by comparing likelihoods at different warping factors. The VTLN can be implemented by direct frequency warping, bilinear transforms [117], or it can be approximated through linear transformation in cepstral space [139, 173]. VTLN is an effective feature normalisation technique and is commonly used with other normalisation techniques in state-of-the-art speech recognition systems [32].

2.3 Acoustic Models

Hidden Markov models (HMMs) are the most popular and successful acoustic models used in large vocabulary state-of-the-art speech recognition systems [40, 145]. An HMM is a finitestate machine, comprising a number of discrete states, with each state associated with an output probability distribution, as shown in figure 2.5. As each time instant, when a state is entered according to the defined transition probabilities between the states, the HMM generates observations according to the state output distribution. The underlying state sequence is hidden, and only the observations can be seen. An HMM is a generative model, and the speech observations are assumed to be generated from it by traversing through its states.

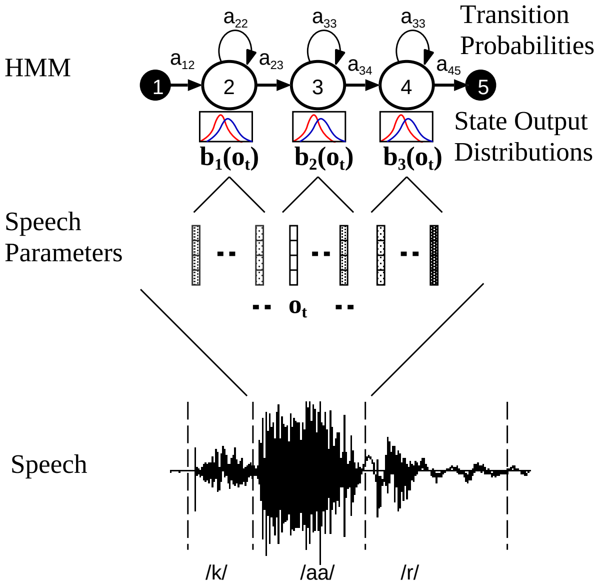


Figure 2.5: A hidden Markov model (HMM) as an acoustic model

An HMM is used to model one acoustic unit such as words or phones. However, subword units such as phones are commonly used as the number of words increases in the speech recognition system. The HMM shown in figure 2.5 is a left-to-right HMM widely used in speech recognition. The filled node at the beginning and the end represent entering and exit non-emitting states, whereas other non-shaded nodes are the states with associated output probability distributions. The connecting arrows represent valid transitions between states. The output probability distribution for state j is specified as $b_j(\mathbf{o}_t)$, and the transition probability from state i to state j is a_{ij}. The state at time t is represented by ψ_t. The complete parameter set of an N-state HMM is characterised by model parameters $\mathcal{M} = \{\boldsymbol{\pi}, \mathcal{A}, \mathcal{B}\}$, where $\boldsymbol{\pi} = \{\pi_i = P(\psi_1 = i) : 1 \leq i \leq N\}$ is the initial state distribution, $\mathcal{A} = \{a_{ij} : 1 \leq i \leq N, 1 \leq j \leq N\}$ is the transition probability matrix, and $\mathcal{B} = \{b_j(\mathbf{o}_t) : 1 \leq j \leq N\}$ is the observation probability of the states. In the above HMM topology with start and end non-emitting states, the initial state distribution is always one for the start state and output probabilities distributions are not required for the start and end states. The start and end non-emitting states facilitates joining HMMs together to form composite HMMs representing longer speech segments. There are two key underlying assumptions in HMMs when modelling speech [145]:

- **First-order Markov process assumption:**
  The probability of making transitions to state ψ_t depends only on the last state ψ_{t-1}, and is independent of the states at time $1, \dots, t-2$. Therefore, the transition probability from state i to state j is given by

$$a_{ij} = P(\psi_t = j | \psi_{t-1} = i) \tag{2.14}$$

- **Output conditional independence assumption:**
  The probability of observation $\mathbf{o}_t$ at time t is conditionally independent of all other states and observations given the current state, ψ_t. In other words, the output probability distribution can be expressed as

$$b_j(\mathbf{o}_t) = p(\mathbf{o}_t | \psi_t = j) \tag{2.15}$$

The output probability distributions $b_j(\mathbf{o}_t)$ may be discrete or continuous thus leading to a discrete HMM (DHMM) or a continuous density HMM (CDHMM). However, most state-of-the-art speech recognition systems are based on CDHMMs, and use a multivariate *Gaussian mixture model* (GMM) as the output probability distribution. This can be written as

$$\begin{aligned} b_j(\mathbf{o}_t) &= \sum_{m=1}^{M_j} c_{jm} \mathcal{N}\left(\mathbf{o}_t; \boldsymbol{\mu}_{jm}, \boldsymbol{\Sigma}_{jm}\right) && (2.16) \\ \sum_{m=1}^{M_j} c_{jm} &= 1 && (2.17) \end{aligned}$$

where M_j is the number of mixture components and c_{jm} is the weight of mixture component m for state j. The multivariate Gaussian distribution for each component is given by

$$\mathcal{N}\left(\mathbf{o}_t; \boldsymbol{\mu}_{jm}, \boldsymbol{\Sigma}_{jm}\right) = \frac{1}{\sqrt{(2\pi)^D \left|\boldsymbol{\Sigma}_{jm}\right|}} \exp\left\{-\frac{1}{2}\left(\mathbf{o}_t - \boldsymbol{\mu}_{jm}\right)^{\mathsf{T}} \boldsymbol{\Sigma}_{jm}^{-1} \left(\mathbf{o}_t - \boldsymbol{\mu}_{jm}\right)\right\} \tag{2.18}$$

$\boldsymbol{\mu}_{jm}$ and $\boldsymbol{\Sigma}_{jm}$ are the mean vector and the covariance matrix for the mth component of the jth state, and D is the dimension of feature vectors. The covariance matrix is normally assumed to be diagonal to reduce the number of parameters and increase decoding speed.

The above assumptions in HMMs imply that speech signal can be split into short stationary segments corresponding to the HMM states, and the transitions between states are instantaneous. However, the speech signal is continuous in nature rather than piecewise stationary and it also shows long-term dependencies. Therefore, the assumptions made are poor for the speech signal, nevertheless, HMMs continue to be the most successful technique for acoustic modelling in speech recognition. In order to use HMMs for speech recognition, there are three fundamental issues to be addressed [145]:

- computing the likelihood of observations given the model
- estimating or training the HMM parameters
- decoding the most likely state sequence for a given observation sequence

The first two are described in the next section. Decoding with HMMs will be presented in section 2.6.

2.3.1 Likelihood Calculation

The calculation of the likelihood of an observation sequence for a given hypothesis is an important aspect in using HMMs as acoustic models. The likelihood of the observation sequence $\mathbf{O} = \{\mathbf{o}_1, \dots, \mathbf{o}_T\}$ can be computed given the state sequence $\boldsymbol{\psi} = \{\psi_1, \dots, \psi_T\}$, as each of the observations is assumed to be generated independently given the state at that time instance. However, the state sequence is hidden in the HMM, and therefore the likelihood of an observation sequence $\mathbf{O}$ for a given hypothesis $\mathcal{H}$ is computed by finding the expected likelihood over all possible state sequences

$$\begin{aligned} p(\mathbf{O}|\mathcal{H}, \mathcal{M}) &= \sum_{\boldsymbol{\psi}} p(\mathbf{O}, \boldsymbol{\psi}|\mathcal{H}, \mathcal{M}) \\ &= \sum_{\boldsymbol{\psi}} P(\boldsymbol{\psi}|\mathcal{H}, \mathcal{M}) p(\mathbf{O}|\boldsymbol{\psi}, \mathcal{M}) \end{aligned} \tag{2.19}$$

Using the first-order Markov and conditional independence assumptions associated with the HMM, the likelihood can be expressed as[1]

$$\begin{aligned} p(\mathbf{O}|\mathcal{H}, \mathcal{M}) &= \sum_{\boldsymbol{\psi}} \prod_{t} P(\psi_t|\psi_{t-1}) b_{\psi_t}(\mathbf{o}_t) \\ &= \sum_{\boldsymbol{\psi}} \pi_{\psi_0} \left(\prod_{t=1}^{T} a_{\psi_{t-1}\psi_t} b_{\psi_t}(\mathbf{o}_t) \right) a_{\psi_T \psi_{T+1}} \end{aligned} \tag{2.20}$$

However, it is not feasible to sum over all possible state sequences as the number of paths grows exponentially as $\mathcal{O}(N^T)$. Therefore, a recursive approach called the *forward-backward algorithm* is used instead, by introducing forward and backward probabilities.

[1] In the form presented here, the observation time index t has been extended from 0 to $T + 1$ with hypothetical observations at both ends to consider the non-emitting end states.

Forward-Backward Algorithm

The *forward probability* $\alpha_j(t)$ is defined by the probability of generating the partial observation sequence up to t and being in state j at time t

$$\alpha_j(t) = p(\mathbf{o}_1, \ldots, \mathbf{o}_t, \psi_t = j|\mathcal{H}, \mathcal{M}) \tag{2.21}$$

The forward probability can be defined recursively as

$$\alpha_j(t) = \left(\sum_{i=2}^{N-1} \alpha_i(t-1)a_{ij} \right) b_j(\mathbf{o}_t), \quad 1 < j < N,\, 1 < t \le T \tag{2.22}$$

with the initial and final conditions imposed as

$$\alpha_j(t) = \left\{ \begin{array}{lll} 1 & j = 1 & t = 0 \\ a_{1j} b_j(\mathbf{o}_t) & 1 < j < N & t = 1 \\ \sum_{i=2}^{N-1} \alpha_i(T) a_{iN} & j = N & t = T+1 \end{array} \right. \tag{2.23}$$

The *backward probability* $\beta_j(t)$ is the likelihood of observing the partial observation sequence from the time instance $t+1$ to the end

$$\beta_j(t) = p(\mathbf{o}_{t+1}, \ldots, \mathbf{o}_T|\psi_t = j, \mathcal{H}, \mathcal{M}) \tag{2.24}$$

The backward probability can be recursively estimated as

$$\beta_j(t) = \sum_{i=2}^{N-1} a_{ji} b_i(\mathbf{o}_{t+1}) \beta_i(t+1), \quad 1 < j < N,\, 0 \le t < T \tag{2.25}$$

with the constraints

$$\beta_j(t) = \left\{ \begin{array}{lll} 1 & j = N & t = T+1 \\ a_{jN} & 1 < j < N & t = T \end{array} \right. \tag{2.26}$$

The likelihood of the observation sequence can be obtained either from the forward or the backward algorithm as

$$p(\mathbf{O}|\mathcal{H}, \mathcal{M}) = \alpha_N(T+1) = \beta_1(0) \tag{2.27}$$

These forward and backward probabilities can be also used to compute the state occupation $\gamma_j(t)$, i.e., the probability of being in state j at time t, as

$$\begin{aligned} \gamma_j(t) &= P(\psi_t = j|\mathbf{O}, \mathcal{H}, \mathcal{M}) \\ &= \frac{P(\mathbf{O}, \psi_t = j|\mathcal{H}, \mathcal{M})}{p(\mathbf{O}|\mathcal{H}, \mathcal{M})} \\ &= \frac{\alpha_j(t)\beta_j(t)}{p(\mathbf{O}|\mathcal{H}, \mathcal{M})} \end{aligned} \tag{2.28}$$

as from the definitions of the forward and backward probabilities

$$\alpha_j(t)\beta_j(t) = P(\mathbf{O}, \psi_t = j|\mathcal{H}, \mathcal{M}) \tag{2.29}$$

They can also give transition posteriors $\chi_{ij}(t)$, i.e. the probability of transitioning from state i to state j at time t, as

$$\begin{aligned} \chi_{ij} &= P(\psi_{t-1} = i, \psi_t = j|\mathbf{O}, \mathcal{H}, \mathcal{M}) \\ &= \frac{\alpha_i(t-1)a_{ij}b_j(\mathbf{o}_t)\beta_j(t)}{p(\mathbf{O}|\mathcal{H}, \mathcal{M})} \end{aligned} \tag{2.30}$$

These state posteriors are used in several algorithms including HMM training as described below.

2.3.2 Maximum Likelihood Training of HMMs

In maximum-likelihood training of HMMs, the model parameters $\mathcal{M}$ are estimated by maximising the likelihood of the training data $\left\{\mathbf{O}^{(1)}, \ldots, \mathbf{O}^{(U)}\right\}$ as

$$\hat{\mathcal{M}} = \arg\max_{\mathcal{M}} \left\{ \sum_{u=1}^{U} \log p(\mathbf{O}^{(u)}|\mathcal{H}^{(u)}, \mathcal{M}) \right\} \tag{2.31}$$

where $\mathbf{O}^{(u)}$ is the training utterance with transcript $\mathcal{H}^{(u)}$ for utterance u. In the following derivations, however, the summation over utterances and superscript (u) are dropped for the sake of simplicity. As a direct optimisation of the ML objective function is difficult, an implementation of the *expectation maximisation (EM) algorithm* [28] called the *Baum-Welch algorithm* [13] is iteratively used to estimate the HMM parameters. In this approach, an auxiliary function is defined at the current model parameters $\mathcal{M}_k$ at the kth iteration, which is a lower-bound to the log-likelihood. The new estimates of the model parameters $\mathcal{M}_{k+1}$ at the $(k+1)$th iteration are then obtained by maximising this lower-bound, which subsequently maximises the log-likelihood.

A lower-bound to the log-likelihood can be derived by introducing a variational distribution $q(\boldsymbol{\psi})$ of the hidden state sequence and applying Jensen's inequality [86], as

$$\log p(\mathbf{O}|\mathcal{H}, \mathcal{M}_{k+1}) = \log \sum_{\boldsymbol{\psi}} q(\boldsymbol{\psi}) \frac{p(\mathbf{O}, \boldsymbol{\psi}|\mathcal{H}, \mathcal{M}_{k+1})}{q(\boldsymbol{\psi})} \tag{2.32}$$

$$\geq \Big\langle \log p(\mathbf{O}, \boldsymbol{\psi}|\mathcal{H}, \mathcal{M}_{k+1}) \Big\rangle_{q(\boldsymbol{\psi})} + \mathtt{H}\big(q(\boldsymbol{\psi})\big) \tag{2.33}$$

where $\mathtt{H}\big(q(\boldsymbol{\psi})\big) = -\sum_{\boldsymbol{\psi}} q(\boldsymbol{\psi}) \log\big(q(\boldsymbol{\psi})\big)$ is the entropy [1] of $q(\boldsymbol{\psi})$ and $< f(x) >_{g(x)}$ represents the expectation of $f(x)$ with respect to $g(x)$. The lower-bound is maximised turning the above inequality into an equality when $q(\boldsymbol{\psi}) = P(\boldsymbol{\psi}|\mathbf{O}, \mathcal{H}, \mathcal{M}_{k+1})$. However, the posterior probability of the state sequence $P(\boldsymbol{\psi}|\mathbf{O}, \mathcal{H}, \mathcal{M}_{k+1})$ cannot be directly known, as $\mathcal{M}_{k+1}$ is to be estimated. Therefore, the current model parameters $\mathcal{M}_k$ are used to compute the lower-bound by setting $q(\boldsymbol{\psi}) = P(\boldsymbol{\psi}|\mathbf{O}, \mathcal{H}, \mathcal{M}_k)$ as

$$\log p(\mathbf{O}|\mathcal{H}, \mathcal{M}_{k+1}) \geq \Big\langle \log p(\mathbf{O}, \boldsymbol{\psi}|\mathcal{H}, \mathcal{M}_{k+1}) \Big\rangle_{P(\boldsymbol{\psi}|\mathbf{O},\mathcal{H},\mathcal{M}_k)} + \mathtt{H}\big(P(\boldsymbol{\psi}|\mathbf{O}, \mathcal{H}, \mathcal{M}_k)\big) \tag{2.35}$$

The first term in the right hand side of above equation is used as the auxiliary function to estimate the model parameters $\mathcal{M}_{k+1}$, as the second term is not a function of $\mathcal{M}_{k+1}$. Therefore, the ML auxiliary function is given by

$$\mathcal{Q}(\mathcal{M}_{k+1}; \mathcal{M}_k) = \Big\langle \log p(\mathbf{O}, \boldsymbol{\psi}|\mathcal{H}, \mathcal{M}_{k+1}) \Big\rangle_{P(\boldsymbol{\psi}|\mathbf{O},\mathcal{H},\mathcal{M}_k)} \tag{2.36}$$

The maximisation of the above auxiliary function is guaranteed not to decrease the likelihood, as it can be shown that

$$\log p(\mathbf{O}|\mathcal{M}_{k+1}, \mathcal{H}) - \log p(\mathbf{O}|\mathcal{M}_k, \mathcal{H}) \geq \mathcal{Q}(\mathcal{M}_{k+1}; \mathcal{M}_k) - \mathcal{Q}(\mathcal{M}_k; \mathcal{M}_k) \tag{2.37}$$

The EM algorithm converges to a local maximum of the likelihood function. The EM algorithm is run iteratively in two steps, as shown in algorithm 1. In the E-step, the hidden state posteriors are estimated and an auxiliary function is formed. In the M-step, the auxiliary function is maximised and new estimates of parameters are obtained.

```
Initialise $\mathcal{M}_k, k = 0$
Do
   E-step: compute $\mathcal{Q}(\mathcal{M}_{k+1}; \mathcal{M}_k)$
   M-step: estimate $\mathcal{M}_{k+1} = \arg\max_{\mathcal{M}} \mathcal{Q}(\mathcal{M}; \mathcal{M}_k)$
   k=k+1
While $\mathcal{Q}(\mathcal{M}_{k+1}; \mathcal{M}_k) - \mathcal{Q}(\mathcal{M}_k; \mathcal{M}_k) >$ threshold
```

Algorithm 1: *The EM algorithm*

As observations are assumed conditionally independent, the ML auxiliary function in equation (2.36) can be expressed as

$$\mathcal{Q}(\mathcal{M}_{k+1}; \mathcal{M}_k) = \sum_{tj} \gamma_j(t) \log b_j(\mathbf{o}_t) + \sum_{tij} \chi_{ij}(t) \log a_{ij} \tag{2.38}$$

[1] The entropy of any discrete distribution $P(z)$ is defined as

$$\mathtt{H}\big(P(z)\big) = -\sum_z P(z) \log\big(P(z)\big) \tag{2.34}$$

where the state posterior occupancy $\gamma_j(t)$ and state pairwise posterior occupancy $\chi_{ij}(t)$ are computed using current model parameters, and are defined in equations (2.28) and (2.30). By maximising the above auxiliary function, the new estimates of transition probabilities are given as

$$\hat{a}_{ij} = \begin{cases} \gamma_j(1) & i = 1 \quad\quad 1 < j < N \\ \frac{\sum_{t=2}^{T} \chi_{ij}(t)}{\sum_{t=1}^{T} \gamma_i(t)} & 1 < i < N \quad 1 < j < N \\ \frac{\gamma_i(T)}{\sum_{t=1}^{T} \gamma_i(t)} & 1 < i < N \quad j = N \end{cases} \tag{2.39}$$

For HMMs with GMMs as state emission distributions, Gaussian mixture components can be regarded as hidden variables, such that the component posterior $\gamma_{jm}(t)$ is given by

$$\gamma_{jm}(t) = \frac{\sum_{i=2}^{N-1} \alpha_i(t-1) a_{ij} c_{jm} b_{jm}(\mathbf{o}_t) \beta_j(t)}{p(\mathbf{O}|\mathcal{H}, \mathcal{M}_k)} \tag{2.40}$$

where $b_{jm}(\mathbf{o}_t)$ is a Gaussian distribution $\mathcal{N}(\mathbf{o}_t; \boldsymbol{\mu}_{jm}, \boldsymbol{\Sigma}_{jm})$ associated with the mth Gaussian component of state j, and c_{jm} is the weight for the component. Therefore, the re-estimation formulae for HMM parameters are given by

$$\hat{c}_{jm} = \frac{\sum_t \gamma_{jm}(t)}{\sum_{mt} \gamma_{jm}(t)} \tag{2.41}$$

$$\hat{\boldsymbol{\mu}}_{jm} = \frac{\sum_t \gamma_{jm}(t) \mathbf{o}_t}{\sum_t \gamma_{jm}(t)} \tag{2.42}$$

$$\hat{\boldsymbol{\Sigma}}_{jm} = \mathrm{diag}\left(\frac{\sum_t \gamma_{jm}(t) \left(\mathbf{o}_t - \hat{\boldsymbol{\mu}}_{jm}\right) \left(\mathbf{o}_t - \hat{\boldsymbol{\mu}}_{jm}\right)^{\mathrm{T}}}{\sum_t \gamma_{jm}(t)} \right) \tag{2.43}$$

In this work, the mean and covariance matrix of the Gaussian components are of primary interest. Therefore, the ML auxiliary function is obtained in terms of Gaussian components. If $\boldsymbol{\theta} = \{\theta_1, \ldots, \theta_T\}$ represents a Gaussian component sequence corresponding to the observation $\mathbf{O} = \{\mathbf{o}_1, \ldots, \mathbf{o}_T\}$, the likelihood can be evaluated by summing over all possible component sequences as

$$\begin{aligned} p(\mathbf{O}|\mathcal{H}, \mathcal{M}) &= \sum_{\boldsymbol{\theta}} p(\mathbf{O}, \boldsymbol{\theta}|\mathcal{H}, \mathcal{M}) \\ &= \sum_{\boldsymbol{\theta}} P(\boldsymbol{\theta}|\mathcal{H}, \mathcal{M}) \prod_t p(\mathbf{o}_t|\mathcal{M}, \theta_t) \end{aligned} \tag{2.44}$$

This leads to an ML auxiliary function using component sequence posteriors as

$$\mathcal{Q}(\hat{\mathcal{M}}; \mathcal{M}) = \left\langle \log p(\mathbf{O}, \boldsymbol{\theta}|\mathcal{H}, \hat{\mathcal{M}}) \right\rangle_{P(\boldsymbol{\theta}|\mathbf{O}, \mathcal{H}, \mathcal{M})} \tag{2.45}$$

The auxiliary function for the update of Gaussian component can be obtained by rearranging the above equation and ignoring the constant terms independent of the component mean and covariance, leading to

$$\mathcal{Q}(\hat{\mathcal{M}};\mathcal{M}) = -\frac{1}{2}\sum_{tm}\gamma_m^{\mathtt{ml}}(t)\left\{\log|\hat{\boldsymbol{\Sigma}}_m| + (\mathbf{o}_t - \hat{\boldsymbol{\mu}}_m)^{\mathsf{T}}\,\hat{\boldsymbol{\Sigma}}_m^{-1}\,(\mathbf{o}_t - \hat{\boldsymbol{\mu}}_m)\right\} \tag{2.46}$$

where $\gamma_m^{\mathtt{ml}}(t) = P(\theta_t = m|\mathbf{O},\mathcal{M},\mathcal{H})$ is the occupation probability for component m[1] at time t, and is computed through a component level forward-backward algorithm using the current model parameters $\mathcal{M}$. The auxiliary function in the above equation can be also expressed in terms of sufficient statistics $\boldsymbol{\Gamma}^{\mathtt{ml}} = \left\{\gamma_m^{\mathtt{ml}},\mathbf{k}_m^{\mathtt{ml}},\mathbf{L}_m^{\mathtt{ml}}\right\}$ as

$$\begin{aligned}\mathcal{Q}(\hat{\mathcal{M}};\mathcal{M}) = \mathcal{G}(\hat{\mathcal{M}};\boldsymbol{\Gamma}^{\mathtt{ml}}) \quad &= \quad -\frac{1}{2}\sum_{m}\left\{\gamma_m^{\mathtt{ml}}\log|\hat{\boldsymbol{\Sigma}}_m| + \operatorname{tr}\left(\mathbf{L}_m^{\mathtt{ml}}\hat{\boldsymbol{\Sigma}}_m^{-1}\right)\right.\\ &\qquad \left. - 2\hat{\boldsymbol{\mu}}_m^{\mathsf{T}}\hat{\boldsymbol{\Sigma}}_m^{-1}\mathbf{k}_m^{\mathtt{ml}} + \hat{\boldsymbol{\mu}}_m^{\mathsf{T}}\hat{\boldsymbol{\Sigma}}_m^{-1}\hat{\boldsymbol{\mu}}_m\right\}\end{aligned} \tag{2.47}$$

where tr$(\cdot)$ represents the trace of a square matrix, and the sufficient statistics are given as

$$\gamma_m^{\mathtt{ml}} = \sum_t \gamma_m^{\mathtt{ml}}(t) \tag{2.48}$$

$$\mathbf{k}_m^{\mathtt{ml}} = \sum_t \gamma_m^{\mathtt{ml}}(t)\mathbf{o}_t \tag{2.49}$$

$$\mathbf{L}_m^{\mathtt{ml}} = \sum_t \gamma_m^{\mathtt{ml}}(t)\mathbf{o}_t\mathbf{o}_t^{\mathsf{T}} \tag{2.50}$$

These sufficient statistics can be used to find the mean and covariance matrix of Gaussian components, by maximising the auxiliary function in equation (2.47). They are given as

$$\hat{\boldsymbol{\mu}}_m = \frac{\mathbf{k}_m^{\mathtt{ml}}}{\gamma_m^{\mathtt{ml}}} \tag{2.51}$$

$$\hat{\boldsymbol{\Sigma}}_m = \operatorname{diag}\left(\frac{\mathbf{L}_m^{\mathtt{ml}}}{\gamma_m^{\mathtt{ml}}} - \hat{\boldsymbol{\mu}}_m\hat{\boldsymbol{\mu}}_m^{\mathsf{T}}\right) \tag{2.52}$$

and lead to the same parameter updates for the component as given by equations (2.42) and (2.43). These sufficient statistics forms of the auxiliary function and parameter updates will be later used for describing discriminative training of HMMs in section 2.3.4, and for other derivations as well.

The maximum likelihood (ML) training as described in this section may not always lead to an optimal recognition performance in the speech recognition systems, as it has several limitations [73, 124]:

- The ML training estimates HMM parameters to maximise the likelihood $p(\mathbf{O}|\mathcal{H},\mathcal{M})$ of observations, but the correlation between the likelihood and word error rate (WER) may be weak.

[1] It should be noted that m here refers to a unique component in the whole component space of HMMs, whereas in the previous state-level derivations, m referred to the mth component of state j.

- In practical situations, where the training data is limited, ML training may lead to the unreliable estimate of parameters (e.g., the variance of the estimated parameters may be large.)

Therefore, it is preferable to employ a training scheme that explicitly aims at reducing the word error rate and that addresses the data sparsity problems. In the following section, Bayesian training is described that addresses the limited data problem. Thereafter, discriminative training approaches are described that explicitly considers the recognition performance in the training criteria.

2.3.3 Bayesian Training of HMMs

The training of HMM parameters through maximum likelihood, as described in the previous section, assumes a sufficient amount of data to obtain robust estimates of the parameters. However, in many cases, the training data is limited and may not lead to reliable estimates. Bayesian approaches [57, 183] can be used for the estimation of HMM parameters from sparse training data to cope with the uncertainty associated with the parameters. In Bayesian approaches, model parameters are regarded as random variables with probability distributions rather than being point estimates. The likelihood of data is given as a marginalisation over the model parameters,

$$p(\mathbf{O}|\mathcal{H}) = \int_{\mathcal{M}} p(\mathbf{O}|\mathcal{H},\mathcal{M})p(\mathcal{M}|\Phi)\, d\mathcal{M} \tag{2.53}$$

where $p(\mathcal{M}|\Phi)$ is the prior distribution over model parameters with hyperparameters Φ.

The goal of the Bayesian training is to estimate the hyperparameters Φ of the prior distribution. Bayesian training attempts to update the prior distribution to the posterior distribution of the parameters for the given training data. The selection of the form of prior is one of the most important issue in Bayesian training. Generally, a conjugate prior is selected that gives the posterior distribution of parameters in the same form as the prior. In this case, updating the hyperparameters of the prior distributions is equivalent to estimating the posterior distribution. Due to hidden parameters involved in HMMs, a conjugate prior to the likelihood of the observation sequence does not exist [57]. However, conjugate priors to the likelihood of the *complete* data set can be obtained [57]. In continuous density HMMs, for individual Gaussian component means and covariances $\{\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m\}$, the joint conjugate prior density is a normal-Wishart distribution given by [57]

$$\begin{aligned} p(\mathcal{M}|\Phi) \quad \propto \quad & |\boldsymbol{\Sigma}_m|^{-\frac{1}{2}(\alpha_m - D)} \exp\Big(-\frac{1}{2}\mathrm{tr}\big(\tilde{\boldsymbol{\Sigma}}_m \boldsymbol{\Sigma}_m^{-1}\big)\Big) \\ & + \exp\Big(-\frac{\tau_m}{2}\big(\boldsymbol{\mu}_m - \tilde{\boldsymbol{\mu}}_m\big)^{\mathsf{T}} \boldsymbol{\Sigma}_m^{-1} \big(\boldsymbol{\mu}_m - \tilde{\boldsymbol{\mu}}_m\big)\Big) \end{aligned} \tag{2.54}$$

where $\Phi = \{\tau_m, \tilde{\boldsymbol{\mu}}_m, \alpha_m, \tilde{\boldsymbol{\Sigma}}_m\}$ is the set of hyperparameters for the prior distribution, with $\alpha_m > D - 1$ and $\tau_m > 0$. Similarly, a Dirichlet density which is the conjugate prior for the multinomial distribution is used as prior for the mixture component weights [57]. Dirichlet densities can be also used for the initial probability $\boldsymbol{\pi}$ and for each row of the transition probability matrix.

In many cases, the hyperparameters of the prior distribution is assumed known from the subjective knowledge about the stochastic process. In this case, Bayesian training updates the hyperparameters using the posterior distribution of parameters obtained from the training data. When no prior information is given, the hyperparameters can be directly estimated from the training data, using the empirical Bayes approach [149, 150]. In the empirical Bayes approach, the prior is obtained by maximising the marginal likelihood of data given in equation (2.53) with respect to Φ. The estimated prior has the same form and hyperparameters as the posterior distribution $p(\mathcal{M}|\mathbf{O}, \mathcal{H})$ estimated on the given training data with a non-informative prior. As the direct estimation of the hyperparameters of the prior distribution is hard due to hidden parameters in HMMs, various approximations have been investigated. In a variational Bayes approach described in [15, 183], a variational posterior distribution is used instead of the true posterior. Similarly, in the quasi-Bayesian approach in [77], the posterior distribution is assumed to be the exponential of the standard auxiliary function. The details of Bayesian learning with variational methods can be found in [58, 79, 89, 107].

Once the prior distribution is obtained,it is used in recognition to compute the marginal likelihood for inference.

2.3.4 Discriminative Training of HMMs

As described in section 2.3.2, the ML training of HMMs maximises the likelihood of data given the reference transcripts. This leads to the models with poor discriminative ability as ML training does not consider competing hypotheses. A number of discriminative criteria has been investigated which consider the likelihood of competing hypotheses and explicitly model the performance metrics in the criteria. The discriminative training of HMMs have been found to outperform ML training [92, 130, 140, 154]. It is widely used in state-of-the-art speech recognition systems. The following sections describes commonly used discriminative criteria and associated optimisation schemes for discriminative training of HMMs.

2.3.4.1 Discriminative Training Criteria

A discriminative training criterion considers the likelihood from competing hypotheses and also integrates metrics related to the recognition or classification performance in the criterion.

Depending upon the metrics used, a number of different discriminative criteria have been investigated for training of HMMs. Some of the commonly used discriminative criteria are given below.

Maximum Mutual Information (MMI)

The maximum mutual information (MMI) [10, 176] criterion is given by the posterior probability of correct transcripts for the given training data and observations,

$$\begin{aligned}\mathcal{F}_{\mathtt{mmi}}(\mathcal{M}) &= \sum_{u=1}^{U} \log P(\mathcal{H}_{\mathtt{r}}^{(u)}|\mathbf{O}^{(u)}, \mathcal{M}) \\ &= \sum_{u=1}^{U} \log \frac{p(\mathbf{O}^{(u)}|\mathcal{H}_{\mathtt{r}}^{(u)}, \mathcal{M})P(\mathcal{H}_{\mathtt{r}}^{(u)})}{\sum_{\breve{\mathcal{H}}} p(\mathbf{O}^{(u)}|\breve{\mathcal{H}}, \mathcal{M})P(\breve{\mathcal{H}})} \qquad (2.55)\end{aligned}$$

where $\mathbf{O}^{(u)}$ is the training utterance corresponding to the transcript $\mathcal{H}_{\mathtt{r}}^{(u)}$ for the utterance u, and $\breve{\mathcal{H}}$ represents all possible hypotheses. Thus the MMI criterion is equivalent to maximising the ratio of the likelihood of the correct hypotheses (numerator) to that of the "composite" hypotheses (denominator). The denominator hypotheses are usually represented by an N-best list [25] or a lattice [155] for compactness. With $P(\mathcal{H})$ fixed, an MMI criterion is equivalent to a *conditional maximum likelihood* (CML) criterion [124]. Compensating for the difference in the dynamic range of the acoustic score and the language model score, it can be expressed as

$$\mathcal{F}_{\mathtt{mmi}}(\mathcal{M}) = \sum_{u=1}^{U} \log \frac{p^{\kappa}(\mathbf{O}^{(u)}|\mathcal{H}_{\mathtt{r}}^{(u)}, \mathcal{M})P(\mathcal{H}_{\mathtt{r}}^{(u)})}{\sum_{\breve{\mathcal{H}}} p^{\kappa}(\mathbf{O}^{(u)}|\breve{\mathcal{H}}, \mathcal{M})P(\breve{\mathcal{H}})} \qquad (2.56)$$

where κ the acoustic scaling factor. It is usually set to the inverse of the language model scaling factor for speech recognition tasks, and it allows proper consideration of the less likely hypotheses in the criterion [155].

Minimum Classification Error (MCE)

A minimum classification error (MCE) criterion [90] is given by

$$\mathcal{F}_{\mathtt{mce}}(\mathcal{M}) = \sum_{u=1}^{U} f\left(\log \frac{p(\mathbf{O}^{(u)}|\mathcal{M}, \mathcal{H}_{\mathtt{r}}^{(u)})P(\mathcal{H}_{\mathtt{r}}^{(u)})}{\sum_{\breve{\mathcal{H}} \notin \mathcal{H}_{\mathtt{r}}^{(u)}} p(\mathbf{O}^{(u)}|\breve{\mathcal{H}}, \mathcal{M})P(\breve{\mathcal{H}})} \right) \qquad (2.57)$$

where f is a smoothing function, usually taken as an identity function $f(z) = z$ or a sigmoid function given by

$$f(z) = \frac{1}{1 + \exp(-az)}. \qquad (2.58)$$

The difference from the MMI criterion is that the denominator term in the MCE criterion contains only incorrect hypotheses rather than all of them, and the posteriors are smoothed with a sigmoid function.

Minimum Bayes Risk (MBR)

The MMI criterion defines the objective function to minimise sentence error rate, as it considers utterances as either correct or incorrect. The MBR criterion incorporates more general error metrics in the objective function. The MBR criterion [123–125] minimises the Bayesian risk or expected loss given by

$$\mathcal{F}_{\mathtt{mbr}}(\mathcal{M}) \quad = \quad \sum_{u=1}^{U}\sum_{\mathcal{H}} P(\mathcal{H}|\mathbf{O}^{(u)},\mathcal{M})\mathcal{L}(\mathcal{H},\mathcal{H}_{\mathtt{r}}^{(u)}) \tag{2.59}$$

where $\mathcal{L}(\mathcal{H},\mathcal{H}_{\mathtt{r}}^{(u)})$ is a loss function that defines the cost between the hypothesis $\mathcal{H}$ and reference transcript $\mathcal{H}_{\mathtt{r}}^{(u)}$. This loss function can be defined at the sentence, word or phone level, thus leading to different discriminative criteria.

When the loss function is defined at the word level, this leads to the *minimum word error* (MWE) criterion [91, 142]. In the MWE criterion, the Levenshtein distance is used as the loss function. Given the word pair alignment $\{\mathcal{W}^{(k)},\mathcal{W}_{\mathtt{r}}^{(uk)}\}$, the word-level loss function $\mathcal{L}_{\mathtt{word}}(\mathcal{H},\mathcal{H}_{\mathtt{r}}^{(u)})$ is given as

$$\mathcal{L}_{\mathtt{word}}(\mathcal{H},\mathcal{H}_{\mathtt{r}}^{(u)}) = \sum_{k=1}^{K} d_{\mathtt{lev}}\left(\mathcal{W}^{(k)},\mathcal{W}_{\mathtt{r}}^{(uk)}\right) \tag{2.60}$$

where

$$d_{\mathtt{lev}}\left(\mathcal{W}^{(k)},\mathcal{W}_{\mathtt{r}}^{(uk)}\right) = \begin{cases} 0 & \mathcal{W}^{(k)} = \mathcal{W}_{\mathtt{r}}^{(uk)} \\ 1 & \mathcal{W}^{(k)} \neq \mathcal{W}_{\mathtt{r}}^{(uk)} \end{cases} \tag{2.61}$$

Similarly, when the loss function is defined at the phone level, this leads to the popular *minimum phone error* (MPE) criterion [141], and is given by

$$\mathcal{F}_{\mathtt{mpe}}(\mathcal{M}) \quad = \quad \sum_{u=1}^{U}\sum_{\mathcal{H}} P(\mathcal{H}|\mathbf{O}^{(u)},\mathcal{M})\mathcal{L}_{\mathtt{phone}}(\mathcal{H},\mathcal{H}_{\mathtt{r}}^{(u)}) \tag{2.62}$$

where $\mathcal{L}_{\mathtt{phone}}(\mathcal{H},\mathcal{H}_{\mathtt{r}}^{(u)})$ is the phone-level loss function between hypothesis $\mathcal{H}$ and reference $\mathcal{H}_{\mathtt{r}}^{(u)}$. However, generally the MPE criterion is defined in terms of phone correctness for implementation [140, 196]. This is the form that will be used in this work, as it allows the consistent use of the concept of maximisation, function concavity and lower bound. Therefore, the MPE criterion is redefined by replacing the loss function $\mathcal{L}_{\texttt{phone}}(\mathcal{H},\mathcal{H}_{\texttt{r}}^{(u)})$ with a closely related phone-level accuracy function $\mathcal{A}(\mathcal{H},\mathcal{H}_{\texttt{r}}^{(u)})$, and is given as

$$\begin{aligned}\mathcal{F}_{\texttt{mpe}}(\mathcal{M}) &= \sum_{u=1}^{U}\sum_{\mathcal{H}} P(\mathcal{H}|\mathbf{O}^{(u)},\mathcal{M})\mathcal{A}(\mathcal{H},\mathcal{H}_{\texttt{r}}^{(u)}) \\ &= \sum_{\mathcal{H}} \frac{p(\mathbf{O}^{(u)}|\mathcal{M},\mathcal{H})P(\mathcal{H})}{\sum_{\breve{\mathcal{H}}} p(\mathbf{O}^{(u)}|\mathcal{M},\breve{\mathcal{H}})P(\breve{\mathcal{H}})}\mathcal{A}(\mathcal{H},\mathcal{H}_{\texttt{r}}^{(u)}) \qquad (2.63)\end{aligned}$$

where $\mathcal{A}(\mathcal{H},\mathcal{H}_{\texttt{r}}^{(u)})$ is computed by aligning the hypothesis with reference at phone level

$$\mathcal{A}(\mathcal{H},\mathcal{H}_{\texttt{r}}^{(u)}) = \sum_{k=1}^{K}\max_{\mathcal{P}_{\text{r}}} a(\mathcal{P}^{(k)},\mathcal{P}_{\texttt{r}}^{(uk)}) \qquad (2.64)$$

The phone level accuracy is given by

$$a(\mathcal{P}^{(k)},\mathcal{P}_{\texttt{r}}^{(uk)}) = \begin{cases} 1 & \mathcal{P}^{(k)} = \mathcal{P}_{\texttt{r}}^{(uk)} \\ 0 & \mathcal{P}^{(k)} \neq \mathcal{P}_{\texttt{r}}^{(uk)} \\ -1 & \mathcal{P}^{(k)} : \text{ insertion} \end{cases} \qquad (2.65)$$

The alignment for computing the loss function is very expensive when the competing hypotheses are represented in the form of lattices. Therefore, when lattices are used, a heuristic method is used for computing the loss function, without explicitly doing the alignment. The arcs of the lattices representing a phone are marked with time stamps, and the phone level accuracy given above is modified as

$$a(\mathcal{P}^{(k)},\mathcal{P}_{\texttt{r}}^{(uk)}) = \begin{cases} -1 + 2e(\mathcal{P}^{(k)},\mathcal{P}_{\texttt{r}}^{(uk)}) & \mathcal{P}^{(k)} = \mathcal{P}_{\texttt{r}}^{(uk)} \\ -1 + e(\mathcal{P}^{(k)},\mathcal{P}_{\texttt{r}}^{(uk)}) & \mathcal{P}^{(k)} \neq \mathcal{P}_{\texttt{r}}^{(uk)} \end{cases} \qquad (2.66)$$

where $e(\mathcal{P}^{(k)},\mathcal{P}_{\texttt{r}}^{(uk)})$ represents overlap between the phones. Considering the probability scaling factor, the MPE criterion in equation (2.63) can be given as

$$\mathcal{F}_{\texttt{mpe}}(\mathcal{M}) = \sum_{u=1}^{U}\sum_{\mathcal{H}} \frac{p^{\kappa}(\mathbf{O}^{(u)}|\mathcal{H},\mathcal{M})P(\mathcal{H})}{\sum_{\breve{\mathcal{H}}} p^{\kappa}(\mathbf{O}^{(u)}|\breve{\mathcal{H}},\mathcal{M})P(\breve{\mathcal{H}})}\mathcal{A}(\mathcal{H},\mathcal{H}_{\texttt{r}}^{(u)}) \qquad (2.67)$$

In this work, the MPE criterion refers to the form with phone correctness as given in equations (2.63) and (2.67), unless stated otherwise. The MPE criterion is used for discriminative training of all acoustic models in the experiments. The next section describes optimisation of the discriminative criteria. The summation over multiple utterances in the criteria will not be shown from this point onwards in this work for the sake of simplicity.

2.3.4.2 Optimisation of Discriminative Criteria

In maximum likelihood estimation, a lower bound to the likelihood can be obtained, which is then used as an auxiliary function. The maximisation of such an auxiliary function is guaranteed not to decrease the objective function, and is also referred as a *strong-sense auxiliary function* [140]. However, a strict lower-bound is difficult to obtain for discriminative criteria due to the denominator term. Therefore, the discriminative training of the state-of-the-art LVCSR systems is usually done by optimising the discriminative criteria using the extended Baum-Welch (EBW) algorithm [62, 130] or a *weak-sense auxiliary function* [140]. The weak-sense auxiliary function gives the same update equations as given by the extended Baum-Welch (EBM) algorithm [62, 130, 190], but from a different perspective. The weak-sense auxiliary function is defined as a function with the same gradient at the current estimate of the parameters $\mathcal{M}$ such that

$$\left.\frac{\partial \mathcal{Q}(\hat{\mathcal{M}};\mathcal{M})}{\partial \hat{\mathcal{M}}}\right|_{\hat{\mathcal{M}}=\mathcal{M}} = \left.\frac{\partial \mathcal{F}(\hat{\mathcal{M}})}{\partial \hat{\mathcal{M}}}\right|_{\hat{\mathcal{M}}=\mathcal{M}} \tag{2.68}$$

In this case, maximising the auxiliary function with respect to new estimate of model parameters $\hat{\mathcal{M}}$ does not guarantee an increase in the objective function, however when the auxiliary function $\mathcal{Q}(\hat{\mathcal{M}};\mathcal{M})$ reaches a local maximum, the objective function $\mathcal{F}(\hat{\mathcal{M}})$ is also at the local maximum, as gradients of both are same.

The weak-sense auxiliary function is first described for the MMI objective function in this section. The MMI objective function can be expressed as

$$\begin{aligned}\mathcal{F}_{\texttt{mmi}}(\hat{\mathcal{M}}) &= \log \frac{p(\mathbf{O}|\mathcal{H},\hat{\mathcal{M}})P(\mathcal{H})}{\sum_{\breve{\mathcal{H}}} p(\mathbf{O}|\breve{\mathcal{H}},\hat{\mathcal{M}})P(\breve{\mathcal{H}})} \\ &= \log p(\mathbf{O}|\mathcal{H},\hat{\mathcal{M}})P(\mathcal{H}) - \log \sum_{\breve{\mathcal{H}}} p(\mathbf{O}|\breve{\mathcal{H}},\hat{\mathcal{M}})P(\breve{\mathcal{H}}) \end{aligned} \tag{2.69}$$

where $\mathbf{O}$ is the observation sequence corresponding to the reference transcript $\mathcal{H}_{\texttt{r}}$, and $\breve{\mathcal{H}}$ represents all possible hypotheses. The weak-sense auxiliary function for the MMI criterion is given by [140]

$$\mathcal{Q}(\hat{\mathcal{M}};\mathcal{M}) = \mathcal{Q}^{\texttt{num}}(\hat{\mathcal{M}};\mathcal{M}) - \mathcal{Q}^{\texttt{den}}(\hat{\mathcal{M}};\mathcal{M}) \tag{2.70}$$

where $\mathcal{Q}^{\texttt{num}}(\hat{\mathcal{M}};\mathcal{M})$ and $\mathcal{Q}^{\texttt{den}}(\hat{\mathcal{M}};\mathcal{M})$ correspond to the numerator and the denominator likelihoods in the objective function, respectively. The numerator and denominator auxiliary functions have a similar form as the ML auxiliary function in equation (2.47). The numerator auxiliary function is given by

$$\begin{aligned}\mathcal{Q}^{\texttt{num}}(\hat{\mathcal{M}};\mathcal{M})=\mathcal{G}(\hat{\mathcal{M}};\boldsymbol{\Gamma}^{\texttt{num}}) &= -\frac{1}{2}\sum_{m}\Big\{\gamma_m^{\texttt{num}}\log|\hat{\boldsymbol{\Sigma}}_m|+\operatorname{tr}\left(\mathbf{L}_m^{\texttt{num}}\hat{\boldsymbol{\Sigma}}_m^{-1}\right)\\ &\quad -2\hat{\boldsymbol{\mu}}_m^{\mathsf{T}}\hat{\boldsymbol{\Sigma}}_m^{-1}\mathbf{k}_m^{\texttt{num}}+\hat{\boldsymbol{\mu}}_m^{\mathsf{T}}\hat{\boldsymbol{\Sigma}}_m^{-1}\hat{\boldsymbol{\mu}}_m\Big\} \qquad (2.71)\end{aligned}$$

where tr(·) is the trace of a square matrix, and m represents the mixture component index. The numerator sufficient statistics $\boldsymbol{\Gamma}^{\texttt{num}}=\{\gamma_m^{\texttt{num}},\mathbf{k}_m^{\texttt{num}},\mathbf{L}_m^{\texttt{num}}\}$ is also given in the same form as the ML sufficient statistics in equations (2.48)-(2.50) as

$$\gamma_m^{\texttt{num}} = \sum_t \gamma_m^{\texttt{num}}(t) \qquad (2.72)$$

$$\mathbf{k}_m^{\texttt{num}} = \sum_t \gamma_m^{\texttt{num}}(t)\mathbf{o}_t \qquad (2.73)$$

$$\mathbf{L}_m^{\texttt{num}} = \sum_t \gamma_m^{\texttt{num}}(t)\mathbf{o}_t\mathbf{o}_t^{\mathsf{T}} \qquad (2.74)$$

The denominator auxiliary function can be also given in the same form as above, thus obtaining the denominator sufficient statistics $\boldsymbol{\Gamma}^{\texttt{den}}=\left\{\gamma_m^{\texttt{den}},\mathbf{k}_m^{\texttt{den}},\mathbf{L}_m^{\texttt{den}}\right\}$. In the MMI criterion based optimisation, the numerator occupations, and thus the statistics, are computed using reference supervision, whereas that for the denominator are computed using all possible transcripts. Therefore,

$$\gamma_m^{\texttt{num}}(t) = P(\theta_t=m|\mathbf{O},\mathcal{M},\mathcal{H}_{\mathbf{r}}) \qquad (2.75)$$

$$\gamma_m^{\texttt{den}}(t) = \sum_{\mathcal{H}} P(\theta_t=m|\mathbf{O},\mathcal{M},\mathcal{H}) \qquad (2.76)$$

They are usually computed using a lattice-based forward-backward algorithm, as the lattices are used to represent the hypotheses in a compact form for the training of models.

Smoothing

As the auxiliary function in equation (2.70) may not be even concave, a smoothing term is added to ensure its concavity. This gives

$$\mathcal{Q}(\hat{\mathcal{M}};\mathcal{M})=\mathcal{Q}^{\texttt{num}}(\hat{\mathcal{M}};\mathcal{M})-\mathcal{Q}^{\texttt{den}}(\hat{\mathcal{M}};\mathcal{M})+\mathcal{Q}^{\texttt{sm}}(\hat{\mathcal{M}};\mathcal{M}) \qquad (2.77)$$

where the smoothing term $\mathcal{Q}^{\texttt{sm}}(\hat{\mathcal{M}};\mathcal{M})$ is chosen such that its maxima lies at the current estimate of parameters,

$$\left.\frac{\partial\mathcal{Q}^{\texttt{sm}}(\hat{\mathcal{M}};\mathcal{M})}{\partial\hat{\mathcal{M}}}\right|_{\hat{\mathcal{M}}=\mathcal{M}}=0 \qquad (2.78)$$

A form of smoothing function commonly used for model parameter estimation is [155]

$$\mathcal{F}_{\texttt{sm}}(\hat{\mathcal{M}};\mathcal{M}) = \sum_m D_m \int_{\mathbf{o}} p(\mathbf{o}|m,\mathcal{M}) \log p(\mathbf{o}|m,\hat{\mathcal{M}})\, d\mathbf{o} \tag{2.79}$$

which can be expressed in the form of a smoothing auxiliary function as

$$\begin{aligned} \mathcal{Q}^{\texttt{sm}}(\hat{\mathcal{M}};\mathcal{M}) &= \sum_m -\frac{D_m}{2}\Big\{ \log|\hat{\boldsymbol{\Sigma}}_m| + \mathrm{tr}\Big(\left(\boldsymbol{\Sigma}_m + \boldsymbol{\mu}_m\boldsymbol{\mu}_m^{\texttt{T}}\right) \hat{\boldsymbol{\Sigma}}_m^{-1}\Big) \\ &\qquad - 2\hat{\boldsymbol{\mu}}_m^{\texttt{T}}\hat{\boldsymbol{\Sigma}}_m^{-1}\boldsymbol{\mu}_m + \hat{\boldsymbol{\mu}}_m^{\texttt{T}}\hat{\boldsymbol{\Sigma}}_m^{-1}\hat{\boldsymbol{\mu}}_m\Big\} \end{aligned} \tag{2.80}$$

where $\boldsymbol{\Sigma}_m$ and $\boldsymbol{\mu}_m$ are the covariance matrix and mean vector of Gaussian component m from the current model set $\mathcal{M}$, respectively. In terms of sufficient statistics,

$$\mathcal{Q}^{\texttt{sm}}(\hat{\mathcal{M}};\mathcal{M}) = \mathcal{G}(\hat{\mathcal{M}};\boldsymbol{\Gamma}^{\texttt{sm}}) \tag{2.81}$$

where the smoothing function statistics are

$$\boldsymbol{\Gamma}^{\texttt{sm}} = \{D_m, D_m\mathbf{k}_m^{\texttt{sm}}, D_m\mathbf{L}_m^{\texttt{sm}}\} \tag{2.82}$$

$$\mathbf{k}_m^{\texttt{sm}} = \boldsymbol{\mu}_m \tag{2.83}$$

$$\mathbf{L}_m^{\texttt{sm}} = \boldsymbol{\Sigma}_m + \boldsymbol{\mu}_m\boldsymbol{\mu}_m^{\texttt{T}} \tag{2.84}$$

The value of the smoothing factor D_m is critical for the optimisation of the discriminative objective function, and is selected as [140, 190],

$$D_m = \max\left(2\tilde{D}_m, E\gamma_m^{\texttt{den}}\right) \tag{2.85}$$

where $\tilde{D}_m$ is the smallest value required to ensure the updated covariance matrix is positive-definite, and E is a user-specified constant. The value of E is usually selected between 1 and 2 for training of acoustic models in LVCSR [140].

I-Smoothing

The model parameters in discriminative training may be overtrained [190]. This problem of overtraining is dealt by a technique called *I-smoothing* [140, 142]. It consists of introducing a prior distribution over the model parameters. Therefore, for the MMI criterion, the modified objective function can be expressed as

$$\mathcal{F}_{\texttt{mmi}}(\hat{\mathcal{M}}) = \log \frac{p(\mathbf{O}|\mathcal{H},\hat{\mathcal{M}})P(\mathcal{H})}{\sum_{\breve{\mathcal{H}}} p(\mathbf{O}|\breve{\mathcal{H}},\hat{\mathcal{M}})P(\breve{\mathcal{H}})} + \log p(\hat{\mathcal{M}}|\Phi) \tag{2.86}$$

This gives an overall auxiliary function for the discriminative criterion as

$$\mathcal{Q}(\hat{\mathcal{M}};\mathcal{M}) = \mathcal{Q}^{\texttt{num}}(\hat{\mathcal{M}};\mathcal{M}) - \mathcal{Q}^{\texttt{den}}(\hat{\mathcal{M}};\mathcal{M}) + \mathcal{Q}^{\texttt{sm}}(\hat{\mathcal{M}};\mathcal{M}) + \log p(\hat{\mathcal{M}}|\Phi) \tag{2.87}$$

A Normal-Wishart distribution [27] is commonly used for the I-smoothing prior. The auxiliary function for the prior term is the function itself, excluding the constant terms

$$\begin{aligned}\mathcal{Q}^{\mathtt{I}}(\hat{\mathcal{M}};\mathcal{M}) \;&=\; -\frac{\tau^{\mathtt{I}}}{2}\sum_{m}\Big\{\log|\hat{\boldsymbol{\Sigma}}_m| + \mathrm{tr}\Big(\tilde{\boldsymbol{\Sigma}}_m\hat{\boldsymbol{\Sigma}}_m^{-1}\Big) \\ &\qquad + \Big(\hat{\boldsymbol{\mu}}_m - \tilde{\boldsymbol{\mu}}_m\Big)^{\mathtt{T}}\hat{\boldsymbol{\Sigma}}_m^{-1}\Big(\hat{\boldsymbol{\mu}}_m - \tilde{\boldsymbol{\mu}}_m\Big)\Big\} \qquad (2.88)\end{aligned}$$

where $\Phi = \{\tau^{\mathtt{I}}, \tilde{\boldsymbol{\mu}}_m, \tilde{\boldsymbol{\Sigma}}_m\}$ is the set of hyperparameters of the I-smoothing prior distribution and $\tau^{\mathtt{I}}$ controls the impact of the prior. The value of $\tau^{\mathtt{I}}$ is normally tuned to specific tasks. This I-smoothing auxiliary function can be expressed in the same form as equation (2.47) using sufficient statistics as

$$\mathcal{Q}^{\mathtt{I}}(\hat{\mathcal{M}};\mathcal{M}) = \mathcal{G}(\hat{\mathcal{M}};\boldsymbol{\Gamma}^{\mathtt{I}}) \qquad (2.89)$$

where the I-smoothing sufficient statistics are given as

$$\begin{aligned}\boldsymbol{\Gamma}^{\mathtt{I}} \;&=\; \{\tau^{\mathtt{I}}, \tau^{\mathtt{I}}\mathbf{k}_m^{\mathtt{I}}, \tau^{\mathtt{I}}\mathbf{L}_m^{\mathtt{I}}\} &(2.90)\\ \mathbf{k}_m^{\mathtt{I}} \;&=\; \tilde{\boldsymbol{\mu}}_m &(2.91)\\ \mathbf{L}_m^{\mathtt{I}} \;&=\; \tilde{\boldsymbol{\Sigma}}_m + \tilde{\boldsymbol{\mu}}_m\tilde{\boldsymbol{\mu}}_m^{\mathtt{T}} &(2.92)\end{aligned}$$

The hyper-parameters $\tilde{\boldsymbol{\mu}}_m$ and $\tilde{\boldsymbol{\Sigma}}_m$ for the I-smoothing distribution may be based on ML statistics $\boldsymbol{\Gamma}^{\mathtt{ml}}$ [142].

Parameter Estimation

As seen above, the overall auxiliary function for the discriminative criterion constitutes of numerator, denominator, smoothing and I-smoothing parts. By using the auxiliary functions in terms of sufficient statistics, the overall auxiliary function can be expressed as

$$\mathcal{Q}(\hat{\mathcal{M}};\mathcal{M}) = \mathcal{G}(\hat{\mathcal{M}};\boldsymbol{\Gamma}^{\mathtt{num}}) - \mathcal{G}(\hat{\mathcal{M}};\boldsymbol{\Gamma}^{\mathtt{den}}) + \mathcal{G}(\hat{\mathcal{M}};\boldsymbol{\Gamma}^{\mathtt{sm}}) + \mathcal{G}(\hat{\mathcal{M}};\boldsymbol{\Gamma}^{\mathtt{I}}) \qquad (2.93)$$

All the constituent terms have the same form in terms of sufficient statistics. The sufficient statistics can be simply combined, giving overall sufficient statistics $\boldsymbol{\Gamma} = \{\gamma_m, \mathbf{k}_m, \mathbf{L}_m\}$ as

$$\begin{aligned}\gamma_m \;&=\; \gamma_m^{\mathtt{num}} - \gamma_m^{\mathtt{den}} + D_m + \tau^{\mathtt{I}} &(2.94)\\ \mathbf{k}_m \;&=\; \mathbf{k}_m^{\mathtt{num}} - \mathbf{k}_m^{\mathtt{den}} + D_m\mathbf{k}_m^{\mathtt{sm}} + \tau^{\mathtt{I}}\mathbf{k}_m^{\mathtt{I}} &(2.95)\\ \mathbf{L}_m \;&=\; \mathbf{L}_m^{\mathtt{num}} - \mathbf{L}_m^{\mathtt{den}} + D_m\mathbf{L}_m^{\mathtt{sm}} + \tau^{\mathtt{I}}\mathbf{L}_m^{\mathtt{I}} &(2.96)\end{aligned}$$

Therefore, maximising the auxiliary function in equation (2.93) with respect to model parameters gives the parameter estimates as [140]

$$\hat{\boldsymbol{\mu}}_m = \frac{\mathbf{k}_m}{\gamma_m} \tag{2.97}$$

$$\hat{\boldsymbol{\Sigma}}_m = \operatorname{diag}\left(\frac{\mathbf{L}_m}{\gamma_m} - \hat{\boldsymbol{\mu}}_m \hat{\boldsymbol{\mu}}_m^{\mathtt{T}}\right) \tag{2.98}$$

In this way, means and covariance matrices of model parameters are estimated using the MMI criterion. The optimisation of other discriminative criteria such as MPE is also the same, except that the computation of occupation probabilities is slightly different, and a different smoothing prior may be used. The MPE objective function is expressed as

$$\mathcal{F}_{\mathtt{mpe}}(\hat{\mathcal{M}}) = \sum_{\mathcal{H}} \frac{p(\mathbf{O}|\mathcal{H}, \hat{\mathcal{M}})P(\mathcal{H})}{\sum_{\breve{\mathcal{H}}} p(\mathbf{O}|\breve{\mathcal{H}}, \hat{\mathcal{M}})P(\breve{\mathcal{H}})} \mathcal{A}(\mathcal{H}, \mathcal{H}_{\mathtt{r}}) \tag{2.99}$$

where $\mathcal{A}(\mathcal{H}, \mathcal{H}_{\mathtt{r}})$ is the raw phone accuracy between hypothesis $\mathcal{H}$ and reference transcript $\mathcal{H}_{\mathtt{r}}$ as described in section 2.3.4.1. The auxiliary function for the MPE criterion is defined in terms of the log-likelihood of phone arcs $\log p(\mathbf{O}|l, \mathcal{M})$ as [140]

$$\begin{aligned} \mathcal{Q}(\hat{\mathcal{M}}; \mathcal{M}) &= \sum_{l} \frac{\partial \mathcal{F}_{\mathtt{mpe}}(\hat{\mathcal{M}})}{\partial \log p(\mathbf{O}|l, \hat{\mathcal{M}})}\bigg|_{\hat{\mathcal{M}}=\mathcal{M}} \log p(\mathbf{O}|l, \hat{\mathcal{M}}) \\ &= \sum_{l} \gamma_l^{\mathtt{mpe}} \log p(\mathbf{O}|l, \hat{\mathcal{M}}) \end{aligned} \tag{2.100}$$

where $\gamma_l^{\mathtt{mpe}}$ is the “posterior probability” of arc l defined by

$$\gamma_l^{\mathtt{mpe}} = \frac{\partial \mathcal{F}_{\mathtt{mpe}}(\hat{\mathcal{M}})}{\partial \log p(\mathbf{O}|l, \hat{\mathcal{M}})}\bigg|_{\hat{\mathcal{M}}=\mathcal{M}} = \gamma_l(\bar{\mathcal{A}}_l - \bar{\mathcal{A}}) \tag{2.101}$$

where γ_l is the occupation probability of arc l calculated from a lattice based forward-backward algorithm. $\bar{\mathcal{A}}_l$ defines average accuracy, $\mathcal{A}(\mathcal{H}, \mathcal{H}_{\mathtt{r}})$, of the hypotheses passing through arc l, and $\bar{\mathcal{A}}$ represents the average accuracy of all the hypotheses in the recognition lattice for each utterance. Depending upon the sign of $\gamma_l^{\mathtt{mpe}}$, it can be divided into numerator and denominator parts. The positive $\gamma_l^{\mathtt{mpe}}$, for which the average arc accuracy is higher than the overall average accuracy, is classified as numerator counts, whereas the arcs with negative $\gamma_l^{\mathtt{mpe}}$ are assigned as denominator. Therefore, the numerator and denominator occupations for the MPE criterion can be given as

$$\gamma_m^{\mathtt{num}}(t) = \sum_{l: s_l \le t \le e_l} \gamma_{lm}(t) \max(0, \gamma_l^{\mathtt{mpe}}(t)) \tag{2.102}$$

$$\gamma_m^{\mathtt{den}}(t) = \sum_{l: s_l \le t \le e_l} \gamma_{lm}(t) \max(0, -\gamma_l^{\mathtt{mpe}}(t)) \tag{2.103}$$

where s_l and e_l are start and end times of phone arc l, respectively, and $\gamma_{lm}(t)$ is the occupation probability of the mth mixture component at time t conditioned on arc l. With the occupation probabilities divided into numerator and denominator groups, the auxiliary function for the MPE criterion can be also expressed in the same form as equation (2.70). The optimisation of the MPE criterion through a weak-sense auxiliary function also leads to the same formulae for updates of model parameters, however occupations given in equations (2.102) and (2.103) are used to compute the sufficient statistics. In the case of the MPE criterion, an MMI prior may be used for I-smoothing, instead of the ML prior [143]. The I-smoothing prior is added to the MPE objective function as

$$\mathcal{F}_{\mathtt{mpe}}(\hat{\mathcal{M}}) = \sum_{\mathcal{H}} \frac{p(\mathbf{O}|\mathcal{H},\hat{\mathcal{M}})P(\mathcal{H})}{\sum_{\breve{\mathcal{H}}} p(\mathbf{O}|\breve{\mathcal{H}},\hat{\mathcal{M}})P(\breve{\mathcal{H}})} \mathcal{A}(\mathcal{H},\mathcal{H}_{\mathtt{r}}) + \log p(\hat{\mathcal{M}}|\Phi) \tag{2.104}$$

The use of I-smoothing is essential for robust MPE training of the models [142] and it is commonly used in state-of-the-art systems.

2.3.5 Context Dependent Models and Parameter Tying

In speech recognition, whole-word acoustic models can be used for small vocabulary tasks such as digit recognition. However as the number of words increases, it becomes difficult to obtain a sufficient amount of training data for each word. Besides, some words or contexts may not be ever seen in the training data. Therefore, a set of subword units such as phones or syllables are used instead which forms significantly smaller set than the words. The words are mapped into the sequence of subword units using a lexicon, described in section 2.4. These subword units are then trained and used in recognition by concatenating them as the *beads-on-a-string* model to represent an utterance. They are context-independent, and when phones are used as the subword units, they are called *monophones*. The problem with monophones is that they fail to capture coarticulatory effects present in speech. The realisation of each phone is dependent upon the phonetic context. Therefore it is essential to consider the context dependency when defining the phone set or acoustic units [11, 132, 158]. *Triphones* are often used in speech recognition systems which consider both the immediate right and left contexts [192]. Similarly, a shorter or longer contexts can be also used, for example, *biphones* consider only either immediate right or left context, and *quinphones* [65] consider two phones to both the right and left of the current phone. The lexicon in this case expands the word into context-dependent models. These context-dependent models can be word-internal or cross-word depending upon whether they consider contexts of preceding and following *words* or not. A word-internal triphone representation of '`pronounce`' can be given as

```
pronounce     p+r p-r+ax r-ax+n ax-n+aw n-aw+n aw-n+s n-s
```

It should be noted that due to the word-internal constraint, the context-dependent expansion in the above example has to use biphones at the word boundary. Cross-word triphones are the most widely used models in speech recognition systems due to the significant reduction in word error rate obtained with them [192].

One problem with the context-dependent models is that the number of models and parameters to train increases exponentially with the size of the phone context considered. For example, with N monophones, the number of possible triphones would be N^3. It is difficult to robustly train all these models even with a large amount of data, as some of the models may be still unseen or may have occurred only a few number of times [71]. To deal with the problem of unseen models and insufficient training data, some of the parameters can be shared or tied across the models [71, 100, 197, 198]. The statistics from all models or states to be shared are then used to estimate the shared parameters. This parameter tying can be performed at different levels such as phones, states, Gaussian components, or even means and covariances of the components [64]. One of the commonly employed tying of model parameters is called state-clustering [197]. In this approach, the states to be tied can be determined by a data-driven or a decision tree based approach, as described below.

Data-driven State Tying

In a data-driven bottom-up state tying scheme [197, 198], all states are initially placed into individual clusters. A distance metric is computed between each pair of states, and the pair with a distance under a given threshold is put into the same cluster. This continues until the size of the largest cluster reaches a threshold, or a desired number of clusters is obtained. The size of the cluster is defined by greatest distance between any two states. The distance is taken as the weighted Euclidean distance between means for output with single Gaussians, whereas for a tied-mixture system, the Euclidean distance between mixture component weights is used. The data-driven clustering is shown in figure 2.6. The problem with this approach is that it is not reliable for contexts with a small amount of training data and it cannot also deal with the unseen contexts.

Decision-tree Based Clustering

The problem of clustering with rarely or unseen contexts can be overcome by using decision-tree based clustering [12, 131, 132, 198]. In this method, a binary tree with yes/no phonetic context questions attached to each node of the decision tree is associated with each state position i of the phone q. A top-down approach is followed to grow the tree by initially

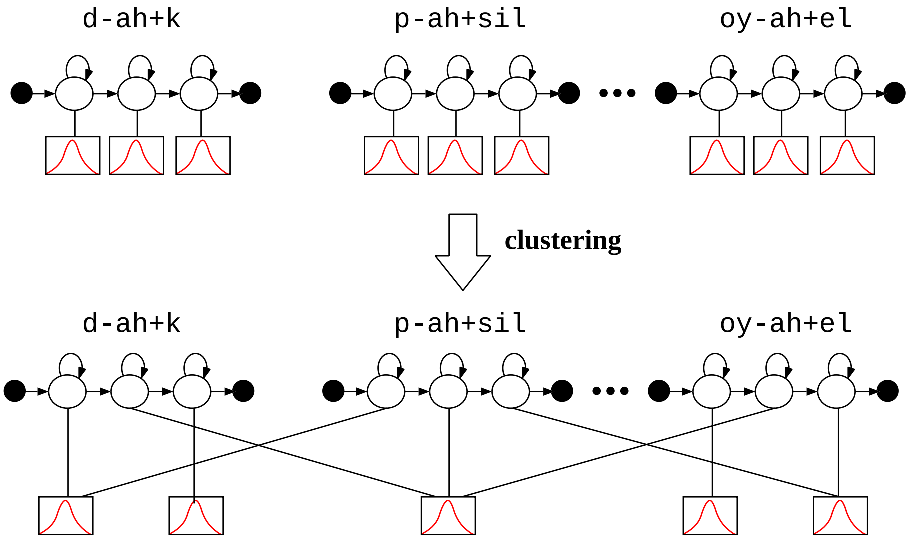


Figure 2.6: The data-driven state tying

assigning all states i of all logical models derived from q to the root node of the tree. This pool of states is successively split depending upon the answer at each node, until the states trickle down to leaf nodes or the amount of data associated with the node falls below a minimum threshold. All states in the same leaf nodes are tied together. The decision-tree based clustering is shown in figure 2.7. The question at each node is selected from a predetermined set. The question is selected to locally maximise the likelihood of training data for the given final state tying. The decision tree can be grown efficiently through a greedy iterative node splitting algorithm. The decision tree can handle unseen contexts or logical models, and is therefore widely used in LVCSR.

Covariance Tying

The use of full covariance matrices greatly increases the numbers of parameters to be estimated, and there may not be a sufficient amount of training data to robustly estimate them. This has motivated the tying of covariance parameters across the classes. One of the commonly used techniques for this is semi-tied covariance (STC) [47] modelling which represents each covariance with two elements: a component-specific diagonal covariance and a semi-tied class dependent non-diagonal matrix. This decomposition is usually done for the precision matrix (inverse of covariance) as the inverse of covariance is used in the likelihood calculation. The precision matrix with STC modelling is given by

$$\boldsymbol{\Sigma}_m^{-1} = \mathbf{A}_{\texttt{stc}}^{\texttt{T}} \boldsymbol{\Sigma}_{\texttt{diag},m}^{-1} \mathbf{A}_{\texttt{stc}} \tag{2.105}$$

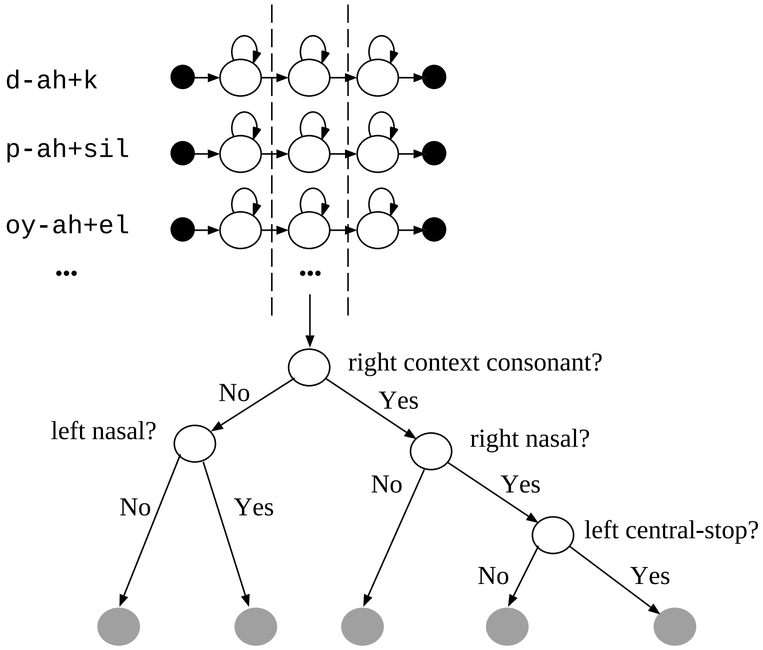


Figure 2.7: A decision-tree based state tying

where $\mathbf{A}_{\texttt{stc}}$ is called semi-tied transform and $\boldsymbol{\Sigma}^{-1}_{\texttt{diag,m}}$ is diagonal. The transform $\mathbf{A}_{\texttt{stc}}$ can be tied globally or across a class. The component likelihood in this case can be given as

$$\mathcal{N}(\mathbf{o}_t; \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m) = |\mathbf{A}_{\texttt{stc}}| \mathcal{N}\left(\mathbf{A}_{\texttt{stc}}\mathbf{o}_t; \mathbf{A}_{\texttt{stc}}\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_{\texttt{diag},m}\right) \tag{2.106}$$

The advantage of this form is that the computation cost reduces to $\mathcal{O}(D)$ with the use of diagonal covariance matrices, compared to $\mathcal{O}(D^2)$ cost with full covariance matrices. The parameter $\mathbf{A}_{\texttt{stc}}\boldsymbol{\mu}_m$ is stored for each component and transformed features $\mathbf{A}_{\texttt{stc}}\mathbf{o}_t$ are cached for each time instance, thus giving almost no increase in computation cost during recognition than the standard diagonal covariance matrix based system. The semi-tied covariance system is trained using EM algorithm [47]. The STC matrix $\mathbf{A}_{\texttt{stc}}$ is initialised to the identity matrix and $\boldsymbol{\Sigma}_{\texttt{diag},m}$ to the current model covariances, and they are estimated in an interleaved fashion. STC is one form of structured covariance modelling. Other types of structured covariance modelling includes precision matrix modelling [134, 162], subspace constrained precision and means (SPAM) [8], mixtures of inverse covariances [177], and extended maximum likelihood linear transforms (EMLLT) [134]. It should be noted that STC is a specific case of structured covariance representation called precision matrix modelling [134, 162], which models the precision matrix as

$$\boldsymbol{\Sigma}^{-1}_m = \sum_{i=1}^{B} \nu_{mi}\mathbf{S}_i \tag{2.107}$$

where ν_{mi} is a component-specific weight that determines the contributions from B global positive semi-definite matrices $\mathbf{S}_i$. When the number of bases is equal to the number of feature dimensions ($B = D$), and the bases are symmetric 1-rank matrices represented as $\mathbf{S}_i = \mathbf{a}_i^{\mathtt{T}}\mathbf{a}_i$ with $\mathbf{a}_i$ being each row of $\mathbf{A}_{\mathtt{stc}}$, the above equation reduces to STC modelling

$$\boldsymbol{\Sigma}_m^{-1} = \sum_{i=1}^{D} \nu_{mi}\mathbf{a}_i^{\mathtt{T}}\mathbf{a}_i = \mathbf{A}_{\mathtt{stc}}^{\mathtt{T}}\boldsymbol{\Sigma}_{\mathtt{diag},m}^{-1}\mathbf{A}_{\mathtt{stc}}^{\mathtt{T}} \tag{2.108}$$

where the weight for a dimension is the inverse variance of the corresponding dimension.

2.3.6 Model Based Feature Projection

There may be a correlation between different dimensions of the feature vectors, for example due to overlapping windows for frames and using delta coefficients. Even after applying DCT transforms, the features are not complete uncorrelated. However, it is desirable to have compact and discriminative features for effective speech recognition. A range of linear projection schemes have been proposed to project the feature vector into an uncorrelated subspace thus increasing discriminative capability. The linear projection schemes transform an n-dimensional observation vector $\mathbf{o}_t$ using a $p \times n$ transform matrix $\mathbf{A}_{[p]}$ to obtain a new p-dimensional observation $\hat{\mathbf{o}}_t$ as

$$\hat{\mathbf{o}}_t = \mathbf{A}_{[p]}\mathbf{o}_t. \tag{2.109}$$

Some of the commonly used linear projection schemes are principal component analysis (PCA) [36], linear discriminant analysis (LDA) [18, 36] and heteroscedastic linear discriminant analysis (HLDA) [98].

In PCA [36], the data covariance matrix of all observations is decorrelated by using an eigen-value decomposition. The PCA transform is estimated by find the p rows of orthogonal matrix $\mathbf{A}$ as

$$\hat{\mathbf{A}}_{\mathtt{pca},[p]} = \arg\max_{\mathbf{A}_{[p]}} \left\{ \left| \mathbf{A}_{[p]}\boldsymbol{\Sigma}_g\mathbf{A}_{[p]}^{\mathtt{T}} \right| \right\} \tag{2.110}$$

where $\boldsymbol{\Sigma}_g$ is the global covariance matrix of the observations. This selects the orthogonal projections of features that maximises the total variance in the projected subspace. However, this may not necessarily lead to a subspace that is discriminative between classes. A linear discriminant analysis (LDA) [18, 36] can be used which is a supervised scheme and assumes each Gaussian component in the model as a separate class to be discriminated. The LDA transform is obtained by maximising the ratio of projected between class covariance $\boldsymbol{B}$ and average within class covariance $\boldsymbol{\Sigma}$, as

$$\hat{\mathbf{A}}_{\mathtt{lda},[p]} = \arg\max_{\mathbf{A}_{[p]}} \left\{ \frac{\left|\mathtt{diag}\left(\mathbf{A}_{[p]}\mathbf{B}\mathbf{A}_{[p]}^{\mathtt{T}}\right)\right|}{\left|\mathtt{diag}\left(\mathbf{A}_{[p]}\boldsymbol{\Sigma}\mathbf{A}_{[p]}^{\mathtt{T}}\right)\right|} \right\} \tag{2.111}$$

where both the between class covariance $\mathbf{B}$ and the within class covariance $\boldsymbol{\Sigma}$ are constrained to be diagonal in the projected subspace. The LDA transform can be obtained by finding eigen-vectors associated with top p eigen-values of $\boldsymbol{\Sigma}^{-1}\mathbf{B}$ [36, 185]. A maximum-likelihood estimation of the LDA transform [18] can be given as

$$\hat{\mathbf{A}}_{\mathtt{lda}} = \arg\max_{\mathbf{A}} \left\{ \sum_{mt} \gamma_{jm}(t) \left(\log|\mathbf{A}|^2 - \log|\tilde{\boldsymbol{\Sigma}}_{\mathtt{diag}}| \right) \right\} \tag{2.112}$$

where $\gamma_m(t)$ is the occupation probability of component m at time t as computed through the forward-backward algorithm in section 2.3, and $\tilde{\boldsymbol{\Sigma}}_{\mathtt{diag}}$ is the transformed average within class diagonal covariance in the feature space defined by $\mathbf{A}$ consisting of p useful rows and $(n-p)$ nuisance rows. LDA suffers from the assumption that the within class covariances for all components are the same. This is relaxed in HLDA, which is estimated as [98]

$$\hat{\mathbf{A}}_{\mathtt{hlda}} = \arg\max_{\mathbf{A}} \left\{ \sum_{mt} \gamma_{m}(t) \left(\log|\mathbf{A}|^2 - \log|\tilde{\boldsymbol{\Sigma}}_{\mathtt{diag},m}| \right) \right\} \tag{2.113}$$

where $\tilde{\boldsymbol{\Sigma}}_{\mathtt{diag},m}$ is the transformed diagonal covariance in the feature space defined by $\mathbf{A}$. HLDA is widely used in the state-of-the-art speech recognition systems [54]. In this work, (39×52) dimensional HLDA transforms are used to project 52-dimensional initial feature vectors to a 39-dimensional space.

2.4 Lexicon

A lexicon or dictionary is one of the building blocks of a speech recognition system, as shown in figure 2.1. It defines the allowed vocabulary set for speech recognition and provides pronunciations for the words. A lexicon consists of one or more pronunciations for a given word, usually given at the phone level. In the case of multiple pronunciations, a pronunciation probability may be also specified. The inflected forms of a word are usually considered different words in the lexicon [73]. The pronunciations in the dictionary may be obtained from difference sources, and can be also derived through rule based or data-driven approaches. It is preferable to have a smaller vocabulary size, as it reduces the potential confusable candidates thus possibly giving better word accuracy. However, it may also introduce *out-of-vocabulary*

(OOV) errors. They occur when the word to be recognised is not in the lexicon. In practice, a fixed-size vocabulary is often used. The words in the dictionary are selected to minimise the expected OOV. For a desired vocabulary size of V, a minimum OOV rate vocabulary can be obtained by selecting the most frequent V words in the dictionary [73].

Speech recognition tasks are often classified according to their vocabulary size. The tasks with less than 1k words are called small vocabulary tasks, between 1k - 10k words are referred as medium vocabulary tasks, and greater than 10k vocabulary are called large vocabulary tasks.

2.5 Language Models

A language model (LM) is used in speech recognition systems as shown in figure 2.1 that represents syntactic and semantic information in spoken word sequences. It gives the probability of hypothesis $\mathcal{H} = \{\mathcal{W}_1, \ldots, \mathcal{W}_K\}$ constituting a sequence of words $\mathcal{W}_k$. The probability of the hypothesis can be expressed as a product of condition probabilities

$$P(\mathcal{H}) = \prod_{k=1}^{K} P(\mathcal{W}_k|\mathcal{W}_{k-1}, \ldots, \mathcal{W}_1) \tag{2.114}$$

This requires consideration of the full history of the words. The number of possible word sequences is very large in LVCSR systems. Consequently, it is not possible to obtain robust estimates of language model probabilities for all possible word sequences.

One solution to the above problem is to restrict the history to the preceding $(N-1)$ words only. This is referred as an *N-gram* language model and is currently the most popular model used in speech recognition. The probability of the hypothesis with the N-gram model is given by

$$P(\mathcal{W}_k|\mathcal{W}_{k-1}, \ldots, \mathcal{W}_1) \approx P(\mathcal{W}_k|\mathcal{W}_{k-1}, \ldots, \mathcal{W}_{k-N+1}) \tag{2.115}$$

In the above equation, when $N = 2$, it yields a *bigram* language model, whereas for $N = 3$, a *trigram* language model is obtained. In this work, bigram and trigram language models are used. The bigram language models are used for generating initial hypotheses in this work.

The ML estimates of N-gram language model probabilities are given by [73]

$$P(\mathcal{W}_k|\mathcal{W}_{k-1}, \ldots, \mathcal{W}_{k-N+1}) = \frac{C(\mathcal{W}_k, \mathcal{W}_{k-1}, \ldots, \mathcal{W}_{k-N+1})}{\sum_{\mathcal{W}} C(\mathcal{W}, \mathcal{W}_{k-1}, \ldots, \mathcal{W}_{k-N+1})} \tag{2.116}$$

where $C(\mathcal{W}_k, \mathcal{W}_{k-1}, \ldots, \mathcal{W}_{k-N+1})$ is the frequency count of the N-gram word sequence occurred in the training data. The major problem with such estimation is that all possible N-grams are required to be covered with sufficient counts for robust estimates of language model probabilities. This is not feasible for even small values of N. This data sparsity problem can be dealt by discounting and backoff techniques. Discounting handles the unobserved N-grams by taking out some of the probability mass from seen N-grams and allocating it to the unseen N-grams. Backoff, on the other hand, discards the low-count estimates of higher-order N-grams and uses more frequently observed shorter context estimates in their place. Some of the discounting and back-off techniques include absolute discounting [128], Good-Turing discounting [61], Witten-Bell discounting [188], Katz-smoothing [93], and Kneser-Ney smoothing [127]. Other approaches for obtaining robust LMs are interpolation [85] and class-based LMs [17, 112, 121]. A review of smoothing techniques for language models can be found in [19, 20].

The language model can become very large when high order N-grams are used. This makes the training and decoding process very slow for a large vocabulary speech recognition system. Therefore, it becomes necessary to *prune* some of the N-grams. A number of different criteria based on information theoretic measure can be used for the purpose. For example, the pruning can be done by minimising the KL-divergence between distributions of the unpruned and pruned models [166]. The optimal pruning of language models can speed up the training and decoding process significantly with only a small degradation in performance.

LMs can be compared by computing their *perplexities* on a test text corpus [73].

2.6 Recognition of Speech Using HMMs

The recognition of speech refers to finding the best word sequence representation for the given speech. The recognition process uses the acoustic and language models and the lexicon described above to decode the test speech, as shown in figure 2.1. Several decoding and search algorithms can be used for the purpose as described below.

2.6.1 MAP Decoding

As discussed in section 2.1, the best hypothesis for the given speech $\mathbf{O}$ is selected as the one with a maximum a-posteriori probability

$$\begin{aligned} \hat{\mathcal{H}} &= \arg\max_{\mathcal{H}} \left\{ P(\mathcal{H}|\mathbf{O},\mathcal{M}) \right\} \\ &= \arg\max_{\mathcal{H}} \left\{ p(\mathbf{O}|\mathcal{H},\mathcal{M}) P(\mathcal{H}) \right\} \end{aligned} \tag{2.117}$$

where $P(\mathcal{H}|\mathbf{O},\mathcal{M})$ is computed by using the acoustic model and $P(\mathcal{H})$ is obtained from the language model as discussed in section 2.1 The number of possible hypothesis or word sequences grows exponentially as the number of words in the hypothesis or vocabulary size increases. If each of the possible hypotheses is considered separately, the search process becomes intractable. A number of strategies can be used for searching for the best possible word sequence. This is often approximated by finding the most likely state sequence. The Viterbi algorithm [145, 178] can be used to find the most likely state sequence efficiently.

The Viterbi algorithm introduces a partial best-path probability $\phi_j(t)$ which represents the likelihood of the most likely state sequence at time t that generated observation sequence from $\{\mathbf{o}_1, \ldots, \mathbf{o}_t\}$ and ended in state j. A recursion is used to compute this partial likelihood as

$$\phi_j(t) \quad = \quad \max_i \left\{ \phi_i(t-1) a_{ij} \right\} b_j(\mathbf{o}_t), \quad 1 < j < N, \quad 1 \leq t \leq T \tag{2.118}$$

where the initial condition is given by

$$\phi_j(0) = \begin{cases} 1 & j = 1 \\ 0 & \text{otherwise} \end{cases} \tag{2.119}$$

The maximum likelihood for the most likely state sequence is then given by

$$\phi_N(T+1) \quad = \quad \max_i \left\{ \phi_i(T) a_{iN} \right\} \tag{2.120}$$

This is also known as the Viterbi likelihood. It should be noted that likelihood computation through the Viterbi algorithm uses the max operation in place of the summation as in the forward-backward algorithm. Thus the forward-backward algorithm gives the total likelihood of all paths, whereas the Viterbi algorithm gives the likelihood of the best path only. This allows easy generalisation of Viterbi decoding to continuous speech recognition. The Viterbi algorithm can be extended for a continuous speech recognition system with a *token-passing algorithm* [196].

The use of N-gram language models and crossword triphones makes it complex to implement a Viterbi decoder for continuous speech recognition systems, as it increases the search space drastically and may cause memory issues. This can be dealt with by dynamically expanding the search space as the contexts are encountered during decoding [133, 136]. Alternative search strategies like stack decoding [84, 138] can be also used.

The search efficiency is an important factor while decoding. The use of complex acoustic and language models can greatly increase the search space of the decoder. This can be dealt with by *pruning* the low likelihood paths, and thus expanding only a certain number of paths at each stage [73]. This is called *beam-search*. It may be implemented by maintaining only a certain number of the most promising paths, or by discarding the paths whose likelihoods are lower than a certain threshold [73]. It can also be implemented by maintaining only the paths that have likelihoods less by a threshold amount than the likelihood of the most promising path [73]. Pruning speeds up the decoding process, but may introduce *search errors* if the likely paths are pruned before reaching the end of the utterance.

Though the task of the speech recognition system is to find the best hypothesis, a number of possible hypotheses can be output for further processing. This is called an *N-best list* [156, 157], and typically the size of the list is 100 to 1000. These hypotheses can be used for rescoring and reranking, without redecoding the data from the beginning. The possible hypotheses can be more compactly represented in a form of word graphs, called *lattices* [136, 148]. A word lattice constitutes of nodes and arcs, with each node representing a point in time and arcs representing hypothesised words. The arc can be also assigned a score such as language and acoustic model scores. The word lattices are very useful and efficient as they can be rescored quickly, for example with new higher order language models, and thus they are widely used in the state-of-the-art speech recognition systems. An example of word lattices is shown in figure 2.8.

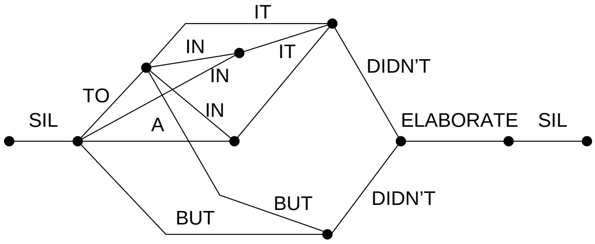


Figure 2.8: A word-lattice of recognised hypotheses (from [52])

In practice, there is a large difference in the dynamic range of acoustic and language model scores due to the modelling assumptions. Therefore, a language model or grammar scaling factor is used to scale the LM score. In addition, the decoder is prone to inserting short words as they tend to have larger likelihoods due to their presence in the training data [73]. This problem can be dealt by using a word insertion penalty for each new word in the recognition hypothesis. Therefore, the inference in many systems is done as

$$\hat{\mathcal{H}} = \arg\max_{\mathcal{H}} \left\{ \log p(\mathbf{O}|\mathcal{H}, \mathcal{M}) + \beta \log P(\mathcal{H}) + \rho|\mathcal{H}| \right\} \tag{2.121}$$

where β is the language model scaling factor, ρ is the word insertion penalty and $|\mathcal{H}|$ is the length of the hypothesis $\mathcal{H}$ in words.

2.6.2 MBR Decoding

MAP decoding finds the most likely sentence. Thus it can be viewed as minimising the expected sentence error rate. However, the performance of speech recognition system is often measured is terms of the word error rate (WER). MBR decoding [60, 110, 167, 187] addresses this issue by integrating the evaluation metrics into the decoding criterion, and finds the best hypothesis as

$$\hat{\mathcal{H}} = \arg\min_{\mathcal{H}_r} \Big\{ \sum_{\mathcal{H}} P(\mathcal{H}|\mathbf{O},\mathcal{M})\mathcal{L}(\mathcal{H},\mathcal{H}_r) \Big\} \tag{2.122}$$

The Levenshtein distance associated with the word error rate can be used as the loss function $\mathcal{L}(\mathcal{H},\mathcal{H}_r)$. The difference with MBR training criteria in section 2.3.4.1 is that in this case $\mathcal{H}_r$ is unknown, and thus searched from the set of possible hypotheses. The search over all hypotheses using the true expected loss in equation (2.122) is computationally very expensive. Therefore, MBR decoding is implemented using only a set of likely hypotheses represented in the form of lattices or N-best lists. A smaller set of hypothesis may be used as a search space compared to the set used for computing the expected loss, to make the search practical. The use of pinched lattices and confusion networks has also been investigated. There are several variations of MBR decoding based on an N-best list, lattices, or a confusion network.

In an N-best list based approach [167], the posterior probabilities of the hypotheses is approximated and the expected word error rate is computed using an N-best list. The hypothesis with the lowest expected word error is selected as the final hypothesis. The problem with this approach is that for a reasonable approximation of the posterior probability, the size of the N-best list should be large. However, this becomes computationally expensive as rescoring N hypotheses require computation of $\mathcal{O}(N^2)$. This can be dealt with to some extent by searching over fewer hypotheses. However, the posterior probability should be still approximated using a larger N-best list, otherwise the approximation becomes poor.

In a lattice-based approach [186], word posteriors are computed using the forward-backward algorithm and the best path through the word graph is directly searched using these word posteriors based on the accumulated score [186].

In confusion network (CN) decoding [110, 111], a linear graph structure called a confusion network is used for finding the best hypothesis. The confusion network is derived from the word lattices. The arc posteriors of word lattices are computed through a forward-backward pass and the arcs with low-posterior links are pruned. The links corresponding to the same word are then merged depending upon overlap, and the links corresponding to different words are clustered into confusion sets, iteratively. A sample confusion network obtained for the lattice shown in figure 2.8 is given in figure 2.9. The best hypothesis from the CN is obtained by selecting the word with highest posterior probably in each confusion set.

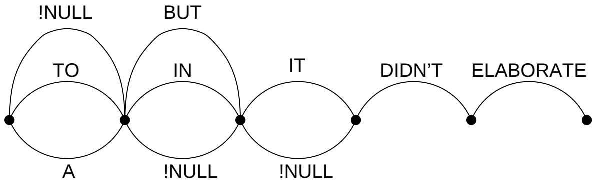


Figure 2.9: A confusion network (from [52])

2.6.3 Bayesian Inference

In the Bayesian framework as described in section 2.3.3, the HMM parameters are themselves random variables with probability distributions. A prior distribution is associated with the HMM parameters. The optimal Bayes solution for speech recognition is given by [78, 125]

$$\hat{\mathcal{H}} \;=\; \arg\max_{\mathcal{H}} \left\{ p(\mathbf{O}|\mathcal{H})P(\mathcal{H}) \right\} \tag{2.123}$$

where the likelihood is computed by marginalising over the model parameters as

$$\begin{aligned} p(\mathbf{O}|\mathcal{H}) &= \int_{\mathcal{M}} p(\mathbf{O}|\mathcal{H},\mathcal{M})p(\mathcal{M}|\Phi)\,d\mathcal{M} \\ &= \int_{\mathcal{M}} \left(\sum_{\boldsymbol{\psi}} P(\boldsymbol{\psi}|\mathcal{H},\mathcal{M}) \prod_t b_{\psi_t}(\mathbf{o}_t) \right) p(\mathcal{M}|\Phi)\,d\mathcal{M}. \end{aligned} \tag{2.124}$$

In the above equation, $p(\mathcal{M}|\Phi)$ is the prior distribution over model parameters. As discussed in section 2.3.3, the posterior distribution of the model parameters $p(\mathcal{M}|\mathbf{O}_{\mathtt{trn}}, \mathcal{H}_{\mathtt{trn}})$, estimated from the training data $\mathbf{O}_{\mathtt{trn}}$ and $\mathcal{H}_{\mathtt{trn}}$, is generally used as the prior during inference. The inference through equations (2.123) and (2.124) is also called Bayesian predictive classification (BPC) [78].

The acoustic score is required for doing inference through equation (2.123). However, the integral for the acoustic score or marginal likelihood in equation (2.124) is intractable, and thus some form of approximation is required for inference.

2.6.3.1 Markov Chain Monte-Carlo

One of the options to approximate the intractable marginal likelihood in equation (2.124) is to use Monte-Carlo methods. The simplest method is to generate random samples $\{\mathcal{M}_1, \ldots, \mathcal{M}_N\}$ from distribution $p(\mathcal{M}|\Phi)$ and use that to compute the approximate marginal likelihood as [76]

$$p(\mathbf{O}|\mathcal{H}) \approx \frac{1}{N} \sum_{i=1}^{N} p(\mathbf{O}|\mathcal{M}_i, \mathcal{H}) \tag{2.125}$$

where N is the number of samples. As the number of samples tends to infinity, it gives the true marginal likelihood. A double-fold Monte-Carlo simulation of HMM parameters and hidden state/component sequences can be also done [76]. The Monte-Carlo methods are computationally very expensive and not suitable for large-vocabulary speech recognition systems. A review of Monte-Carlo methods can be found in [108, 109, 126, 151].

2.6.3.2 Frame-Independence Assumption

The intractable marginal likelihood in equation (2.124) constrains the model parameters to be constant for all frames of the utterance. If the model parameters at each of the frames are assumed independent and allowed to vary from one frame to other, then the integration in equation (2.124) can be done at the frame level, rather than for the whole observation sequence [87, 88]. This is expressed as [88]

$$p(\mathbf{O}|\mathcal{H}) \quad \approx \quad \sum_{\boldsymbol{\psi}} P(\boldsymbol{\psi}|\mathcal{H}, \mathcal{M}) \prod_{t} \bar{b}_{\psi_t}(\mathbf{o}_t) \qquad (2.126)$$

where

$$\bar{b}_{\psi_t}(\mathbf{o}_t) = \int_{\mathcal{M}} b_{\psi_t}(\mathbf{o}_t) p(\mathcal{M}|\Phi) \, d\mathcal{M} \qquad (2.127)$$

Therefore, in this approach, each of the state/component output is marginalised for the associated model uncertainties and acts as a compensated distribution. Thus this method is also referred as Bayesian predictive model compensation [87, 88]. Given the appropriate form of prior for model parameters, the frame-level integration in equation (2.127) is tractable.

It should be noted that with the frame-independence assumption, the conditional independence assumption of HMMs is valid and Viterbi algorithm can be used. The difference is that instead of the original state output distribution, the predictive density in equation (2.127) is used. In [88], a modified frame-synchronous Viterbi algorithm with predictive density has been investigated.

2.6.3.3 Laplace Approximation

The Laplace approximation or normal approximation can be also used for approximating the marginal likelihood in equation (2.124). The approximated marginal likelihood is given by [76, 78, 169]

$$p(\mathbf{O}|\mathcal{H}) \approx p(\mathbf{O}|\hat{\mathcal{M}}_{\mathtt{map}}, \mathcal{H}) p(\hat{\mathcal{M}}_{\mathtt{map}}|\Phi)(2\pi)^{\frac{n}{2}} |\boldsymbol{\Sigma}_{\mathtt{map}}|^{\frac{1}{2}} \qquad (2.128)$$

where $\hat{\mathcal{M}}_{\mathtt{map}}$ is the MAP estimate given by

$$\hat{\mathcal{M}}_{\mathtt{map}} = \arg\max_{\mathcal{M}} \left\{ p(\mathbf{O}|\mathcal{M}, \mathcal{H}) p(\mathcal{M}|\Phi) \right\} \qquad (2.129)$$

In the above equations, n is the total number of parameters of $\mathcal{M}$, and $\boldsymbol{\Sigma}_{\texttt{map}} = (-\mathbf{V})^{-1}$, where $\mathbf{V}$ is the Hessian matrix of $\log\big(p(\mathbf{O}|\mathcal{M},\mathcal{H})p(\mathcal{M}|\Phi)\big)$ evaluated at $\mathcal{M} = \hat{\mathcal{M}}_{\texttt{map}}$. The Laplace method approximates the integrand in (2.124) with a Gaussian density at its mode matching its value, first derivative and second derivative. The mean of the Gaussian is the MAP estimate $\hat{\mathcal{M}}_{\texttt{map}}$ given in equation (2.129), and can be estimated using EM algorithm [57]. The covariance matrix for the Gaussian is related to the Hessian matrix $\mathbf{V}$, which is computationally expensive to find directly. Quasi-Bayesian approaches can be used with further approximations to approximate covariance matrices [75, 77], but the method is still computationally expensive.

2.6.3.4 Variational Bayes

The variational Bayes [7, 14, 15] approach provides a lower bound to the marginal likelihood in equation (2.124) which can be used in equation (2.123) to compute inference evidence, provided the bound is tight. The lower-bound to the marginal likelihood is computed by introducing a joint variational distribution $q(\boldsymbol{\theta},\mathcal{M})$ and applying Jensen's inequality as

$$\begin{aligned}\log p(\mathbf{O}|\mathcal{H}) &= \log \int_{\mathcal{M}} \sum_{\boldsymbol{\theta}} p(\mathbf{O},\boldsymbol{\theta}|\mathcal{H},\mathcal{M})p(\mathcal{M})\, d\mathcal{M} \\ &= \log \int_{\mathcal{M}} \sum_{\boldsymbol{\theta}} q(\boldsymbol{\theta},\mathcal{M})\frac{p(\mathbf{O},\boldsymbol{\theta}|\mathcal{H},\mathcal{M})p(\mathcal{M})}{q(\boldsymbol{\theta},\mathcal{M})}\, d\mathcal{M} \\ &\geq \int_{\mathcal{M}} \sum_{\boldsymbol{\theta}} q(\boldsymbol{\theta},\mathcal{M}) \log \frac{p(\mathbf{O},\boldsymbol{\theta}|\mathcal{H},\mathcal{M})p(\mathcal{M})}{q(\boldsymbol{\theta},\mathcal{M})}\, d\mathcal{M} \end{aligned} \tag{2.130}$$

Maximising this lower-bound with respect to $q(\boldsymbol{\theta},\mathcal{M})$ gives $q(\boldsymbol{\theta},\mathcal{M}) = p(\boldsymbol{\theta},\mathcal{M}|\mathbf{O})$, which turns the above inequality into an equality. However, evaluating the true posterior $p(\boldsymbol{\theta},\mathcal{M}|\mathbf{O})$ involves computing the marginal likelihood for the normalisation constant, which is intractable. Therefore, the key idea in variational Bayes approximation is to use the factored approximation

$$q(\boldsymbol{\theta},\mathcal{M}) = q(\boldsymbol{\theta})q(\mathcal{M}) \tag{2.131}$$

such that the the above bound is expressed as

$$\log p(\mathbf{O}|\mathcal{H}) \geq \int_{\mathcal{M}} \sum_{\boldsymbol{\theta}} q(\boldsymbol{\theta})q(\mathcal{M}) \log \frac{p(\mathbf{O},\boldsymbol{\theta}|\mathcal{H},\mathcal{M})p(\mathcal{M})}{q(\boldsymbol{\theta})q(\mathcal{M})}\, d\mathcal{M} \tag{2.132}$$

The variational Bayes algorithm iteratively maximises the lower-bound in equation (2.132) with respect to $q(\boldsymbol{\theta})$ and $q(\mathcal{M})$ in an interleaved fashion. Using variational calculus, the update equations can be derived as [7, 15]

$$q_{k+1}(\boldsymbol{\theta}) \propto \exp\left(\int_{\mathcal{M}} \log p(\boldsymbol{\theta}, \mathbf{O}|\mathcal{M}) q_k(\mathcal{M})\, d\mathcal{M}\right) \tag{2.133}$$

$$q_{k+1}(\mathcal{M}) \propto p(\mathcal{M}) \exp\left(\sum_{\boldsymbol{\theta}} \log p(\boldsymbol{\theta}, \mathbf{O}|\mathcal{M}) q_{k+1}(\boldsymbol{\theta})\right) \tag{2.134}$$

where subscripts k and $k+1$ represent iteration numbers. The maximisation of the lower-bound in (2.132) is equivalent to approximating true posterior as well as obtaining the tightest lower-bound to the marginal likelihood [7]. When the estimated distributions converge, they are used to compute the final lower-bound to the marginal likelihood. The method is computationally efficient and fast, however not as accurate as Markov chain Monte-Carlo (MCMC) and often inferior to the Laplace method as well [119, 120]. One of the techniques that has been reported to produce more accurate approximations is expectation propagation [119]. The method will be applied for doing inference in adaptive speech recognition in chapter 5, and is described in section 5.3 and appendix A in detail.

2.6.4 Multipass Decoding and System Combination

State-of-the-art systems generally use a multipass strategy to refine the search space and output hypotheses in several stages [40]. The decoding is performed over the test data successively by more complex models. A typical multipass system may involve a fast first pass using a simple speaker independent model to obtain the 1-best transcript for adaptation. The second pass uses this transcript to adapt models and redecode the data with a bigram or trigram language model to generate word lattices. This may be followed by rescoring of lattices with more complex language models (4-gram or 5-gram), or different sets of models and adaptation schemes to obtain a number of candidate outputs. Some of the examples of multipass systems include [54, 55].

A speech recognition system generally makes different errors in recognising the same speech compared to other systems with different models and adaptation schemes [40]. The outputs from different systems are thus combined in state-of-the-art transcription systems to obtain a better recognition hypothesis. This is usually done by using recogniser output voting error reduction (ROVER) [35] or confusion network combination (CNC) [31]. Also, the use of N-best lists or lattices generated from another system and cross-adaption by using the adaptation transcript generated from another system also provide an implicit method of system combination.

2.7 Evaluating ASR

The performance of speech recognition systems is usually evaluated in terms of the word error rate of the hypotheses. An optimal string matching, generally using dynamic programming, is performed between the reference transcript and the generated hypothesis. This is based on Levenshtein distance, and assigns scores for insertion, deletion and substitution errors with respect to the reference transcript. In HTK [196], scores of 7, 7 and 10 are used for insertions, deletions and substitutions, whereas NIST scoring [34] uses scores of 3, 3, and 4, respectively. Once the total number for substitution (S), deletion (D) and insertion (I) errors are computed through the optimal alignment, the word error rate (WER) is given as

$$WER\% = \frac{S + D + I}{N} \times 100\% \tag{2.135}$$

where N is the total number of words in the reference transcript. Sometimes, the change in WER obtained may be small between two systems. In such cases, a matched-pair significant test can be used to check the statistical significance of performance differences [34, 59].

2.8 Summary

This chapter has reviewed HMM-based automatic speech recognition systems, with a detailed description of each module. The speech signal captured from the microphone is first converted into a sequence of speech features, usually MFCC or PLP. The features are usually normalised by applying CMN, CVN, Gaussianisation, VTLN or LDA transforms, to make them robust to speaker or environmental variations. The acoustic variabilities in speech features are modelled through HMMs, by using training samples of speech. The HMMs can be trained using the ML criterion with the Baum-Welch algorithm. However, the ML training has certain limitations, and to deal with them, Bayesian approaches and discriminative training criteria can be used. The HMMs can be trained using a discriminative criterion such as minimum phone error (MPE) using a weak-sense auxiliary function. The selection of suitable acoustic units and parameters tying are also described. The acoustic model is used in conjunction with a language model and a lexicon for decoding of test speech. A word N-gram model trained from linguistic text corpora is commonly used as the language model. The ML estimation of language models is also described, along with the other techniques to obtain robust estimates of language model probabilities. The best hypothesis for a given test speech can be found by MAP or MBR decoding. Several search techniques including the Viterbi algorithm are discussed to decode speech and find the best hypothesis. The performance of a speech recognition system is usually given in terms of word error rate, by comparing the generated hypothesis to the reference hypothesis.

Chapter 3
Adaptation and Adaptive Training

In speech recognition systems, there may be a mismatch between training and test acoustic conditions that degrades the performance [40]. Therefore, the trained acoustic models are adapted to the test speaker or acoustic condition to obtain a better performance [189]. Moreover, large vocabulary speech recognition systems are usually trained on a large corpus of speech data collected from several speakers and different recording conditions. The speech signal does not contain only the relevant acoustic information required for speech recognition but also unwanted variations from speakers and the environment. The training data is thus non-homogeneous in nature, and the acoustic models need to be trained and extracted from such data. This chapter describes the techniques to adapt acoustic models to a test speaker, and also the training of acoustic models from non-homogeneous data. Several speaker adaptation techniques [189] are described to reduce the mismatch between training and test acoustic conditions. Thereafter, an adaptive training framework [5] is described that models speech and non-speech variabilities in the non-homogeneous training data separately. In this chapter, the maximum likelihood criterion is used for adaptation and adaptive training. The discriminative adaptation and adaptive training schemes are separately described in the next chapter.

3.1 Speaker Adaptation

A significant difference is observed in speech due to its dynamic and versatile nature when even the same words are uttered by different speakers. These variations may result from the speaker's voice, age, gender, dialect, intonation, speaking rate and style [73]. In addition, the background noise, different microphones, transmission channel, and noise-induced stress also introduce variations in the speech even from the same speaker uttering same words under different conditions [73].

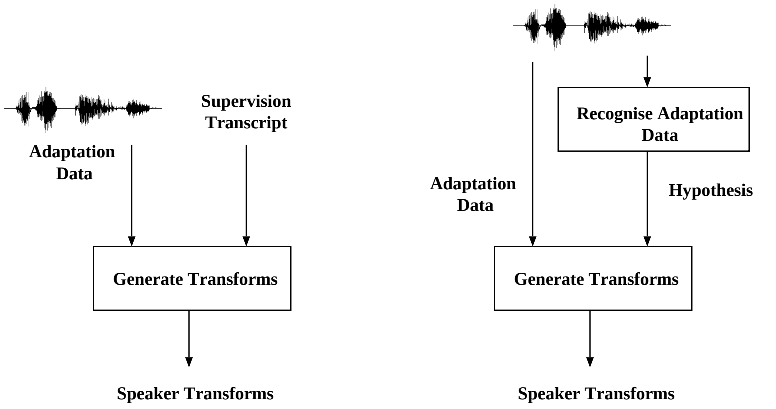
The performance of speech recognition systems may be severely degraded when there is a mismatch between the training and the testing speakers and acoustic conditions [74, 96]. The performance can be improved by reducing this mismatch between the trained models and the test condition. For example, it has been found that a speaker dependent (SD) system trained from data of a specific speaker performs much better than a speaker-independent (SI) system trained on the same amount of data but from different speakers [189]. However, it is not possible to build large speaker-dependent systems due to lack of training data. Therefore, in practice, an effort is made to either reduce the speaker or environmental dependent variations in speech, or to adapt the trained models to the specific test condition. The first one is commonly referred as *speaker or environmental normalisation*[1], and attempts to model inherent variabilities in speech by removing or reducing the speaker and environment induced variabilities. The later technique transforms the trained models to the target test condition so that the transformed model represents the test condition. This is usually called *speaker adaptation* [189] and is of interest in this work.

A small amount of data from the target speaker is generally used to adapt the trained acoustic models to the target speaker. This data is called adaptation data. Adaptation can be performed in different modes, depending upon the availability of the transcript for the adaptation data, and the time when the adaptation data becomes available [204].

- **Supervised and Unsupervised Adaptation:** In a supervised mode of adaptation, the transcript corresponding to the adaptation data is known. On the other hand, in unsupervised adaptation, the correct supervision transcript for the adaptation data is not given. In this case, the supervision transcript is generated by decoding the adaptation data with the available acoustic model. The quality of adaptation depends both on the amount of the adaptation data as well as the quality of the generated supervision hypothesis which may contain several errors. If the test data itself is used

[1] Cepstral mean and variance normalisation, Gaussianisation, and vocal tract length normalisation described in the last chapter are examples of the speaker or environmental normalisation.

for adaptation as well, the method is also called *self-adaptation* [43]. The supervised and unsupervised mode of adaptation is also illustrated in figure 3.1.



(a) Supervised adaptation (b) Unsupervised adaptation

Figure 3.1: Supervised and unsupervised adaptation

- **Offline and Online Adaptation:** In an offline mode of adaptation, all adaptation data is assumed to be available at once, before the adaptation and recognition process starts. This is also referred as a *static* or *batch* mode of adaptation. On the other hand, in an online mode, adaptation is performed as soon as the adaptation data becomes available, and the adaptation data becomes available in stages. This is also referred to as *rapid* or *instantaneous* adaptation. The online adaptation can be also done in an *incremental* fashion, in which adaptation information is propagated from one stage to another for effective adaptation [204].

A number of techniques have been developed over the years to adapt HMM parameters to a target speaker. They include maximum-a-posteriori adaptation, linear transforms and speaker-cluster based adaptation techniques [189]. Some of them are described in the next section.

3.1.1 Maximum a Posteriori (MAP) Adaptation

A straightforward way to adapt the models given the adaptation data would be to retrain the model using the ML criterion. However, as the amount of adaptation data is usually small, this leads to the overtraining of HMMs that would not generalise. Therefore, a maximum-a-posteriori (MAP) approach [57] was proposed in which model parameters are viewed as random variables. The MAP estimate of the adapted model parameters are obtained by maximising the posterior distribution of HMM parameters as

$$\mathcal{M}_{\texttt{map}} = \arg\max_{\mathcal{M}} \Big\{ p(\mathcal{M}|\mathbf{O},\mathcal{H}) \Big\} = \arg\max_{\mathcal{M}} \Big\{ p(\mathbf{O}|\mathcal{M},\mathcal{H}) p(\mathcal{M}|\Phi) \Big\} \tag{3.1}$$

where $\mathbf{O}$ and $\mathcal{H}$ are the adaptation data and the corresponding supervision transcript, and $p(\mathcal{M}|\Phi)$ is a prior distribution over HMM parameters with hyperparameters Φ. The prior term $p(\mathcal{M}|\Phi)$ prevents the HMM parameters being overtrained on the supervision data.

The MAP estimates of model parameters in equation (3.1) are obtained by defining an auxiliary function and using the EM algorithm. The auxiliary function for the MAP estimation can be obtained by adding a prior term to the ML auxiliary function in equation (2.46). The MAP auxiliary function is expressed as

$$\mathcal{Q}(\hat{\mathcal{M}};\mathcal{M}) = -\frac{1}{2}\sum_{tm}\gamma_m(t)\Big\{\log|\hat{\boldsymbol{\Sigma}}_m| + (\mathbf{o}_t - \hat{\boldsymbol{\mu}}_m)^{\mathsf{T}}\hat{\boldsymbol{\Sigma}}_m^{-1}(\mathbf{o}_t - \hat{\boldsymbol{\mu}}_m)\Big\} + \log p(\hat{\mathcal{M}}|\Phi) \tag{3.2}$$

where $\hat{\mathcal{M}}$ is the new estimate of the model parameters, and $\gamma_m(t)$ is the ML posterior occupancy of component m at time t computed using current model parameters $\mathcal{M}$.

An important issue for MAP estimation is the choice of the prior distribution. A closed form solution for the MAP estimation can be obtained if a conjugate prior to the likelihood is chosen as the prior distribution. However, for HMMs with GMMs as the state output distribution, a finite-dimensional conjugate prior does not exist. In [57], parameters of mixture components are assumed independent of the component weights, giving the joint conjugate prior $p(\hat{\mathcal{M}}|\Phi)$ to the likelihood of complete data as the product of Dirichlet and normal-Wishart distributions. The form of the prior for individual Gaussian component is shown in equation 2.54. In this case, the MAP estimate of a mean vector is given by

$$\hat{\boldsymbol{\mu}}_m = \frac{\sum_t \gamma_m(t)\mathbf{o}_t}{\sum_t \gamma_m(t) + \tau} + \frac{\tau}{\sum_t \gamma_m(t) + \tau}\tilde{\boldsymbol{\mu}}_m \tag{3.3}$$

where $\tilde{\boldsymbol{\mu}}_m$ is the mean of the prior $p(\hat{\mathcal{M}}|\Phi)$ and the scaling factor τ controls the balance between the ML estimate and the prior. It can be observed that when only a small amount of data is available for adaptation, the MAP estimate is closer to the prior. As additional data becomes available, the MAP estimate tends towards the ML estimate.

A major drawback of the MAP estimation is that only the models whose speech units are observed in the adaptation data can be adapted. State-of-the-art speech recognition systems usually have many thousands of Gaussians, and a large number of components will not be adapted as they are unseen in the adaptation data. Several methods including regression model prediction [2, 3] and structured MAP [160] have been proposed to overcome this limitation.

3.1.2 Linear Transforms

In this approach, linear transforms are used to adapt the means and/or covariance matrix of Gaussian components of HMM output probability distributions to obtain a better representation of the target speaker. Several Gaussians can share the same transform and thus the method is effective for a small amount of adaptation data as well.

The transforms can have several forms: a diagonal [94], a full [69, 103, 104] or a block-diagonal matrix [29]. Generally, a bias term is also used in the transform. The choice of the particular form and the number of transforms to be generated depends upon the amount of adaptation data available [49, 50]. A diagonal transform can be robustly estimated from a comparatively small amount of adaptation data. However, using full matrices can lead to powerful transforms when they can be robustly estimated. A block-diagonal transform is an intermediate form that transforms the parameters in blocks. For example, separate transforms can be used for each blocks of parameters corresponding to static and dynamic coefficients. A form of block-diagonal transform using separate transforms for static, first derivative and second derivative parameter blocks can be expressed as

$$\begin{pmatrix} \mathbf{A} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}^{(\Delta)} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{A}^{(\Delta^2)} \\ & & \end{pmatrix} \tag{3.4}$$

where $\mathbf{A}$, $\mathbf{A}^{(\Delta)}$ and $\mathbf{A}^{(\Delta^2)}$ are full matrix transforms for blocks corresponding to static, first derivative and second derivative coefficients, respectively, and $\mathbf{0}$ is a null matrix.

Some of the popular forms of linear transforms used in model adaptation are described below.

3.1.2.1 Mean MLLR

In a maximum likelihood linear regression (MLLR) [103], the mean vector of the mth Gaussian component is adapted as

$$\hat{\boldsymbol{\mu}}_m = \mathbf{A}\boldsymbol{\mu}_m + \mathbf{b} = \mathbf{W}\boldsymbol{\xi}_m \tag{3.5}$$

where $\hat{\boldsymbol{\mu}}_m$ represents the adapted mean vector, $\boldsymbol{\xi}_m = [\boldsymbol{\mu}_m^{\mathsf{T}} \;\; 1]^{\mathsf{T}}$ is the extended mean vector, and $\mathbf{W} = [\mathbf{A} \;\; \mathbf{b}]$ is an affine linear transform. MLLR transforms are estimated by maximising the likelihood of adaptation data as

$$\hat{\mathbf{W}} = \arg\max_{\mathbf{W}} \left\{ \log p(\mathbf{O}|\mathcal{H}; \mathbf{W}, \mathcal{M}) \right\} \tag{3.6}$$

The ML objective function is maximised using the EM algorithm by defining an auxiliary function as [103]

$$\mathcal{Q}(\hat{\mathbf{W}};\mathbf{W},\mathcal{M}) \quad = \quad \tilde{K} + \sum_{mt} \gamma_m^{\mathtt{ml}}(t) \log \mathcal{N}(\mathbf{o}_t;\hat{\mathbf{W}}\boldsymbol{\xi}_m,\boldsymbol{\Sigma}_m) \tag{3.7}$$

where $\gamma_m^{\mathtt{ml}}(t)$ is the occupation probability for component m at time t computed using the current model $\mathcal{M}$ and current estimate of the transform $\mathbf{W}$, and $\tilde{K}$ includes constant terms independent of $\hat{\mathbf{W}}$. The above auxiliary function can be re-expressed ignoring the constant term as

$$\mathcal{Q}(\hat{\mathbf{W}};\mathbf{W},\mathcal{M}) = -\frac{1}{2}\sum_{tm}\gamma_m^{\mathtt{ml}}(t)\left(\mathbf{o}_t - \hat{\mathbf{W}}\boldsymbol{\xi}_m\right)^{\mathtt{T}}\boldsymbol{\Sigma}_m^{-1}\left(\mathbf{o}_t - \hat{\mathbf{W}}\boldsymbol{\xi}_m\right) \tag{3.8}$$

The above auxiliary function is similar to the standard ML auxiliary function in equation (2.46) used to estimate component parameters, except that adapted means are used, and $\gamma_m^{\mathtt{ml}}(t)$ is now computed using current estimate of transform and model parameters. Assuming covariance matrices to be diagonal, the ML estimate of the dth row of transform $\hat{\mathbf{w}}_d$ is given by

$$\hat{\mathbf{w}}_d = \left(\mathbf{G}_d^{\mathtt{ml}}\right)^{-1}\mathbf{k}_d^{\mathtt{ml}} \tag{3.9}$$

where for the dth row of the transform, the sufficient statistics are given as

$$\mathbf{G}_d^{\mathtt{ml}} \quad = \quad \sum_{tm}\frac{\gamma_m^{\mathtt{ml}}(t)}{\sigma_{md}^2}\boldsymbol{\xi}_m\boldsymbol{\xi}_m^{\mathtt{T}} \tag{3.10}$$

$$\mathbf{k}_d^{\mathtt{ml}} \quad = \quad \sum_{tm}\frac{\gamma_m^{\mathtt{ml}}(t)o_{td}}{\sigma_{md}^2}\boldsymbol{\xi}_m \tag{3.11}$$

In the above equations, o_{td} represents the dth element of observation vector $\mathbf{o}_t$, and σ_{md}^2 is the dth diagonal element of $\boldsymbol{\Sigma}_m$.

3.1.2.2 Variance MLLR

In a variance MLLR [44, 51], the covariance matrix of the mth component is adapted as

$$\hat{\boldsymbol{\Sigma}}_m = \mathbf{L}_m^{\mathtt{T}}\mathbf{H}\mathbf{L}_m \tag{3.12}$$

where $\mathbf{H}$ is a linear transform, $\mathbf{L}_m$ is the inverse of the Choleski factor of $\boldsymbol{\Sigma}_m^{-1}$ (i.e. $\mathbf{L}_m = \mathbf{C}_m^{-1}$, where $\boldsymbol{\Sigma}_m^{-1} = \mathbf{C}_m\mathbf{C}_m^{\mathtt{T}}$). Due to the high computation cost involved with this form, an alternative form was proposed in [44] as

$$\hat{\boldsymbol{\Sigma}}_m = \mathbf{H}\boldsymbol{\Sigma}_m\mathbf{H}^{\mathtt{T}} \tag{3.13}$$

In this form, the likelihood can be expressed with modified mean vectors and observations, as

$$\log \mathcal{N}(\mathbf{o}_t; \boldsymbol{\mu}_m, \hat{\boldsymbol{\Sigma}}_m) = \log \mathcal{N}\left(\mathbf{H}^{-1}\mathbf{o}_t; \mathbf{H}^{-1}\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m\right) - \log|\mathbf{H}| \tag{3.14}$$

This leads to the efficient computation of likelihoods for diagonal covariance matrices, simply by modifying means and observations. The transform estimation formulae for the variance adaptation are given in [44].

3.1.2.3 Constrained MLLR

In constrained MLLR [29, 44], both the mean vector and the covariance matrix of Gaussian components are adapted by using a linear transform, which is constrained to be the same in both cases, as

$$\hat{\boldsymbol{\mu}}_m = \tilde{\mathbf{A}}\boldsymbol{\mu}_m - \tilde{\mathbf{b}} \tag{3.15}$$

$$\hat{\boldsymbol{\Sigma}}_m = \tilde{\mathbf{A}}\boldsymbol{\Sigma}_m\tilde{\mathbf{A}}^{\mathtt{T}} \tag{3.16}$$

where $\tilde{\mathbf{A}}$ represents a constrained linear transform and $\tilde{\mathbf{b}}$ is a bias on the mean vector. The constrained MLLR in the model domain can be equivalently applied in the feature space, which is computationally more efficient. This equivalence can be written as

$$\log \mathcal{N}(\mathbf{o}_t; \hat{\boldsymbol{\mu}}_m, \hat{\boldsymbol{\Sigma}}_m) = \log \mathcal{N}\left(\hat{\mathbf{o}}_t; \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m\right) + \log|\mathbf{A}| \tag{3.17}$$

which gives

$$\hat{\mathbf{o}}_t = \tilde{\mathbf{A}}^{-1}\mathbf{o}_t + \tilde{\mathbf{A}}^{-1}\tilde{\mathbf{b}} = \mathbf{A}\mathbf{o}_t + \mathbf{b} = \mathbf{W}\boldsymbol{\zeta}_t \tag{3.18}$$

where $\boldsymbol{\zeta}_t$ is the extended observation vector $[\mathbf{o}_t^{\mathtt{T}} \quad 1]^{\mathtt{T}}$. The full CMLLR transform in feature space can be estimated using the EM algorithm by defining an auxiliary function as [44]

$$\mathcal{Q}(\hat{\mathbf{W}}; \mathbf{W}, \mathcal{M}) = -\frac{1}{2}\sum_{tm}\gamma_m^{\mathtt{ml}}(t)\left\{\left(\hat{\mathbf{W}}\boldsymbol{\zeta}_t - \boldsymbol{\mu}_m\right)^{\mathtt{T}}\boldsymbol{\Sigma}_m^{-1}\left(\hat{\mathbf{W}}\boldsymbol{\zeta}_t - \boldsymbol{\mu}_m\right) - \log\left(|\hat{\mathbf{A}}|^2\right)\right\} \tag{3.19}$$

Given sufficient statistics [44]

$$\mathbf{G}_d^{\mathtt{ml}} = \sum_m \frac{1}{\sigma_{md}^2}\sum_t \gamma_m^{\mathtt{ml}}(t)\boldsymbol{\zeta}_t\boldsymbol{\zeta}_t^{\mathtt{T}} \tag{3.20}$$

$$\mathbf{k}_d^{\mathtt{ml}} = \sum_m \frac{\mu_{md}}{\sigma_{md}^2}\sum_t \gamma_m^{\mathtt{ml}}(t)\boldsymbol{\zeta}_t \tag{3.21}$$

where σ_{md}^2 is the dth diagonal element of covariance matrix $\boldsymbol{\Sigma}_m$ and μ_{md} is the dth element of $\boldsymbol{\mu}_m$, the dth row of the transform $\hat{\mathbf{w}}_d$ can be estimated by

$$\hat{\mathbf{w}}_d = \left(\mathbf{G}_d^{\mathtt{ml}}\right)^{-1}\left(\alpha\mathbf{p}_d + \mathbf{k}_d^{\mathtt{ml}}\right) \tag{3.22}$$

In the above equation, $\mathbf{p}_d$ is the extended cofactor vector $[c_{d1} \ \dots \ c_{dD} \ 0]^{\mathtt{T}}$, with cofactor $c_{ij} = \mathrm{cof}(\mathbf{A}_{ij})$, and α satisfies the quadratic expression given as

$$\alpha^2 \mathbf{p}_d^{\mathtt{T}} \left(\mathbf{G}_d^{\mathtt{ml}}\right)^{-1} \mathbf{p}_d + \alpha \mathbf{p}_d^{\mathtt{T}} \left(\mathbf{G}_d^{\mathtt{ml}}\right)^{-1} \mathbf{k}_d^{\mathtt{ml}} - \beta = 0 \tag{3.23}$$

where $\beta = \sum_{tm} \gamma_m^{\mathtt{ml}}(t)$ is the total occupancy. This leads to an iterative solution over the rows, as the estimation of the one row of the transform is dependent upon all other rows through the cofactors.

3.1.2.4 MAP Linear Regression (MAPLR)

The linear transforms are quite effective for adaptation however they may have unreliable estimates when the amount of adaptation data is very small. This may distort the underlying structure of the acoustic space [21]. Therefore, a prior distribution over the transforms can be introduced as a constraint similar to MAP adaptation of model parameter in section 3.1.1. This is called MAP linear regression (MAPLR) [21, 164]. The MAPLR transform $\hat{\mathbf{W}}$ is estimated as[1]

$$\hat{\mathbf{W}} = \arg\max_{\mathbf{W}} \left\{ p(\mathbf{O}|\mathcal{H}; \mathbf{W}, \mathcal{M}) p(\mathbf{W}|\phi) \right\} \tag{3.24}$$

where $p(\mathbf{W}|\phi)$ is the prior over transform with hyperparameters ϕ. The above MAP objective function is optimised using the EM algorithm by defining an auxiliary function. The auxiliary function for MAPLR estimation can be obtained by adding the prior term to the auxiliary function for ML transforms given in equation (3.7). This is expressed as

$$\mathcal{Q}(\hat{\mathbf{W}}; \mathbf{W}, \mathcal{M}) \quad = \quad \sum_{mt} \gamma_m^{\mathtt{ml}}(t) \log p(\mathbf{o}_t | \hat{\mathbf{W}}, \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m) + \log p(\hat{\mathbf{W}}|\phi) \tag{3.25}$$

The choice of the form of prior and its estimation is an important consideration in MAP estimation. As noted earlier, if a conjugate prior is chosen, the MAP optimisation problem can be greatly simplified. In [21], a matrix variate normal prior is used for the mean transform. This is expressed as

$$p(\mathbf{W}|\phi) = \frac{1}{\sqrt{(2\pi)^{D(D+1)} \, |\boldsymbol{\Omega}^{\mathtt{W}}|^{D} \, |\boldsymbol{\Sigma}^{\mathtt{W}}|^{(D+1)}}} \exp\left(-\frac{1}{2} \mathrm{tr}\left(\boldsymbol{\Omega}^{\mathtt{W}\,-1} (\mathbf{W} - \mathbf{M}^{\mathtt{W}})^{\mathtt{T}} \boldsymbol{\Sigma}^{\mathtt{W}\,-1} (\mathbf{W} - \mathbf{M}^{\mathtt{W}})\right)\right) \tag{3.26}$$

where $\mathbf{M}^{\mathtt{W}}$ is the mean matrix of the transform $\mathbf{W}$ and

$$\boldsymbol{\Sigma} \quad = \quad \mathcal{E}\left\{ (\mathbf{W} - \mathbf{M}^{\mathtt{W}})(\mathbf{W} - \mathbf{M}^{\mathtt{W}})^{\mathtt{T}} \right\} \tag{3.27}$$

$$\boldsymbol{\Omega} \quad = \quad \mathcal{E}\left\{ (\mathbf{W} - \mathbf{M}^{\mathtt{W}})^{\mathtt{T}} (\mathbf{W} - \mathbf{M}^{\mathtt{W}}) \right\} / K \tag{3.28}$$

[1] A prior scaling factor $\alpha^{\mathtt{P}}$ may be used in practice to control the contribution of the transform prior.

In the above equation, K is a constant that ensures appropriate power normalisation. This distribution is closely related to multivariate normal distribution as

$$\mathrm{vec}(\mathbf{W}) \sim \mathcal{N}\left(\mathrm{vec}(\mathbf{W}); \mathrm{vec}(\mathbf{M}^{\mathtt{W}}), \boldsymbol{\Omega}^{\mathtt{W}} \otimes \boldsymbol{\Sigma}^{\mathtt{W}}\right) \tag{3.29}$$

where $\otimes$ is the Kronecker product. In this work, the rows of transforms are assumed independent to be consistent with the diagonal covariance matrices of the HMM components, and a Gaussian prior is imposed over the transform as

$$p(\mathbf{W}|\phi) = \prod_{d=1}^{D} \mathcal{N}(\mathbf{w}_d; \boldsymbol{\mu}_d^{\mathtt{W}}, \boldsymbol{\Sigma}_d^{\mathtt{W}}) \tag{3.30}$$

The hyperparameters of the transform prior $\phi = \{\boldsymbol{\mu}_d^{\mathtt{W}}, \boldsymbol{\Sigma}_d^{\mathtt{W}}; 1 \leq d \leq D\}$ are estimated using an empirical Bayesian approach using a set of transforms. The set of transforms is obtained from the training data set speakers in this work.

The auxiliary function in equation (3.25) with this prior for MLLR transforms leads to sufficient statistics for dth row of the transform as

$$\mathbf{G}_d^{\mathtt{map}} = \sum_{tm} \frac{\gamma_m^{\mathtt{ml}}(t)}{\sigma_{md}^2} \boldsymbol{\xi}_m \boldsymbol{\xi}_m^{\mathtt{T}} + \boldsymbol{\Sigma}_d^{\mathtt{W}\,-1} \tag{3.31}$$

$$\mathbf{k}_d^{\mathtt{map}} = \sum_{tm} \frac{\gamma_m^{\mathtt{ml}}(t) o_{td}}{\sigma_{md}^2} \boldsymbol{\xi}_m + \boldsymbol{\Sigma}_d^{\mathtt{W}\,-1} \boldsymbol{\mu}_d^{\mathtt{W}} \tag{3.32}$$

This yields the MAPLR transform for the dth row as

$$\hat{\mathbf{w}}_d = \left(\mathbf{G}_d^{\mathtt{map}}\right)^{-1} \mathbf{k}_d^{\mathtt{map}} \tag{3.33}$$

It should be noted that for CMLLR transforms, the likelihood computation in equation (3.17) involves $|\mathbf{A}|$ and it is difficult to find a conjugate prior for it. Therefore, constrained transforms are not investigated further in the Bayesian framework in this work.

3.1.3 Cluster Based Adaptation

The methods described above are based on a standard set of HMMs and do not explicitly use information about characteristics of an HMM set for particular speakers. An alternative approach is to perform adaptation on a number of HMM sets corresponding to different speaker groups or clusters [189]. One of the simplest examples is the use of gender dependent models. In the traditional cluster based approach [37, 95], several cluster-dependent models are built, and the appropriate one is chosen for a particular speaker during recognition. Instead of such a hard assignment to clusters, the adapted model for a particular speaker can be obtained by a linear combination of a set of cluster-dependent models [41, 45, 97]. In these methods, a set of cluster-dependent models need to be estimated first, and is thus related to training of multiple cluster HMMs. This will be described in section 3.2.2.

3.1.4 Regression Classes

As more adaptation data become available, the adaptation can be improved by increasing the number of transforms, rather than using a single global transform. A *regression class tree* [49, 102] is often used to cluster the components into different hierarchical groups so that similar components can be transformed in a similar way. A transform is generated for each group (node) of components with sufficient adaptation data, rather than using an identical global transform for all components. In this case, rather than accumulating statistics over all components, statistics for Gaussian components within each base class are accumulated separately. Then a transform is estimated for each of the base classes using corresponding statistics, which can adapt the parameters more effectively.

An example regression class tree based on expert knowledge is shown in figure 3.2. The regression class tree has five terminal nodes (base classes), and if there is a sufficient amount of data associated with each of them, five different transforms are generated. This is determined by comparing the occupation counts for each node to a predefined threshold value. When there is not a sufficient amount of data associated with a node, the data from sibling nodes are pooled into the parent node, and a transform is generated for it, provided the data becomes sufficient. The transform generated for a parent node is used for its children nodes with an insufficient amount of adaptation data. In the example regression class tree shown, the data for the unvoiced component group, marked by a dotted circle, is not sufficient, and a transform will not be generated for it. Rather the transform for the consonant group will be used for the unvoiced components, which is estimated by pooling the data from the voiced and unvoiced constant groups.

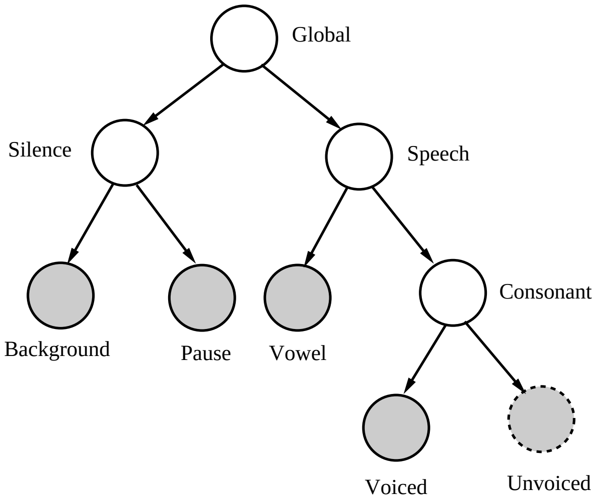


Figure 3.2: A regression class tree for adaptation transforms

In practice, the regression class trees are generally built by automatically clustering the Gaussian components which are close in the acoustic space [49, 161]. This can be obtained through k-means clustering using the Kullback-Leibler distance measure [161, 184] or using a centroid-splitting algorithm with a Euclidean distance [196].

The regression class tree provides an elegant way to scale the number of transforms generated to the available adaptation data, and is widely used in state-of-the-art systems.

3.1.5 Extensions of Standard Techniques

The linear transform based adaptation schemes described above are widely used in large vocabulary speech recognition systems. In unsupervised adaption, the transcript for the adaptation data is not known and is usually generated using SI models. The generated supervision hypotheses may contain several errors, and transforms cannot be reliably estimated by using such supervision hypotheses. The performance of the speech recognition system may degrade due to over-tuning to the erroneous supervision hypothesis. One way to improve the supervision hypothesis is to iteratively refine it, using multiple iterations of decoding and adaptation [70, 194]. In the iterative MLLR [194], estimated MLLR transforms are used to adapt the models and regenerate the supervision hypothesis, which are then used for re-estimating the transforms. The process is also illustrated in figure 3.3, and can be compared to the unsupervised adaptation process in figure 3.1.

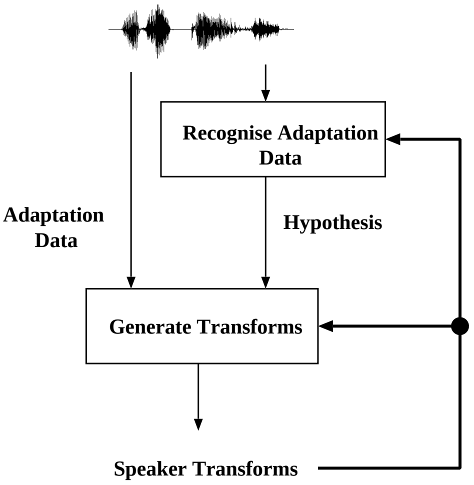


Figure 3.3: Iterative MLLR

A number of other techniques have been investigated as well, to deal with the problem of erroneous supervision hypothesis. Some of them are described below.

3.1.5.1 Confidence Based Adaptation

In confidence score based adaptation [4, 31, 70, 165, 172, 205], a confidence score associated with each word of the supervision hypothesis is used to judge and determine the high quality adaptation data. In this approach, a confidence score is computed for each word of the supervision hypothesis, and the adaptation data corresponding to words with high confidence scores only is used to accumulate the statistics to generate the transforms. The words with high confidence scores are assumed less prone to errors. The adaptation data corresponding to words with confidence scores below a threshold is discarded. The word posterior probabilities from decoding run can be used as confidence scores [31]. The word-level posterior probabilities are computed from a word lattice containing the acoustic and language model likelihoods as well as start and end times for words. A forward-backward algorithm is first used to compute lattice arc posterior probabilities

$$P(l|\mathbf{O}) = \frac{\sum_{q(l,\mathcal{W}))} p^{1/\kappa}(\mathbf{O}|q(l,\mathcal{W}))P(\mathcal{W})}{p(\mathbf{O})} \tag{3.34}$$

where $q(l,\mathcal{W}))$ is the path through the arc l that corresponds to word $\mathcal{W}$, and κ is an acoustic score scaling factor. In the above equation, $p(\mathbf{O}|q(l,\mathcal{W}))$ is the likelihood of path $q(l,\mathcal{W})$, $P(\mathcal{W})$ is LM probability, and $p(\mathbf{O})$ is the data likelihood approximated by summing over all the paths through the lattice. The arc posteriors corresponding to the same word at a given time are summed to obtain time-dependent word posteriors. The final word posterior probability of a word for particular start and end times is taken as the geometric mean of the time-dependent posteriors for the word in the interval, which is used a confidence score for the word. The lattice-based methods tend to overestimate posterior probabilities of words. This may lead to poor confidence scores, specially when lattices are small and contain only a small part of likely word sequences. A decision tree trained using scoring results of hypotheses is used for piece-wise linear mapping of posterior probabilities to confidence scores in [31].

The confidence based adaptation is useful for the scenario where the generated supervision hypothesis has a high word error rate, as it discards the words with high error rate from the supervision. It has been found to improve the performance of speech recognition system compared to the standard adaptation techniques [4, 31, 174, 205]. However, it also reduces the amount of adaptation data, and if the amount of data becomes very small, it may not give reliable estimates of transforms. This problem can be dealt by using the confidence score to linearly weight the statistics for transform generation, rather than discarding segments [33].

3.1.5.2 N-best Adaptation

In the N-best list based adaptation approach [115, 116, 199], a number of possible hypotheses are used for adaptation purpose rather than just using the 1-best hypothesis. The N-best list may be the output produced by a multipass framework for rescoring or it can be generated using an SI model. N-best lists are generally generated at the utterance level or short segments, as their size increases exponentially with the number of words in the hypothesis. So there is only a small amount of data associated with the hypotheses in the N-best list and therefore N-best list based adaptation generally uses MAP estimation [115, 199]. In [115, 116], only a bias to the mean is considered, whereas a full transform is used in [199]. The N-best list based adaptation and rescoring framework in [199] is given in algorithm 2. This framework is used in a self-adaptation mode to adapt and decode given speech segments using N-best lists. In the framework, a separate transform is estimated corresponding to each of the hypothesis in the N-best list and the hypothesis giving the best inference criteria is selected as the recognition output. The approach in [199] is motivated from an approximation to Bayesian adaptive inference as described in section 5.2.3.

Step 1: Start with N-best hypotheses.

$$\mathcal{H} \in \{\mathcal{H}_1, \ldots, \mathcal{H}_N\} \tag{3.35}$$

Step 2: Estimate transforms for each hypothesis $\mathcal{H}$.

$$\hat{\mathbf{W}}^{(\mathcal{H})} = \arg\max_{\mathbf{W}} \Big\{ p(\mathbf{O}|\mathcal{H};\mathbf{W},\mathcal{M})p(\mathbf{W}|\phi) \Big\} \tag{3.36}$$

Step 3: Select the best hypothesis.

$$\hat{\mathcal{H}} = \arg\max_{\mathcal{H}} \Big\{ p(\mathbf{O}|\mathcal{H};\mathbf{W}^{(\mathcal{H})},\mathcal{M})p(\mathbf{W}|\phi)P(\mathcal{H}) \Big\} \tag{3.37}$$

Algorithm 2: *The N-best adaptation and decoding framework*

The N-best list can be used in other ways for adaptation as well. In [116], a final transform is obtained by smoothing the transforms corresponding to the N-best hypotheses and is subsequently used for decoding. This can be given for full transforms as

$$\hat{\mathbf{W}} = \frac{\sum_{\mathcal{H}} \mathcal{C}^{(\mathcal{H})}\mathbf{W}^{(\mathcal{H})}}{\sum_{\mathcal{H}} \mathcal{C}^{(\mathcal{H})}} \tag{3.38}$$

where $\mathcal{C}^{(\mathcal{H})}$ is a weight associated with each of the hypothesis. These weights are selected to be some confidence measures such as likelihood ratio given as

$$\mathcal{C}^{(\mathcal{H})} = \exp\Big(\eta\big(\log p(\mathbf{O}|\mathcal{H}) - \log p(\mathbf{O}|\mathcal{H}_1)\big)\Big) \tag{3.39}$$

where η is a heuristic control parameter. In [129], all hypotheses in the N-best list are used as supervision to estimate a transform as

$$\hat{\mathbf{W}} = \arg\max_{\mathbf{W}} \left\{ \sum_{\mathcal{H}} \mathcal{C}^{(\mathcal{H})} p(\mathbf{O}|\mathcal{H};\mathbf{W},\mathcal{M}) p(\mathbf{W}|\phi) \right\} \tag{3.40}$$

This approach is similar to the lattice-based adaptation described below.

The advantage of the N-best adaptation approach is that it can be used for instantaneous adaptation and decoding of speech. However, the use of an N-best list prevents utilising large amounts of data for adaptation due to exponentially increasing size of the N-best list.

3.1.5.3 Lattice Based Adaptation

An alternative solution to the problem of the erroneous hypothesis is to use lattice based adaptation [137, 174, 190, 191], as lattices have much lower oracle error rate than that of 1-best hypotheses. A lattice-based forward-backward algorithm is run over the lattice hypothesis, and the occupation probabilities are computed for each component. In this case, occupation probabilities represent the posterior probability of the component given all possible hypotheses in the lattice. The accumulated statistics are then used to estimate the transforms as in the standard MLLR or CMLLR estimation. The method has been found to give significant improvement in the performance, and is widely used in the multipass framework in state-of-art systems.

3.2 Adaptive Training

The training of speech recognition systems requires a large amount of speech data. As it is difficult to obtain large amounts of data recorded in a controlled environment, recently there has been a growing trend towards building a speech recognition system on *found* data, like broadcast news and conversation. Such data usually consists of utterances from different acoustic environments and several hundred speakers, and is inherently *non-homogeneous* in nature.

The standard approach for such a case is to build a system on all the data, treating them as a single homogeneous block of data independent of the source acoustic environment or speakers. This approach is referred as *multistyle training* [43]. The problem with this approach is that the trained acoustic models may not extract and represent the speech variabilities properly from non-homogeneous data that contains many other non-speech variabilities as well. The HMMs are forced to model the non-speech variabilities in speech as well, across a large number of speakers or environments. Therefore, the resulting multistyle models have large within-class variances and reduced discriminative ability. To alleviate the problem, some feature normalisation techniques[1] like cepstral mean and variance normalisation [6, 193], Gaussianisation [53, 105, 153] and vocal tract length normalisation [101] as described in section 2.2.3 are used to remove the non-speech variabilities from features to some extent. But still these methods cannot completely remove the non-speech variabilities from the given speech data.

An alternative way to deal with the training on such non-homogeneous data is to model speech and non-speech variabilities separately. This approach is called *adaptive training* [5, 44]. In an adaptive training scheme, speech variabilities are modelled by canonical HMMs, whereas the non-speech variabilities are modelled by a set of transforms. The commonly used forms of adaptive training are described below.

3.2.1 Speaker Adaptive Training (SAT)

Speaker adaptive training (SAT)[2] [5, 44] is a useful technique for building speech recognition systems on non-homogeneous data. The structure of SAT is shown in figure 3.4. In SAT, a set of speaker transforms, as well as a set of canonical models, is trained. The canonical models are speaker or environment independent, whereas a specific set of transforms is associated with each homogeneous block. The models for each homogeneous block are obtained by adapting the canonical models with the specific set of transforms corresponding to the block.

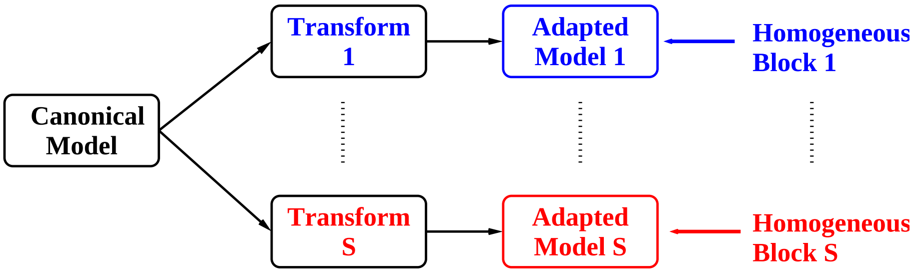


Figure 3.4: The speaker adaptive training (SAT) on non-homogeneous data

The canonical models and transforms set are estimated in an iterative fashion: first, the speaker-specific transforms are found; and then the canonical models are updated given these transforms. In ML-based SAT, both the transforms and the canonical models are estimated using the ML criterion. The iterative training procedure for ML-SAT is given in algorithm 3.

[1] The feature normalisation techniques can be also regarded as one form of adaptive training, as they attempt to separate or isolate non-speech variabilities across different non-homogeneous blocks.

[2] The term 'speaker' refers to one homogeneous block of data in a broad sense.

Step 1: Initialise canonical model set and transforms.

$\mathcal{M}_{\mathtt{ml}}$: SI Model

$\mathbf{W}^{(s)}_{\mathtt{ml}}$: $\mathbf{A}^{(s)}_{\mathtt{ml}} = \mathbf{I}, \mathbf{b}^{(s)}_{\mathtt{ml}} = \mathbf{0}$

Step 2: Estimate transforms for each speaker.

$$\mathbf{W}^{(s)}_{\mathtt{ml}} = \arg\max_{\mathbf{W}} \left\{ \log p(\mathbf{O}^{(s)} | \mathcal{H}^{(s)}; \mathbf{W}, \mathcal{M}_{\mathtt{ml}}) \right\} \tag{3.41}$$

Step 3: Update model parameters.

$$\mathcal{M}_{\mathtt{ml}} = \arg\max_{\mathcal{M}} \left\{ \sum_{s=1}^{S} \log p(\mathbf{O}^{(s)} | \mathcal{H}^{(s)}; \mathbf{W}^{(s)}_{\mathtt{ml}}, \mathcal{M}) \right\} \tag{3.42}$$

Step 4: Go to step (2) unless converged.

Algorithm 3: *The ML-SAT algorithm*

In the ML-SAT algorithm shown, the canonical model $\mathcal{M}_{\mathtt{ml}}$ is first initialised with the ML speaker-independent (SI) model, and speaker-specific affine transforms $\mathbf{W}^{(s)}_{\mathtt{ml}}$ are initialised with an identity transform ($\mathbf{I}$) and zero bias ($\mathbf{0}$). MLLR [103] transforms, $\mathbf{W}^{(s)}_{\mathtt{ml}}$, for each speaker s are estimated using equation (3.41). In the equation, $\mathbf{O}^{(s)}$ and $\mathcal{H}^{(s)}$ are the observation and the corresponding transcripts for data from speaker s, respectively and $\mathcal{M}_{\mathtt{ml}}$ is the current canonical model set. Given the set of estimated transforms, the model parameters are updated by maximising the log-likelihood over the training data from all S speakers as in equation (3.42). The expectation maximisation (EM) algorithm is used for estimating the transform parameters and canonical models through an iterative process. The required auxiliary function can be derived from the expression for the likelihood, which can be given for speech data from all speakers $\mathbb{O} = \{\mathbf{O}^{(1)}, \ldots, \mathbf{O}^{(S)}\}$ as[1]

$$p(\mathbb{O}|\mathbb{H}, \mathcal{M}, \mathbb{W}) = \prod_{s=1}^{S} p(\mathbf{O}^{(s)}|\mathcal{H}^{(s)}, \mathcal{M}, \mathbf{W}^{(s)}) \tag{3.43}$$

where s represents a speaker or homogeneous block of data, $\mathbb{W} = \{\mathbf{W}^{(1)}, \ldots, \mathbf{W}^{(S)}\}$ is the set of transforms, and $\mathbb{H} = \{\mathcal{H}^{(1)}, \ldots, \mathcal{H}^{(S)}\}$ is the set of transcripts for the corresponding observation sequences. The likelihood for each speaker can be expressed as a marginalisation over all possible component sequences in a similar way to equation (2.44) as

$$p(\mathbf{O}^{(s)}|\mathcal{H}^{(s)}, \mathcal{M}, \mathbf{W}^{(s)}) = \sum_{\boldsymbol{\theta}} P(\boldsymbol{\theta}|\mathcal{H}^{(s)}, \mathbf{W}^{(s)}, \mathcal{M}) \prod_{t} p(\mathbf{o}_t|\mathcal{M}, \mathbf{W}^{(s)}, \theta_t) \tag{3.44}$$

[1] The label $\mathtt{ml}$ has been dropped from the model and transforms hereafter in this chapter.

However, due to the hidden component sequence, the model parameters cannot be directly estimated from the above equations. Therefore, a lower-bound to the likelihood in the above equation is used as an auxiliary function, and the EM algorithm is used to iteratively update the parameters. The auxiliary function for estimating the model parameters given the current estimate of transform set $\mathbb{W}$ and model $\mathcal{M}$ is given in a similar way to equation (2.45) as

$$\mathcal{Q}(\hat{\mathcal{M}};\mathcal{M},\mathbb{W}) = \left\langle \log p(\mathbb{O},\boldsymbol{\theta}|\hat{\mathcal{M}},\mathbb{W},\mathbb{H}) \right\rangle_{P(\boldsymbol{\theta}|\mathbb{O},\mathbb{H},\mathcal{M},\mathbb{W})} \tag{3.45}$$

Similarly, the auxiliary function for estimating transforms for each speaker is given by

$$\mathcal{Q}\big(\hat{\mathbf{W}}^{(s)};\mathbf{W}^{(s)},\mathcal{M}\big) = \left\langle \log p(\mathbf{O}^{(s)},\boldsymbol{\theta}|\mathcal{M},\hat{\mathbf{W}}^{(s)},\mathcal{H}^{(s)}) \right\rangle_{P(\boldsymbol{\theta}|\mathbf{O}^{(s)},\mathcal{H}^{(s)},\mathcal{M},\mathbf{W}^{(s)})} \tag{3.46}$$

The models and transforms are estimated in an interleaved fashion as shown in algorithm 3, using the auxiliary functions given in equations (3.45) and (3.46). It should be noted that each of the estimation steps is itself an iterative procedure. Moreover, the model parameter estimation itself involves interleaved updates for mean vectors and covariance matrices.

Canonical models estimated with SAT cannot be directly used for recognition. As unsupervised adaptation is being used in this work, an initial supervision hypothesis must be obtained. An SI model is often used for generating the supervision hypothesis for the given test data. Given this hypothesis, test-set speaker transforms are estimated in a similar fashion to the training procedure in algorithm 3, except that the model update stage in step (3) is omitted. The recognition procedure using the ML-SAT system is also illustrated in figure 3.5. As it can be seen in the figure, the ML transform for test data is first estimated using the ML-SI model and then subsequently using each iteration's ML-SAT models, given the transforms estimated using the model at previous iteration. The final ML-SAT models and estimated test set transforms are used for decoding the test data.

Several forms of transforms are possible for SAT [5, 44]. The MLLR and CMLLR based ML-SAT schemes are described in the next sections.

3.2.1.1 MLLR-based SAT

In MLLR-based speaker adaptive training [5], a distinct transform is estimated for each speaker and applied to mean vectors of canonical models to obtain the adapted model parameters for the speaker. The form of the transform and adaptation has been described in section 3.1.2.1. However, as data from a number of speakers are involved in SAT, the MLLR transform is indexed with a speaker index s, and re-expressed as

$$\boldsymbol{\mu}_m^{(s)} = \mathbf{A}^{(sr_m)}\boldsymbol{\mu}_m + \mathbf{b}^{(sr_m)} = \mathbf{W}^{(sr_m)}\boldsymbol{\xi}_m \tag{3.47}$$

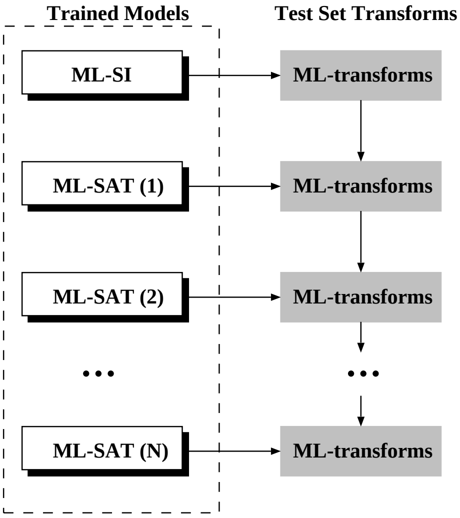


Figure 3.5: The recognition setup for test data using the ML-SAT system. ML-SAT(k) represents canonical models from the kth iteration of the ML-SAT procedure.

where r_m is the regression base class of the Gaussian component m, and $\mathbf{W}^{(sr)} = [\mathbf{A}^{(sr)} \;\; \mathbf{b}^{(sr)}]$ is the transform for speaker s and regression base class r. $\boldsymbol{\mu}_m^{(s)}$ is the adapted mean of component m for speaker s. The estimation of MLLR transforms is also the same as given in equations (3.8) to (3.11), though the estimation now uses the current canonical models and the current estimate of the MLLR transform to obtain component posteriors. The auxiliary function for updating the canonical model parameters using the ML criterion can be obtained from equation (3.45), and is given by

$$\begin{aligned}\mathcal{Q}(\hat{\mathcal{M}};\mathcal{M},\mathbb{W}) &= -\frac{1}{2}\sum_{sm}\sum_{t_s}\gamma_m^{\mathtt{ml}}(t_s)\Big\{\log|\hat{\boldsymbol{\Sigma}}_m| \\ &\quad + \left(\mathbf{o}_{t_s}^{(s)} - \mathbf{A}^{(sr_m)}\hat{\boldsymbol{\mu}}_m - \mathbf{b}^{(sr_m)}\right)^{\mathtt{T}}\hat{\boldsymbol{\Sigma}}_m^{-1}\left(\mathbf{o}_{t_s}^{(s)} - \mathbf{A}^{(sr_m)}\hat{\boldsymbol{\mu}}_m - \mathbf{b}^{(sr_m)}\right)\Big\} \quad (3.48)\end{aligned}$$

where $\hat{\mathcal{M}}$ is the new estimate of canonical models, $\mathbb{W}$ represents the MLLR transform set consisting of current estimates of transforms for all speakers and baseclasses, and $\gamma_m^{\mathtt{ml}}(t_s)$ is the posterior occupancy of component m at time t_s for speaker s based on the current canonical model $\mathcal{M}$ and the transform estimates $\mathbb{W}$. The update to the means and the covariance is done one by one in an interleaved fashion, assuming one to be fixed while updating the other. The mean can be first updated by assuming the covariance matrix to be fixed at the current estimate $\boldsymbol{\Sigma}_m$. This gives the sufficient statistics for mean update as [5]

$$\mathbf{G}_m^{\mathtt{ml}} = \sum_{st_s} \gamma_m^{\mathtt{ml}}(t_s)\mathbf{A}^{(sr_m)\mathtt{T}}\boldsymbol{\Sigma}_m^{-1}\mathbf{A}^{(sr_m)} \tag{3.49}$$

$$\mathbf{k}_m^{\mathtt{ml}} = \sum_{st_s} \gamma_m^{\mathtt{ml}}(t_s)\mathbf{A}^{(sr_m)\mathtt{T}}\boldsymbol{\Sigma}_m^{-1}\left(\mathbf{o}_{t_s}^{(s)} - \mathbf{b}^{(sr_m)}\right) \tag{3.50}$$

The new estimates of the mean is given by

$$\hat{\boldsymbol{\mu}}_m = \left(\mathbf{G}_m^{\mathtt{ml}}\right)^{-1}\mathbf{k}_m^{\mathtt{ml}} \tag{3.51}$$

The covariance is estimated after the mean update, and is given by

$$\hat{\boldsymbol{\Sigma}}_m = \operatorname{diag}\left(\frac{\sum_{st_s}\gamma_m^{\mathtt{ml}}(t_s)\left(\mathbf{o}_{t_s}^{(s)} - \mathbf{W}^{(sr_m)}\hat{\boldsymbol{\xi}}_m\right)\left(\mathbf{o}_{t_s}^{(s)} - \mathbf{W}^{(sr_m)}\hat{\boldsymbol{\xi}}_m\right)^{\mathtt{T}}}{\sum_{st_s}\gamma_m^{\mathtt{ml}}(t_s)}\right) \tag{3.52}$$

where $\hat{\boldsymbol{\xi}}_m = [\hat{\boldsymbol{\mu}}_m^{\mathtt{T}} \;\; 1]^{\mathtt{T}}$. In MLLR-based SAT, re-estimation of means using sufficient statistics in equation (3.49) requires considerable memory when using a full transform [113]. Besides, the means and variances cannot be updated in a single pass.

3.2.1.2 CMLLR-based SAT

In CMLLR based SAT [44], constrained transforms as described in section 3.1.2.3 are used for each speaker or homogeneous block, along with the canonical models. Using the speaker and the regression base class index, the form of adaptation with CMLLR can be re-expressed as

$$\mathbf{o}_{t_s}^{(sr_m)} = \mathbf{A}^{(sr_m)}\mathbf{o}_{t_s}^{(s)} + \mathbf{b}^{(sr_m)} = \mathbf{W}^{(sr_m)}\boldsymbol{\zeta}_{t_s}^{(s)} \tag{3.53}$$

where r_m is the regression base class the Gaussian component m belongs to, and $\mathbf{W}^{(sr)} = [\mathbf{A}^{(sr)} \;\; \mathbf{b}^{(sr)}]$ is the CMLLR transform for speaker s and regression base class r, $\mathbf{o}_{t_s}^{(sr_m)}$ is the transformed observation, adapted using $\mathbf{W}^{(sr_m)}$, and thus is dependent upon the regression base class. These CMLLR transforms for the SAT system can be estimated using the auxiliary function in equation (3.48), and leads to the same update formulae as given in section 3.1.2.3, however now the occupation probabilities are based on current estimate of the canonical models and the transforms. The auxiliary function for update of canonical model parameters in the CMLLR based SAT can be derived from equation (3.48), and is given by [44]

$$\mathcal{Q}(\hat{\mathcal{M}};\mathcal{M},\mathbb{W}) = -\frac{1}{2}\sum_{sm}\sum_{t_s}\gamma_m^{\mathtt{ml}}(t_s)\left\{\log|\hat{\boldsymbol{\Sigma}}_m| + \left(\mathbf{o}_{t_s}^{(sr_m)} - \hat{\boldsymbol{\mu}}_m\right)^{\mathtt{T}}\hat{\boldsymbol{\Sigma}}_m^{-1}\left(\mathbf{o}_{t_s}^{(sr_m)} - \hat{\boldsymbol{\mu}}_m\right)\right\} \tag{3.54}$$

where $\mathbb{W}$ is now the set of current estimates of CMLLR transforms for each speaker, and $\gamma_m^{\texttt{ml}}(t_s)$ is the posterior occupancy based on current estimate of models and transforms (transformed observations). This auxiliary function differs from the standard ML auxiliary function for model estimation in equation 2.46 only in that the transformed observations, $\mathbf{o}_{t_s}^{(sr_m)}$, are used. The maximisation of the above auxiliary function yields the update for mean and covariance matrices

$$\begin{aligned}\hat{\boldsymbol{\mu}}_m &= \frac{\sum_{st_s}\gamma_m^{\texttt{ml}}(t_s)\mathbf{o}_{t_s}^{(sr_m)}}{\sum_{st_s}\gamma_m^{\texttt{ml}}(t_s)} \\ \hat{\boldsymbol{\Sigma}}_m &= \text{diag}\left(\frac{\sum_{st_s}\gamma_m^{\texttt{ml}}(t_s)(\hat{\mathbf{o}}_{t_s}^{(sr_m)}-\hat{\boldsymbol{\mu}}_m)(\mathbf{o}_{t_s}^{(sr_m)}-\hat{\boldsymbol{\mu}}_m)^{\texttt{T}}}{\sum_{st_s}\gamma_m^{\texttt{ml}}(t_s)}\right)\end{aligned}$$

These update formulae are similar to the standard ML estimation in equations (2.42) and (2.43), and thus leads to similar storage requirements to the standard ML training.

3.2.2 Cluster Adaptive Training (CAT)

In cluster adaptive training [41, 46], multiple sets of HMMs, one for each cluster of training data, are used as canonical models. A set of interpolation weights are used to combine them to obtain the models for the target speaker or environment. Therefore, a model for a target environment is given as the weighted sum of the multiple sets of HMMs from different clusters, as shown in figure 3.6. When these weights are binary 1/0, the method reduces to a cluster dependent modelling, where each cluster has its own set of models.

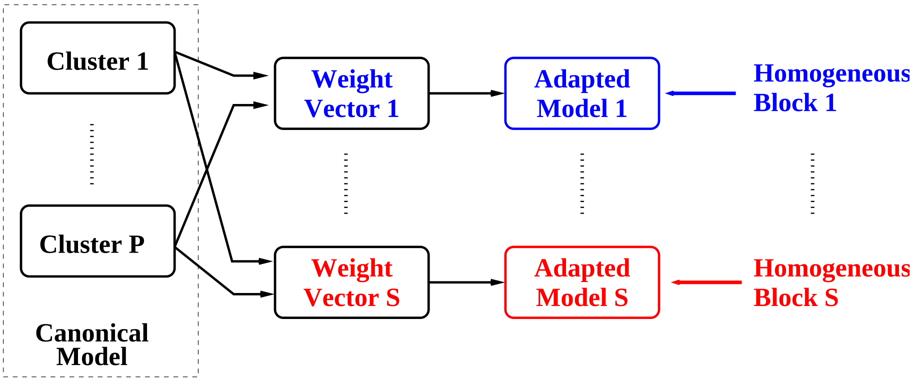


Figure 3.6: The cluster adaptive training (CAT) on non-homogeneous data

In the commonly used CAT systems, only means of the components are assumed distinct for each cluster, and other parameters including covariances, mixture component weights and transition matrices are assumed to be the same for all clusters. The canonical model parameters in CAT for each component m consist of a prior c_m, a covariance matrix $\boldsymbol{\Sigma}_m$, and a set of P means given by

$$\mathbf{M}_m = \left[\begin{array}{ccc} \boldsymbol{\mu}_m^{(1)} & \ldots & \boldsymbol{\mu}_m^{(P)} \end{array} \right]$$

where $\boldsymbol{\mu}_m^{(p)}$ is the mean associated with cluster p and P is the total number of clusters. The transform parameters in CAT are cluster weights vector $\boldsymbol{\lambda}^{(s)}$ for each speaker s can be expressed as

$$\boldsymbol{\lambda}^{(s)} = \left[\begin{array}{ccc} \lambda_1^{(s)} & \ldots & \lambda_P^{(s)} \end{array} \right]^{\mathsf{T}} \tag{3.55}$$

where $\lambda_p^{(s)}$ is the interpolation weight associated with cluster p. The adapted mean vector corresponding to a particular speaker s is given by

$$\boldsymbol{\mu}_m^{(s)} = \mathbf{M}_m \boldsymbol{\lambda}^{(s)} \tag{3.56}$$

The parameter estimation for maximum-likelihood CAT has been described in [41]. The expectation maximisation algorithm is used to estimate the canonical models and cluster weights in an interleaved fashion. The trained canonical models are used in recognition by estimating the cluster weights for the given adaptation data and the corresponding transcript. The CAT scheme described above is also referred as model-based CAT, where the clusters are represented as a distinct set of mean vectors. An alternative form of CAT known as transform-based CAT has been also described in [41], in which the clusters are represented by a set of cluster-specific transforms of a common *set* of canonical means. This gives a more compact representation of clusters.

A closely related technique to CAT is eigenvoices [97], which also finds the means for the adapted model as a weighted sum of the cluster-dependent HMMs. However, this method, in its original form, finds the clusters, called eigenvoices, by using principal component analysis (PCA) of a set of supervectors constructed from all the mean values in the set of speaker dependent HMM systems [97]. A maximum-likelihood eigen-decomposition algorithm is used to estimate weights for eigenvoices during adaptation [97], which is identical to the model-based CAT. In [16], the use of MAP and MLLR for estimating the required speaker dependent models has been investigated.

3.3 Summary

In this chapter, the techniques for adaptation and adaptive training have been reviewed. The mismatch between training and testing acoustic condition is reduced by adapting the trained acoustic models to the test environment to obtain a better performance. This is done by using different adaptation schemes like MAP, MLLR, CMLLR, MALPR or the cluster-based approach to adapt mean and/or covariance of Gaussian components of HMMs. A regression class tree is also commonly used to generate multiple transforms depending upon the amount of adaptation data. The adaptation can be supervised or unsupervised depending upon the availability of supervision transcripts for given adaptation data and it can be performed either in offline or online mode. In unsupervised adaptation, the gain is reduced compared to the supervised adaptation due to erroneous supervision hypotheses. Confidence, N-best list and lattice based adaptation can be used to partly deal with this problem of erroneous supervision hypotheses. Similarly, to deal with the non-homogeneous data in training, adaptive training schemes such as speaker adaptive training (SAT) and cluster adaptive training (CAT) can be used. In SAT, a speaker-independent canonical model set and speaker-specific transforms are used, and the model for the target environment is obtained by adapting the canonical model with a target transform. On the other hand, CAT has one HMM set for each cluster of training data, and the model for the target environment is obtained through their interpolation. In this way, adaptive training attempts to model speech and non-speech variabilities separately.

CHAPTER 4
Discriminative Adaptation and Adaptive Training

Adaptation and adaptive training schemes play an important role in speech recognition systems. In the previous chapter, the training criteria to estimate models and transforms were based on maximum likelihood. However, there are limitations of maximum likelihood estimation as described in section 2.3.2, and discriminative training of HMMs has been found to improve the performance of speech recognition systems [92, 130, 140, 154]. Therefore, the use of discriminative criteria such as MMI and MPE has been also investigated for estimating adaptation transforms and canonical models [189]. This chapter describes several discriminative transforms as well as commonly used discriminative adaptive training of acoustic models. The discriminative linear transforms (DLTs) [63, 118, 171, 175, 182] and discriminative mapping transforms (DMTs) [202, 203] based adaptation are described in section 4.1. This is followed by the description of MLLR and DLT based discriminative speaker adaptive training [106, 118, 171, 180], in section 4.2. Finally, an adaptive training scheme based on discriminative mapping transforms is proposed in section 4.3 to deal with the problem of unsupervised discriminative adaptation.

4.1 Discriminative Adaptation

The traditional adaptation schemes use ML transforms that maximise the likelihood of adaptation data. However, maximising the likelihood of data is not closely related to word error rate in the speech recognition task. ML estimation assumes HMMs as a true generative model of speech such that maximising the likelihood of adaptation data is expected to give good performance on unseen test data. Moreover, ML estimation can give an optimal consistent estimate with minimum variance only when there is sufficient amount of adaptation data and HMMs are the true source of data. Otherwise, the ML criterion may lead to unreliable estimates. However, most of the practical speech recognition systems may have only a small amount of adaptation data. Therefore, the use of discriminative criteria in model adaptation is very reasonable as discriminative training of models have led to performance improvements in LVCSR compared to traditional ML training. A number of approaches for discriminative adaption has been investigated using linear transforms to adapt Gaussian means and covariances. These transforms are estimated using one of the discriminative criteria such as maximum mutual information (MMI) [10] or minimum phone error (MPE) [140]. Discriminative linear transforms (DLTs) [63, 118, 171, 175, 182] and discriminative mapping transforms [202, 203] are described in the next sections.

4.1.1 Discriminative Linear Transforms (DLT)

The discriminative linear transforms (DLTs) [63, 118, 171, 175, 182] use one of the discriminative criteria such as MMI, MPE or MWE to estimate linear transforms which are then used to adapt models. These transforms may be mean, diagonal or full covariance transforms, or constrained transforms. A discriminative mean transform, based on the MPE criterion, is described in this section. The transforms based on other discriminative criteria can be also estimated in a similar manner. A regression class tree as described in section 3.1.4 can be also used in the same manner as used for ML transforms to generate multiple transforms by grouping Gaussians to yield separate sufficient statistics. The form of the DLT described in this section is same as given in equation (3.5) for transformation of means and is reproduced here

$$\hat{\boldsymbol{\mu}}_m = \mathbf{A}\boldsymbol{\mu}_m + \mathbf{b} = \mathbf{W}\boldsymbol{\xi}_m \tag{4.1}$$

where $\hat{\boldsymbol{\mu}}_m$ represents the adapted mean vector, $\boldsymbol{\xi}_m = [\boldsymbol{\mu}_m^{\mathtt{T}} \;\; 1]^{\mathtt{T}}$ is the extended mean vector, and $\mathbf{W} = [\mathbf{A} \;\; \mathbf{b}]$ is an affine linear transform, though now estimated using a discriminative criterion.

The MPE criterion based mean transform is obtained by optimising the MPE objective function given as [182]

$$\begin{aligned} \mathcal{F}(\mathbf{W}) &= \sum_{\mathcal{H}} P(\mathcal{H}|\mathbf{O}, \mathbf{W}, \mathcal{M}) \mathcal{A}(\mathcal{H}, \mathcal{H}_{\mathtt{r}}) \\ &= \sum_{\mathcal{H}} \frac{p(\mathbf{O}|\mathcal{H}, \mathbf{W}, \mathcal{M}) P(\mathcal{H})}{\sum_{\breve{\mathcal{H}}} p(\mathbf{O}|\breve{\mathcal{H}}, \mathbf{W}, \mathcal{M}) P(\breve{\mathcal{H}})} \mathcal{A}(\mathcal{H}, \mathcal{H}_{\mathtt{r}}) \end{aligned} \tag{4.2}$$

where the raw phone accuracy $\mathcal{A}(\mathcal{H}, \mathcal{H}_{\mathtt{r}})$ between the hypothesis $\mathcal{H}$ and reference supervision $\mathcal{H}_{\mathtt{r}}$ has been used in place of loss function for convenience in description. The difference with equation (2.63) is that the adaptation transform has been included in the criterion. A weak-sense auxiliary function [140] as used in the discriminative training of HMMs in section 2.3.4.2, is used to optimise the above objective function as well and obtain new estimates of transforms $\hat{\mathbf{W}}$. The auxiliary function is based on the log-likelihood conditioned on phone arcs and is given as [180]

$$\begin{aligned} \mathcal{Q}(\hat{\mathbf{W}}; \mathbf{W}) &= \sum_{l} \frac{\partial \mathcal{F}(\hat{\mathbf{W}})}{\partial \log p(\mathbf{O}|l, \hat{\mathbf{W}}, \mathcal{M})} \bigg|_{\hat{\mathbf{W}}=\mathbf{W}} \log p(\mathbf{O}|l, \hat{\mathbf{W}}, \mathcal{M}) \\ &= \sum_{l} \gamma_l^{\mathtt{mpe}} \log p(\mathbf{O}|l, \hat{\mathbf{W}}, \mathcal{M}). \end{aligned} \tag{4.3}$$

In the above equation, $\log p(\mathbf{O}|l, \hat{\mathbf{W}}, \mathcal{M})$ is the log-likelihood for given phone arc l given the transform and models parameters, and $\gamma_l^{\mathtt{mpe}}$ is the “posterior probability” of arc l defined by

$$\gamma_l^{\mathtt{mpe}} = \frac{\partial \mathcal{F}(\hat{\mathbf{W}})}{\partial \log p(\mathbf{O}|l, \hat{\mathbf{W}}, \mathcal{M})} \bigg|_{\hat{\mathbf{W}}=\mathbf{W}} \tag{4.4}$$

where $\mathbf{W}$ is the current estimate of the transform. The MPE posterior occupancy $\gamma_l^{\mathtt{mpe}}$ is computed in a similar way as described in section 2.3.4.2. $\gamma_l^{\mathtt{mpe}}$ can be split into numerator and denominator parts depending upon its sign, as in equations (2.102) and (2.103), thus leading to numerator and denominator auxiliary functions. A smoothing term $\mathcal{Q}^{\mathtt{sm}}(\hat{\mathbf{W}}; \mathbf{W})$ is also added to the auxiliary function to ensure its stability [180], which is maximum at current estimate of transforms, so that

$$\frac{\partial \mathcal{Q}^{\mathtt{sm}}(\hat{\mathbf{W}}; \mathbf{W})}{\partial \hat{\mathbf{W}}} \bigg|_{\hat{\mathbf{W}}=\mathbf{W}} = 0. \tag{4.5}$$

The use of an I-smoothing prior has been also found to be useful in discriminative training [142], and is thus included in the auxiliary function as $\mathcal{Q}^{\mathtt{I}}(\hat{\mathbf{W}}; \mathbf{W})$. Therefore, the overall auxiliary function including smoothing terms can be given as

$$\begin{aligned}\mathcal{Q}(\hat{\mathbf{W}};\mathbf{W}) &= \mathcal{Q}^{\mathtt{num}}(\hat{\mathbf{W}};\hat{\mathbf{W}}) - \mathcal{Q}^{\mathtt{den}}(\hat{\mathbf{W}};\mathbf{W}) + \mathcal{Q}^{\mathtt{sm}}(\hat{\mathbf{W}};\mathbf{W}) + \mathcal{Q}^{\mathtt{I}}(\hat{\mathbf{W}};\mathbf{W}) \\ &= \sum_d \Big(\mathcal{Q}^{\mathtt{num}}(\hat{\mathbf{w}}_d;\mathbf{w}_d) - \mathcal{Q}^{\mathtt{den}}(\hat{\mathbf{w}}_d;\mathbf{w}_d) + \mathcal{Q}^{\mathtt{sm}}(\hat{\mathbf{w}}_d;\mathbf{w}_d) + \mathcal{Q}^{\mathtt{I}}(\hat{\mathbf{w}}_d;\mathbf{w}_d)\Big) \quad (4.6)\end{aligned}$$

where $\hat{\mathbf{w}}_d$ is the dth row of the transform $\hat{\mathbf{W}}$ arranged as a column vector, and each rows of the transform are assumed independent of each other.

The numerator auxiliary function $\mathcal{Q}^{\mathtt{num}}(\hat{\mathbf{W}};\mathbf{W})$ can be expressed in the same form as the MLLR auxiliary function in equation (3.8) and is expressed as

$$\mathcal{Q}^{\mathtt{num}}(\hat{\mathbf{W}};\mathbf{W}) = \tilde{K}^{\mathtt{num}} + \sum_{mt} \gamma_m^{\mathtt{num}}(t) \log \mathcal{N}(\mathbf{o}_t; \hat{\mathbf{W}}\boldsymbol{\xi}_m, \boldsymbol{\Sigma}_m) \quad (4.7)$$

where $\boldsymbol{\xi}_m = [\boldsymbol{\mu}_m^{\mathtt{T}} \;\; 1]^{\mathtt{T}}$ is an extended mean vector, $\tilde{K}^{\mathtt{num}}$ is a constant and $\gamma_m^{\mathtt{num}}(t)$ is numerator occupancy of the mth mixture component at time t as defined in equations (2.102) and (2.103). Considering each row $\hat{\mathbf{w}}_d$ independently for a diagonal covariance matrix case, the numerator auxiliary function can be expressed in a sufficient statistics form as

$$\begin{aligned}\mathcal{Q}^{\mathtt{num}}(\hat{\mathbf{w}}_d;\mathbf{w}_d) &= \mathcal{G}(\hat{\mathbf{w}}_d;\boldsymbol{\Gamma}^{\mathtt{num}}) \\ &= K_d^{\mathtt{num}} - \frac{1}{2}\hat{\mathbf{w}}_d^{\mathtt{T}}\mathbf{G}_d^{\mathtt{num}}\hat{\mathbf{w}}_d + \hat{\mathbf{w}}_d^{\mathtt{T}}\mathbf{k}_d^{\mathtt{num}} \quad (4.8)\end{aligned}$$

where $\boldsymbol{\Gamma}_d^{\mathtt{num}} = \{\mathbf{G}_d^{\mathtt{num}}, \mathbf{k}_d^{\mathtt{num}}\}$ is the sufficient statistics, given by

$$\begin{aligned}\mathbf{G}_d^{\mathtt{num}} &= \sum_{mt} \gamma_m^{\mathtt{num}}(t) \frac{\boldsymbol{\xi}_m\boldsymbol{\xi}_m^{\mathtt{T}}}{\sigma_{md}^2} \quad (4.9) \\ \mathbf{k}_d^{\mathtt{num}} &= \sum_{mt} \gamma_m^{\mathtt{num}}(t) o_{td} \frac{\boldsymbol{\xi}_m}{\sigma_{md}^2} \quad (4.10)\end{aligned}$$

In the above equations, σ_{md}^2 is the dth diagonal element of the covariance matrix. The auxiliary function and the statistics for the denominator term are also given in the same form.

The smoothing auxiliary function in equation (4.6) is selected satisfying the criterion in equation (4.5) as [180]

$$\mathcal{Q}^{\mathtt{sm}}(\hat{\mathbf{W}};\mathbf{W}) = \tilde{K}^{\mathtt{sm}} + \sum_m D_m \Big(-\frac{1}{2}(\hat{\mathbf{W}}\boldsymbol{\xi}_m - \mathbf{W}\boldsymbol{\xi}_m)^{\mathtt{T}} \boldsymbol{\Sigma}_m^{-1} (\hat{\mathbf{W}}\boldsymbol{\xi}_m - \mathbf{W}\boldsymbol{\xi}_m)\Big) \quad (4.11)$$

where D_m is a smoothing factor and $\tilde{K}^{\mathtt{sm}}$ includes all terms independent of transform $\hat{\mathbf{W}}$. This smoothing auxiliary function can also be expressed in a sufficient statistics form for each row of $\hat{\mathbf{W}}$ as

$$\begin{aligned}\mathcal{Q}^{\mathtt{sm}}(\hat{\mathbf{w}}_d;\mathbf{w}_d) &= \mathcal{G}(\hat{\mathbf{w}}_d;\boldsymbol{\Gamma}^{\mathtt{sm}}) \\ &= K_d^{\mathtt{sm}} - \frac{1}{2}\hat{\mathbf{w}}_d^{\mathtt{T}}\mathbf{G}_d^{\mathtt{sm}}\hat{\mathbf{w}}_d + \hat{\mathbf{w}}_d^{\mathtt{T}}\mathbf{k}_d^{\mathtt{sm}} \quad (4.12)\end{aligned}$$

where $K_d^{\texttt{sm}}$ is a constant and sufficient statistics $\mathbf{\Gamma}_d^{\texttt{sm}} = \{\mathbf{G}_d^{\texttt{sm}}, \mathbf{k}_d^{\texttt{sm}}\}$ is given by

$$\mathbf{G}_d^{\texttt{sm}} = \sum_m D_m \frac{\boldsymbol{\xi}_m \boldsymbol{\xi}_m^{\texttt{T}}}{\sigma_{md}^2} \tag{4.13}$$

$$\mathbf{k}_d^{\texttt{sm}} = \sum_m D_m \frac{\boldsymbol{\xi}_m \boldsymbol{\xi}_m^{\texttt{T}} \mathbf{w}_d}{\sigma_{md}^2}. \tag{4.14}$$

For the remaining I-smoothing term, the ML-statistics scaled by a factor $\alpha^{\texttt{I}}$ is generally used. Therefore, the auxiliary function $\mathcal{Q}^{\texttt{I}}(\hat{\mathbf{W}}; \mathbf{W})$ for the I-smoothing prior $p(\hat{\mathbf{W}}|\phi_{\texttt{ml}})$ can be also expressed in a similar way as

$$\begin{aligned}\mathcal{Q}^{\texttt{I}}(\hat{\mathbf{w}}_d; \mathbf{w}_d) &= \mathcal{G}(\hat{\mathbf{w}}_d; \mathbf{\Gamma}^{\texttt{I}}) \\ &= K_d^{\texttt{I}} - \frac{1}{2}\hat{\mathbf{w}}_d^{\texttt{T}} \mathbf{G}_d^{\texttt{I}} \hat{\mathbf{w}}_d + \hat{\mathbf{w}}_d^{\texttt{T}} \mathbf{k}_d^{\texttt{I}} \end{aligned} \tag{4.15}$$

with the sufficient statistics $\mathbf{\Gamma}_d^{\texttt{I}} = \{\mathbf{G}_d^{\texttt{I}}, \mathbf{k}_d^{\texttt{I}}\}$ given by

$$\mathbf{G}_d^{\texttt{I}} = \alpha^{\texttt{I}} \mathbf{G}_d^{\texttt{ml}} \tag{4.16}$$

$$\mathbf{k}_d^{\texttt{I}} = \alpha^{\texttt{I}} \mathbf{k}_d^{\texttt{ml}} \tag{4.17}$$

where $\alpha^{\texttt{I}}$ is a constant controlling the impact of the prior.

Therefore, the overall auxiliary function in equation (4.6) can be rewritten in the sufficient statistics form as

$$\mathcal{Q}(\hat{\mathbf{w}}_d; \mathbf{w}_d) = \mathcal{G}(\hat{\mathbf{w}}_d; \mathbf{\Gamma}^{\texttt{num}}) - \mathcal{G}(\hat{\mathbf{w}}_d; \mathbf{\Gamma}^{\texttt{den}}) + \mathcal{G}(\hat{\mathbf{w}}_d; \mathbf{\Gamma}^{\texttt{sm}}) + \mathcal{G}(\hat{\mathbf{w}}_d; \mathbf{\Gamma}^{\texttt{I}}) \tag{4.18}$$

which gives the overall sufficient statistics $\mathbf{\Gamma}_d = \{\mathbf{G}_d, \mathbf{k}_d\}$ as

$$\mathbf{G}_d = \mathbf{G}_d^{\texttt{num}} - \mathbf{G}_d^{\texttt{den}} + \mathbf{G}_d^{\texttt{sm}} + \alpha^{\texttt{I}} \mathbf{G}_d^{\texttt{ml}} \tag{4.19}$$

$$\mathbf{k}_d = \mathbf{k}_d^{\texttt{num}} - \mathbf{k}_d^{\texttt{den}} + \mathbf{k}_d^{\texttt{sm}} + \alpha^{\texttt{I}} \mathbf{k}_d^{\texttt{ml}} \tag{4.20}$$

The new estimate of DLT parameters can be obtained in terms of these sufficient statistics by maximising the auxiliary function in equation (4.18). This is given by

$$\hat{\mathbf{w}}_d = \mathbf{G}_d^{-1} \mathbf{k}_d. \tag{4.21}$$

The smoothing factor D_m in equations (4.13) and (4.14) required for the DLT estimation is set as

$$D_m = E_d \gamma_m^{\texttt{den}}; \quad E_d = \max(E, 2\hat{E}_d) \tag{4.22}$$

where the value of E_d is separately chosen for each row of transforms. In the above equation, E is a user-defined global constant and $\hat{E}_d$ is the minimum value to make $\mathbf{G}_d$ positive-definite.

The choice of an appropriate value of the smoothing factor is very important to obtain reliable estimates of DLTs. In [180], a value between 0.5 and 2.5 is chosen for E.

DLTs have been found to give significant performance gain over ML for supervised adaptation [180, 202]. However, the gain is significantly reduced in the unsupervised mode of adaptation [147, 202]. This is because discriminative criteria are based on phone or word error metrics. Transforms estimated using discriminative criteria are very sensitive to errors in the supervision hypothesis. The sensitivity of discriminative transforms to such errors limits the performance gain with DLTs. To reduce the impact of hypothesis errors, confidence scores based approaches [179, 180, 182] and lattice-based adaptation [137, 174, 202] have also been investigated for discriminative adaptation as well. A discriminative version of MAP, and the use of N-best list for accumulating the weighted statistics for discriminative transforms have been investigated in [56]. However, they yield only a little improvement, if any. To address this problem, discriminative mapping transforms (DMTs) [202, 203] have been also proposed, which is a speaker-independent transform and is applied to speaker-specific ML transforms. DMTs do not directly depend upon the supervision hypothesis of test data and thus are not sensitive to any errors in it. The form and estimation of DMTs are described in the next section.

4.1.2 Discriminative Mapping Transforms (DMT)

A discriminative mapping transform [202, 203] aims to transform a speaker specific ML transform into a discriminative one. This transform mapping is estimated in a speaker-independent fashion. The DMTs are discriminatively-estimated speaker independent transforms, and thus the same transforms can be used for the training and the test data. There is no need to estimate speaker-specific discriminative transforms on the test data. Thus the sensitivity to errors in the supervision hypothesis that has a severe impact on the performance of DLTs for unsupervised adaptation should not be a problem. A general form of the DMT [202] is expressed as [1]

$$\text{vec}(\mathbf{W}_{\texttt{d}}^{(s)}) = \mathbf{H}_{\texttt{d}}\text{vec}(\mathbf{W}_{\texttt{ml}}^{(s)}) + \mathbf{c}_{\texttt{d}} \tag{4.23}$$

where $\mathbf{W}_{\texttt{d}}^{(s)}$ is the final discriminative-like speaker transform, $\mathbf{H}_{\texttt{d}}$ and $\mathbf{c}_{\texttt{d}}$ are the speaker independent parameters of the DMT and $\mathbf{W}_{\texttt{ml}}^{(s)}$ is the speaker specific ML transform. The operator 'vec()' maps a matrix to a vector. The matrix $\mathbf{H}_{\texttt{d}}$ is of size $D(D+1) \times D(D+1)$ and the vector $\mathbf{c}_{\texttt{d}}$ is of size $D(D+1)$ for D-dimensional features.

[1] The subscripts `ml` and `d` have been used to distinguish ML and discriminative transforms, as both of them are involved in this section.

A simple form of the transformation can be obtained by restricting $\mathbf{H}_{\mathtt{d}}$ to be block-diagonal with each block being tied and restricting $\mathbf{c}_{\mathtt{d}}$ to yield a bias on the mean. In this case, the final adapted mean obtained using the MLLR-based DMT adaptation may be expressed as

$$\hat{\boldsymbol{\mu}}^{(s)} = \mathbf{A}_{\mathtt{d}}\hat{\boldsymbol{\mu}}_{\mathtt{ml}}^{(s)} + \mathbf{b}_{\mathtt{d}} = \mathbf{W}_{\mathtt{d}}\hat{\boldsymbol{\xi}}_{\mathtt{ml}}^{(s)} \tag{4.24}$$

where $\mathbf{W}_{\mathtt{d}} = [\mathbf{A}_{\mathtt{d}} \ \ \mathbf{b}_{\mathtt{d}}]$ is the DMT transform, and $\hat{\boldsymbol{\xi}}_{\mathtt{ml}}^{(s)} = [\hat{\boldsymbol{\mu}}_{\mathtt{ml}}^{(s)\mathtt{T}} \ \ 1]^{\mathtt{T}}$ with the speaker-adapted mean $\hat{\boldsymbol{\mu}}_{\mathtt{ml}}^{(s)}$ given by

$$\hat{\boldsymbol{\mu}}_{\mathtt{ml}}^{(s)} = \mathbf{A}_{\mathtt{ml}}^{(s)}\boldsymbol{\mu} + \mathbf{b}_{\mathtt{ml}} = \mathbf{W}_{\mathtt{ml}}^{(s)}\boldsymbol{\xi}. \tag{4.25}$$

In the above equation, $\mathbf{W}_{\mathtt{ml}}^{(s)} = [\mathbf{A}_{\mathtt{ml}}^{(s)} \quad \mathbf{b}_{\mathtt{ml}}]$ is the MLLR transform for speaker s, obtained as described in section 3.1.2.1. The above simplification leads the DMT adaptation to have the same form as the DLT adaptation and DMTs can be estimated in a similar manner as the DLT.

Considering the MPE criterion, the parameters of the DMT are estimated as

$$\hat{\mathbf{W}}_{\mathtt{d}} = \arg\max_{\mathbf{W}} \left\{ \sum_{s}\sum_{\mathcal{H}} P(\mathcal{H}|\mathbf{O}^{(s)}; \mathbf{W}, \mathbf{W}_{\mathtt{ml}}^{(s)}, \mathcal{M})\mathcal{A}(\mathcal{H}, \mathcal{H}^{(s)}) \right\} \tag{4.26}$$

This form of optimisation is related to the DLT optimisation in equation (4.2). The optimisation for the DMT objective function in equation (4.26) is done using a weak-sense auxiliary function and the parameter estimation turns into a slightly modified version of DLT transform estimation given in the previous section. However, as DMT is a speaker independent transform, data from all speakers are used to estimate it. Therefore the sufficient statistics are obtained by summing over over all training speakers, and they are accumulated using speaker-specific MLLR adapted models. The sufficient statistics for the DMT estimation can be thus written as

$$\begin{aligned}
\gamma_m(t_s) &= \gamma_m^{\mathtt{num}}(t_s) - \gamma_m^{\mathtt{den}}(t_s) + \alpha^{\mathtt{I}}\gamma_m^{\mathtt{ml}}(t_s) && (4.27)\\
\mathbf{G}_d &= \sum_{sm}\sum_{t_s}\gamma_m(t_s)\frac{\hat{\boldsymbol{\xi}}_{\mathtt{ml},m}^{(s)}\hat{\boldsymbol{\xi}}_{\mathtt{ml},m}^{(s)\mathtt{T}}}{\sigma_{md}^2} + \sum_{sm} D_m^{(s)}\frac{\hat{\boldsymbol{\xi}}_{\mathtt{ml},m}^{(s)}\hat{\boldsymbol{\xi}}_{\mathtt{ml},m}^{(s)\mathtt{T}}}{\sigma_{md}^2} && (4.28)\\
\mathbf{k}_d &= \sum_{sm}\sum_{t_s}\gamma_m(t_s)o_{t_sd}^{(s)}\frac{\hat{\boldsymbol{\xi}}_{\mathtt{ml},m}^{(s)}}{\sigma_{md}^2} + \sum_{sm} D_m^{(s)}\frac{\hat{\boldsymbol{\xi}}_{\mathtt{ml},m}^{(s)}\hat{\boldsymbol{\xi}}_{\mathtt{ml},m}^{(s)\mathtt{T}}\mathbf{w}_d}{\sigma_{md}^2} && (4.29)
\end{aligned}$$

where t_s is the time index for speaker s, and $\gamma_m^{\mathtt{num}}(t_s)$ and $\gamma_m^{\mathtt{den}}(t_s)$ are posterior occupancies for the component m being at time t_s. The smoothing constant $D_m^{(s)}$ can be computed as

$$D_m^{(s)} = E\sum_{t_s}\gamma_m^{\mathtt{den}}(t_s) \tag{4.30}$$

where E is a constant usually selected between 0.5 and 2.5 [180]. With these statistics, each row of the DMT is estimated by equation (4.21) as used for estimating DLTs.

In the same way as MLLR, DMTs can make use of multiple regression classes. An interesting aspect of DMTs is that the number of transform parameters can be made very large compared to the number of speaker-specific linear transforms. This is because all the acoustic model training data are used to estimate the DMT parameters, rather than using the data just from a specific speaker.

4.2 Discriminative Speaker Adaptive Training (DSAT)

As described in the previous chapter, speaker adaptive training (SAT) is an important technique to build a speech recognition system from non-homogeneous training data. In SAT, the speech and non-speech variabilities are modelled separately through canonical models and adaptation transforms. Originally, the canonical models and transforms were both estimated using the maximum likelihood criterion [5, 43]. However, state-of-the-art systems use discriminative training criteria such as minimum phone error (MPE). Therefore, the use of these discriminative criteria has also been investigated for estimating the canonical models and transforms in the SAT framework [118, 171, 180, 181]. Similarly, as discriminative linear transforms (DLTs) are highly sensitive to errors in the supervision hypothesis, an alternative approach uses ML transforms with discriminatively estimated canonical models [106, 181]. In the following sections, MLLR transforms and DLTs based discriminative speaker adaptive training (DSAT) schemes are described.

4.2.1 MLLR-based DSAT

The ML transform based DSAT is the most commonly used form of discriminative adaptive training for unsupervised adaptation tasks [106, 180, 181]. In this approach, ML-based transforms are used in conjunction with the discriminatively trained canonical models. In MLLR-based DSAT, the ML-SAT scheme is initially run as shown in algorithm 3 in section 3.2.1. A final set of speaker-specific MLLR transforms is estimated using the final ML canonical model set in equation (3.41). These transforms are then fixed and used for all subsequent discriminative canonical model updates. The canonical models are updated using discriminative criteria given the set of speaker transforms. This may be expressed for the MPE criterion as

$$\hat{\mathcal{M}}_{\mathtt{d}} = \arg\max_{\mathcal{M}} \left\{ \sum_{s} \sum_{\mathcal{H}} P(\mathcal{H}|\mathbf{O}^{(s)}; \mathbf{W}_{\mathtt{ml}}^{(s)}, \mathcal{M}) \mathcal{A}(\mathcal{H}, \mathcal{H}^{(s)}) \right\} \tag{4.31}$$

where $P(\mathcal{H}|\mathbf{O}^{(s)};\mathbf{W}_{\mathtt{ml}}^{(s)},\mathcal{M})$ is the posterior probability of hypothesis $\mathcal{H}$ for the given observation and transform for speaker s, $\mathcal{A}(\mathcal{H},\mathcal{H}^{(s)})$ is the raw phone accuracy of the hypothesis $\mathcal{H}$ for the given supervision $\mathcal{H}^{(s)}$, and $\mathbf{W}_{\mathtt{ml}}^{(s)}$ is the MLLR transform for speaker s estimated through equation (3.41) using final ML-SAT canonical models.

The optimisation of discriminative SAT objective function in equation (4.31) can be done by defining a weak-sense auxiliary function as done in section 2.3.4 for discriminative training of models. The auxiliary function for the discriminative canonical model update can be expressed as [180]

$$\begin{aligned}\mathcal{Q}(\hat{\mathcal{M}};\mathcal{M},\mathbb{W}) &= \mathcal{Q}^{\mathtt{num}}(\hat{\mathcal{M}};\mathcal{M},\mathbb{W}) - \mathcal{Q}^{\mathtt{den}}(\hat{\mathcal{M}};\mathcal{M},\mathbb{W}) + \mathcal{Q}^{\mathtt{sm}}(\hat{\mathcal{M}};\mathcal{M},\mathbb{W}) \\ &\quad + \mathcal{Q}^{\mathtt{I}}(\hat{\mathcal{M}};\mathcal{M},\mathbb{W}) \qquad (4.32)\end{aligned}$$

where $\mathbb{W}$ is the set of transforms for all S speakers. The subscript $\mathtt{ml}$ has been dropped from the transforms, as the optimisation for model parameters update being described is not dependent on whether the transforms are ML or discriminative estimates, and is thus applicable to both.

The numerator auxiliary function in the above equation can be expressed in the same form as the ML-SAT auxiliary function in equation (3.48) though using numerator occupancies as

$$\begin{aligned}\mathcal{Q}^{\mathtt{num}}(\hat{\mathcal{M}};\mathcal{M},\mathbb{W}) &= -\frac{1}{2}\sum_{sm}\sum_{t_s}\gamma_m^{\mathtt{num}}(t_s)\Big\{\log|\hat{\boldsymbol{\Sigma}}_m| \\ &\quad + \left(\mathbf{o}_{t_s}^{(s)} - \mathbf{W}^{(sr_m)}\hat{\boldsymbol{\xi}}_m\right)^{\mathtt{T}}\hat{\boldsymbol{\Sigma}}_m^{-1}\left(\mathbf{o}_{t_s}^{(s)} - \mathbf{W}^{(sr_m)}\hat{\boldsymbol{\xi}}_m\right)\Big\} \qquad (4.33)\end{aligned}$$

where $\mathbf{W}^{(sr_m)}$ is the MLLR transform for speaker s and regression base class r_m of the mth component. The numerator occupation $\gamma_m^{\mathtt{num}}(t_s)$ is already defined in equation (2.102) for the MPE criterion, and they are computed using current estimate of model parameters and transforms. The above numerator auxiliary function can be expressed in a sufficient statistics form for the *mean* update as

$$\begin{aligned}\mathcal{Q}^{\mathtt{num}}(\hat{\mathcal{M}};\mathcal{M},\mathbb{W}) &= \mathcal{G}^{\mathtt{sat}}(\mathcal{M};\boldsymbol{\Gamma}^{\mathtt{num}}) \\ &= -\frac{1}{2}\sum_m\left\{\hat{\boldsymbol{\mu}}_m^{\mathtt{T}}\mathbf{G}_m^{\mathtt{num}}\hat{\boldsymbol{\mu}}_m - 2\hat{\boldsymbol{\mu}}_m^{\mathtt{T}}\mathbf{k}_m^{\mathtt{num}}\right\} \qquad (4.34)\end{aligned}$$

where the sufficient statistics is given as

$$\boldsymbol{\Gamma}^{\mathtt{num}} = \{\mathbf{G}_m^{\mathtt{num}},\mathbf{k}_m^{\mathtt{num}}\} \qquad (4.35)$$

These are defined in the same way as for the maximum-likelihood case given in equation (3.49) and equation (3.50). The auxiliary function and statistics for the denominator term are also defined in the same form as numerator, however using the denominator occupation probability $\gamma_m^{\mathtt{den}}(t_s)$ as defined in equation (2.103).

The smoothing term in equation (4.36) for the mean update is given by [180]

$$\mathcal{Q}^{\mathtt{sm}}(\hat{\mathcal{M}};\mathcal{M},\mathbb{W}) = -\frac{1}{2}\sum_{sm} D_m \nu_m^{(s)} \left(\hat{\boldsymbol{\xi}}_m - \boldsymbol{\xi}_m\right)^{\mathtt{T}} \mathbf{W}^{(sr_m)\mathtt{T}} \boldsymbol{\Sigma}_m^{-1} \mathbf{W}^{(sr_m)} \left(\hat{\boldsymbol{\xi}}_m - \boldsymbol{\xi}_m\right) \tag{4.36}$$

where $\boldsymbol{\xi}_m$ represents the extended mean at the current model estimate and D_m is a smoothing factor. The smoothing for each speaker is usually made proportional to the amount of data available for the speaker, rather than using the same smoothing for all speakers. This is done by using a scaling constant $\nu_m^{(s)}$ in the above equation representing the proportion of the data for a speaker, which is generally taken as [199]

$$\nu_m^{(s)} = \frac{\sum_{t_s} \gamma_m^{\mathtt{num}}(t_s)}{\sum_{st_s} \gamma_m^{\mathtt{num}}(t_s)} \tag{4.37}$$

In terms of sufficient statistics, the auxiliary function for the smoothing term in equation (4.36) can be expressed as

$$\mathcal{Q}^{\mathtt{sm}}(\hat{\mathcal{M}};\mathcal{M},\mathbb{W}) = \mathcal{G}^{\mathtt{sat}}(\hat{\mathcal{M}};\boldsymbol{\Gamma}^{\mathtt{sm}}) \tag{4.38}$$

where

$$\boldsymbol{\Gamma}^{\mathtt{sm}} = \left\{D_m \mathbf{G}_m^{\mathtt{sm}}, D_m \mathbf{k}_m^{\mathtt{sm}}\right\} \tag{4.39}$$

$$\mathbf{G}_m^{\mathtt{sm}} = \sum_s \nu_m^{(s)} \mathbf{A}^{(sr_m)\mathtt{T}} \boldsymbol{\Sigma}_m^{-1} \mathbf{A}^{(sr_m)} \tag{4.40}$$

$$\mathbf{k}_m^{\mathtt{sm}} = \mathbf{G}_m^{\mathtt{sm}} \boldsymbol{\xi}_m \tag{4.41}$$

An I-smoothing prior [142] $p(\hat{\mathcal{M}}|\Phi)$ can be also used in a similar way as in the standard discriminative training of HMMs in section 2.3.4 [180]. The I-smoothing can be done to the ML statistics, which can be expressed in terms of sufficient statistics as

$$\mathcal{Q}^{\mathtt{I}}(\hat{\mathcal{M}};\mathcal{M},\mathbb{W}) = \mathcal{G}^{\mathtt{sat}}(\mathcal{M};\boldsymbol{\Gamma}^{\mathtt{I}}) \tag{4.42}$$

$$\boldsymbol{\Gamma}^{\mathtt{I}} = \left\{\mathbf{G}_m^{\mathtt{I}}, \mathbf{k}_m^{\mathtt{I}}\right\} \tag{4.43}$$

$$\mathbf{G}_m^{\mathtt{I}} = \alpha^{\mathtt{I}} \mathbf{G}_m^{\mathtt{ml}} \tag{4.44}$$

$$\mathbf{k}_m^{\mathtt{I}} = \alpha^{\mathtt{I}} \mathbf{k}_m^{\mathtt{ml}} \tag{4.45}$$

where $\alpha^{\mathtt{I}}$ determines the contribution of I-smoothing prior. The ML statistics for SAT are already defined in equation (3.49) and equation (3.50). In case of the MPE criterion based training, as being described here, an MMI prior can be used as the I-smoothing prior.

Defining each of the terms of the auxiliary function in equation (4.32) in terms of sufficient statistics as above, the overall sufficient statistics for the mean update is given by

$$\begin{aligned}\mathbf{G}_m &= \mathbf{G}_m^{\texttt{num}} - \mathbf{G}_m^{\texttt{den}} + D_m\mathbf{G}_m^{\texttt{sm}} + \mathbf{G}_m^{\texttt{I}} && (4.46)\\ \mathbf{k}_m &= \mathbf{k}_m^{\texttt{num}} - \mathbf{k}_m^{\texttt{den}} + D_m\mathbf{k}_m^{\texttt{sm}} + \mathbf{k}_m^{\texttt{I}} && (4.47)\end{aligned}$$

The new estimate of the mean vector can be thus obtained by maximising the auxiliary function in equation (4.32) as

$$\hat{\boldsymbol{\mu}}_m = \mathbf{G}_m^{-1}\mathbf{k}_m \tag{4.48}$$

As in the ML-SAT in section 3.2.1, covariance matrices are estimated only after the mean update. The auxiliary function for the covariance matrix update is derived from the same discriminative SAT model update auxiliary function in equation (4.32), by expressing its parts in sufficient statistics form relevant to the covariance update and ignoring other independent terms. First, the numerator part in equation (4.33) can be expressed using sufficient statistics for the covariance matrix update as

$$\begin{aligned}\mathcal{Q}^{\texttt{num}}(\hat{\mathcal{M}};\mathcal{M},\mathbb{W}) &= \mathcal{G}^{\texttt{sat}}(\hat{\mathcal{M}};\boldsymbol{\Gamma}^{\texttt{num}})\\ &= -\frac{1}{2}\sum_m\left\{\gamma_m^{\texttt{num}}\log\left|\hat{\boldsymbol{\Sigma}}_m\right| + \operatorname{tr}\left(\mathbf{L}_m^{\texttt{num}}\hat{\boldsymbol{\Sigma}}_m^{-1}\right)\right\} && (4.49)\end{aligned}$$

where the sufficient statistics are given as

$$\begin{aligned}\boldsymbol{\Gamma}^{\texttt{num}} &= \left\{\gamma_m^{\texttt{num}}, \mathbf{L}_m^{\texttt{num}}\right\} && (4.50)\\ \gamma_m^{\texttt{num}} &= \sum_s\sum_{t_s}\gamma_m^{\texttt{num}}(t_s) && (4.51)\\ \mathbf{L}_m^{\texttt{num}} &= \sum_s\sum_{t_s}\gamma_m^{\texttt{num}}(t_s)\left(\mathbf{o}_{t_s}^{(s)} - \mathbf{W}^{(sr_m)}\hat{\boldsymbol{\xi}}_m\right)\left(\mathbf{o}_{t_s}^{(s)} - \mathbf{W}^{(sr_m)}\hat{\boldsymbol{\xi}}_m\right)^{\texttt{T}} && (4.52)\end{aligned}$$

The denominator statistics can be also given in the same form. Similarly, the smoothing term for the covariance update is given by

$$\mathcal{Q}^{\texttt{sm}}(\hat{\mathcal{M}};\mathcal{M},\mathbb{W}) = \mathcal{G}^{\texttt{sat}}(\hat{\mathcal{M}};\boldsymbol{\Gamma}^{\texttt{sm}}) = -\frac{1}{2}\sum_m D_m\left\{\log\left|\hat{\boldsymbol{\Sigma}}_m\right| + \operatorname{tr}\left(\boldsymbol{\Sigma}_m\hat{\boldsymbol{\Sigma}}_m^{-1}\right)\right\} \tag{4.53}$$

with the sufficient statistics

$$\boldsymbol{\Gamma}^{\texttt{sm}} = \left\{D_m, D_m\boldsymbol{\Sigma}_m\right\} \tag{4.54}$$

where $\boldsymbol{\Sigma}_m$ is the covariance matrix of the mth component of the current canonical model. In this case also, the I-smoothing to ML statistics can be done as in the mean update. The I-smoothing sufficient statistics for the covariance update can be expressed as

$$\boldsymbol{\Gamma}^{\texttt{I}} = \left\{\alpha^{\texttt{I}}\gamma_m^{\texttt{ml}}, \alpha^{\texttt{I}}\mathbf{L}_m^{\texttt{ml}}\right\} \tag{4.55}$$

where $\alpha^{\mathtt{I}}$ controls the impact of the prior. In this way, once the terms of the auxiliary function in equation (4.32) are expressed in terms of sufficient statistics for the covariance update, the new estimate of the covariance can be obtained by maximising the same auxiliary function. This gives the new estimate of the covariance matrix as

$$\hat{\boldsymbol{\Sigma}}_m = \operatorname{diag}\left(\frac{\mathbf{L}_m^{\mathtt{num}} - \mathbf{L}_m^{\mathtt{den}} + D_m\boldsymbol{\Sigma}_m + \alpha^{\mathtt{I}}\mathbf{L}_m^{\mathtt{ml}}}{\gamma_m^{\mathtt{num}} - \gamma_m^{\mathtt{den}} + D_m + \alpha^{\mathtt{I}}\gamma_m^{\mathtt{ml}}}\right) \tag{4.56}$$

In this way, both means and covariances of canonical models can be estimated for the DSAT system.

The testing procedure for the MLLR-based DSAT has the same starting point as the ML-SAT scheme. The ML-SAT test procedure, as shown in figure 3.5 is first run to obtain the final ML-SAT speaker transforms. Based on these final ML-SAT transforms and the final DSAT canonical models, additional ML-based transform estimations can be performed to obtain the final testset transforms. These are then used with the final DSAT models to decode the test data.

The ML-transforms based DSAT scheme is applicable to both supervised and unsupervised tasks and has been found to yield consistent reductions in word error rate [106, 181]. In this approach, as ML-based speaker-specific transforms are used, they are relatively robust to errors in the supervision hypothesis, and thus the system can be used for unsupervised adaptation as well. However, an adaptive training scheme based on the discriminative linear transforms can be also formulated. The next section describes a DLT-based DSAT scheme, and the problems associated with it are also discussed.

4.2.2 DLT-based DSAT

As previously mentioned, it is possible to estimate both the transforms and the canonical models using discriminative criteria. Hence, the use of discriminative linear transforms in adaptive training has been investigated [118, 171, 180]. In the DLT-based DSAT scheme, again the ML-SAT procedure is initially run and a set of ML speaker transforms are estimated using the final ML canonical models, using ML-SAT algorithm 3 presented in section 3.2.1. Thereafter, the DLT estimation and model parameters update is performed in an interleaved fashion as given in algorithm 4. The DLT estimation involves optimising the same discriminative objective function as given in equation (4.2), and involves accumulating statistics as described in section 4.1.1. The model parameters are updated using equation (4.58) in a similar way to the standard MLLR-based DSAT scheme described in the previous section, thus giving the same update formulae though using DLTs in place of the MLLR transforms.

Step 1: Initialise canonical model set and transforms.

$\mathcal{M}_{\mathtt{d}} = \mathcal{M}_{\mathtt{ml}}$ ML Canonical Model

$\mathbf{W}_{\mathtt{d}}^{(s)} = \mathbf{W}_{\mathtt{ml}}^{(s)}$ ML-SAT Transforms

Step 2: Estimate DLT transforms for each speaker.

$$\mathbf{W}_{\mathtt{d}}^{(s)} = \arg\max_{\mathbf{W}} \left\{ \sum_{\mathcal{H}} P(\mathcal{H}|\mathbf{O}^{(s)}; \mathbf{W}, \mathcal{M}_{\mathtt{d}}) \mathcal{A}(\mathcal{H}, \mathcal{H}^{(s)}) \right\} \tag{4.57}$$

Step 3: Update model parameters.

$$\mathcal{M}_{\mathtt{d}} = \arg\max_{\mathcal{M}} \left\{ \sum_{s} \sum_{\mathcal{H}} P(\mathcal{H}|\mathbf{O}^{(s)}; \mathbf{W}_{\mathtt{d}}^{(s)}, \mathcal{M}) \mathcal{A}(\mathcal{H}, \mathcal{H}^{(s)}) \right\} \tag{4.58}$$

Step 4: Go to step (2) unless converged.

Algorithm 4: *The DLT-based DSAT algorithm*

The testing procedure for DLT-based DSAT again uses the ML-SAT testing procedure first as described in section 3.2.1, to obtain initial set of speaker transforms. A modified version of the DLT-DSAT training procedure given in the algorithm 4 is then run, omitting the model-update stage to obtain testset discriminative transforms corresponding to the successive training iterations. The testset DLTs corresponding to the final DSAT iteration are then used for decoding of test data.

As previously discussed, discriminative transforms are sensitive to errors in the supervision hypothesis. During training, the DLTs are estimated using the reference transcripts, so there are no supervision errors. If used in a supervised adaptation mode, DLTs can be robustly estimated and reductions in the error rate are obtained [180]. However, this is not the case for unsupervised adaptation. The errors in the supervision hypothesis in unsupervised mode of adaptation greatly degrade the performance of DLT-based DSAT scheme [147]. To reduce the impact of hypothesis errors, it is possible to use confidence scores and lattice-based adaptation as investigated in [180, 202]. Though these approaches yield slightly greater robustness to hypothesis errors, the improvements over MLLR-based DSAT are still normally small [180, 202].

4.3 Adaptive Training using Discriminative Mapping Transforms

As described in the last section, the DLT-based DSAT approach is not used with unsupervised adaptation, as the testset transforms are sensitive to supervision hypothesis errors. In practice, ML transforms are used in the adaptive training framework, and only canonical models are trained using discriminative criteria as described in section 4.2.1. This does not yield a full and consistent discriminative adaptive training framework. However, in the same way as discriminative training of canonical models leads to performance gains, if discriminative transforms could be *robustly* estimated, additional gains should be possible by using them in a SAT framework. This would also yield a full DSAT system, where both the transforms and canonical models are updated discriminatively. One of the contributions of this thesis is to address this problem, by using discriminative mapping transforms in the adaptive training framework. A DMT is a discriminatively estimated speaker-independent transform based on speaker-specific ML transforms, as described in section 4.1.2. The use of DMT in adaptive training framework makes it possible to use with unsupervised adaptation, as DMTs do not depend upon the testset supervision hypothesis. The DMT-based DSAT framework is described in algorithm 5. The algorithm can be compared to the DLT-based DSAT procedure described in algorithm 4, where the DLT estimation step has been replaced by the estimation of MLLR transforms and DMTs. It should be noted that each of the steps in algorithm 5 for estimating MLLR transforms, DMTs and updating model parameters is itself an iterative process involving estimation through the EM-algorithm.

The starting point for the DMT-based DSAT procedure is the same as the other DSAT approaches. The final ML-SAT models and corresponding speaker-specific MLLR transforms are used to initialise canonical models and ML transforms for the DMT-based DSAT system. An identity transform and zero bias is used for initialising the DMTs. The DMT-based DSAT procedure is iterated by estimating DMTs constrained on current ML transforms (Step 2), and updating the model parameters (Step 3) in an interleaved fashion as shown in algorithm 5. The iteration is stopped when the estimated parameters converge, or the desired number of iterations are completed.

Step 1: **Initialise canonical model set and transforms.**

$\mathcal{M}_{\mathtt{d}} = \mathcal{M}_{\mathtt{ml}}$, ML Canonical Models

$\mathbf{W}^{(s)}_{\mathtt{ml}}$: ML-SAT Transforms

$\mathbf{W}_{\mathtt{d}}$: $\mathbf{A}_{\mathtt{d}} = \mathbf{I}, \mathbf{b}_{\mathtt{d}} = \mathbf{0}$

Step 2: **Estimate DMTs.**

$$\mathbf{W}_{\mathtt{d}} = \arg\max_{\mathbf{W}} \left\{ \sum_{s} \sum_{\mathcal{H}} P(\mathcal{H}|\mathbf{O}^{(s)}; \mathbf{W}, \mathbf{W}^{(s)}_{\mathtt{ml}}, \mathcal{M}_{\mathtt{d}}) \mathcal{A}(\mathcal{H}, \mathcal{H}^{(s)}) \right\} \tag{4.59}$$

constrained to

$$\mathbf{W}^{(s)}_{\mathtt{ml}} = \arg\max_{\mathbf{W}} \left\{ \log p(\mathbf{O}^{(s)}|\mathcal{H}^{(s)}; \mathbf{W}_{\mathtt{d}}, \mathbf{W}, \mathcal{M}_{\mathtt{d}}) \right\} \tag{4.60}$$

Step 3: **Update model parameters.**

$$\mathcal{M}_{\mathtt{d}} = \arg\max_{\mathcal{M}} \left\{ \sum_{s} \sum_{\mathcal{H}} P(\mathcal{H}|\mathbf{O}^{(s)}; \mathbf{W}_{\mathtt{d}}, \mathbf{W}^{(s)}_{\mathtt{ml}}, \mathcal{M}) \mathcal{A}(\mathcal{H}, \mathcal{H}^{(s)}) \right\} \tag{4.61}$$

Step 4: **Go to step (2) unless converged.**

Algorithm 5: *The DMT-based DSAT algorithm*

In the DMT-based DSAT, DMTs are estimated using equation (4.59), constrained to MLLR transforms in (4.60). The optimisation of the DMT objective function is described in section 4.1.2, and the sufficient statistics required for its estimation are given in equations (4.28) and (4.29). The only difference is that the models used are the available canonical models at the current iteration of the DSAT procedure.

The MLLR transforms for the DMT-based DSAT are estimated using equation (4.60), given the current set of canonical models, DMTs and speaker-specific MLLR transforms. The MLLR transforms are estimated by maximising the objective function in equation (4.60) through the EM algorithm using an auxiliary function given as

$$\mathcal{Q}(\hat{\mathbf{W}}^{(sr)}_{\mathtt{ml}}; \mathbf{W}^{(sr)}_{\mathtt{ml}}) = -\frac{1}{2} \sum_{t_s, m \in \mathcal{R}} \gamma^{\mathtt{ml}}_{m}(t_s) \log \mathcal{N}(\mathbf{o}^{(s)}_{t_s}; \mathbf{A}^{(\rho_m)}_{\mathtt{d}} \hat{\mathbf{W}}^{(sr_m)}_{\mathtt{ml}} \boldsymbol{\xi}_m + \mathbf{b}^{(\rho_m)}_{\mathtt{d}}, \boldsymbol{\Sigma}_m) \tag{4.62}$$

where $\gamma^{\mathtt{ml}}_{m}(t_s)$ are component occupancies computed using current MLLR transforms $\mathbf{W}^{(sr_m)}_{\mathtt{ml}}$, DMTs $\mathbf{W}_{\mathtt{d}}{}^{(\rho_m)}$ and model parameters $\mathcal{M}_{\mathtt{d}}$. In the equation, r_m and ρ_m represent the regression base class of mixture component m for MLLR transforms and DMTs, respectively, and $\mathcal{R}$ represents a set of all mixture components belonging to regression class r of the MLLR transform. The above auxiliary function can be re-expressed by ignoring the terms indepedent of $\hat{\mathbf{W}}_{\mathtt{ml}}^{(sr)}$ as

$$\begin{aligned}
&\mathcal{Q}(\hat{\mathbf{W}}_{\mathtt{ml}}^{(sr)};\mathbf{W}_{\mathtt{ml}}^{(sr)}) \\
&\quad= -\frac{1}{2}\sum_{t_s,m\in\mathcal{R}}\gamma_m^{\mathtt{ml}}(t_s)\left(\mathbf{o}_{t_s}^{(s)}-\mathbf{A}_{\mathtt{d}}^{(\rho_m)}\hat{\mathbf{W}}_{\mathtt{ml}}^{(sr_m)}\boldsymbol{\xi}_m-\mathbf{b}_{\mathtt{d}}^{(\rho_m)}\right)^{\mathtt{T}}\boldsymbol{\Sigma}_m^{-1}\left(\mathbf{o}_{t_s}^{(s)}-\mathbf{A}_{\mathtt{d}}^{(\rho_m)}\hat{\mathbf{W}}_{\mathtt{ml}}^{(sr_m)}\boldsymbol{\xi}_m-\mathbf{b}_{\mathtt{d}}^{(\rho_m)}\right) \\
&\quad= -\frac{1}{2}\sum_{t_s,m\in\mathcal{R}}\gamma_m^{\mathtt{ml}}(t_s)\left(\tilde{\mathbf{o}}_{t_s}^{(s)}-\mathbf{A}_{\mathtt{d}}^{(\rho_m)}\hat{\mathbf{W}}_{\mathtt{ml}}^{(sr_m)}\boldsymbol{\xi}_m\right)^{\mathtt{T}}\boldsymbol{\Sigma}_m^{-1}\left(\tilde{\mathbf{o}}_{t_s}^{(s)}-\mathbf{A}_{\mathtt{d}}^{(\rho_m)}\hat{\mathbf{W}}_{\mathtt{ml}}^{(sr_m)}\boldsymbol{\xi}_m\right) \qquad (4.63)
\end{aligned}$$

where $\tilde{\mathbf{o}}_{t_sm}^{(s)}=\mathbf{o}_{t_s}^{(s)}-\mathbf{b}_{\mathtt{d}}^{(\rho_m)}$. The above equation can be rewritten by keeping only the terms dependent on $\hat{\mathbf{W}}_{\mathtt{ml}}^{(sr)}$ as

$$\begin{aligned}
\mathcal{Q}(\hat{\mathbf{W}}_{\mathtt{ml}}^{(sr)};\mathbf{W}_{\mathtt{ml}}^{(sr)}) \;=\; -\frac{1}{2}\sum_{t_s,m\in\mathcal{R}}\gamma_m^{\mathtt{ml}}(t_s)\Big(&-2\tilde{\mathbf{o}}_{t_sm}^{(s)\mathtt{T}}\boldsymbol{\Sigma}_m^{-1}\mathbf{A}_{\mathtt{d}}^{(\rho_m)}\hat{\mathbf{W}}_{\mathtt{ml}}^{(sr_m)}\boldsymbol{\xi}_m \\
&+\boldsymbol{\xi}_m^{\mathtt{T}}\hat{\mathbf{W}}_{\mathtt{ml}}^{(sr_m)\mathtt{T}}\mathbf{A}_{\mathtt{d}}^{(\rho_m)\mathtt{T}}\boldsymbol{\Sigma}_m^{-1}\mathbf{A}_{\mathtt{d}}^{(\rho_m)}\mathbf{A}_{\mathtt{ml}}^{(sr_m)}\boldsymbol{\xi}_m\Big) \qquad (4.64)
\end{aligned}$$

Taking the derivative with respect to $\hat{\mathbf{W}}_{\mathtt{ml}}^{(sr_m)}$ and equating to zero leads to

$$\sum_{t_s,m\in\mathcal{R}}\gamma_m^{\mathtt{ml}}(t_s)\mathbf{A}_{\mathtt{d}}^{(\rho_m)\mathtt{T}}\boldsymbol{\Sigma}_m^{-1}\tilde{\mathbf{o}}_{t_sm}^{(s)}\boldsymbol{\xi}_m^{\mathtt{T}}=\sum_{t_s,m\in\mathcal{R}}\gamma_m^{\mathtt{ml}}(t_s)\mathbf{A}_{\mathtt{d}}^{(\rho_m)\mathtt{T}}\boldsymbol{\Sigma}_m^{-1}\mathbf{A}_{\mathtt{d}}^{(\rho_m)}\hat{\mathbf{W}}_{\mathtt{ml}}^{(sr_m)}\boldsymbol{\xi}_m\boldsymbol{\xi}_m^{\mathtt{T}} \qquad (4.65)$$

Representing left-hand side of equation (4.65) as

$$\mathbf{Z}=\sum_{t_s,m\in\mathcal{R}}\gamma_m^{\mathtt{ml}}(t_s)\mathbf{A}_{\mathtt{d}}^{(\rho_m)\mathtt{T}}\boldsymbol{\Sigma}_m^{-1}\tilde{\mathbf{o}}_{t_sm}^{(s)}\boldsymbol{\xi}_m^{\mathtt{T}} \qquad (4.66)$$

and defining

$$\begin{aligned}
\mathbf{V}_m &= \sum_{t_s}\gamma_m^{\mathtt{ml}}(t_s)\mathbf{A}_{\mathtt{d}}^{(\rho_m)\mathtt{T}}\boldsymbol{\Sigma}_m^{-1}\mathbf{A}_{\mathtt{d}}^{(\rho_m)} \qquad (4.67)\\
\mathbf{Y}_m &= \boldsymbol{\xi}_m\boldsymbol{\xi}_m^{\mathtt{T}} \qquad (4.68)
\end{aligned}$$

for right-hand side, the equation can be re-expressed as[1]

$$\mathrm{vec}(\mathbf{Z})=\left(\sum_m\mathrm{kron}(\mathbf{V}_m,\mathbf{Y}_m)\right)\mathrm{vec}(\hat{\mathbf{W}}_{\mathtt{ml}}^{(sr)}) \qquad (4.70)$$

where 'vec' represents vectorisation of a matrix and 'kron' is Kronecker tensor product. The estimation of $\hat{\mathbf{W}}_{\mathtt{ml}}^{(sr)}$ through a direct solution of equation (4.70) involves inverting a $(D^2+D)\times(D^2+D)$ matrix and thus is computationally expensive. These equations for estimating MLLR

[1] The following identity is used [80]

$$\mathrm{vec}(\mathbf{C})=\mathrm{vec}(\mathbf{AXB})=(\mathbf{A}\otimes\mathbf{B}^{\mathtt{T}})\mathrm{vec}(\mathbf{X}) \qquad (4.69)$$

where $\otimes$ is the Kronecker product and 'vec' is the vectorisation of the matrix formed by stacking rows into a single vector.

transforms given DMTs can be compared to the MLLR estimation equation for full covariance matrices in [51, 103] [1]. However, in the standard MLLR estimation described in [51, 103], for diagonal covariance matrices each row of transforms can be estimated efficiently, that involves inverting matrices of size $(D+1) \times (D+1)$ only. By contrast, in this case, even though the covariance matrices are diagonal, the multiplication with a full $\mathbf{A}_{\mathtt{d}}^{(\rho_m)}$ matrix as seen in equation (4.67) makes the result a full matrix, and thus the transform estimation equation involves tensor products. Thus even for the diagonal covariance matrices, the estimation of MLLR transforms through equation (4.60) is computationally very expensive. It should be noted that when a diagonal transform $\mathbf{A}_{\mathtt{d}}^{(\rho_m)}$ is used along with the diagonal covariance matrices, a simple solution for estimating each row of the transform can be still obtained that involves inverting matrices of size $(D+1) \times (D+1)$ only. In the experiments in this work, however, a full DMT is used, and to deal with the computational complexities of the MLLR transform estimation in the DMT-based DSAT procedure, further assumptions are made as described later in this section.

Once the MLLR transforms and DMTs are estimated, the canonical models are updated using equation (4.61). By using equations (4.24) and (4.25), it is possible to combine the effects of the DMT and MLLR transform into a single linear transform of the means for each speaker. The mean of mth Gaussian component is first adapted by an MLLR transform and then DMT to yield the final adapted mean

$$\hat{\boldsymbol{\mu}}_m^{(s)} = \mathbf{A}_{\mathtt{d}}^{(\rho_m)} \mathbf{W}_{\mathtt{ml}}^{(sr_m)} \boldsymbol{\xi}_m + \mathbf{b}_{\mathtt{d}}^{(\rho_m)} \tag{4.72}$$

where r_m and ρ_m represent the regression base class of component m for MLLR transforms and DMTs, respectively, and $\boldsymbol{\xi}_m = [\boldsymbol{\mu}_m^{\mathtt{T}} \;\; 1]^{\mathtt{T}}$ is the extended mean vector for component m. $\mathbf{W}_{\mathtt{d}}^{(\rho_m)} = [\mathbf{A}_{\mathtt{d}}^{(\rho_m)} \;\; \mathbf{b}_{\mathtt{d}}^{(\rho_m)}]$ and $\mathbf{W}_{\mathtt{ml}}^{(sr_m)} = [\mathbf{A}_{\mathtt{ml}}^{(sr_m)} \quad \mathbf{b}_{\mathtt{ml}}^{(sr_m)}]$ are the DMT and MLLR transforms. The above expression can be re-expressed as

$$\begin{aligned}\hat{\boldsymbol{\mu}}_m^{(s)} &= \mathbf{A}_{\mathtt{d}}^{(\rho_m)} \left(\mathbf{A}_{\mathtt{ml}}^{(sr_m)} \boldsymbol{\mu}_m + \mathbf{b}_{\mathtt{ml}}^{(sr_m)} \right) + \mathbf{b}_{\mathtt{d}}^{(\rho_m)} \\ &= \mathbf{A}_{\mathtt{d}}^{(\rho_m)} \mathbf{A}_{\mathtt{ml}}^{(sr_m)} \boldsymbol{\mu}_m + \left(\mathbf{A}_{\mathtt{d}}^{(\rho_m)} \mathbf{b}_{\mathtt{ml}}^{(sr_m)} + \mathbf{b}_{\mathtt{d}}^{(\rho_m)} \right)\end{aligned} \tag{4.73}$$

Thus the MLLR and DMT adaptation for a mixture component can be combined into a single

[1] The equation for estimating the standard MLLR transform for the full covariance matrices is given by [51, 103]

$$\sum_{t_s, m \in \mathcal{R}} \gamma_m^{\mathtt{ml}}(t_s) \boldsymbol{\Sigma}_m^{-1} \mathbf{o}_{t_s}^{(s)} \boldsymbol{\xi}_m^{\mathtt{T}} = \sum_{t_s, m \in \mathcal{R}} \gamma_m^{\mathtt{ml}}(t_s) \boldsymbol{\Sigma}_m^{-1} \hat{\mathbf{W}}_{\mathtt{ml}}^{(sr_m)} \boldsymbol{\xi}_m \boldsymbol{\xi}_m^{\mathtt{T}} \tag{4.71}$$

This should be compared to equation (4.65).

transform with parameters given as

$$\mathbf{A}_{\mathtt{d}}^{(s)} = \mathbf{A}_{\mathtt{d}}^{(\rho_m)}\mathbf{A}_{\mathtt{ml}}^{(sr_m)} \tag{4.74}$$

$$\mathbf{b}_{\mathtt{d}}^{(s)} = \mathbf{A}_{\mathtt{d}}^{(\rho_m)}\mathbf{b}_{\mathtt{ml}}^{(sr_m)} + \mathbf{b}_{\mathtt{d}}^{(\rho_m)} \tag{4.75}$$

This allows to use the standard procedure for discriminative canonical model estimation as described in section 4.2.1, and the model update equations can be obtained by replacing the ML transforms parameters with the transform parameters in equations (4.74) and (4.75). The mean and variance update equations are given in equations (4.48) and (4.56).

The recognition procedure for the DMT-based DSAT system follows its training procedure, however only MLLR transforms need to be re-estimated for a testset. The evaluation procedure for the DMT-based DSAT scheme is shown in figure 4.1. First, the ML-SAT evaluation procedure is followed and the final MLLR transforms for the testset are obtained as shown in figure 3.5. The DMT-based DSAT procedure is started with these ML transforms and identity DMTs. Using the canonical models of the DMT-based DSAT system for each iteration, the testset MLLR transforms are re-estimated given the DMTs and the test set MLLR transforms from the previous iteration. Once the final test set MLLR transforms are obtained, they are used along with the canonical models and DMTs from the final iteration of the DSAT system for decoding the test data.

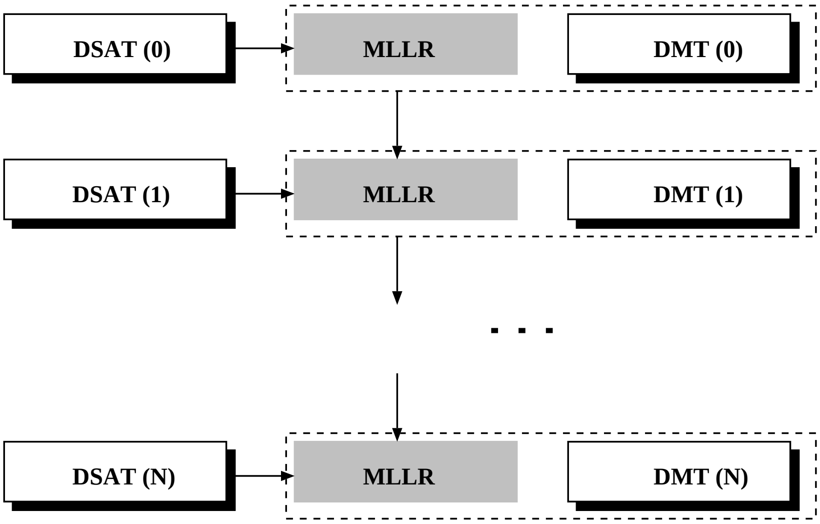


Figure 4.1: A recognition setup for test data using the DMT-based DSAT system. DSAT(k) and DMT(k) represent canonical models and DMTs from the kth iteration of the DSAT procedure. Only the shaded blocks (MLLR transforms) need to be estimated for the test data, others are estimated during the training of the DMT-based DSAT system.

In the DMT-based DSAT procedure described above, other forms of MLLR transform contraints can be also used for estimating DMTs. In the experiments in this work, the MLLR transforms for the DMT-based DSAT iterations are estimated through the standard MLLR estimation procedure described in section 3.1.2.1. This leads to the sufficient statistics given in equations (3.10) and (3.11) for MLLR transforms, however the posterior occupations for components based on the current estimates of MLLR transforms and DMTs are used. The advantage of this approach is that it avoids the computational complexities involved with estimating MLLR transforms through equation (4.60) leading to equation (4.65). However, this estimation approach does not fully consider the current estimates of DMTs in accumulating sufficient statistics for MLLR transforms estimation, as only the occupation probabilities are based on the current estimates of both MLLR transforms and DMTs. The recognition procedure in this case also is same as described above and shown in figure 4.1.

In an another alternative DMT-based DSAT scheme, the MLLR transforms can be also kept fixed while updating the canonical models and DMTs only. This is similar to the commonly used DSAT procedure described in section 4.2.1 where ML-transforms are kept fixed while updating only the canonical models discriminatively. When the final canonical models are obtained, MLLR transforms are estimated using the standard procedure given current estimates of MLLR transforms and DMTs. The final set of DMTs are trained with these estimated MLLRs and final canonical models. This method is simple and computationally inexpensive, however MLLR transforms used to estimate DMTs (and model parameters) are not updated at each iteration of the DSAT procedure, and thus it may not lead to the best possible estimates for DMTs. In this case also, for recognition of test data, the ML-SAT evaluation procedure as shown in figure 3.5 is run to obtain final ML-SAT test set transforms. The final DSAT test set MLLR transforms are obtained using the canonical models from the last iteration of the DMT-based DSAT, following the same procedure for estimating final iteration MLLR transforms as in training.

In the DMT-based DSAT system, DMTs used are speaker-independent transforms, and the same DMTs are used during training and test. There is no need to re-estimate the DMTs on testset and thus they are not affected by the testset supervision hypothesis errors. Therefore the DMT-based DSAT scheme should be able to deal with the unsupervised adaptation.

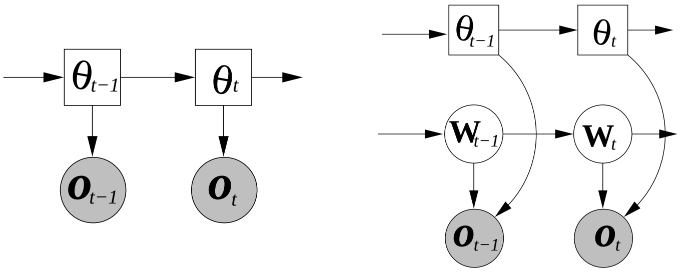
4.4 Summary

This chapter has presented techniques for discriminative adaptation and adaptive training. To overcome the limitations of the conventional maximum likelihood based adaptation, discriminative criteria such as minimum phone error can be used to estimate the transforms.

However, discriminative linear transforms (DLTs) are biased towards the supervision hypothesis and very sensitive to any errors in the supervision hypothesis. This problem can be dealt by using a discriminative mapping transform (DMT), which is a speaker-independent transform applied to speaker-specific ML transforms. Both the forms and the estimation of DLTs and DMTs have been described. Similarly, adaptive training can be performed using discriminative criteria to estimate both the canonical models and transforms. Discriminative adaptive training has been described using both MLLR and DLT. However, maximum-likelihood transforms are commonly preferred in the discriminative adaptive training framework, due to the same problem of sensitivity of discriminative transforms to errors in the supervision hypothesis. To deal with this issue, a DMT-based adaptive training scheme was proposed to give a complete discriminative adaptive training framework that can be effectively used for unsupervised adaptation as well.

Chapter 5
Bayesian Adaptive Training and Inference

As seen in chapter 3, adaptive training is an important technique when building speech recognition systems on non-homogeneous data. The adaptive training scheme yields canonical models and a set of speaker-specific transforms. However a major problem with the adaptive training approach is that this canonical models must always be used in conjunction with the transforms for recognition [40]. When no transform is available or the transform is poorly estimated, the performance of the system may be degraded. This problem may occur for online adaptation, for example, where there is only a small amount of adaptation data and reliable estimates of the transforms cannot be obtained. This issue is addressed by formulating a Bayesian framework for adaptive training where both the model parameters and the transforms are regarded as random variables [43, 170, 200]. In this chapter, the form of Bayesian adaptive training and inference based on the likelihood criterion is described. Adaptive training is first described from a Bayesian perspective in section 5.1. This is followed by several approximation schemes used for Bayesian inference in section 5.2. Thereafter, an expectation propagation based Bayesian inference scheme for adaptive training is proposed in section 5.3.



(a) A Standard HMM (b) An Adaptive HMM

Figure 5.1: The dynamic Bayesian networks (DBNs) of a standard HMM and an adaptive HMM

5.1 Bayesian Adaptive Training

In adaptive training, speech and non-speech variabilities are modelled separately, using canonical models and a set of transforms. In a Bayesian framework for adaptive training [43, 200], both the model parameters and the transforms are regarded as random variables. The interaction of the model parameters and transforms with the observations can be seen in a dynamic Bayesian network for adaptive training shown in figure 5.1(b). The DBN for a standard HMM is also shown in figure 5.1(a). In the DBN for the standard HMM, the observation at time t depends on the component θ_t, and is conditionally independent of the observations or components at any other time, given the component θ_t. In contrast, in the DBN for the adaptive HMM, the observation at the time t depends both on the component θ_t and the transform $\mathbf{W}_t$ at time t. Moreover, the transforms in an adaptive HMM are forced to be constant for each homogeneous block. This is expressed for one homogeneous block as

$$p(\mathbf{W}_t|\mathbf{W}_{t-1}) = \delta(\mathbf{W}_t - \mathbf{W}_{t-1}) \tag{5.1}$$

where $\delta(.)$ represents a Dirac-delta distribution. In this section, Bayesian adaptive training based on maximum-likelihood is described following [43, 200].

In the Bayesian framework for adaptive training, as both the canonical models and the transforms are regarded as random variables, the likelihood of the training data is given by a marginalisation over the distribution of the model parameters. Therefore, the marginal likelihood for the observation set $\mathbb{O} = \{\mathbf{O}^{(1)}, \ldots, \mathbf{O}^{(S)}\}$ from all S speakers is given by

$$p(\mathbb{O}|\mathbb{H}) = \int_{\mathcal{M}} p(\mathbb{O}|\mathbb{H}, \mathcal{M})p(\mathcal{M}|\Phi)\, d\mathcal{M} \tag{5.2}$$

where $\mathbb{H} = \{\mathcal{H}^{(1)}, \ldots, \mathcal{H}^{(S)}\}$ is the set of transcripts for corresponding observations, and $p(\mathcal{M}|\Phi)$ is the prior distribution for the canonical model parameters $\mathcal{M}$ with Φ as the hyper-parameters of the prior. Each homogeneous block of data $\mathbf{O}^{(s)}$ is regarded as conditionally independent of all others. Thus the likelihood of all blocks of the data, inside the integral in the above equation, can be expressed as

$$p(\mathbb{O}|\mathbb{H},\mathcal{M}) = \prod_{s=1}^{S}\int_{\mathbf{W}} p(\mathbf{O}^{(s)}|\mathcal{H}^{(s)},\mathcal{M},\mathbf{W})p(\mathbf{W}|\phi)\,d\mathbf{W} \tag{5.3}$$

where $p(\mathbf{W}|\phi)$ is a prior distribution over the transform parameters with hyperparameters ϕ. The likelihood of each homogeneous block of data, used inside the integral in the above equation, is given as

$$p(\mathbf{O}^{(s)}|\mathcal{H}^{(s)},\mathcal{M},\mathbf{W}) = \sum_{\boldsymbol{\theta}} P(\boldsymbol{\theta}|\mathcal{H}^{(s)},\mathcal{M})\prod_{t} p(\mathbf{o}_t^{(s)}|\mathcal{M},\mathbf{W},\theta_t) \tag{5.4}$$

where $P(\boldsymbol{\theta}|\mathcal{H}^{(s)},\mathcal{M})$ is the probability of component sequence $\boldsymbol{\theta}$, and $p(\mathbf{o}_t^{(s)}|\mathcal{M},\mathbf{W},\theta_t)$ is the likelihood of observation vector $\mathbf{o}_t^{(s)}$ given the Gaussian component θ_t at the time t.

Thus, Bayesian adaptive training involves the prior distributions over the model parameters as well as the transforms. They should be estimated from the training data. However, the forms of the prior distributions are first defined before estimating them. The form of the prior is generally limited to a conjugate prior. Once the forms of the priors are defined, the hyperparameters of these priors are estimated by maximising the marginal likelihood in equation (5.2), using an empirical Bayes approach, as described next.

The hyperparameters for the model prior $p(\mathcal{M}|\Phi)$ are estimated by maximising a lower bound to the marginal likelihood in equation (5.2), as its direct optimisation is impractical [200]. The lower bound to the marginal likelihood in equation (5.2) is obtained by introducing a variational distribution $q(\mathcal{M})$ and using Jensen's inequality as

$$\begin{aligned}\log p(\mathbb{O}|\mathbb{H}) &\geq \left\langle \log \frac{p(\mathbb{O}|\mathbb{H},\mathcal{M})p(\mathcal{M}|\Phi)}{q(\mathcal{M})}\right\rangle_{q(\mathcal{M})} \\ &= \big\langle \log p(\mathbb{O}|\mathbb{H},\mathcal{M})\big\rangle_{q(\mathcal{M})} - \mathtt{KL}\big(q(\mathcal{M})||p(\mathcal{M}|\Phi)\big)\end{aligned} \tag{5.5}$$

where $\mathtt{KL}(\cdot||\cdot)$ is the Kullback-Leibler (KL) divergence between the two distributions [1] . The inequality in the above equation turns out to be an equality when

$$q(\mathcal{M}) = p(\mathcal{M}|\mathbb{O},\mathbb{H}) \tag{5.6}$$

The maximisation of the lower-bound in equation (5.5) with respect to model prior hyperparameters Φ is equivalent to minimising the KL-divergence between $q(\mathcal{M})$ and $p(\mathcal{M}|\Phi)$.

[1] The KL divergence between two distributions is defined as

$$\mathtt{KL}\big(q(z)||p(z)\big) = \int_z q(z)\log\frac{q(z)}{p(z)}$$

Therefore, the empirical Bayesian estimate of the canonical model prior is given by [199]

$$p(\mathcal{M}|\Phi) = q(\mathcal{M}) = p(\mathcal{M}|\mathbb{O},\mathbb{H}) \tag{5.7}$$

The required posterior $p(\mathcal{M}|\mathbb{O},\mathbb{H})$ can be estimated by maximising $\log p(\mathbb{O}|\mathbb{H})$ with respect to $p(\mathcal{M}|\mathbb{O},\mathbb{H})$ for a non-informative prior.

Similarly, the hyperparameters for the transform prior $p(\mathbf{W}|\phi)$ are estimated by maximising the marginal likelihood. First, a lower-bound of the conditional likelihood for a given model set as defined in equation (5.3) for independent blocks of data is obtained. The lower-bound to the conditional likelihood is found by introducing a variational distribution $q^{(s)}(\mathbf{W})$ and applying Jensen's inequality as

$$\begin{aligned}\log p(\mathbb{O}|\mathbb{H},\mathcal{M}) &\geq \sum_{s=1}^{S}\left\langle \log \frac{p(\mathbf{O}^{(s)}|\mathcal{H}^{(s)},\mathcal{M},\mathbf{W})p(\mathbf{W}|\phi)}{q^{(s)}(\mathbf{W})}\right\rangle_{q^{(s)}(\mathbf{W})} \\ &= \sum_{s=1}^{S}\left\langle \log p(\mathbf{O}^{(s)}|\mathcal{H}^{(s)},\mathcal{M},\mathbf{W})\right\rangle_{q^{(s)}(\mathbf{W})} - \sum_{s=1}^{S}\mathtt{KL}\left(q^{(s)}(\mathbf{W})||p(\mathbf{W}|\phi)\right) \quad (5.8)\end{aligned}$$

The above lower-bound is maximum when

$$q^{(s)}(\mathbf{W}) = p(\mathbf{W}|\mathbf{O}^{(s)},\mathcal{H}^{(s)},\mathcal{M}) \tag{5.9}$$

The transform variational distribution $q^{(s)}(\mathbf{W})$ in above equation is associated with each homogeneous block of data. However, the transform prior $p(\mathbf{W}|\phi)$ is taken as independent of the acoustic conditions. Furthermore, the above equations are for a given specific model set, and if the distribution over model parameters is to be considered, the transform posterior estimate should also consider marginalisation over model parameters. This makes it difficult to directly minimise the KL divergence included in equation (5.8). Thus further assumptions are made to estimate the transform prior distribution. A sufficient amount of data is assumed during *training*, such that the posteriors of the model and transforms parameters reduce to the Dirac-delta distributions as

$$\begin{aligned} p(\mathcal{M}|\mathbb{O},\mathbb{H}) &\approx \delta(\mathcal{M}-\hat{\mathcal{M}}_{\mathtt{ml}}) \quad (5.10)\\ p(\mathbf{W}|\mathbf{O}^{(s)},\mathcal{H}^{(s)}) &\approx \delta(\mathbf{W}-\hat{\mathbf{W}}^{(s)}_{\mathtt{ml}}) \quad (5.11)\end{aligned}$$

where $\hat{\mathbf{W}}^{(s)}_{\mathtt{ml}}$ and $\hat{\mathcal{M}}_{\mathtt{ml}}$ are the point estimates of the transform for the homogeneous block s and canonical models, respectively. This yields the canonical model prior as

$$p(\mathcal{M}|\Phi) \approx \delta(\mathcal{M}-\hat{\mathcal{M}}_{\mathtt{ml}}) \tag{5.12}$$

The transform estimate for each acoustic condition s becomes an ML estimate $\hat{\mathbf{W}}^{(s)}_{\texttt{ml}}$ given by

$$\hat{\mathbf{W}}^{(s)}_{\texttt{ml}} = \arg\max_{\mathbf{W}} \left\{ p(\mathbf{O}^{(s)}|\mathcal{H}^{(s)}, \hat{\mathcal{M}}, \mathbf{W}) \right\} \tag{5.13}$$

These point estimates of the transforms are used to find the point estimate of the canonical models as well as the hyperparameters of the prior distribution of the transform $p(\mathbf{W}|\phi)$. The point estimate of the canonical model is given as

$$\hat{\mathcal{M}}_{\texttt{ml}} = \arg\max_{\mathcal{M}} \left\{ p(\mathbb{O}|\mathbb{H}, \mathcal{M}, \hat{\mathbb{W}}_{\texttt{ml}}) \right\} \tag{5.14}$$

where $\hat{\mathbb{W}}_{\texttt{ml}} = \left\{ \hat{\mathbf{W}}^{(1)}_{\texttt{ml}}, \dots, \hat{\mathbf{W}}^{(S)}_{\texttt{ml}} \right\}$ is the set of the transforms for all S homogeneous blocks. Similarly, using $p(\mathcal{M}|\mathbb{O}, \mathbb{H}) \approx \delta(\mathcal{M} - \hat{\mathcal{M}}_{\texttt{ml}})$ and $p(\mathbf{W}|\mathbf{O}^{(s)}, \mathcal{H}^{(s)}) \approx \delta(\mathbf{W} - \hat{\mathbf{W}}^{(s)}_{\texttt{ml}})$, the lower bound in equation (5.8) can be re-expressed as

$$\begin{aligned} &\log p(\mathbb{O}|\mathbb{H}, \hat{\mathcal{M}}_{\texttt{ml}}) \\ &\geq \sum_{s=1}^{S} \left(\log p(\mathbf{O}^{(s)}|\mathcal{H}^{(s)}, \hat{\mathcal{M}}_{\texttt{ml}}, \hat{\mathbf{W}}^{(s)}_{\texttt{ml}}) + \texttt{H}\left(\delta(\mathbf{W} - \hat{\mathbf{W}}^{(s)}_{\texttt{ml}})\right) + \log p(\hat{\mathbf{W}}^{(s)}_{\texttt{ml}}|\phi) \right) \end{aligned} \tag{5.15}$$

where $\texttt{H}(.)$ is the entropy of a function. Maximisation of above auxiliary function ignoring the $-\infty$ entropy term[1] with respect to hyperparameters ϕ of the transform prior leads to

$$\hat{\phi} = \arg\max_{\phi} \left\{ \sum_{s=1}^{S} \log p(\hat{\mathbf{W}}^{(s)}_{\texttt{ml}}|\phi) \right\} \tag{5.16}$$

Therefore, Bayesian adaptive training under the sufficient data assumption leads to a point estimate of the canonical model as in equation (5.14) and a non-point transform prior distribution with hyperparameters given in equation (5.16). It is thus similar to the standard SAT procedure described in section 3.2.1 except that a transform prior is also estimated. Once the canonical model and the prior distribution for the transform are obtained from the training data, they can be used for Bayesian adaptive inference, as described in the next section.

5.2 Bayesian Adaptive Inference

Bayesian adaptive inference attempts to find the best hypothesis by using the marginal likelihood $p(\mathbf{O}|\mathcal{H})$ and the language model score $P(\mathcal{H})$ as [200]

$$\hat{\mathcal{H}} = \arg\max_{\mathcal{H}} \left\{ p(\mathbf{O}|\mathcal{H}) P(\mathcal{H}) \right\} \tag{5.17}$$

[1] The entropy of a delta function is $-\infty$.

The inference equation is same as the standard decoding equation (2.117), however it now uses the marginal likelihood as the acoustic score. The marginal likelihood for the given point estimate of the canonical model is given by[1]

$$p(\mathbf{O}|\mathcal{H}) = \int_{\mathbf{W}} p(\mathbf{O}|\mathcal{H}, \mathbf{W})p(\mathbf{W})\, d\mathbf{W} \tag{5.18}$$

In supervised adaptation, a transform posterior distribution is estimated from the given observation and corresponding supervision transcript as

$$p(\mathbf{W}|\mathbf{O}_{\texttt{supv}}, \mathcal{H}_{\texttt{supv}}) = \frac{p(\mathbf{O}_{\texttt{supv}}|\mathbf{W}, \mathcal{H}_{\texttt{supv}})p(\mathbf{W}|\phi)}{p(\mathbf{O}_{\texttt{supv}}|\mathcal{H}_{\texttt{supv}})} \tag{5.19}$$

where $\mathbf{O}_{\texttt{supv}}$ and $\mathcal{H}_{\texttt{supv}}$ are the observations and the hypothesis of the supervision data, respectively, and $p(\mathbf{W}|\phi)$ is the prior transform distribution. This posterior distribution over the transform is then used as $p(\mathbf{W})$ to compute the acoustic score. This is also called a posterior adaptation [43]. When there is no supervision transcript available in an unsupervised mode, the marginal likelihood of the test data is computed using the transform prior distribution.

The goal of the Bayesian adaptive inference is to compute the inference evidence for each possible hypothesis and select the one with the best evidence. The computation of the inference evidence involves finding the marginal likelihood in equation (5.18), which is intractable due to coupling between transform parameters and hidden state/component sequences. Therefore, several approximations are used to estimate the inference evidence. Some of them are described in the next sections.

5.2.1 Monte-Carlo Approximation

In the Monte-Carlo approximation, a large number of samples are drawn from the transform distribution, and the average of the integral function values is used to approximate the marginal integral in equation (5.18) as

$$p(\mathbf{O}|\mathcal{H}) \approx \frac{1}{N}\sum_{n=1}^{N} p(\mathbf{O}|\mathcal{H}, \hat{\mathbf{W}}_n) \tag{5.20}$$

where N is the total number of samples and $\hat{\mathbf{W}}_n$ is the nth sample drawn from $p(\mathbf{W})$. As $N \rightarrow \infty$, the value obtained by the sampling approximation in above equation will tend to the true value of the marginal likelihood. However, a transform has large number of parameters. For example, a full transform with a bias for a 39-dimensional feature has 1560 dimensional distribution in a vectorised form. With a large number of transform parameters, the number

[1] The canonical model $\hat{\mathcal{M}}_{\texttt{ml}}$ has been dropped from the equations, when there is no confusion.

of samples required for a reasonable approximation increases dramatically. Therefore, the method is computationally expensive, as inference evidence is computed for each sample $\hat{\mathbf{W}}_n$ of the transform using a forward/backward algorithm. Therefore, this method is seldom used for doing inference in a large speech recognition system, rather other computationally efficient methods are investigated. The sampling approach has been also investigated in [159] using Gibbs sampling. The Bayesian speaker adaptive training (BSAT) approach given in [170] uses a mixture of transforms instead of a continuous distribution for transforms, and can be regarded as a case of sampling approaches. The likelihood of an utterance is given as weighted sum of likelihoods obtained by applying each of the transform in the mixture with weights set to the transform priors [170]. The method uses feature domain transforms.

5.2.2 Frame-Independence Approximation

In a frame-independence approximation, the transform at each frame is assumed to be independent, and is allowed to vary at every frame. This assumption is inherent in Bayesian prediction approaches [88, 183], and has been used for adaptation as well [23, 42, 168]. This assumption turns the DBN of an adaptive HMM in figure 5.2(a) into a modified DBN as shown in figure 5.2(b) with the links between transform states removed as they are no longer constrained to be the same. The likelihood in this case is approximated as [199]

$$p(\mathbf{O}|\mathcal{H}) \approx \sum_{\boldsymbol{\theta}} P(\boldsymbol{\theta}|\mathcal{H},\mathcal{M}) \prod_{t} \bar{p}(\mathbf{o}_t|\mathcal{M},\theta_t) \tag{5.21}$$

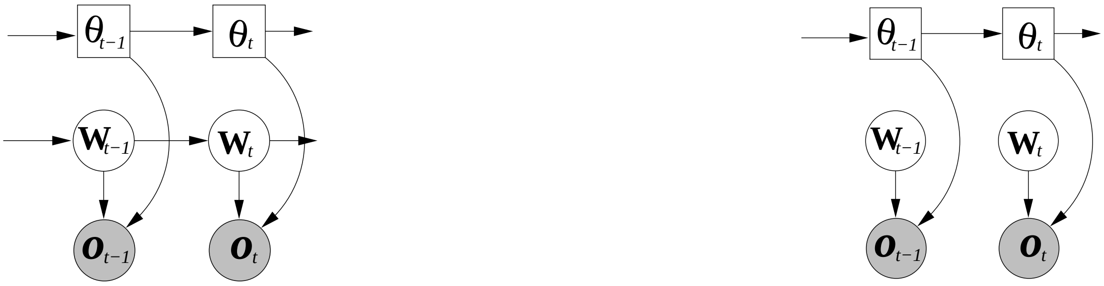
where $\bar{p}(\mathbf{o}_t|\mathcal{M},\theta_t)$ is the predictive distribution given by

$$\bar{p}(\mathbf{o}_t|\mathcal{M},\theta_t) = \int_{\mathbf{W}} p(\mathbf{o}_t|\mathcal{M},\mathbf{W},\theta_t)p(\mathbf{W})\,d\mathbf{W} \tag{5.22}$$

When the form of $p(\mathbf{W})$ is selected as a conjugate prior to the likelihood of the observations, this integral at the frame-level becomes tractable and a standard Viterbi algorithm can be used to compute the likelihood. For the mean transform, a Gaussian distribution is often used for the transform prior which is a conjugate prior to the likelihood [23, 42]. A single-component Gaussian transform prior is often used [23, 42, 199] as given in (3.30). This is reproduced here as

$$p(\mathbf{W}) = \prod_{d=1}^{D} \mathcal{N}(\mathbf{w}_d; \boldsymbol{\mu}_d^{\mathtt{W}}, \boldsymbol{\Sigma}_d^{\mathtt{W}}) \tag{5.23}$$

where the rows of transforms are assumed independent. This is in consistent with the diagonal covariance matrix for the HMM components. In this case, the likelihood in equation (5.22)



(a) Constrained $\mathbf{W}$

(b) Frame-independent $\mathbf{W}$

Figure 5.2: The dynamic Bayesian networks for constrained and frame-independent transforms

can be simply computed by using the parameters of the resulting predictive distribution. The parameters of the resulting predictive distributions are given as

$$\begin{aligned} \bar{\mu}_{md} &= \boldsymbol{\mu}_d^{\mathtt{W}\,\mathtt{T}}\boldsymbol{\xi}_m \\ \bar{\sigma}^2_{md} &= \sigma^2_{md} + \boldsymbol{\xi}_m^{\mathtt{T}}\boldsymbol{\Sigma}_d^{\mathtt{W}}\boldsymbol{\xi}_m \end{aligned}$$

where $\bar{\mu}_{md}$ and $\bar{\sigma}^2_{md}$ are the predictive component mean and variance for component m and dimension d. A GMM can be also used as the transform prior, however the resulting predictive distribution for a component will also turn out to be a GMM, as a GMM is not a conjugate prior for the likelihood [199].

The frame-independence approximation gives a computationally efficient and simple method to calculate the marginal likelihood. The disadvantage of this approximation is that it breaks the assumption in the original DBN for the adaptive HMMs in figure 5.2(a), and the transform is allowed to change at every frame, rather than constraining it to be same for one homogeneous block. This makes it more similar to training of multistyle or speaker-independent models [42]. Thus it may degrade the performance of the speech recognition systems, compared to the standard adaptation approaches.

5.2.3 Lower Bound Approximations

In a lower bound approximation, a lower bound to the marginal likelihood $\mathcal{L}(\mathbf{O}|\mathcal{H})$ is found and used in place of the marginal likelihood $\log p(\mathbf{O}|\mathcal{H})$ in the inference criteria in equation (5.17). In this approach, the lower-bounds to the likelihood are *assumed* to give the same rank ordering as the real marginal likelihood, when used for the inference as in equation (5.17) [199]. Therefore, for two different hypotheses $\mathcal{H}_i$ and $\mathcal{H}_j$, if

$$\mathcal{L}(\mathbf{O}|\mathcal{H}_i) + \log P(\mathcal{H}_i) \quad > \quad \mathcal{L}(\mathbf{O}|\mathcal{H}_j) + \log P(\mathcal{H}_j)$$

this assumption implies that

$$\log p(\mathbf{O}|\mathcal{H}_i) + \log P(\mathcal{H}_i) \quad > \quad \log p(\mathbf{O}|\mathcal{H}_j) + \log P(\mathcal{H}_j). \tag{5.24}$$

The lower bound to the marginal likelihood in equation (5.18) is obtained by introducing a joint variational distribution $q(\boldsymbol{\theta}, \mathbf{W})$ over the component sequence $\boldsymbol{\theta}$ and transform parameters $\mathbf{W}$ and applying Jensen's inequality, and is given by

$$\log p(\mathbf{O}|\mathcal{H}) \geq \mathcal{L}(\mathbf{O}|\mathcal{H}) = \left\langle \log \frac{p(\mathbf{O}, \boldsymbol{\theta}|\mathbf{W}, \mathcal{H})p(\mathbf{W})}{q(\boldsymbol{\theta}, \mathbf{W})} \right\rangle_{q(\boldsymbol{\theta},\mathbf{W})} \tag{5.25}$$

where $\mathcal{L}(\mathbf{O}|\mathcal{H})$ represents the lower bound to $\log p(\mathbf{O}|\mathcal{H})$. The above lower-bound is maximum when

$$q(\boldsymbol{\theta}, \mathbf{W}) = p(\boldsymbol{\theta}, \mathbf{W}|\mathbf{O}, \mathcal{H}) = P(\boldsymbol{\theta}|\mathbf{O}, \mathcal{H}, \mathbf{W})p(\mathbf{W}|\mathbf{O}, \mathcal{H}) \tag{5.26}$$

However, the computation of the transform posterior distribution $p(\mathbf{W}|\mathbf{O}, \mathcal{H})$ in the above equation requires the marginal likelihood $p(\mathbf{O}|\mathcal{H})$. Hence, further approximations are required to make the ideal joint variational distribution in equation (5.26) tractable. The next sections describe two different approximations for it.

5.2.3.1 MAP Point Estimates

In the MAP point-estimate approximations, a sufficient amount data is assumed for transform estimation, such that the transform posterior can be approximated by a Dirac-delta distribution. Thus the joint variational distribution in equation (5.26) becomes

$$q(\boldsymbol{\theta}, \mathbf{W}) = P(\boldsymbol{\theta}|\mathbf{O}, \mathcal{H}, \mathbf{W})\delta(\mathbf{W} - \hat{\mathbf{W}}) \tag{5.27}$$

where $\hat{\mathbf{W}}$ represents the point estimate of the transform. Hence, the lower-bound in equation (5.25) can be expressed as

$$\log p(\mathbf{O}|\mathcal{H}) \geq \mathcal{L}_{\mathtt{map}}(\mathbf{O}|\mathcal{H}, \hat{\mathbf{W}}) = \log p(\mathbf{O}|\mathcal{H}, \hat{\mathbf{W}}) + \log p(\hat{\mathbf{W}}) + \mathtt{H}\left(\delta(\mathbf{W} - \hat{\mathbf{W}})\right) \tag{5.28}$$

As the entropy of Dirac-delta function is $-\infty$ for all transforms $\hat{\mathbf{W}}$ [30], the rank-ordering can be simply obtained by avoiding the entropy term. Therefore, the MAP objective function can be given as

$$\mathcal{F}_{\mathtt{map}}(\hat{\mathbf{W}}) = \log p(\mathbf{O}|\mathcal{H}, \hat{\mathbf{W}}) + \log p(\hat{\mathbf{W}}) \tag{5.29}$$

Thus the MAP estimation as described in section 3.1.2.4 can be derived from the lower-bound approach. As seen in section 3.1.2.4, the MAP objective function can be maximised by using the auxiliary function [24, 28]

$$\mathcal{Q}(\hat{\mathbf{W}}; \mathbf{W}) \quad = \quad \left\langle \log p(\mathbf{O}, \boldsymbol{\theta}|\hat{\mathbf{W}}, \mathcal{H}) \right\rangle_{P(\boldsymbol{\theta}|\mathbf{O},\mathcal{H},\mathbf{W})} + \log p(\hat{\mathbf{W}}) \tag{5.30}$$

where $\mathbf{W}$ represents the current transform. Thus by iteratively maximising the above MAP auxiliary function, the MAP estimate of the transform can be obtained. They are then used in equation (5.29) to compute the acoustic score to be used in doing inference using equation (5.17).

It should be noted that the MAP approximation produces a very loose lower bound as given in equation (5.28), as there is $-\infty$ term present in it. Therefore, it may not produce a good rank ordering of the hypotheses, as produced by other tighter bounds. The next section describes a variational Bayes lower-bound.

5.2.3.2 Variational Bayes

Variational Bayes can be used to find a lower-bound to the marginal likelihood using equation (5.25) [14, 199]. In the variational Bayes approximation, the component sequence posterior and the transform posterior are assumed to be conditionally independent [199]. Thus the joint distribution of the component sequence and the transform in equation (5.26) is approximated as

$$q(\boldsymbol{\theta}, \mathbf{W}) = q(\boldsymbol{\theta}|\mathbf{O}, \mathcal{H})q(\mathbf{W}|\mathbf{O}, \mathcal{H}) \tag{5.31}$$

In the VB approximation, the uncoupled posteriors [1] $q(\boldsymbol{\theta})$ and $q(\mathbf{W})$ are then iteratively refined to make the lower-bound in equation (5.18) tighter. The variational posteriors can be updated using an auxiliary function obtained by re-expressing the lower bound for the kth iteration as [199]

$$\begin{aligned}\log p(\mathbf{O}|\mathcal{H}) &\geq \mathcal{L}_{\texttt{vb}}\big(\mathbf{O}|\mathcal{H}\big) = \mathcal{Q}_{\texttt{vb}}\big(q_{k+1}(\boldsymbol{\theta}), q_k(\mathbf{W})\big) \\ &= \big\langle \log p(\mathbf{O}, \boldsymbol{\theta}|\mathbf{W}, \mathcal{H})\big\rangle_{q_{k+1}(\boldsymbol{\theta})q_k(\mathbf{W})} + \texttt{H}\big(q_{k+1}(\boldsymbol{\theta})\big) - \texttt{KL}\big(q_k(\mathbf{W})||p(\mathbf{W})\big) \end{aligned} \tag{5.32}$$

A variational Bayes expectation maximisation (VBEM) algorithm is used for updating the component and transform posteriors in an interleaved fashion as described in [199]. The algorithm is summarised below.

1. **Initialise:** $q_0(\mathbf{W}) = p(\mathbf{W})$, $k = 1$.

2. **VB Expectation (VBE):** The variational component sequence posterior distribution is estimated by

$$q_k(\boldsymbol{\theta}) = \frac{1}{\mathcal{Z}_{\boldsymbol{\Theta}}(\mathbf{O}, \mathcal{H})} \exp\Big(\langle \log p(\mathbf{O}, \boldsymbol{\theta}|\mathbf{W}, \mathcal{H})\rangle_{q_{k-1}(\mathbf{W})}\Big) \tag{5.33}$$

[1] $q(\boldsymbol{\theta})$ and $q(\mathbf{W})$ are used as the short-hand notations for $q(\boldsymbol{\theta}|\mathbf{O}, \mathcal{H})$ and $q(\mathbf{W}|\mathbf{O}, \mathcal{H})$, respectively.

where $\mathcal{Z}_{\boldsymbol{\Theta}}(\mathbf{O},\mathcal{H})$ is a normalisation constant that makes $q_k(\boldsymbol{\theta})$ a valid distribution, and $\boldsymbol{\Theta}$ is a set of all possible component sequences. The expectation of the joint distribution $\log p(\mathbf{O},\boldsymbol{\theta}|\mathbf{W},\mathcal{H})$ can be computed by factoring it to the frame level as

$$\langle \log p(\mathbf{O},\boldsymbol{\theta}|\mathbf{W},\mathcal{H})\rangle_{q_{k-1}(\mathbf{W})} = \langle \log P(\boldsymbol{\theta})\rangle_{q_{k-1}(\mathbf{W})} + \sum_t \langle \log p(\mathbf{o}_t|\mathbf{W},\theta_t)\rangle_{q_{k-1}(\mathbf{W})} \quad (5.34)$$

The required normalisation constant is given as

$$\mathcal{Z}_{\boldsymbol{\Theta}}(\mathbf{O},\mathcal{H}) = \sum_{\boldsymbol{\theta}} P(\boldsymbol{\theta}|\mathcal{H},\mathcal{M}) \prod_t \tilde{p}(\mathbf{o}_t|\theta_t) \quad (5.35)$$

where

$$\tilde{p}(\mathbf{o}_t|\theta_t) = \exp\left(< \log p(\mathbf{o}_t|\mathbf{W},\theta_t) >_{q_{k-1}(\mathbf{W})}\right) \quad (5.36)$$

is a pseudo-distribution (unnormalised) distribution for the components.

3. **VB Maximisation (VBM):** The optimal transform variational posterior $q_k(\mathbf{W})$ for given variational component sequence posterior $q_k(\boldsymbol{\theta})$ is expressed as

$$q_k(\mathbf{W}) = \frac{1}{\mathcal{Z}_{\mathbf{W}}(\mathbf{O},\mathcal{H})} p(\mathbf{W}) \exp\left(\langle \log p(\mathbf{O},\boldsymbol{\theta}|\mathbf{W},\mathcal{H})\rangle_{q_k(\boldsymbol{\theta})}\right) \quad (5.37)$$

where $\mathcal{Z}_{\mathbf{W}}(\mathbf{O},\mathcal{H})$ is a normalisation constant.

4. $k = k + 1$. **If not converged, go to step 2.**

Finally, with the estimated variational transform distribution $q(\mathbf{W})$, the component sequence distribution $q(\boldsymbol{\theta})$ is computed based on $q(\mathbf{W})$ using equation (5.33). By substituting it into equation (5.32), the resulting lower bound is expressed as

$$\log p(\mathbf{O}|\mathcal{H}) \geq \mathcal{L}_{\texttt{vb}}\left(q(\mathbf{W})\right) = \log \mathcal{Z}_{\boldsymbol{\Theta}}(\mathbf{O},\mathcal{H}) - \texttt{KL}\left(q(\mathbf{W})||p(\mathbf{W})\right) \quad (5.38)$$

This lower bound is then used as the acoustic score for inference.

The variational Bayes approximation avoids the negative entropy term as in the MAP point estimates, and gives a tighter bound. However, it still gives a lower bound to the marginal likelihood, not the true marginal likelihood. Therefore, in the VB based Bayesian inference, the lower-bounds are assumed to produce the same rank-ordering of N-best hypotheses as obtained with true marginal likelihood. However, if the bound is not very tight, this may not be the true and the performance of the speech recognition system may be affected badly. Therefore, it is important to investigate more accurate approximations to the marginal likelihood and thus the inference evidence. Expectation propagation (EP) [67, 119] is an attractive choice, which has been generally reported to give more accurate estimates for the state-posteriors than VB. In the next section, an expectation propagation based approach is proposed for Bayesian adaptive inference.

5.3 Expectation Propagation Based Bayesian Adaptive Inference

Expectation propagation [67, 119] is an iterative algorithm for doing approximate Bayesian inference through tractable approximations to complex distributions. The approximating distribution is usually chosen in the exponential family for tractability, and is constrained to have the same moments as the distributions to be approximated. In this section, expectation propagation is used for inference in an adaptive HMM.

A DBN for the adaptive HMM is shown in figure 5.3. In the DBN, ψ_t and $\mathbf{W}_t$ represent the discrete HMM state and the continuous transform state at time t, respectively. The observation $\mathbf{o}_t$ at time t depends on both the speech and the transform states at time t. A state in the adaptive HMM comprises of both the discrete speech state and continuous transform state, therefore the state of the HMM at time t is represented as $\{\psi_t, \mathbf{W}_t\}$. This also makes all state posteriors and forward/backward messages[1] a function of both the speech state ψ_t and transform state $\mathbf{W}_t$. Therefore, for example, the forward probability is represented as $\alpha_t(\psi_t, \mathbf{W}_t)$, as function of both ψ_t and $\mathbf{W}_t$, and is no more a discrete probability as in the standard HMM described in section 2.3. This notation has been used in this section for describing the expectation propagation based inference.

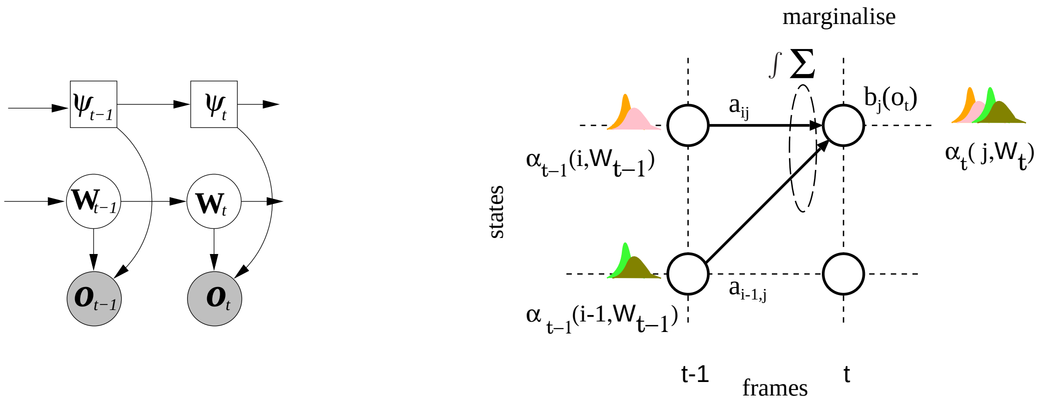


Figure 5.3: A DBN for the adaptive HMM and the computation of forward messages for exact inference in it

The exact inference in the adaptive system shown in figure 5.3 leads to the exponential growth of mixture components. If an attempt is made to compute the likelihood through the forward/backward message passing algorithm, the number of mixture components will

[1] The word 'message' rather than probability is used in this case as generalisation, as they can be also unnormalised.

increase due to the summation of the messages from incoming states as shown in figure 5.3. As this process is repeated over the subsequent speech frames, it will give rise to an exponential growth in the number of mixture components in the messages. This can be prevented by using EP, by approximating the resulting complex mixture into a Gaussian with the same moments. This will make the inference in the adaptive system tractable. However, it is the state-belief (comparable to the state-posterior in the standard forward-backward algorithm) rather than the messages (comparable to the forward/backward probabilities) that is projected or approximated in the expectation propagation algorithm [119, 122]. The projection operation, denoted by 'proj(.)' in this work, implies approximating a complex probability distribution by a simple distribution, in this case by a Gaussian with the same moments.

The expectation propagation has been applied for estimating state posteriors and likelihood in the adaptive system in appendix A. The EP-based forward-backward algorithm has been derived for iteratively updating the forward and backward messages. The EP-based forward-backward iteration is summarised in algorithm 6. The algorithm is given for a left-to-right HMM with non-emitting end-states and T frames of observation vectors in total[1]. The output probability distribution is assumed to be GMMs as in equation (2.17) such that

$$p(\mathbf{o}_t|\psi_t = j, \mathbf{W}_t) = \sum_{m=1}^{M} c_{jm}\, \mathcal{N}(\mathbf{o}_t; \mathbf{W}_t \boldsymbol{\xi}_{jm}, \boldsymbol{\Sigma}_{jm}) \tag{5.39}$$

where c_{jm} is the mixture component weight, $\boldsymbol{\xi}_{jm}$ is the extended mean vector $\boldsymbol{\xi}_{jm} = [\boldsymbol{\mu}_{jm}^{\mathtt{T}}\ 1]^{\mathtt{T}}$, and $\boldsymbol{\Sigma}_{jm}$ is the covariance matrix for the mth mixture component of the jth state. Also, the transition parameters are given as

$$\begin{aligned} P(\psi_t = j|\psi_{t-1} = i) &= a_{ij} && (5.40) \\ p(\mathbf{W}_t|\mathbf{W}_{t-1}) &= \begin{cases} p(\mathbf{W}|\phi) & t = 0 \\ \delta(\mathbf{W}_t - \mathbf{W}_{t-1}) & t = 1, \ldots, T+1 \end{cases} && (5.41) \end{aligned}$$

The rows of the transform are assumed independent, and the prior distribution for the transforms is given by

$$p(\mathbf{W}) = \mathcal{N}(\text{vec}(\mathbf{W}); \boldsymbol{\mu}^{\mathtt{W}}, \boldsymbol{\Sigma}^{\mathtt{W}}) = \prod_{1}^{D} \mathcal{N}(\mathbf{w}_d; \boldsymbol{\mu}_d^{\mathtt{W}}, \boldsymbol{\Sigma}_d^{\mathtt{W}}) \tag{5.42}$$

where $\mathbf{w}_d$ is the dth row of the transform $\mathbf{W}$. The rows of the transforms are assumed independent. As described before, the forward and backward messages in this case are a

[1] In the form presented, hypothetical observations are assumed to be generated by non-emitting states thus extending the sequence from 0 to $T+1$. This allows the same formula to be used for the end-states also, however care should be taken to assign appropriate output probabilities to states at the both ends.

function of both the speech state ψ_t and transform $\mathbf{W}_t$. The forms of forward and backward messages are constrained as

$$\alpha_t(\psi_t = i, \mathbf{W}_t) = P^{\alpha}_{t,i}\ \mathcal{N}(\mathrm{vec}(\mathbf{W}_t); \boldsymbol{\mu}^{\alpha}_{t,i}, \boldsymbol{\Sigma}^{\alpha}_{t,i}) \tag{5.43}$$

$$\beta_t(\psi_t = i, \mathbf{W}_t) = P^{\beta}_{t,i}\ \mathcal{N}(\mathrm{vec}(\mathbf{W}_t); \boldsymbol{\mu}^{\beta}_{t,i}, \boldsymbol{\Sigma}^{\beta}_{t,i}) \tag{5.44}$$

As the rows of transforms are assumed independent, they have block-diagonal covariance matrices. These parameter specifications are used for doing inference in the adaptive system through expectation propagation.

In the EP-based forward-backward algorithm given in algorithm 6, the forward and backward messages are first initialised as given in step (1) of the algorithm. Once the messages are initialised, the forward-pass is run from from $t = 1$ to $T + 1$ using the equation given in step (2). It is worthwhile to look at the equation and find similarity to the equation (2.22) used in the standard forward algorithm. In the equation in step(2), the quantity inside the integral/summation is called the current estimate of the two-slice marginal $\hat{p}_{t-1,t}(\psi_{t-1}, \psi_t, \mathbf{W}_{t-1}, \mathbf{W}_t)$[1] , which can be compared to the transition posterior $P(\psi_{t-1} = i, \psi_t = j|\mathbf{O}, \mathcal{H}, \mathcal{M})$ in equation (2.30) for the standard HMM. Both equations consist of the forward message/probability, the transition probability, the observation probability and the backward message/probability. After marginalisation, the quantity in the numerator in equation (2.22) is called the state-belief $\hat{p}_t(\psi_t, \mathbf{W}_t)$, which can be compared to the state-occupancy $P(\psi_t = j|\mathbf{O}, \mathcal{H}, \mathcal{M}) \equiv \gamma_j(t)$ in equation (2.28). However, the main difference is that the forward and backward messages are no more discrete distributions as in equations (2.28) and (2.30), but a function of continuous transform parameters. As in the standard HMM, the state-occupancy is the product of forward and backward messages for a state, therefore the forward message is obtained by dividing the state-belief by the backward message. However, this is done only after projection (proj) in the EP-based forward backward algorithm, as after marginalisation, the state-belief has a larger number of mixture components. The resulting mixture for the state-belief is projected to a Gaussian, by minimising the KL-divergence through moment matching.

[1] It should be noted that the two-slice marginal $\hat{p}_{t-1,t}(\psi_{t-1}, \psi_t, \mathbf{W}_{t-1}, \mathbf{W}_t)$ is only an approximate value, based on the current estimates of messages, which is refined in successive iterations and therefore, a $\hat{}$ (hat) has been used in the notation.

Step 1. Initialisation
The forward and backward messages are initialised as follows:

$$\begin{array}{lll} t=0: & \alpha_0(\psi_0=1,\mathbf{W}_0)=p(\mathbf{W}) & \\ & \alpha_0(\psi_0=i,\mathbf{W}_0)=0 & i=2,\dots,N \\ t=T+1: & \beta_{T+1}(\psi_{T+1}=N,\mathbf{W}_{T+1})=1 & \\ & \beta_{T+1}(\psi_{T+1}=j,\mathbf{W}_{T+1})=0 & j=1,\dots,N-1 \end{array}$$

All other messages are initialised to *one*.

Step 2. Forward Pass, for $t=1$ to $T+1$
As described in section A.1, the forward message $\alpha_t(\psi_t=j,\mathbf{W}_t)$ is obtained by projecting the state-belief $\hat{p}_t(\psi_t,\mathbf{W}_t)$ at time t and conditioning it with respect to the backward message as[1]

$$\alpha_t(j,\mathbf{W}_t)=\frac{\mathrm{proj}\Big(\sum_i\int_{\mathbf{W}_{t-1}}\frac{1}{k_t}\alpha_{t-1}(i,\mathbf{W}_{t-1})\ a_{ij}\delta(\mathbf{W}_t-\mathbf{W}_{t-1})p(\mathbf{o}_t|j,\mathbf{W}_t)\ \beta_t(j,\mathbf{W}_t)\Big)}{\beta_t(j,\mathbf{W}_t)}$$

Step 3. Backward Pass, for $t=T+1$ to 1
As described in section A.1, the backward message $\beta_{t-1}(\psi_{t-1}=i,\mathbf{W}_{t-1})$ is obtained by projecting the state-belief $\hat{p}_{t-1}(\psi_{t-1},\mathbf{W}_{t-1})$ at time $t-1$ and conditioning it with respect to the forward message as

$$\beta_{t-1}(i,\mathbf{W}_{t-1})=\frac{\mathrm{proj}\Big(\sum_j\int_{\mathbf{W}_t}\frac{1}{k_t}\alpha_{t-1}(i,\mathbf{W}_{t-1})\ a_{ij}\delta(\mathbf{W}_t-\mathbf{W}_{t-1})p(\mathbf{o}_t|j,\mathbf{W}_t)\ \beta_t(j,\mathbf{W}_t)\Big)}{\alpha_{t-1}(j,\mathbf{W}_{t-1})}$$

Step 4. Go to step (2), until converged.

Algorithm 6: *The EP-based forward-backward algorithm*

For the im mixture component in the state-belief for state j, each with weight $\hat{P}_{imj}$, mean $\hat{\boldsymbol{\mu}}_{imj}$ and covariance $\hat{\boldsymbol{\Sigma}}_{imj}$, the parameters for the projected state-belief is given as [67]

$$\bar{P}_j = \sum_{im}\hat{P}_{imj} \tag{5.45}$$

$$\bar{\boldsymbol{\mu}}_j = \sum_{im}\frac{\hat{P}_{imj}}{\hat{P}_j}\hat{\boldsymbol{\mu}}_{imj} \tag{5.46}$$

$$\bar{\boldsymbol{\Sigma}}_j = \sum_{im}\frac{\hat{P}_{imj}}{\hat{P}_j}\hat{\boldsymbol{\Sigma}}_{imj}+\sum_{im}\frac{\hat{P}_{imj}}{\hat{P}_j}(\hat{\boldsymbol{\mu}}_{imj}-\hat{\boldsymbol{\mu}}_j)(\hat{\boldsymbol{\mu}}_{imj}-\hat{\boldsymbol{\mu}}_j)^{\mathtt{T}} \tag{5.47}$$

Though this projection leads to full covariance matrices, they are constrained to be block-diagonal. Once the state-belief is projected, it can be conditioned with respect to the backward message to obtain the forward message. It should be noted that the multiplication and division

[1] The same formula can be used for $t=T+1$ also, by removing the $p(\mathbf{o}_t|j,\mathbf{W}_t)$ term.

of the messages are easily done in the canonical representation of the distributions. Therefore, the messages are converted from moment form to canonical form, and vice-versa during the recursive estimate of forward messages. This can be done as described in appendix A.2. Similarly, once the forward pass is completed and all forward messages are obtained, the backward pass is run as in step (3) of the algorithm. This also involves a similar procedure as when estimating the forward message in step (2).

It should be noted that in the forward/backward message estimation in step (2) and step (3) of the algorithm, both messages interact with each other. The current estimate of the backward message is used while finding new values of the forward message and vice-versa. This allows iterative refinement of the forward and backward messages, by running interleaved forward and backward passes. After iterating the forward and backward passes until the messages are converged, the value of the posteriors/state beliefs can be obtained and likelihood can be also estimated. The likelihood is estimated from the normalisation constants in equations in step (2) or (3) of the algorithm as equation (A.15) as

$$p(\mathbf{O}|\mathcal{H}) = \prod_{t=1}^{T} k_t \tag{5.48}$$

In this way, the marginal likelihood is estimated using the EP-based forward-backward algorithm. The beauty of the above EP based algorithm is that for a point estimate of a transform, it reduces to the standard forward-backward algorithm. In other words, given $p(\mathbf{W}) = \delta(\mathbf{W}-\hat{\mathbf{W}})$, the forward/backward messages again becomes discrete distributions. In this case, the projection step will not be required and thus no approximation step is involved. The forward/backward messages do not interfere and therefore one iteration will be sufficient to compute the posteriors or likelihood.

Though the EP-based approach is computationally much superior to the exact inference approach, and makes Bayesian inference tractable, the storage requirement for the method is still very high. This is because a transform distribution is associated with each (valid) node (each state j for each time t) in the forward-backward trellis. As the dimensionality of the transform is large, it requires a large amount of space for the computation of forward/backward messages, as the messages need to be stored for use in the next pass. The conversion of distributions from the canonical form to the moment form and vice-versa is also computationally expensive.

5.4 Summary

In this chapter, a Bayesian framework for adaptive training has been presented, in which both the model parameters and transforms are regarded as random variables. In the Bayesian framework, the adaptation and inference becomes an integral process, and the canonical model can be directly used for the inference. The Bayesian framework can deal with a small amount of adaptation data, thus making instantaneous adaptation feasible. However, the Bayesian inference leads to an intractable integral for the marginal likelihood, and some forms of approximations are required. Several forms of approximations including sampling approaches, frame-independence assumption, and lower bound approaches like variational Bayes and maximum-a-posteriori estimation can be used. The accuracy of the approximation or tightness of the bound in the lower-bound approaches is very important for accurate ranking of the hypotheses and doing inference. The lower bound approaches may not be able to produce a good rank ordering of the hypotheses if the bound is not very tight. Therefore, to deal with this problem, an expectation propagation based approach was proposed to approximate the marginal likelihood for doing Bayesian adaptive inference.

Chapter 6
Bayesian Discriminative Adaptive Training and Inference

In chapter 4, discriminative adaptation and adaptive training schemes were described which use discriminative criteria such as minimum phone error to estimate adaptation transforms and acoustic models. Though discriminative transforms can give performance gains for supervised adaptation, they are seldom used for unsupervised adaptation for which the correct transcript is not known [147, 180, 202]. This is because discriminative transforms are biased towards the supervision hypothesis and are highly sensitive to errors in it. In this chapter, approaches to handle the issue of bias and limited amount of data in discriminative adaptation are proposed. The maximum-likelihood Bayesian framework for adaptation and adaptive training [43] described in the last chapter is extended to discriminative criteria to handle these issues. First, discriminative adaptive training and inference is described in a Bayesian framework in sections 6.1 and 6.2. This is followed by the investigation of various forms of the maximum-a-posteriori estimation of discriminative transforms, and also the use of discriminative mapping transforms for Bayesian adaptation, in section 6.3.

6.1 Bayesian Discriminative Adaptive Training

In section 5.1 in the last chapter, a Bayesian framework for adaptive training was described for the maximum-likelihood criterion. This can be extended to discriminative criteria as well [43, 199]. In the experiments in [199], transforms and the associated prior are still based on the ML criterion and the used framework is thus not fully discriminative. In this work, a complete discriminative Bayesian framework is considered with both the canonical models and transforms using discriminative criteria. The Bayesian discriminative adaptive training is described in this section using the MMI criterion as given in equation (2.55). However, it can be similarly formulated for other discriminative criteria such as MPE or MWE.

In the Bayesian framework for adaptive training, as noted before, both the model parameters and the transforms are regarded as random variables with valid probability distributions. Thus the posterior of the hypothesis set $\mathbb{H} = \{\mathcal{H}^{(1)}, \dots, \mathcal{H}^{(S)}\}$ for the corresponding observation set $\mathbb{O} = \{\mathbf{O}^{(1)}, \dots, \mathbf{O}^{(S)}\}$ for all S homogeneous blocks is expressed as a marginal given by

$$\log P(\mathbb{H}|\mathbb{O}) \quad = \quad \log \int_{\mathcal{M}} P(\mathbb{H}|\mathbb{O}, \mathcal{M}) p(\mathcal{M}|\Phi)\, d\mathcal{M} \qquad (6.1)$$

where $p(\mathcal{M}|\Phi)$ is a prior over model $\mathcal{M}$ with hyperparameters Φ. The conditional independence of different homogeneous blocks allows the conditional posterior for the specific model, used inside the above equation, to be expressed as

$$P(\mathbb{H}|\mathbb{O}, \mathcal{M}) \quad = \quad \prod_{s=1}^{S} \int_{\mathbf{W}} P(\mathcal{H}^{(s)}|\mathbf{O}^{(s)}, \mathcal{M}, \mathbf{W}) p(\mathbf{W}|\phi)\, d\mathbf{W} \qquad (6.2)$$

where s represents each speaker or homogeneous block, and $p(\mathbf{W}|\phi)$ is the transform prior with hyperparameters ϕ. In the above equation, the posterior of the hypothesis for each homogeneous block of data s is given by

$$P(\mathcal{H}^{(s)}|\mathbf{O}^{(s)}, \mathcal{M}, \mathbf{W}) \quad = \quad \frac{p(\mathbf{O}^{(s)}|\mathcal{H}^{(s)}, \mathcal{M}, \mathbf{W}) P(\mathcal{H}^{(s)})}{\sum_{\breve{\mathcal{H}}} p(\mathbf{O}^{(s)}|\breve{\mathcal{H}}, \mathcal{M}, \mathbf{W}) P(\breve{\mathcal{H}})} \qquad (6.3)$$

where $\breve{\mathcal{H}}$ represents a set of all possible hypotheses for the homogeneous block s.

The priors $p(\mathcal{M}|\Phi)$ and $p(\mathbf{W}|\phi)$ used in equations (6.1) and (6.2) are estimated from the training data, after determining their appropriate forms. The estimation of the priors is a goal of the Bayesian adaptive training and is described next. The hyperparameters of these priors are estimated by using discriminative criteria. Thus in this case, the hyperparameter Φ for the prior over model parameters is estimated by maximising the marginal posterior for all data. This is done by finding a lower-bound to the marginal posterior in equation (6.1) by introducing a variation distribution $q(\mathcal{M})$ and applying Jensen's inequality, which gives

$$\log P(\mathbb{H}|\mathbb{O}) \quad \geq \quad \big\langle \log P(\mathbb{H}|\mathbb{O}, \mathcal{M})\big\rangle_{q(\mathcal{M})} - \mathtt{KL}\big(q(\mathcal{M})||p(\mathcal{M}|\Phi)\big) \tag{6.4}$$

where $\mathtt{KL}\big(q(\mathcal{M})||p(\mathcal{M}|\Phi)\big)$ is the Kullback-Leibler (KL) divergence between the two distributions. The above lower-bound is maximised when

$$q(\mathcal{M}) \quad = \quad p(\mathcal{M}|\mathbb{O}, \mathbb{H}) \tag{6.5}$$

The maximisation of the lower bound in equation (6.4) with respect to model prior hyperparameters Φ is equivalent to minimising the KL-divergence between $p(\mathcal{M}|\Phi)$ and $q(\mathcal{M})$, as noted earlier in section 5.1. This yields the estimate for the model prior as

$$p(\mathcal{M}|\Phi) = p(\mathcal{M}|\mathbb{O}, \mathbb{H}) \tag{6.6}$$

Similarly, for estimating the hyperparameter ϕ of the transform prior $p(\mathbf{W}|\phi)$, a lower bound to the conditional posterior in equation (6.2) is first obtained by introducing a variational transform distribution $q^{(s)}(\mathbf{W})$ for each homogeneous block, and applying Jensen's inequality, thus leading to

$$\log P(\mathbb{H}|\mathbb{O}, \mathcal{M}) \quad \geq \quad \sum_{s=1}^{S} \Big\langle \log P(\mathcal{H}^{(s)}|\mathbf{O}^{(s)}, \mathcal{M}, \mathbf{W})\Big\rangle_{q^{(s)}(\mathbf{W})} - \sum_{s=1}^{S} \mathtt{KL}\left(q^{(s)}(\mathbf{W})||p(\mathbf{W}|\phi)\right) \tag{6.7}$$

The above inequality is maximum when

$$q^{(s)}(\mathbf{W}) \quad = \quad p(\mathbf{W}|\mathbf{O}^{(s)}, \mathcal{H}^{(s)}, \mathcal{M}) \tag{6.8}$$

Again, the maximisation of the lower bound in equation (6.7) is equivalent to minimising the KL-divergence in the equation between the variational distributions and the prior. However, for the same problem as described for the maximum likelihood Bayesian adaptive training, a simple form as obtained for the model prior cannot be obtained for the transform prior in terms of $p(\mathbf{W}|\mathbf{O}^{(s)}, \mathcal{H}^{(s)}, \mathcal{M})$. This is because there are a set of transform variational/posterior distributions, one for each homogeneous block of data whereas the transform prior is tied across all the blocks. Moreover, the lower-bound in equation (6.7) is for the conditional posterior, for a specific estimate of canonical models, and therefore the estimation of the transform prior should also consider the marginalisation over model parameters. Therefore, further assumptions are made.

A sufficient amount of data is assumed during *training* for the given complexity of models, such that both canonical models and transforms for each homogeneous block reduce to point estimates. The point estimate of the canonical model is given by

$$\hat{\mathcal{M}}_{\mathtt{d}} = \arg\max_{\mathcal{M}} \left\{ P(\mathbb{H}|\mathbb{O}, \mathcal{M}, \hat{\mathbb{W}}_{\mathtt{d}}) \right\} \tag{6.9}$$

where $\hat{\mathbb{W}}_{\mathtt{d}} = \left\{ \hat{\mathbf{W}}_{\mathtt{d}}^{(1)}, \ldots, \hat{\mathbf{W}}_{\mathtt{d}}^{(S)} \right\}$ is the set of the transforms for all S homogeneous blocks. The point estimate of the transform $\hat{\mathbf{W}}_{\mathtt{d}}^{(s)}$ for each homogeneous block s is expressed as

$$\hat{\mathbf{W}}_{\mathtt{d}}^{(s)} = \arg\max_{\mathbf{W}} \left\{ P(\mathcal{H}^{(s)}|\mathbf{O}^{(s)}, \hat{\mathcal{M}}_{\mathtt{d}}, \mathbf{W}) \right\} \tag{6.10}$$

It should be noted that the point estimate of the transform in this case is discriminative, obtained by maximising the posterior probability of the hypothesis for each speaker. As there is a point estimate of the transform associated with each homogeneous block of data, the transform prior is a non-point distribution, and is estimated using transforms from all homogeneous blocks. The estimation formula for the transform prior can be shown to be same as in equation (5.16), however using the *discriminative* transform $\hat{\mathbf{W}}_{\mathtt{d}}$ for each homogeneous block. This can be derived by re-expressing equation (6.7) using $p(\mathcal{M}|\mathbb{O}, \mathbb{H}) \approx \delta(\mathcal{M} - \hat{\mathcal{M}}_{\mathtt{d}})$ and $p(\mathbf{W}|\mathbf{O}^{(s)}, \mathcal{H}^{(s)}) \approx \delta(\mathbf{W} - \hat{\mathbf{W}}_{\mathtt{d}}^{(s)})$, as

$$\begin{aligned} &\log P(\mathbb{H}|\mathbb{O}) \approx \log P(\mathbb{H}|\mathbb{O}, \hat{\mathcal{M}}_{\mathtt{d}}) \geq \\ &\quad \sum_{s=1}^{S} \left(\log P(\mathcal{H}^{(s)}|\mathbf{O}^{(s)}, \hat{\mathcal{M}}_{\mathtt{d}}, \hat{\mathbf{W}}_{\mathtt{d}}^{(s)}) + \mathtt{H}\left(\delta(\mathbf{W} - \hat{\mathbf{W}}_{\mathtt{d}}^{(s)})\right) + \log p(\hat{\mathbf{W}}_{\mathtt{d}}^{(s)}|\phi) \right) \end{aligned} \tag{6.11}$$

The hyperparameter of the transform prior distribution can be obtained by maximising the above lower bound and is given as

$$\hat{\phi} = \arg\max_{\phi} \left\{ \sum_{s=1}^{S} \log p(\hat{\mathbf{W}}_{\mathtt{d}}^{(s)}|\phi) \right\} \tag{6.12}$$

Thus, for the sufficient training data condition, the Bayesian discriminative adaptive training gives a point estimate of the discriminatively trained canonical model as in equation (6.9) and a prior over discriminative transforms with hyperparameters given in equation (6.12). Both of them are subsequently used for doing inference. The next section describes techniques for Bayesian inference in a discriminative adaptive system.

6.2 Bayesian Inference in Discriminative Adaptive Systems

Speech recognition systems use different criteria such as maximum-a-posteriori (MAP) or minimum Bayes risk (MBR) for decoding, as described in section 2.6. This section describes inference in adaptive speech recognition systems using the maximum-a-posteriori criterion. However, the approaches described in this section can be also extended for the MBR criterion. In MAP decoding, the inference on a homogeneous block of data is done by searching the hypothesis that gives the maximum a-posterior probability value as[1]

$$\hat{\mathcal{H}} = \arg\max_{\mathcal{H}} \Big\{ P(\mathcal{H}|\mathbf{O}) \Big\} \tag{6.13}$$

where $\mathbf{O}$ is the observation sequence of the test data, and $\mathcal{H}$ is one possible hypothesis sequence. In a non-adaptive system, applying Bayes' rule and ignoring the normalisation constant in the denominator of the resulting expression leads to the standard decoding criterion commonly used in speech recognition systems as described in section 2.6.1

$$\hat{\mathcal{H}} = \arg\max_{\mathcal{H}} \Big\{ p(\mathbf{O}|\mathcal{H})P(\mathcal{H}) \Big\} \tag{6.14}$$

However, in an adaptive system, the form of the inference evidence depends upon the assumption of the underlying adaptation process as generative or discriminative. The following sections distinguish generative and discriminative processes, and describe inference with them.

6.2.1 Generative and Discriminative Processes

A generative system models the data likelihood directly rather than modelling the posterior probability of the hypothesis. The data is assumed to be "generated" by the model. Thus in an adaptive system with a generative process assumption, the data likelihood is marginalised with respect to the transform to obtain the total or marginal likelihood. This is given as

$$p(\mathbf{O}|\mathcal{H}) = \int p(\mathbf{O}|\mathcal{H}, \mathbf{W})p(\mathbf{W}|\phi)d\mathbf{W} \tag{6.15}$$

The posterior probability of the hypothesis is computed by using the Bayes' rule as

$$P(\mathcal{H}|\mathbf{O}) \quad = \quad \frac{p(\mathbf{O}|\mathcal{H})P(\mathcal{H})}{p(\mathbf{O})} \tag{6.16}$$

Thus the posterior can be split into a acoustic score and a language model score parts, and the marginal likelihood in equation (6.15) is used for the acoustic score.

On the other hand, with a discriminative process assumption, the posterior probability of the hypothesis is directly modelled (not split with Bayes' rule). Therefore, with this assumption in the adaptive system, the posterior of the hypothesis is directly marginalised with respect to the adaptation transform to obtain the marginal posterior as

$$P(\mathcal{H}|\mathbf{O}) = \int P(\mathcal{H}|\mathbf{O}, \mathbf{W})p(\mathbf{W}|\phi)d\mathbf{W} \tag{6.17}$$

[1] The superscript (s) has been dropped during the discussion of inference in the following sections.

These different levels of marginalisation with respect to the adaptation transform lead to different inference evidences in the discriminative adaptive system, as described in the next sections.

6.2.2 Generative Adaptive Inference

With a generative model assumption for adaptation, the best hypothesis according to the maximum-a-posteriori criterion is searched as

$$\begin{aligned}\hat{\mathcal{H}} &= \arg\max_{\mathcal{H}} \left\{ P(\mathcal{H}|\mathbf{O}) \right\} \\ &= \arg\max_{\mathcal{H}} \left\{ \frac{p(\mathbf{O}|\mathcal{H})P(\mathcal{H})}{\sum_{\breve{\mathcal{H}}} p(\mathbf{O}|\breve{\mathcal{H}})P(\breve{\mathcal{H}})} \right\} \end{aligned} \tag{6.18}$$

where the summation in the denominator is over all possible hypotheses $\breve{\mathcal{H}}$. The generative system models the data likelihood, and uses Bayes rule in equation (6.16) as above to form the posterior of the hypothesis. The inference evidence in equation (6.18) can be re-expressed for an adaptive system with a generative model assumption as

$$\hat{\mathcal{H}} = \arg\max_{\mathcal{H}} \left\{ \frac{\int p(\mathbf{O}|\mathcal{H},\mathbf{W})p(\mathbf{W}|\phi)d\mathbf{W}\;\; P(\mathcal{H})}{\sum_{\breve{\mathcal{H}}} \;\int p(\mathbf{O}|\breve{\mathcal{H}},\mathbf{W})p(\mathbf{W}|\phi)d\mathbf{W}\;\; P(\breve{\mathcal{H}})} \right\} \tag{6.19}$$

where the data likelihood is a marginal over the transform prior distribution with hyperparameters ϕ. The denominator term in the above equation is same for all possible hypotheses in the search space. Therefore, it does not affect the ranking of the hypotheses and thus can be dropped. The best hypothesis can be simply searched as

$$\hat{\mathcal{H}} = \arg\max_{\mathcal{H}} \left\{ \int p(\mathbf{O}|\mathcal{H},\mathbf{W})p(\mathbf{W}|\phi)d\mathbf{W}\;\; P(\mathcal{H}) \right\}. \tag{6.20}$$

This form of inference is referred as *generative adaptive inference* in this work. This is the form of Bayesian adaptive inference used in section 5.2 for the maximum-likelihood case. However, the integral in equation (6.20) is generally intractable, and some forms of approximation are required. Several approaches including the sampling approach, the frame-independence assumption, lower-bound approaches and expectation propagation have been described in section 5.2 to approximate the inference evidence in equation (6.20).

6.2.3 Discriminative Adaptive Inference

With the discriminative process assumption, the best hypothesis is selected by looking at the marginal posterior of the hypothesis as

$$\begin{aligned}\hat{\mathcal{H}} &= \arg\max_{\mathcal{H}} \left\{ P(\mathcal{H}|\mathbf{O}) \right\} \\ &= \arg\max_{\mathcal{H}} \left\{ \int P(\mathcal{H}|\mathbf{O},\mathbf{W}) p(\mathbf{W}|\phi) d\mathbf{W} \right\}. \end{aligned} \tag{6.21}$$

This is referred as *discriminative adaptive inference* in this work, and the evidence used is called *discriminative inference evidence*. Using Bayes' rule to express the conditional posterior in the above equation, the marginal posterior can be re-expressed as

$$\begin{aligned} P(\mathcal{H}|\mathbf{O}) &= \int P(\mathcal{H}|\mathbf{O},\mathbf{W}) p(\mathbf{W}|\phi) d\mathbf{W} \\ &= \int \frac{p(\mathbf{O}|\mathcal{H},\mathbf{W})P(\mathcal{H})}{\sum_{\breve{\mathcal{H}}} p(\mathbf{O}|\breve{\mathcal{H}},\mathbf{W})P(\breve{\mathcal{H}})} p(\mathbf{W}|\phi) d\mathbf{W} \end{aligned} \tag{6.22}$$

where the language model probability is assumed independent of the transform, i.e. $P(\mathcal{H}|\mathbf{W}) = P(\mathcal{H})$. It should be noted that the denominator term in this case cannot be ignored as in equation (6.19).

The integral for the marginal posterior in equation (6.22) used as the inference evidence is again intractable. Therefore, some form of approximation is required, as used for approximating the marginal likelihood in equation (5.18) in section 5.2. However, one important difference compared to the marginal likelihood approximations is that a Variational Bayes lower-bound based inference scheme cannot be used in this case. This is due to the denominator term present in equation (6.22). Some of the applicable forms of the approximations for the marginal posterior are described in the following sections.

6.2.3.1 Monte-Carlo Approximation

In this approach, the marginal posterior in equation (6.22) required for the inference is approximated by using a large number of samples drawn from the transform distribution. The approximate marginal posterior of the hypothesis is given by

$$\begin{aligned} P(\mathcal{H}|\mathbf{O}) &\approx \frac{1}{N} \sum_{n=1}^{N} P(\mathcal{H}|\mathbf{O},\hat{\mathbf{W}}_n) \\ &= \frac{1}{N} \sum_{n=1}^{N} \left(\frac{p(\mathbf{O}|\mathcal{H},\hat{\mathbf{W}}_n)P(\mathcal{H})}{\sum_{\breve{\mathcal{H}}} p(\mathbf{O}|\breve{\mathcal{H}},\hat{\mathbf{W}}_n)P(\breve{\mathcal{H}})} \right) \end{aligned} \tag{6.23}$$

where N is the total number of samples and $\hat{\mathbf{W}}_n$ is the nth sample drawn from $p(\mathbf{W}|\phi)$. As $N \rightarrow \infty$, this approximation will tend to the true integral value for the marginal posterior.

As described in section 5.2.1, a large number of transform parameters makes it hard to obtain a reasonable estimate of the integral in an LVCSR system. The method is even more computationally expensive than for the marginal likelihood estimation case, as computing the inference evidence for each sample $\hat{\mathbf{W}}_n$ also involves evaluating the denominator term in equation (6.23).

6.2.3.2 Maximum-a-Posteriori (MAP) Approximation

This approximation is derived by formulating a lower-bound to the marginal posterior in equation (6.22) and then making further assumptions to obtain point estimates of discriminative transforms. A lower-bound to the marginal posterior in equation (6.22) can be obtained by introducing a variational transform distribution $q(\mathbf{W})$ and applying Jensen's inequality. This gives

$$\log P(\mathcal{H}|\mathbf{O}) \;\geq\; \left\langle \log \frac{P(\mathcal{H}|\mathbf{O},\mathbf{W})p(\mathbf{W}|\phi)}{q(\mathbf{W})} \right\rangle_{q(\mathbf{W})} \tag{6.24}$$

where the lower-bound in the right hand side is maximum when

$$q(\mathbf{W}) = p(\mathbf{W}|\mathbf{O},\mathcal{H}). \tag{6.25}$$

When there is sufficient amount of adaptation data, the transform posterior in the above equation can be approximated by the Dirac-delta distribution thus giving the point estimates for the transform $\hat{\mathbf{W}}$. This is expressed as

$$q(\mathbf{W}) \approx \delta(\mathbf{W} - \hat{\mathbf{W}}). \tag{6.26}$$

This distribution can be used in equation (6.24), which gives

$$\log P(\mathcal{H}|\mathbf{O}) \;\geq\; \log\Big(P(\mathcal{H}|\mathbf{O},\hat{\mathbf{W}})p(\hat{\mathbf{W}}|\phi)\Big) + \mathtt{H}\Big(\delta(\mathbf{W} - \hat{\mathbf{W}})\Big) \tag{6.27}$$

where $\mathtt{H}(.)$ is the entropy of the function. As the entropy of delta function $\mathtt{H}\big(\delta(\mathbf{W} - \hat{\mathbf{W}})\big)$ is always $-\infty$ [30], it does not affect the rank ordering of the hypotheses. The only part that plays a role in inference is the first term on the right hand side of equation (6.27). Hence, the best hypothesis according to the maximum-a-posteriori approximation for the marginal posterior is selected using the first term in equation (6.27) as

$$\begin{aligned} \hat{\mathcal{H}} &= \arg\max_{\mathcal{H}} \left\{ P(\mathcal{H}|\mathbf{O},\hat{\mathbf{W}})p(\hat{\mathbf{W}}|\phi) \right\} \\ &= \arg\max_{\mathcal{H}} \left\{ \frac{p(\mathbf{O}|\mathcal{H},\hat{\mathbf{W}})P(\mathcal{H})}{\sum_{\breve{\mathcal{H}}} p(\mathbf{O}|\breve{\mathcal{H}},\hat{\mathbf{W}})P(\breve{\mathcal{H}})} p(\hat{\mathbf{W}}|\phi) \right\} \end{aligned} \tag{6.28}$$

The point estimates of the transforms in the above equation are obtained by maximising a discriminative MAP objective function, also obtained from the same equation (6.27) by using the first-term in the right hand side. The discriminative MAP objective function is expressed as

$$\mathcal{F}_{\texttt{dmap}}(\mathbf{W}) = P(\mathcal{H}|\mathbf{O},\mathbf{W})p(\mathbf{W}|\phi) \tag{6.29}$$

Thus the maximum-a-posteriori estimate of the discriminative transforms (based on the MMI criterion) is given as

$$\hat{\mathbf{W}}_{\texttt{dmap}} = \arg\max_{\mathbf{W}} \Big\{ P(\mathcal{H}|\mathbf{O},\mathbf{W})p(\mathbf{W}|\phi) \Big\} \tag{6.30}$$

Therefore, in the MAP approximation for discriminative Bayesian inference, a point estimate of the discriminative transform should be obtained using equation (6.30) for a given transform prior, and the best hypothesis is searched using it to compute the inference evidences as in equation (6.28).

One issue with the inference through equation (6.28) is the marginalisation over the hypotheses in the denominator and computing the posteriors for the hypotheses. It is not practical to marginalise over all possible hypotheses in practice. Therefore usually a lattice or an N-best list is used to represent the set of possible hypotheses. The size of the lattice or the N-best list is an important factor to obtain reasonable estimates of the posteriors, as described in section 2.6.2. The size of the N-best list for computing the marginal in the denominator, and for the search space can be made different. A larger N-best list is used to estimate the posteriors to obtain reasonable estimates for them, whereas the best hypothesis is searched over usually a smaller list.

Non-informative Prior

In the discriminative MAP approximation described above, when a non-informative prior is used for the transform distribution, the point estimates of the transform in equation (6.30) turns into the standard discriminative transform described in section 4.1.1

$$\hat{\mathbf{W}}_{\texttt{d}} = \arg\max_{\mathbf{W}} \Big\{ P(\mathcal{H}|\mathbf{O},\mathbf{W}) \Big\}. \tag{6.31}$$

Similarly, for a non-informative prior, the discriminative adaptive inference as done through equation (6.28) can be rewritten as

$$\hat{\mathcal{H}} \quad = \quad \arg\max_{\mathcal{H}} \left\{ \frac{p(\mathbf{O}|\mathcal{H},\hat{\mathbf{W}})P(\mathcal{H})}{\sum_{\breve{\mathcal{H}}} p(\mathbf{O}|\breve{\mathcal{H}},\hat{\mathbf{W}})P(\breve{\mathcal{H}})} \right\}. \tag{6.32}$$

This inference scheme in equation (6.32) gives a discriminative way to rank possible hypotheses and select the best one even when no prior information for the transform is available. This can be compared to the work in [111, 167, 206] for ranking hypotheses using discriminative criteria, however they use non-adaptive inference evidences. It should be noted that the denominator term in the inference evidence above is different for each possible hypothesis. Therefore, the denominator term in equation (6.32) should be always considered for computing the discriminative inference evidence for ranking the hypotheses. The corresponding generative adaptive inference can be given as

$$\hat{\mathcal{H}} = \arg\max_{\mathcal{H}} \left\{ p(\mathbf{O}|\mathcal{H}, \hat{\mathbf{W}}) P(\mathcal{H}) \right\} \tag{6.33}$$

which is the standard inference procedure commonly used.

6.3 Bayesian Discriminative Adaptation and Inference

The previous section described several approximations for Bayesian inference in discriminative adaptive systems, including maximum-a-posteriori point estimates. In this section, the maximum-a-posteriori estimation of discriminative transforms is investigated in detail, discussing the issues involved with it. Thereafter, discriminative mapping transform based Bayesian adaptation and inference is described.

6.3.1 Maximum-a-Posteriori Discriminative Adaptation

The objective function for the maximum-a-posteriori estimate of discriminative transforms using the MMI criterion is given in equation (6.30). The discriminative MAP objective function for the MPE criterion, which is used in the experiments in this work, can be similarly given as

$$\mathcal{F}_{\texttt{dmap}}(\mathbf{W}) = \sum_{\mathcal{H}} \frac{p^{\kappa}(\mathbf{O}|\mathcal{H}, \mathcal{M}, \mathbf{W}) P(\mathcal{H})}{\sum_{\breve{\mathcal{H}}} p^{\kappa}(\mathbf{O}|\breve{\mathcal{H}}, \mathcal{M}, \mathbf{W}) P(\breve{\mathcal{H}})} \mathcal{A}(\mathcal{H}, \mathcal{H}_{\texttt{r}}) + \alpha^{\texttt{P}} \log p(\mathbf{W}|\phi) \tag{6.34}$$

where $\breve{\mathcal{H}}$ represents all possible hypotheses, and $p(\mathbf{W}|\phi)$ is the discriminative transform prior with hyperparameters ϕ obtained through equation (6.12). κ and $\alpha^{\texttt{P}}$ are the acoustic model and the prior scaling factors, respectively. An I-smoothing to the ML transform $\log p(\mathbf{W}|\phi_{\texttt{ml}})$ scaled by $\alpha^{\texttt{I}}$ can be also added to the above objective function, as described for the discriminative transforms estimation in section 4.1.1.

The optimisation of the MAP objective function for ML transforms in equation 3.24 is straightforward, as a strict-lower bound can be obtained for it and the EM algorithm can be used. However, the same is not true for the discriminative MAP objective function in equation (6.34). Discriminative objective functions are optimised using a weak-sense auxiliary function [140] or extended Baum-Welch algorithm [62], as seen for discriminative training of HMMs in section 2.3.4.2 and estimating discriminative transforms in section 4.1.1. The same approach is first investigated to optimise the discriminative MAP objective function in equation (6.34). In addition, a reverse-Jensen inequality based approach and gradient based optimisations are also described.

6.3.1.1 Weak-Sense Auxiliary Function Based Optimisation

The optimisation of the objective function for discriminative MAP transforms, through a weak-sense auxiliary function, is similar to that for obtaining discriminative transforms in section 4.1.1, however with an additional transform prior term. The auxiliary function for the discriminative MAP objective function in equation (6.34) can be expressed as

$$\mathcal{Q}(\hat{\mathbf{W}};\mathbf{W}) = \mathcal{Q}^{\texttt{num}}(\hat{\mathbf{W}};\mathbf{W}) - \mathcal{Q}^{\texttt{den}}(\hat{\mathbf{W}};\mathbf{W}) + \mathcal{Q}^{\texttt{sm}}(\hat{\mathbf{W}};\mathbf{W}) + \mathcal{Q}^{\texttt{p}}(\hat{\mathbf{W}};\mathbf{W}) \qquad (6.35)$$

where $\mathbf{W}$ is the current estimate of the transform. The rows of the transforms are assumed to be independent, and the numerator ($\texttt{num}$), the denominator ($\texttt{den}$) and the smoothing ($\texttt{sm}$) terms are expressed in terms of row-wise sufficient statistics $\{\mathbf{G}_d^{\texttt{num/den/sm}}, \mathbf{k}_d^{\texttt{num/den/sm}}\}$ for the dth row of transforms, as given in section 4.1.1. The log-likelihood term for the transform over prior distribution in equation (6.34) itself is used as the auxiliary function $\mathcal{Q}^{\texttt{p}}(\hat{\mathbf{W}};\mathbf{W})$ for the prior term. The transform prior is assumed to be Gaussian with mean $\boldsymbol{\mu}_d^{\texttt{W}}$ and covariance $\boldsymbol{\Sigma}_d^{\texttt{W}}$ for each dth row of the transform $\mathbf{w}_d$. The form of prior is given in equation (5.23). The row-wise sufficient statistics corresponding to the prior term is given by

$$\begin{aligned}\mathbf{G}_d^{\texttt{p}} &= \alpha^{\texttt{p}}\boldsymbol{\Sigma}_d^{\texttt{W}\,-1} &(6.36)\\ \mathbf{k}_d^{\texttt{p}} &= \alpha^{\texttt{p}}\boldsymbol{\Sigma}_d^{\texttt{W}\,-1}\boldsymbol{\mu}_d^{\texttt{W}} &(6.37)\end{aligned}$$

The overall sufficient statistics $\{\mathbf{G}_d, \mathbf{k}_d\}$ is the summation of sufficient statistics of all terms, and is given as

$$\begin{aligned}\gamma_m(t) &= \gamma_m^{\texttt{num}}(t) - \gamma_m^{\texttt{den}}(t) &(6.38)\\ \mathbf{G}_d &= \sum_m\sum_t \gamma_m(t)\frac{\boldsymbol{\xi}_m\boldsymbol{\xi}_m^{\texttt{T}}}{\sigma_{md}^2} + \sum_m D_m\frac{\boldsymbol{\xi}_m\boldsymbol{\xi}_m^{\texttt{T}}}{\sigma_{md}^2} + \alpha^{\texttt{p}}\boldsymbol{\Sigma}_d^{\texttt{W}\,-1} &(6.39)\\ \mathbf{k}_d &= \sum_m\sum_t \gamma_m(t)o_{td}\frac{\boldsymbol{\xi}_m}{\sigma_{md}^2} + \sum_m D_m\frac{\boldsymbol{\xi}_m\boldsymbol{\xi}_m^{\texttt{T}}\mathbf{w}_d}{\sigma_{md}^2} + \alpha^{\texttt{p}}\boldsymbol{\Sigma}_d^{\texttt{W}\,-1}\boldsymbol{\mu}_d^{\texttt{W}} &(6.40)\end{aligned}$$

where $\gamma_m^{\texttt{num}}(t)$ and $\gamma_m^{\texttt{den}}(t)$ are numerator and denominator occupancies of the mth mixture component at time t as defined in equations (2.102) and (2.103). $\boldsymbol{\xi}_m = [\boldsymbol{\mu}_m^{\texttt{T}}\ \ 1]^{\texttt{T}}$ is the extended mean vector and σ^2_{md} is dth diagonal element of the covariance matrix for component m. The smoothing factor D_m is chosen in the same way as for the DLT estimation and is given by

$$D_m = E_d \gamma_m^{\texttt{den}}; \quad E_d \;\; = \;\; \max(E, 2\hat{E}_d) \tag{6.41}$$

where the value of E_d is separately chosen for each row of transforms. In the above equation, E is a user-defined global constant and $\hat{E}_d$ is the minimum value to make $\mathbf{G}_d$ positive-definite. It should be noted that for the DLT estimation [180], a value between 0.5 and 2.5 is chosen for E.

The overall statistics in the above equations differ from the standard DLT sufficient statistics only by the additional terms for the transform prior. Once the sufficient statistics are accumulated as in equations (6.39) and (6.40), the MAP estimate of the dth row of the discriminative transform is obtained as

$$\hat{\mathbf{w}}_d = \mathbf{G}_d^{-1}\mathbf{k}_d \tag{6.42}$$

In the weak-sense auxiliary function given in equation (6.35), $\mathcal{Q}^{\texttt{num}}(\hat{\mathbf{W}};\mathbf{W})$ and $\mathcal{Q}^{\texttt{den}}(\hat{\mathbf{W}};\mathbf{W})$ are effectively the lower bounds of the numerator and the denominator likelihoods of the discriminative objective function in equation (6.34). They are obtained by applying the Jensen's inequality to the logarithm of summations over component sequences. As the lower bound of the denominator term is subtracted in equation (6.35), the resulting expression is not guaranteed to be a lower bound to the discriminative objective function. This implies that maximising the auxiliary function is not guaranteed to maximise the objective function. As the resulting expression may not even be concave, a smoothing term $\mathcal{Q}^{\texttt{sm}}(\hat{\mathbf{W}};\mathbf{W})$ is added, which is tunable by a smoothing factor D_m for each component m, as seen above.

The smoothing factor plays a similar role to the learning parameter or step factor in a gradient ascent or descent method, and controls the amount of update to the estimated parameters. This is illustrated in figure 6.1. A small value of smoothing factor gives large update, and the new estimate may be quite far from the current transform parameters and the value of objective function may decrease leading to the unstable update of the parameters. On the other hand, higher values of the smoothing factor tend to hold the new estimates closer to the current parameters. Therefore, with small smoothing factors, the optimisation may diverge, whereas very high values of smoothing factors may not give sufficiently large updates to the transform parameters. It should be noted that even after adding the smoothing term, the weak-sense auxiliary function is not a lower-bound. This is true when adding the prior term as well. The next section investigates the possibility of a lower-bound for the discriminative MAP objective function.

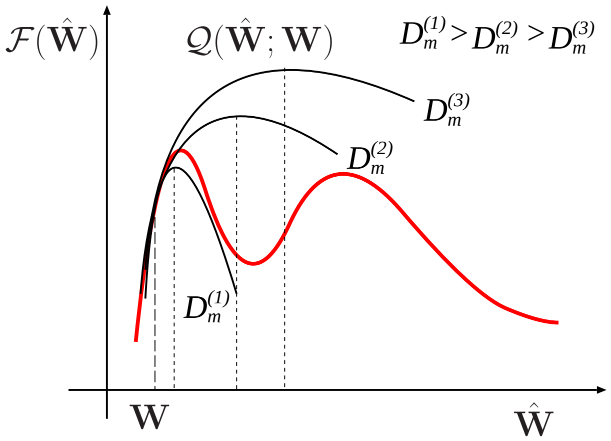


Figure 6.1: The effect of the smoothing factor on the transform updates. A smaller value of smoothing factor leads to large updates in transform parameters and may decrease the value of objective function.

6.3.1.2 Reverse-Jensen Inequality Based Optimisation

Rather than using the weak-sense auxiliary function in the previous section, a strict lower-bound should yield similar attributes to the lower-bounds successfully used with the ML-criterion. When a strict lower-bound can be found to the given objective function, an EM-like algorithm can be applied iteratively to estimate the parameters as in maximum-likelihood estimation. Maximising such a lower-bounded auxiliary function is guaranteed not to decrease the value of objective function. However, due to the denominator term in the objective function, finding a lower-bound has been problematic for discriminative criteria. To obtain an overall lower-bound to the discriminative objective function, a lower bound on the numerator term is required, whereas the denominator term needs to be *upper-bounded*, as shown in figure 6.2. This can be expressed by splitting the discriminative objective function into numerator and denominator parts and then expressing the bounds for them as

$$\begin{aligned}\mathcal{F}(\hat{\mathbf{W}}) &= \mathcal{F}^{\texttt{num}}(\hat{\mathbf{W}}) - \mathcal{F}^{\texttt{den}}(\hat{\mathbf{W}}) \\ &\geq \mathcal{Q}^{\texttt{num}}_{\texttt{LB}}(\hat{\mathbf{W}};\mathbf{W}) - \mathcal{Q}^{\texttt{den}}_{\texttt{UB}}(\hat{\mathbf{W}};\mathbf{W}) \qquad (6.43)\end{aligned}$$

where $\mathcal{Q}^{\texttt{num}}_{\texttt{LB}}(\hat{\mathbf{W}};\mathbf{W})$ is a lower-bound to the numerator component $\mathcal{F}^{\texttt{num}}(\hat{\mathbf{W}})$, and $\mathcal{Q}^{\texttt{den}}_{\texttt{UB}}(\hat{\mathbf{W}};\mathbf{W})$ is an upper-bound to the denominator component $\mathcal{F}^{\texttt{den}}(\hat{\mathbf{W}})$ of the objective function.

The lower-bound to the numerator term can be easily derived by applying Jensen's inequality, as in the maximum-likelihood case. The upper-bound for the denominator term can be obtained by so-called reverse-Jensen inequality [81, 82].

A review of the reverse-Jensen has been provided in appendix B. A reverse-Jensen inequality finds an upper bound to the log-summation of likelihoods by exploiting the convexity

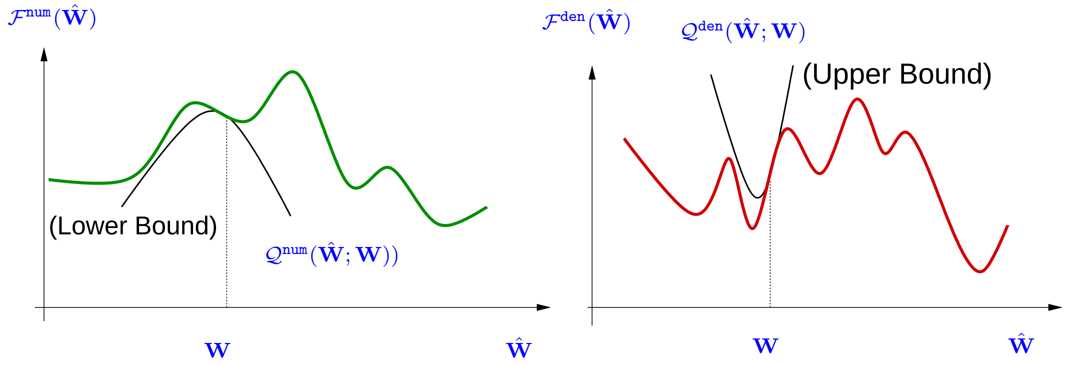

$\mathcal{F}^{\text{num}}(\hat{\mathbf{W}})$
(Lower Bound)
$\mathcal{Q}^{\text{num}}(\hat{\mathbf{W}};\mathbf{W}))$
$\mathbf{W}$
$\hat{\mathbf{W}}$
$\mathcal{F}^{\text{den}}(\hat{\mathbf{W}})$
$\mathcal{Q}^{\text{den}}(\hat{\mathbf{W}};\mathbf{W})$
(Upper Bound)
$\mathbf{W}$
$\hat{\mathbf{W}}$

Figure 6.2: The required bounds for numerator and denominator terms of a discriminative objective function for an overall lower-bound

of the cumulant function of the Gaussian component [81]. However, obtaining an upper-bound directly on the complete denominator term of the discriminative objective function is highly complicated, due to mixture of multinomials and GMM distributions involved in HMMs. Therefore, further assumptions are made to simplify the derivation of the bounds. In standard MPE training, after computing the occupation probabilities for components and grouping them into numerator and denominator, the discriminative training can be assumed as training a bunch of Gaussians. The reverse Jensen inequality is applied to denominator mixtures as in [81] and [1] to obtain its upper bound. With these bounds in place, the auxiliary function can be expressed in the same form as the weak-sense auxiliary function in equation (6.35) including the smoothing term [1]. The upper-bound to the denominator mixtures requires computing the appropriate values of smoothing factors, as described in appendix B. The value of the smoothing factor is given by

$$\begin{aligned} D_m &= \sum_t \gamma_m^{\mathsf{den}}(t) \\ &+ \sum_t \max\left[\gamma_m^{\mathsf{den}}(t)\left(\mathbf{o}_t^{\mathsf{T}}(\hat{\boldsymbol{\mu}}_m\hat{\boldsymbol{\mu}}_m^{\mathsf{T}} + \hat{\boldsymbol{\Sigma}}_m)^{-1}\mathbf{o}_t - 1\right), 0\right] \\ &+ 4\sum_t f\left(\gamma_m^{\mathsf{den}}(t)/2\right)\left(\mathbf{o}_t - \hat{\boldsymbol{\mu}}_m\right)^{\mathsf{T}}\hat{\boldsymbol{\Sigma}}_m^{-1}\left(\mathbf{o}_t - \hat{\boldsymbol{\mu}}_m\right) \\ &+ 4\sum_t f\left(\gamma_m^{\mathsf{den}}(t)/2\right)\left(\left(\mathbf{o}_t - \hat{\boldsymbol{\mu}}_m\right)^{\mathsf{T}}\hat{\boldsymbol{\Sigma}}_m^{-1}\left(\mathbf{o}_t - \hat{\boldsymbol{\mu}}_m\right) - 1\right)^2 \end{aligned} \tag{6.44}$$

where $f(\gamma)$ is a function controlling the tightness of the bound and is given by

$$f(\gamma) = \begin{cases} \gamma + \frac{1}{4\log(6)} + \frac{25/36}{\log(6)^2} - 1/6 & \gamma \geq 1/6 \\ \frac{1}{4\log(1/\gamma)} + \frac{(\gamma-1)^2}{\log(1/\gamma)^2} & \gamma \leq 1/6 \end{cases} \tag{6.45}$$

Therefore, the reverse-Jensen inequality based optimisation of discriminative MAP objective function can be done simply by using the values of the smoothing factor given in equation

(6.44). However, as it will be seen in the experiments, the values of smoothing factors given by the reverse-Jensen inequality is usually very high. Therefore, the update to the parameters using a reverse-Jensen inequality based auxiliary function is very slow, as also noted in [1] for updating model parameters.

6.3.1.3 Hessian and Gradient Based Optimisations

As it will be shown in the experimental results in section 9.3.1, the weak-sense auxiliary function tend to give unstable updates. On the other hand, the reverse-Jensen based auxiliary function gives extremely large values of smoothing factors, thus not giving sufficiently large updates to the transform parameters. Therefore, alternative optimisation schemes for the discriminative MAP objective function in equation (6.29) are explored. It should be noted a Newton's method can be also used for optimising the MAP objective function, by using the Hessian and gradient of the objective function. In the gradient based approaches, the rows of the transform are assumed independent, and are updated separately.

In Newton's method, the new estimate of the transform for the dth row can be given as

$$\hat{\mathbf{w}}_d = \mathbf{w}_d + \eta[\nabla^2\mathcal{F}(\mathbf{w}_d)]^{-1}\nabla\mathcal{F}(\mathbf{w}_d) \tag{6.46}$$

where $\nabla^2\mathcal{F}(\mathbf{w}_d)$ and $\nabla\mathcal{F}(\mathbf{w}_d)$ are the Hessian and gradient of the objective function respectively at the current estimate of the dth row of the transform, $\mathbf{w}_d$. The gradient and Hessian of the log-likelihood is given in equation (C.3) and (C.7) in appendix C. In the equation, $\eta > 0$ is a learning parameter. The direct computation of the curvature of the objective function requires second order statistics as described in appendix C, and is computationally very costly for a large speech recognition system. For this reason, the second-order Newton's method may not be practical for the optimisation of the discriminative objective function. Therefore, a gradient ascent method may be preferred method of optimisation which only requires the gradient of the discriminative objective function. The gradient of the discriminative objective function can be easily computed using functions similar to equation (C.3). The new estimate for dth row of the transform $\hat{\mathbf{w}}_d$ by the gradient ascent algorithm is given as

$$\hat{\mathbf{w}}_d = \mathbf{w}_d + \eta\nabla\mathcal{F}(\mathbf{w}_d) \tag{6.47}$$

where the gradient is computed at the current estimate of the transform. Similarly, other optimisation schemes such as conjugate gradient, BFGS (Broyden-Fletcher-Goldfarb-Shanno) or other quasi-Newton's method [30] can be also used.

The updates to the transform parameters through Newton's and gradient ascent methods are shown in figure 6.3. As it can be seen, a higher value of the learning parameter η gives a large update to the transform parameters and may decrease the objective function leading to unstable updates. On the other hand, a very small value of the learning parameter may not give sufficient update to the transforms. This problem is similar to the one associated with a weak-sense auxiliary function. Thus the main issue with these methods also lies in selecting appropriate learning parameters so as to give stable updates of the transform parameters.

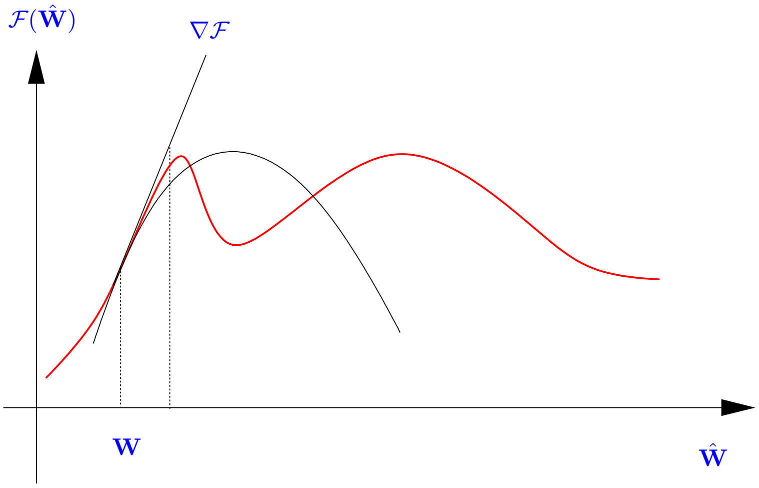


Figure 6.3: The updates to transform parameters through Newton's and gradient ascent method. The learning parameter should be selected appropriately to obtain a consistent increase in the objective function.

6.3.2 Bayesian Adaptive Inference with Discriminative Mapping Transforms

In this section, an alternative approach to discriminative MAP adaptation is proposed. Instead of directly estimating the discriminative MAP transforms as above, a discriminative mapping transform (DMT) is used in the Bayesian framework, as described below.

As discussed in section 4.1.2, instead of directly estimating discriminative transforms, a DMT maps speaker-specific ML transforms into discriminative ones. The mapping itself is speaker-independent. Therefore, the overall adaptation process in a DMT-based system has two stages: MLLR-adaptation as in equation (4.25) and DMT-adaptation as in equation (4.24). The MLLR adaptation can be integrated into a Bayesian framework for inference by using a prior $p(\mathbf{W}|\phi_{\mathtt{ml}})$[1] as described in section 5.2, and leads to the marginalisation of likelihood over the ML transform prior. The DMT adaptation process, on the other hand, is

[1] The subscripts ml and d have been used for distinction, as both ML and discriminative transform components are involved in this section. In this case, the subscript d represents discriminative component of the transform, i.e. DMT. Similarly, $\phi_{\mathtt{ml}}$ represents the hyperparameters for the ML transform prior in this case (not the hyperparameters of the I-smoothing prior).

discriminative in nature, and it is the posterior that is marginalised for any variations in the discriminative transform, as described in section 6.2.3. Thus the marginal posterior required for the inference in equation (6.22) can be re-expressed considering the prior over the DMT transform $\mathbf{W}_{\mathtt{d}}$ as

$$\begin{aligned} P(\mathcal{H}|\mathbf{O}) &= \int P(\mathcal{H}|\mathbf{O},\mathbf{W}_{\mathtt{d}})p(\mathbf{W}_{\mathtt{d}}|\phi_{\mathtt{d}})d\mathbf{W}_{\mathtt{d}} \\ &= \int \frac{p(\mathbf{O}|\mathcal{H},\mathbf{W}_{\mathtt{d}})P(\mathcal{H})}{\sum_{\breve{\mathcal{H}}} p(\mathbf{O}|\breve{\mathcal{H}},\mathbf{W}_{\mathtt{d}})P(\breve{\mathcal{H}})} p(\mathbf{W}_{\mathtt{d}}|\phi_{\mathtt{d}})d\mathbf{W}_{\mathtt{d}} \end{aligned} \tag{6.48}$$

where

$$p(\mathbf{O}|\mathcal{H},\mathbf{W}_{\mathtt{d}}) = \int p(\mathbf{O}|\mathcal{H},\mathbf{W}_{\mathtt{ml}},\mathbf{W}_{\mathtt{d}})p(\mathbf{W}_{\mathtt{ml}}|\phi_{\mathtt{ml}})d\mathbf{W}_{\mathtt{ml}} \tag{6.49}$$

is the marginal likelihood for given $\mathbf{W}_{\mathtt{d}}$. However, the DMT essentially remains constant independent of segments or speakers and a point estimate is used for it such that the posterior distribution for the DMT given the training data, to be used as prior in equation (6.48), is approximated by a Dirac-delta distribution. Therefore, for the point estimates of DMT $\hat{\mathbf{W}}_{\mathtt{d}}$, this can be expressed as

$$p(\mathbf{W}_{\mathtt{d}}|\phi_{\mathtt{d}}) \approx \delta(\mathbf{W}_{\mathtt{d}} - \hat{\mathbf{W}}_{\mathtt{d}}) \tag{6.50}$$

Substituting this in equation (6.48) yields

$$P(\mathcal{H}|\mathbf{O}) = \frac{p(\mathbf{O}|\mathcal{H},\hat{\mathbf{W}}_{\mathtt{d}})P(\mathcal{H})}{\sum_{\breve{\mathcal{H}}} p(\mathbf{O}|\breve{\mathcal{H}},\hat{\mathbf{W}}_{\mathtt{d}})P(\breve{\mathcal{H}})} \tag{6.51}$$

where $p(\mathbf{O}|\mathcal{H},\hat{\mathbf{W}}_{\mathtt{d}})$ for each hypothesis $\mathcal{H}$ is defined in equation (6.49). Therefore, the inference equation for searching the best hypothesis can be re-expressed using equations (6.51) and (6.49) as

$$\hat{\mathcal{H}} = \arg\max_{\mathcal{H}} \Big\{ P(\mathcal{H}|\mathbf{O}) \Big\} = \arg\max_{\mathcal{H}} \left\{ \frac{\int p(\mathbf{O}|\mathcal{H},\mathbf{W}_{\mathtt{ml}},\hat{\mathbf{W}}_{\mathtt{d}})p(\mathbf{W}_{\mathtt{ml}}|\phi_{\mathtt{ml}})d\mathbf{W}_{\mathtt{ml}}\ P(\mathcal{H})}{\sum_{\breve{\mathcal{H}}} \int p(\mathbf{O}|\breve{\mathcal{H}},\mathbf{W}_{\mathtt{ml}},\hat{\mathbf{W}}_{\mathtt{d}})p(\mathbf{W}_{\mathtt{ml}}|\phi_{\mathtt{ml}})d\mathbf{W}_{\mathtt{ml}}\ P(\breve{\mathcal{H}})} \right\} \tag{6.52}$$

In the above equation, the denominator term remains constant for all possible hypotheses $\mathcal{H}$, and does not play a role in the rank ordering of the hypotheses. Therefore, the best hypothesis is selected as

$$\hat{\mathcal{H}} = \arg\max_{\mathcal{H}} \left\{ \int p(\mathbf{O}|\mathcal{H},\mathbf{W}_{\mathtt{ml}},\mathbf{W}_{\mathtt{d}})p(\mathbf{W}_{\mathtt{ml}}|\phi_{\mathtt{ml}})d\mathbf{W}_{\mathtt{ml}}\ P(\mathcal{H}) \right\} \tag{6.53}$$

The marginal likelihood used in the above equation can be lower-bounded as described in section 5.2.3. Variational Bayes as well as the MAP approximation can be used for doing inference using equation (6.53). Using the MAP approximation, the best hypothesis is selected as

$$\hat{\mathcal{H}} \;=\; \arg\max_{\mathcal{H}} \left\{ p(\mathbf{O}|\mathcal{H}, \hat{\mathbf{W}}_{\mathtt{map}}, \hat{\mathbf{W}}_{\mathtt{d}})\; p(\hat{\mathbf{W}}_{\mathtt{map}}|\phi_{\mathtt{ml}})\; P(\mathcal{H}) \right\} \tag{6.54}$$

where the MAP estimates of the ML transforms in this case are obtained as

$$\hat{\mathbf{W}}_{\mathtt{map}} = \arg\max_{\mathbf{W}} \left\{ P(\mathbf{O}|\mathcal{H}, \mathbf{W}, \mathbf{W}_{\mathtt{d}}) p(\mathbf{W}|\phi_{\mathtt{ml}}) \right\} \tag{6.55}$$

Similarly, other approximations such as expectation propagation as described in section 5.3 can be also used for doing inference through equation (6.53) by approximating the marginal likelihood used in it. It should be noted that compared to other cases of discriminative transforms in the Bayesian framework, in this case the discriminative (component of the) transform has been assumed speaker-independent, thus allowing a point-estimate for the transform from the use of a large training data set. Therefore, the problems associated with the discriminative MAP estimation have been avoided, with the Dirac-delta distribution for the DMT.

It should be noted that the inference using equation (6.54) is done using an N-best list based rescoring framework in this work, as described in algorithm 2 for MAPLR. In this framework, N-best hypotheses are used as supervision for generating transforms. A MAP estimate of ML transform is obtained for each hypothesis, which is then used to compute the inference evidence for the hypothesis. The hypothesis with the best inference evidence is selected as output. Thus equation (6.54) can be re-expressed for inference on the N-best rescoring framework as

$$\hat{\mathcal{H}} \;=\; \arg\max_{\mathcal{H}} \left\{ p(\mathbf{O}|\mathcal{H}, \hat{\mathbf{W}}^{(\mathcal{H})}_{\mathtt{map}}, \hat{\mathbf{W}}_{\mathtt{d}})\; p(\hat{\mathbf{W}}^{(\mathcal{H})}_{\mathtt{map}}|\phi_{\mathtt{ml}})\; P(\mathcal{H}) \right\} \tag{6.56}$$

where $\hat{\mathbf{W}}^{(\mathcal{H})}_{\mathtt{map}}$ is the MAP estimate of ML transform obtained using hypothesis $\mathcal{H}$ as supervision. This N-best based rescoring approach is used in this work for doing inference. As N-best hypotheses are used as supervision, it can reduce the hypothesis bias problem associated with transform estimation.

6.4 Summary

This chapter presented a Bayesian framework for discriminative adaptive training and inference. The discriminative adaptive training framework was described from a Bayesian perspective where both the model and the transform parameters are regarded as random variables. The Bayesian inference schemes for discriminative adaptive systems are then detailed to find the best hypothesis for a given observation. The sampling and the MAP approximation approaches were described for doing Bayesian inference in a discriminative adaptive system. Several methods were presented to obtain the MAP estimates of discriminative transforms by optimising a discriminative MAP objective function. They include the weak-sense auxiliary function and reverse-Jensen inequality based approaches. A discriminative mapping transform based Bayesian inference scheme has been also described, which avoids the optimisation problem involved with a discriminative MAP objective function. A variational Bayes or MAP approximation can be used for doing inference with the DMT-based Bayesian approach.

Chapter 7
Experiments on Discriminative Adaptive Training

This chapter presents results from the experimental evaluation of speaker adaptive training using discriminative mapping transforms as described in section 4.3. The experiments were conducted on an LVCSR English conversational telephone speech (CTS) task. The experimental setup for training and evaluation of speech recognition systems is described in section 7.1. This is followed by the evaluation of adaptive training using discriminative mapping transforms in section 7.2, contrasting the performance with other commonly used adaptive training schemes.

7.1 Experimental Setup

The experiments were conducted for large vocabulary continuous speech recognition on a conversational telephone speech (CTS) task. The baseline experimental setup and training data set are identical to those used in [199].

The training data set comprised of about 296 hours of data from three speech corpora distributed by Linguistic Data Consortium (LDC): Call Home English, Switchboard I and Switchboard Cellular. These training corpora were recorded in slightly different acoustic conditions, and consist of 5446 conversational sides or "speakers" (2699 male, 2747 female) in total. Two test sets were used to evaluate the performances of the systems:

1. `dev01sub`: The `dev01sub` testset is a three hour subset of data from the 2001 development data set `dev01` distributed by NIST. It is taken from Switchboard I, Switchboard II and Switchboard Cellular corpora, and consists of 2663 utterances (30k words) from 59 sides (29 male, 30 female).

2. `eval03`: The `eval03` testset consists of six hours of data taken from Switchboard Cellular and Fisher corpora, with 7074 utterances (76k words) from 144 sides (67 male, 77 female).

The speech data had sampling rate of 8 kHz and was encoded with 8-bit μ-law encoding. Speech features were extracted with the MF-PLP front-end [195]. The speech data was parameterised using 12 PLP cepstral coefficients plus the 0th order (`C0`) coefficient. The first, second and third derivatives of the cepstra were also appended. A heteroscedastic linear discriminant analysis (HLDA) transform was used to project this 52-dimensional feature-vector down to 39 dimensions. Speaker-level cepstral mean and variance normalisation as well as a vocal tract length normalisation (VTLN) was applied to the features.

All HMM systems were based on state-clustered triphones with 6189 distinct states. Each speech state had an average of 16 Gaussian components (32 Gaussian components for the silence models). Starting from the ML-SI system, further four iterations were used to build an MPE-SI system using the MPE criterion with a dynamic ML prior for I-smoothing. An MLLR-based ML-SAT system was also built starting from the ML-SI system with four iterations of interleaved transform and model updates. Thereafter, an MLLR-based DSAT system (also called MPE-SAT system in this work) was built starting from the ML-SAT system with further four iterations of MPE model training while keeping the MLLR transforms fixed[1]. The procedure used for building the MLLR-based ML-SAT and DSAT systems is shown in figure 7.1. The MLLR transforms used were speaker-specific transformation of component means. The transform had two base classes: one for speech and another for silence.

A trigram language model trained on 1044M words and a multiple pronunciation dictionary with a vocabulary size of 58k words were used for performing a single-pass decoding.

[1] An alternative version of the MLLR-based DSAT system updating the MLLR transforms at each iteration of the DSAT procedure was also investigated. However, the performance was slightly degraded (by 0.1% absolute on `eval03` testset) than keeping the MLLR transforms fixed during the DSAT procedure.

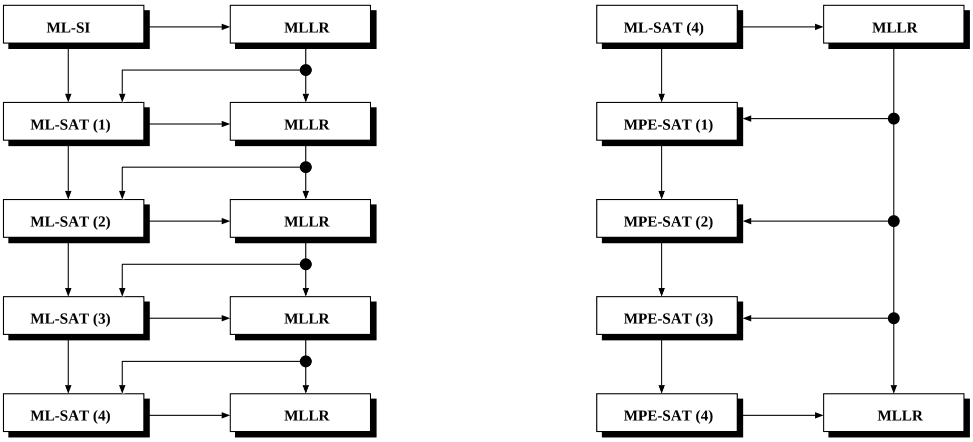


Figure 7.1: The standard MLLR-based ML-SAT and DSAT scheme used in the experiments. The numbers in the bracket represent SAT iterations.

An extended version of HTK [196] was used for the experiments, and the evaluation was done with the NIST speech recognition scoring toolkit `sctk-1.2` [34]. Where significant differences in the performance are mentioned, this was assessed using the NIST matched pairs sentence-segment word error (MAPSSWE) test using the same toolkit at a significance level of 5% (95% confidence) [34, 59].

The WER performance of the systems is given in table 7.1 for the `dev01sub` and `eval03` testsets. Comparing the overall performance of the systems, the MPE training is giving a gain of about 3% absolute compared to the ML training. This demonstrates the effectiveness of discriminative training. Besides, the overall performance of the SAT systems are found to be significantly better than the corresponding multistyle-trained SI systems using the same form of adaptation. For example, the MPE-SAT system is giving a gain of 1.1% absolute on the `dev01sub` testset compared to the MPE-SI system, when both are using MLLR adaptation. This signifies the importance of adaptive training.

System	Adaptation		`dev01sub`		`eval03`	
	Training	Testing	ML	MPE	ML	MPE
SI (hyp)		-	33.4	30.4	32.6	29.2
SI	-	MLLR	31.1	28.5	30.2	27.0
		DLT	31.0	28.3	29.9	26.8
SAT	MLLR	MLLR	30.4	27.4	29.3	26.4
		DLT	30.2	27.2	29.4	26.3

Table 7.1: The performance of MLLR and DLT based *speaker level* adaptation under the standard SI and SAT framework on the `dev01sub` and `eval03` testsets

The table 7.1 further contrasts the performance of speaker-level adaptation by maximum likelihood and discriminative transforms. It should be noted that for MLLR adaptation in both ML and MPE systems, the hypotheses generated from the corresponding SI systems are used as supervision. The WER for the supervision hypothesis is given in the table as 'SI (hyp)'. On the other hand, the lattices for DLT estimation for each model are obtained from the corresponding MLLR-adapted models. A bigram LM is used for generating the denominator lattices. Similarly, the output transcripts obtained by using the MLLR-adapted models and the trigram LM (with WERs shown for MLLR adaptation in table 7.1) are used for generating the numerator lattices using the bigram LM. The smoothing scale factor of $E = 0.8$ and the I-smoothing scale factor of $\alpha^{\mathrm{I}} = 0.01$ were used for the DLT generation. It can be seen from the table that the MLLR adaptation is giving an average gain of 2.2% absolute for the SI systems. In contrast, though discriminative transforms have generally improved the performance on both the SI and SAT systems compared to the MLLR adaptation, the improvement is very small. This is due to the fact that discriminative transforms are more sensitive to the incorrect 1-best hypotheses used as supervision in unsupervised adaptation.

The next section describes the experimental results of discriminative adaptive training using DMT which is expected to deal with this sensitivity to the supervision hypothesis errors.

7.2 Discriminative Adaptive Training

The performance of adaptive training using discriminative mapping transforms was investigated on the CTS task and compared to that of other commonly used approaches. For this purpose, the DMT-based DSAT models were built using four iterations of MPE training based on the ML-SAT models as described in section 4.3. A DLT-based DSAT system was also built for the sake for comparison, in addition to the standard MLLR-based DSAT models. Both MLLR and DLT had the same number of transforms: one for speech and one for silence. The smoothing factors for DLTs and DMTs for adaptive training were the same as used for the DLT adaptation with other systems in this work. The training procedure for DLT and DMT based DSAT schemes are shown in figure 7.2. In the DMT-based DSAT scheme, as the exact estimation of MLLR transforms as given in equation (4.70) was not used due to computational reasons. Rather, the variant of the DMT-based DSAT scheme that uses the standard MLLR estimation as described in section 4.3 was implemented. Another variant with the fixed MLLR transforms for the DMT-based DSAT procedure was also investigated. In the later variant, the MLLR transforms are kept fixed and only the canonical models and DMTs are re-estimated during the DSAT procedure, as described in 4.3. The MLLR-style mean-based linear transforms were used in all experiments. For the DMT, 1000 regression base classes were used. The next sections investigate the training criteria with different DSAT schemes and evaluate the speech recognition performance for each of them.

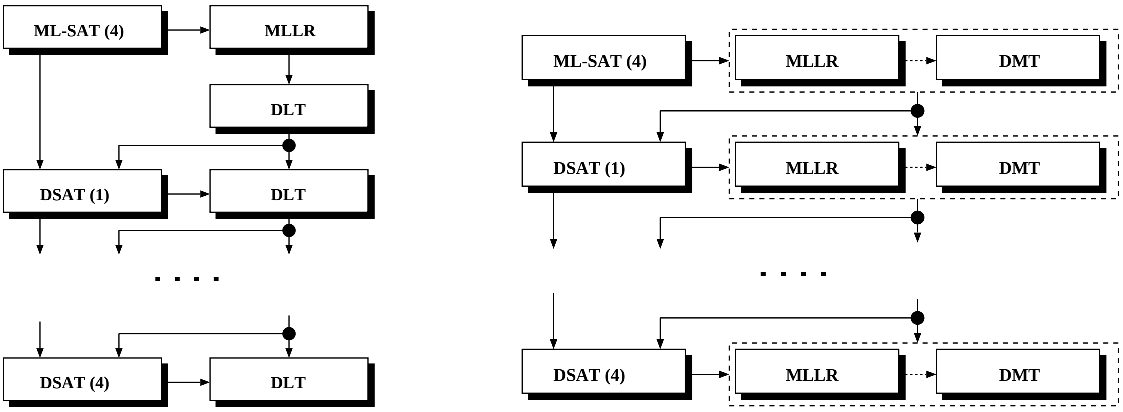


Figure 7.2: The DLT and DMT-based DSAT schemes used in the experiments

7.2.1 Training Criteria

The training criteria for the DSAT schemes are given in table 7.2 for different iterations. The criterion shown is the normalised expected phone correctness (related to the MPE criteria) given by

$$\bar{\mathcal{F}}_{\mathtt{mpe}}(\mathcal{M}) \quad = \quad \frac{1}{N_{\mathtt{phone}}} \sum_{\mathcal{H}} P(\mathcal{H}|\mathbf{O},\mathcal{M})\mathcal{A}(\mathcal{H},\mathcal{H}_{\mathtt{r}}) \tag{7.1}$$

where $\bar{\mathcal{F}}_{\mathtt{mpe}}(\mathcal{M})$ represents the normalised expected phone correctness, $N_{\mathtt{phone}}$ is the number of phones in the reference hypothesis, and $\mathcal{A}(\mathcal{H},\mathcal{H}_{\mathtt{r}})$ is raw phone accuracy of hypothesis $\mathcal{H}$ compared to the reference hypothesis $\mathcal{H}_{\mathtt{r}}$ as given in equation (2.64). At each iteration, the criterion value was obtained during the update of the model parameters. Thus the zeroth iteration shows the criterion after applying MLLR, MLLR+DMT, or DLT to the final ML-SAT acoustic models. The training criteria has been plotted in figure 7.3 as well. As it can be seen from the figure, all schemes show an increase in the correctness as the number of iterations increases. The lowest correctness value was obtained with the MLLR-based DSAT scheme. Using MLLR+DMT during adaptive training shows consistent gains in correctness. However, the largest correctness values were obtained with the DLTs. This indicates that the DLTs perform slightly better on the training data than the other schemes. However, this may not give the best performance on the test data set. It should be noted that both variants of the DMT-based DSAT implemented give similar training criteria, with the updated MLLR version giving slightly higher criteria gain than the fixed-MLLR version.

#Iteration	DSAT Transform			
	MLLR	MLLR(updated)+DMT	MLLR(fixed)+DMT	DLT
0	0.783	0.803	0.803	0.821
1	0.817	0.840	0.840	0.863
2	0.836	0.861	0.860	0.887
3	0.848	0.874	0.873	0.902

Table 7.2: Normalised expected phone correctness given in equation (7.1) for different DSAT schemes during training[1]

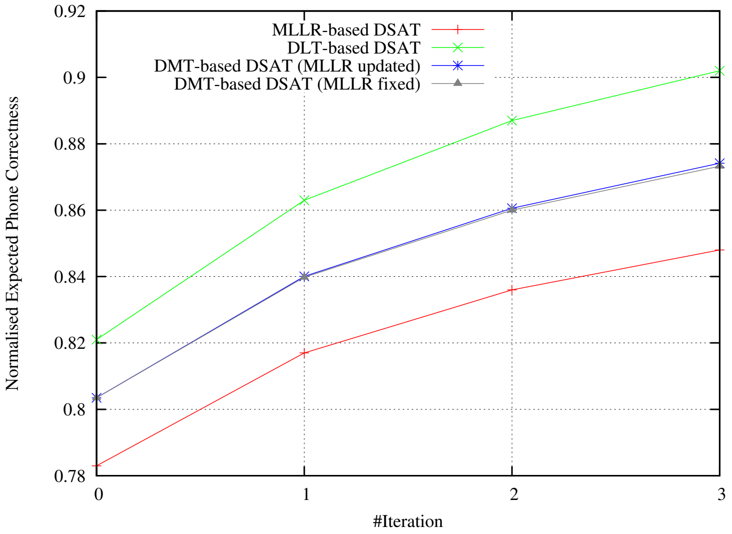


Figure 7.3: A plot of normalised expected phone correctness in equation (7.1) for different DSAT schemes during training

7.2.2 WER Performance

The speech recognition performance on the `eval03` testset for the various DSAT schemes is shown in table 7.3. The results from the MPE-SI system have been also provided for comparison. As the CTS task is an unsupervised adaptation task, an initial hypothesis is required. This was obtained from the MPE-SI model and had a word error rate (WER) of 29.2%. The results using this hypothesis as supervision are labelled as *hyp* in the table. In order to investigate the performance degradation resulting from errors in this supervision hypothesis, the reference transcription itself was also used as the supervision. The results for using the reference transcript as supervision are labelled as *ref* in the table.

Using MLLR adaptation on the SI system with the hypothesis shows large gains in performance, a reduction in WER of 2.2% absolute. MLLR-based DSAT scheme gave an additional

[1] The same phone-marked lattices were used in all these experiments.

0.6% absolute reduction in WER using the hypothesis. If the reference was used to estimate the transform instead, additional consistent gains are observed with both systems compared to using the hypothesis. The most striking result is the difference in performance of the DLT-based system, between using the reference or the hypothesis for the supervision. Using the reference, the DLT-based system yielded the best performance, whereas it had the worst performance among all DSAT schemes when using the hypothesis. This illustrates the sensitivity of DLTs to errors in the hypothesis. On the other hand, the DMT-based DSAT scheme gave the best performance when using the hypothesis as supervision. Both variants of the DMT-based DSAT schemes were found to give similar WER performance. A statistically significant gain of 1.1% absolute was obtained compared to the standard MLLR-based DSAT approach.

Training Scheme	Transform		Supervision	
	Training	Testing	ref	hyp
SI (hyp)	—	—	—	29.2
SI	—	MLLR	24.3	27.0
		DLT	21.7	26.8
		MLLR+DMT	23.4	26.2
DSAT	MLLR	MLLR	23.6	26.4
	DLT	DLT	18.4	28.1
	MLLR+DMT	MLLR+DMT	22.5	25.3

Table 7.3: Comparison of WER% of different DSAT schemes on `eval03` testset

The sensitivity of the DSAT systems to the supervision hypothesis errors is further illustrated in figure 7.4. The figure shows the gain obtained with the DSAT schemes compared to the MLLR-adapted SI system, at the different supervision WER. This is obtained by grouping the speakers in `eval03` testset according to their SI WER, and averaging the gain for each group. It should be noted that most of the speakers fall in the left half region of the plot, and the characteristics in this region should be emphasised. As it can be seen from the figure, as the supervision WER increases, the performance of the DLT-based DSAT scheme degrades, whereas the MLLR-based DSAT system is less affected. The DMT-based DSAT scheme, on the other hand, consistently shows a better performance than both. The trend of higher performance gain with the increasing supervision WER in the DMT-based DSAT scheme is due to the fact that there is more room for the improvement at the higher WER.

It is not necessary to use the same transform for adaptive training as used during recognition. From table 7.3, using MLLR+DMT appears to be a good candidate for testset adaptation due to its robustness to the supervision hypothesis errors. Therefore, the use of MLLR+DMT as a testing transform with other DSAT models was investigated, rather

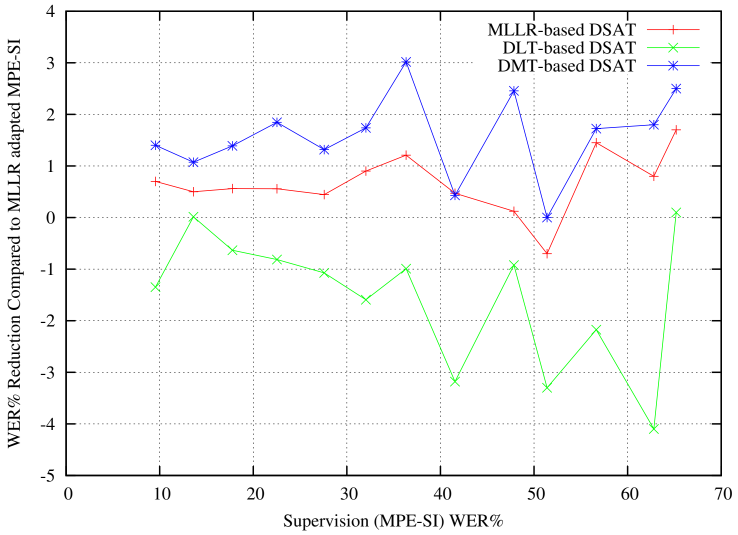


Figure 7.4: Improvement (absolute) obtained with the DSAT schemes compared to the MLLR adapted MPE-SI system at different supervision WERs

than using the same transform as used for their training. Table 7.4 shows the performance of the various DSAT schemes using MLLR+DMT for test-set adaptation. As previously observed in [202], using DMTs in addition to MLLR yields a gain of about 0.8% absolute for both the MPE-SI model and the MLLR-based DSAT model compared to MLLR. However, the performance of both systems is still significantly worse than the proposed DMT-based DSAT system. Using MLLR+DMT with the DLT-based DSAT system shows large gains over using the DLT as the test-set transform. Despite the DLT-based DSAT system having the best criterion on the training data, it is still significantly worse than the DMT-based DSAT system even with the robust MLLR+DMT test-set transform. This may have been caused by the inconsistency between the training transforms, DLT, and the test-set transforms, MLLR+DMT.

Training Scheme	Transform		Supervision hyp
	Training	Testing	
SI	—	MLLR+DMT	26.2
DSAT	MLLR	MLLR+DMT	25.6
	DLT		25.6
	MLLR+DMT		25.3

Table 7.4: Comparison of `eval03` WER% of different DSAT models with MLLR+DMT as testing transforms

7.3 Summary

This chapter has presented experimental results for speaker adaptive training based on discriminative mapping transforms (DMTs) and compared it with other commonly used adaptive training schemes. Adaptively trained systems are generally expected to perform better than the corresponding SI systems, both for maximum-likelihood and discriminative systems. This trend was observed in our experiments proving the superiority of the adaptively trained systems, except for the DLT-based DSAT scheme. The DLT-based DSAT system with unsupervised adaptation was inferior to the adapted SI system. The DLT-based DSAT system was affected by the sensitivity of discriminative transforms to supervision hypothesis errors. The use of the speaker-independent DMTs reduces this sensitivity, and the proposed DMT-based DSAT scheme was found to significantly outperform the standard approaches to speaker adaptive training.

Chapter 8
Experiments on Discriminative Adaptive Inference

This chapter presents the experimental results for discriminative adaptive inference as described in section 6.2. The experiments were conducted on the same conversational telephone speech (CTS) task as described in chapter 7. The experimental setup for training and evaluation is described in section 8.1. Subsequently, section 8.2 presents the results from the investigation of discriminative adaptive inference.

8.1 Experimental Setup and Baseline

The experiment for discriminative adaptive inference was conducted on the same LVCSR conversational telephone speech (CTS) task with a similar experimental setup as described in chapter 7. However, a different adaptation and decoding strategy is used based on an N-best rescoring framework as described in the next paragraph. The training data (about 296 hours), frontend processing and the trained models were identical to that used in the experiments of chapter 7. However, only the MPE-SI model was used in this experiments, as the standard

MPE-SAT models use ML transforms and is not consistent with a complete discriminative framework. In this case also, all transformations perform mean-based adaptation. All speaker-specific transforms used two base classes: one for speech and another for silence. A trigram language model trained on 1044M words and a multiple pronunciation dictionary with a vocabulary size of 58k words were used for decoding in this case also. The `eval03` testset containing about 6 hours of data with 7074 utterances (76K words) from 144 speakers was used for evaluation.

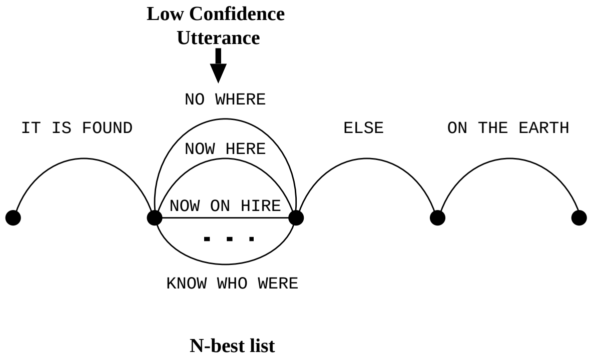


Figure 8.1: The speaker-level incremental N-best framework for adaptive inference

An experimental framework was designed to investigate discriminative adaptive inference as described in section 6.2.3 by ranking hypotheses in an N-best list. An optimal N-best list of suitable depth and small oracle error rate can be obtained only at the utterance level. However, adaptation and inference at the utterance level would be problematic as there is only a small amount of adaptation data for robust estimate of the transforms at the utterance level. On the other hand, if the transform is estimated at the speaker level, there is an exponential growth of hypotheses in the N-best list with the increased number of utterances for a speaker. Therefore, an experimental framework was designed in which only one segment per speaker is expanded into an N-best list at one time as shown in figure 8.1. This limits the possible supervision hypothesis to N for each speaker at a time, which are then used to estimate adaptation transforms. The transforms are still estimated at the speaker level and thus the problem associated with a small amount of adaptation data is isolated. The segment for the N-best expansion is selected by looking at the confidence scores. The segments with low confidence score are selected, one by one, for reranking as they are the ones that need to be redecoded. This is in contrast to the standard adaptation process using 1-best list where using low-confidence segments in supervision has an adverse effect. However, in this case, N-best lists are used as supervision and reranking, and this is not the problem. In this case, selecting low-confidence segments for rescoring is desirable as they are expected to have relatively higher word error rates and thus more room for improvement. After expanding a segment to an N-best list for each speaker, a transform is estimated using each hypothesis in the context of others for each speaker. The best hypothesis in the N-best list is searched by looking at the discriminative inference evidence for each of them. In this case, equation (6.32) is used for doing discriminative adaptive inference. This N-best expansion and adaptive inference is done for segments one by one incrementally on low-confidence segments and the best hypotheses are selected for each of them, obtaining the final set of best hypotheses.

The experimental framework described uses an N-best list for inference. Therefore, baseline results were obtained for the N-best list and compared to the decoding using full search space. The performance of the standard MLLR and DLT adaptation on the MPE-SI model is given in table 8.1 both for N-best list and full search space. In the experiment, the MPE criterion with an I-smoothing prior to ML statistics was used for DLT estimation. A weak-sense auxiliary function was used to optimise the discriminative objective function and estimate transforms. The N-best list used in the experiment are regenerated after adapting the MPE-SI model with MLLR transforms. When all hypotheses were reranked with the same MPE-SI model, the result was slightly different (due to slight difference in the criteria used for generating the N-best list). All the lattices were regenerated and phone-marked with a bigram language model for discriminative transform estimation.

Transforms	Search Space	WER%
—	Full	29.2
MLLR	Full	27.0
	150-best	26.9
DLT	Full	26.8
	150-best	26.9

Table 8.1: The `eval03` baseline performance for MLLR and DLT adapted MPE-SI models with 1-best supervision

The results in table 8.1 show that MLLR 1-best adaptation gives a performance gain of 2.3%, whereas DLT 1-best adaptation gives no additional gain on the N-best list scoring. As discussed earlier, this is due to the hypothesis bias and sensitivity issue of discriminative transforms to the erroneous 1-best supervision. This is reflected from the fact that, for supervised adaptation when there is no errors in the supervision hypothesis, 1-best DLT adaptation has been found to give significant performance gain compared to corresponding MLLR adaptation, as seen in the last chapter. This further motivates the investigation of the N-best list based discriminative adaptive inference. The results are described in the next section.

8.2 Discriminative Adaptive Inference

This section presents the results for discriminative adaptive inference in the N-best rescoring framework described in the previous section. The N-best framework is expected to reduce the hypothesis bias problem of discriminative adaptation. The next sections show the nature of selected segments for expansion and results from reranking the expanded hypotheses.

8.2.1 Segments Selection

For the N-best list based adaptive inference, a segment with low confidence score was selected per speaker and expanded into a 150-best list for each speaker. This yielded 150 supervision hypotheses for estimating each speaker-level transform. The current best hypotheses are used for the rest of the segments not expanded. The segments for each speaker was selected for expansion by looking at the confidence score for the segment. The segment confidence scores were obtained by averaging the word confidences from the confusion network obtained from the denominator lattice for the segment. A segment with the lowest confidence score was selected for the first pass of the N-best list expansion in this work. Figure 8.2 shows WERs of the selected segments compared to the average WER for their corresponding speaker. In the figure, a point below the diagonal line represents a segment with a higher WER than the average WER for its speaker. It can be seen that a majority of the selected segments had a higher WER rate than their corresponding speaker's WER. However, there are segments which have a WER lower than their average speaker WER. This occurs specially in the low WER region.

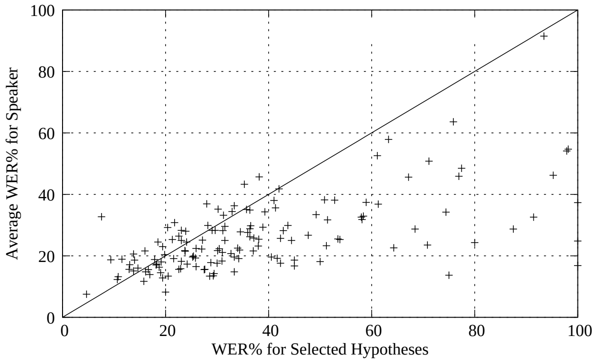


Figure 8.2: The WER% of selected segments compared to average WER% of the corresponding speaker.

8.2.2 WER Performance

In the experiments for the N-best based adaptive inference, the transforms were estimated for each speaker corresponding to each supervision hypothesis that differed at the expanded segment. The discriminative transforms were based on the MPE criterion with I-smoothing prior to ML statistics and without any transform prior. The transform prior is not used as the transforms are being estimated at the speaker level. A weak-sense auxiliary function was used for optimisation with an I-smoothing constant $\alpha^{\text{I}} = 0.01$ and a scaling to smoothing factor $E = 0.8$. The equations (6.33) and (6.32) were used for generative and discriminative adaptive inference, respectively, for ranking the hypotheses in an N-best list.

The result for the 150-best list based adaptive inference with speaker-level adaptation is shown in table 8.2. The standard MLLR 1-best adaptation gives a baseline, that is being compared to the N-best adaptation using MLLR and DLTs. In N-best based MLLR adaptation, hypotheses are ranking using the standard inference procedure as in (6.33). On the other hand, in the N-best based DLTs adaptation, discriminative inference evidence given in (6.32) is used.

Adaptation		Ranking	WER%
Transforms	Supervision		
–	–	–	29.2
MLLR	1-best	–	26.9
MLLR	N-best	gen.	26.9
DLT		dis.	26.9

Table 8.2: The performance of adaptive inference on the MPE-SI model for the `eval03` testset

In the table, no improvement is observed from the N-best adaptation with MLLR. Unfortunately, the same is true for using the discriminative adaptive inference for ranking the N-best hypotheses with the DLT adaptation. This may be due to several reasons including the approximations involved in arriving at the discriminative inference evidence of equation (6.32). It should be noted that the inference evidence in equation (6.32) is only an approximation to the marginal posterior, used in the real discriminative adaptive inference in equation (6.21). As discussed in section 6.2.3, this approximation is derived from a lower-bound to the marginal posterior, and is a poor approximation, not even a tight bound to the marginal posterior. In addition, the size of the N-best list is also an important issue. A sufficiently large N-best list is required for computing the normalisation constant in equation (6.32) to obtain reasonable estimates of the posteriors of hypotheses. Besides, it may be the case that the significant change in the performance was not observed because no sufficient segments were redecoded.

8.3 Summary

This chapter has presented results from the experimental investigation of discriminative adaptive inference using an N-best based rescoring framework. An adaptation transform is estimated for each of the possible hypotheses in the N-best list and the hypotheses are ranked based on discriminative inference criteria to obtain the best hypothesis. Though the method was expected to reduce the hypothesis bias problem of the conventional 1-best based discriminative adaptation, the experimental result shows no significant change in the performance compared to the standard decoding procedure. This may be due to several approximations involved in determining the discriminative inference evidence given in equation (6.32), including the size of the N-best list for computing the normalisation constant for the posterior. It may be also possible that the number of segments decoded in the experiment may not be sufficient to yield a remarkable change in the WER. However, redecoding each segment using discriminative inference evidence requires $\mathcal{O}(N^2)$ calculations of likelihoods of hypotheses using the forward-backward algorithm, and is thus computationally expensive. This implies that for an N-best list of size 150, for rescoring *one* segment using discriminative adaptive inference evidence requires the computation of the likelihood using the forward-backward algorithm 22500 times. For this reason, the method was not further investigated.

CHAPTER 9

Experiments on Bayesian Adaptation and Inference

This chapter presents experimental results for expectation propagation based Bayesian inference as described in section 5.3, as well as the results for discriminative Bayesian adaptation as described in section 6.3. The LVCSR experiments were conducted for the conversational speech task (CTS) with a similar setup as described in the previous chapters, except that the adaptation is done at the *utterance-level* and an N-best rescoring framework is used for inference. The utterance-level adaptation simulates the scenario of online or instantaneous adaptation where there is only a small amount of adaptation data available. The effectiveness of Bayesian adaptive inference is investigated in such a scenario. An N-best list based rescoring framework is used instead of normal Viterbi decoding due to the nature of the Bayesian adaptive inference. The experimental setup and baseline is briefly described in section 9.1. Section 9.2 describes experimental investigation for Bayesian inference based on expectation propagation. This is followed by experimental investigation of MAP estimation of discriminative transforms and the use of discriminative mapping transforms for Bayesian adaptation in section 9.3.

9.1 Experimental Setup and Baseline

The experiments for Bayesian adaptation and inference were conducted on an LVCSR conversational telephone speech (CTS) task, with the experimental setup similar to as described in chapter 7. However, an utterance-level adaptation and N-best rescoring framework is used.

The training data set and trained models were identical to that used in the experiments of chapter 7. The PLP-based front-end parametrisation of speech was also identical yielding the final feature dimension of 39. Cepstral mean and variance normalisation as well as vocal tract length normalisation was also applied to the features. However, it should be noted that though the adaptation and inference being considered is at the utterance level, speaker level CMN and CVN were used, which are more robust. This implies that any improvement obtained with adaptation will be less than the actual improvement that can be obtained. Both SI and SAT model sets were trained using ML and MPE criteria. MPE-SAT model in this case uses MLLR-transforms estimated using the ML-SAT system, and only model parameters are updated during training. An affine transformation of component means was used in all experiments for adaptation in this case also. The number of baseclasses in speaker-specific transforms and DMTs were same as before. A single Gaussian prior for the MLLR transform was estimated using equation (5.16), from the training set transforms for both ML and MPE systems. The prior for speech and silence transforms were independently estimated and had the forms as given in equation (3.30). The transform priors were estimated both for SI and SAT systems. The supervision hypothesis for adaptation was obtained from the corresponding SI model for both ML and MPE systems. All adaptation is done at the utterance level reflecting the scenario of the instantaneous adaptation. The average length of utterances was 3.13s, in contrast to the average length of 153.75s per speaker-side for the `eval03` testset. The N-best list size was 150 for the rescoring experiments. A trigram language model trained on 1044M words and a multiple pronunciation dictionary with a vocabulary size of 58k words were used for decoding in this case also. As noted before, the `eval03` testset used for evaluation consists of 7074 utterances from 144 speakers.

The baseline performance of the utterance-level MLLR-based adaptation is shown in table 9.1 for different Bayesian approximations and is identical to the results in [199]. The performance was evaluated with a size of N-best list as 150, which was found to be satisfactory as noted in [199], as increasing the size of the N-best list to 300 for unadapted ML-SI system was found to give no further gain (or loss). It should be noted that there is a slight difference in performance compared to the results in table 7.1, for example 32.8% for unadapted ML-SI system instead of 32.6%. This difference is caused by using N-best based rescoring using forward-backward likelihoods, rather than using the Viterbi decoding as used for results

Bayesian Approx	ML Train		MPE Train	
	SI	SAT	SI	SAT
—	32.8	—	29.2	—
ML	35.5	35.2	32.4	32.3
MAP	32.2	31.8	29.0	28.8
VB	31.8	31.5	28.8	28.6

Table 9.1: The performance of the *utterance-level* Bayesian adaptive inference with 150-best rescoring using MLLR based adaptation on the `eval03` testset

in table 7.1. In the above table, for the variational Bayes (VB) lower-bound approximation, a single iteration was used for transform distribution update.

It can be observed in the above table that doing MLLR adaptation at the utterance level degrades the performance severely, even compared to the SI performance. This is because the transforms have been generated using only a small amount of data for each utterances. This can be contrasted to the significant gain obtained with the speaker-level MLLR adaptation in table 7.1. With the utterance level adaptation in table 9.1, both MAP and variational Bayes (VB) have been found to improve the performance of the system significantly in case of the ML trained systems, however the same level of improvement is not obtained with the MPE systems. This is possibly due to the use of maximum likelihood based transforms with the MPE systems, and not using the transform prior consistently with the complete discriminative Bayesian framework described in section 6.1. This motivates the use of discriminative transforms to estimate priors and consistently use those prior in a discriminative fashion for discriminative Bayesian inference. Moreover, the lower-bound approximations used assume that the rank ordering of the hypotheses using the lower-bound to likelihood is same as given by the exact likelihood. However, this may not be true if a bound is not very tight and may lead to performance degradation, as described in chapter 5. This can be improved by using a method that can closely approximate the marginal likelihood. The next section describes the experimental investigation of the use of expectation propagation for Bayesian inference that attempts to find more accurate estimates of the marginal likelihood. It is then followed by the experimental investigation of the Bayesian approach to discriminative adaption.

9.2 Expectation Propagation Based Bayesian Inference

This section describes the experimental results for Bayesian adaptive inference using expectation propagation based approximation to the intractable marginal likelihood given in equation (6.15). The next section first compares the EP approximated marginal likelihood with other methods, on a toy problem. Thereafter, the performance of the EP approximation is evaluated on the speech recognition task in terms of word error rate.

9.2.1 Marginal Likelihood Approximations

It is important to compare the marginal likelihood given by different methods to assess the quality of approximations. The approximated marginal likelihood should be as close as possible to the true marginal likelihood. However, it is difficult to compute the exact marginal likelihood for a speech utterance on a complex system. The large number of dimensions in transforms, the number of mixture components used for state output distributions, and exponentially growing number of possible component sequences make it intractable to obtain the exact value of marginal likelihood on an LVCSR task. Therefore, to initially investigate the quality of different approximations, the Old Faithful Geyser data set [9] was used. The data set has a dimensionality of two. A simple left-to-right HMM with three emitting states and Gaussian output probability distributions was used as a model. An artificial Gaussian prior was used for the transform distribution. The marginal likelihood was computed on this system for a given observation sequence using different approaches. The likelihood estimates for a test sequence of 100 frames are plotted in figure 9.1.

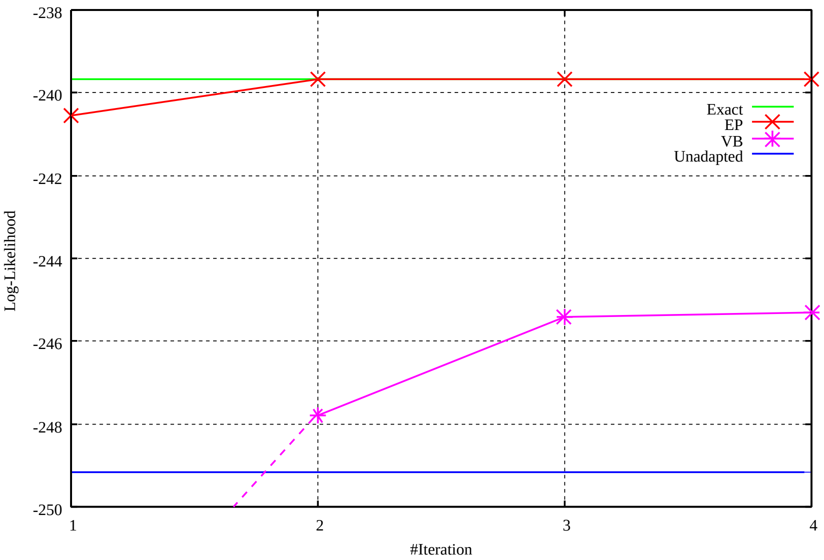


Figure 9.1: The likelihood estimates by EP and VB compared to the exact and the unadapted likelihoods on the Old Faithful Geyser data set

In figure 9.1, the exact likelihood was estimated by enumerating all possible state sequences in the HMM for given observation, and then doing exact marginalisation for the given transform distribution. The estimation of VB lower-bound and EP approximations are described in sections 5.2.3.2 and 5.3, respectively. As it can be seen, the variational Bayes gives a lower-bound to the exact marginal likelihood. Though the VB lower-bound estimates increases as more iterations are completed, they are still far-off than the exact marginal likelihood. On other hand, expectation propagation gives a near exact marginal likelihood just after performing one forward and backward pass. The EP approximated likelihood converges to the exact likelihood only in few iterations. Thus the use of EP approximated marginal likelihood for inference in speech recognition systems is likely to produce the same rank-ordering as the exact marginal likelihood giving a performance improvement.

The implementation of EP for computing marginal likelihoods on the speech recognition task described in section 9.1, however, requires a large amount of memory. This is because in the EP-based forward-backward algorithm described in 5.3, each forward and backward probabilities are not discrete distributions as in the standard HMM, but a probability function of transforms $\mathbf{W}_t$. The dimensionality of the transform is large, $D \times (D + 1)$. Therefore, a large amount of space is required for each node in the forward-backward trellis. Unlike the standard forward/backward algorithm for computing likelihood, these forward and backward distributions are required to be stored for the whole utterances, as they are used in the next pass of the message computation and refinement as described in algorithm 6. Moreover, an EP iteration involves converting the distributions from the canonical form to the moment form and vice-versa for approximating each message. This involves matrix inversion and is computationally very expensive and slow. This may also give rise to ill-conditioned matrices with high condition numbers, and may lead to inaccurate results. These issues make it infeasible to use the EP-based forward-backward algorithm even for utterances with few seconds of data.

To deal with the memory and computational requirement of the algorithm, the messages in the EP-based algorithm were constrained to be diagonal for further experiments on the speech recognition task. EP in this form is then used for approximating marginal likelihoods of hypotheses in the N-best rescoring framework. This approximation may be very inaccurate. This is because even the observation message, in equation A.34, which represents the likelihood of data at a particular frame for a given component is not properly accounted for, as they are diagonalised[1]. Table 9.2 investigates the effect of the this diagonal message approximation by comparing the EP approximated likelihood with full and diagonal messages. It should be noted that 'full' here refers to the form of messages as defined in equation (5.43)

[1] Note that $\mathbf{\Lambda}_d = \frac{\boldsymbol{\xi}\boldsymbol{\xi}^{\mathsf{T}}}{\sigma_d^2}$ in equation (A.34) will be forced to be diagonal in the observation message. By doing so, one can observe the difference it will make to the value of the observation likelihood on the left-hand side of the equation.

and (5.44) with full covariance matrices (actually block-diagonal covariance matrices, as rows of transforms are assumed independent), whereas ‘diagonal’ refers to all messages, including the observation message, constrained to be completely diagonal. As it can be seen from the table, the diagonal message approximation is indeed very crude in nature and gives a quantity far from the real marginal likelihood. However, as the absolute values of the marginal likelihood may not be important themselves for ranking of hypotheses, this assumption is still used for doing inference in speech recognition task. Besides, the approximated likelihood with diagonal message assumption shows difficulty in refining with more iterations. This may be because of the lack of proper interaction between forward and backward messages in successive iterations without the full messages. However, due to the relative computational efficiency, EP-based on this diagonal message assumption is used for ranking hypotheses on the speech recognition task. The results are presented in the next section.

#iter	Exact	EP Approximation	
		Full	Diagonal
1	-2.3967e+02	-2.4056e+02	2.1827e+03
2		-2.3967e+02	2.1827e+03
3		-2.3967e+02	2.9433e+03
4		-2.3967e+02	2.1827e+03

Table 9.2: The comparison of EP approximations with full and diagonal messages on the Old Faithful Geyser data set

9.2.2 Performance on the CTS Task

The EP approximation based on the diagonal message assumption as described above was used to compute the marginal likelihood or acoustic score of the hypotheses in the N-best list for doing inference in a speech recognition system. The performance of the EP-based approach was evaluated using the ML-SAT models on the CTS task described before. Adaptation was done at the utterance level to reflect the scenario of online adaptation and N-best list based rescoring was used to do the inference through equation (6.20). However, in this case the size of N-best lists was truncated to 5 as computing the likelihood through EP approximation was still quite time-consuming for all 7074 utterances in the testset. The effect of this truncating of N-best list is shown in table 9.3 for the VB lower-bound inference approach. At the first iteration of VB when the approximated likelihood tends to be much different from the exact likelihood thus making the system more prone to errors, the performance is improved after truncating the list from 150 to 5. This is because by limiting the search-space to only 5-best hypotheses, the possibility of selecting more errorful hypotheses in the 150-best list has been reduced. Variational Bayes, at the third iteration, gives an improvement of 1.0% absolute compared to SI system with the 5-best list, which is slightly less than that obtained with the 150-best list. The reduction is due to restriction in finding the best hypothesis within the limited 5-best list.

#iter	SI	SAT/VB	
		150-best	5-best
1	32.8	34.1	32.9
2		31.5	31.9
3		31.6	31.8

Table 9.3: The performance of VB based inference for utterance-level adaptation for 5-best and 150-best rescoring on the ML-SAT system

The EP-based approach was also evaluated using the above truncated 5-best list. The EP approximation used full transforms with two-base classes, however the covariance matrices (for each row) of transforms and messages were assumed diagonal. This makes the computation much faster, as described before, by eliminating large number of matrices inversion. A total of three iterations (each forward or backward sweep regarded as one iteration in EP) were used to estimate the marginal likelihood. The estimated likelihoods were then used to rank the hypotheses in the N-best list as in equation (6.20). The results from the ranking are given in table 9.4 for `eval03` testset on ML-SAT system using 5-best rescoring, and is compared to the performance of VB. As it can be seen from the table 9.4, the WER performance of the EP-based approach using the diagonal message assumption is much worse than the VB lower-bound. Though the VB lower-bound performance is also poorer than that of even the SI system initially, with further iterations, it improves significantly. However, the EP-based approach was not found to improve the performance even with further iterations. This can be seen as a consequence of using the diagonal message assumption in the EP-based approach. As already seen in table 9.2, in the diagonal message based EP approximation, the estimate for the marginal likelihood is far from the the exact likelihood (or the actual EP approximated likelihood), and it was not guaranteed to improve with further iterations.

#iter	SI	SAT (5-best)	
		VB	EP
1	32.8	32.9	35.6
2		31.9	35.7
3		31.8	35.6

Table 9.4: The performance of VB and EP based adaptive inference for utterance-level adaptation with 5-best rescoring

Therefore, it may be worthwhile to investigate the EP approximation with full messages for ranking hypotheses on the LVCSR task, as the initial results on the Old Faithful Geyser data set for approximating the marginal likelihood have been encouraging. However, due to the computational cost and memory requirement involved with the EP approximation with full messages, it was not further investigated on the LVCSR task for ranking hypotheses.

9.3 Discriminative Bayesian Adaptation and Inference

This section describes experimental evaluation of discriminative Bayesian adaption. The MAP estimation of discriminative transforms as described in section 6.3.1 is first investigated. This is then followed by the evaluation of the use of discriminative mapping transforms in a Bayesian framework as described in section 6.3.2.

9.3.1 MAP Estimation of Discriminative Transforms

The MAP estimation of discriminative transforms was investigated at the utterance-level, where there is only a small amount of adaptation data that may not give robust estimates of the transforms if estimated directly. It involves optimising a discriminative MAP objective function, and is examined below.

A weak-sense auxiliary function is commonly used for optimising discriminative objective functions both for training of HMMs and estimating discriminative transforms. The same approach was investigated to optimise the discriminative MAP objective function, as described in section 6.3.1.1. Its suitability for the optimisation was investigated by examining its characteristics. The relationship between the auxiliary function and the discriminative objective function was examined first for the case of estimating the standard discriminative transforms using equation (4.2), without a transform prior. Table 9.5 lists the values of objective function as well as auxiliary function at different iterations of DLT estimation. As it can be seen in the table, the weak-sense auxiliary function was found to increase the objective function with an increase in the auxiliary function, however the auxiliary function was not a lower-bound to the objective function, i.e. $\Delta\mathcal{F}(\mathbf{W}) \geq \Delta\mathcal{Q}(\hat{\mathbf{W}};\mathbf{W})$ is not true. This is despite the fact that the auxiliary function uses a smoothing constant of $E = 2$ as a more stable configuration, than the normal value of $E = 0.8$ used for estimating discriminative transforms in this work. Similar trends were observed even after further increasing the values of smoothing constants. However, as described before, higher values of the smoothing constant lead to small changes in the objective function.

#Iteration	$\mathcal{F}(\mathbf{W})$	$\Delta\mathcal{Q}(\hat{\mathbf{W}};\mathbf{W})$	$\Delta\mathcal{F}(\mathbf{W})$
1	29.361012	278.540059	6.301773
2	35.662785	79.976290	5.061386
3	40.724171	41.410955	0.428100
4	41.152271	15.840854	-

Table 9.5: The values of the MPE objective function and corresponding auxiliary function at different iterations of DLT estimation with a weak-sense auxiliary function

The weak-sense auxiliary function described in section 6.3.1.1 was used to estimate the discriminative MAP transforms for the utterance level adaptation. With the normally used values of smoothing factors for estimating discriminative transforms, the discriminative MAP objective function was found to generally oscillate with the iterations leading to very unreliable estimates of the transforms. The change in auxiliary function and the discriminative MAP criteria given in equation (6.34) is given in table 9.6 for a typical MAP estimation of discriminative transforms. In the table, the objective function has decreased at some iterations despite the increase in the auxiliary function. This is possible due to the fact that the weak-sense auxiliary function is not a lower-bound to the objective function, as described before.

#Iteration	$\mathcal{F}(\mathbf{W})$	$\Delta\mathcal{Q}(\hat{\mathbf{W}};\mathbf{W})$	$\Delta\mathcal{F}(\mathbf{W})$
1	4540.199288	67.266195	-28.280020
2	4511.919268	22.369408	19.304500
3	4531.223768	6.520218	-7.416561
4	4523.807207	4.755905	–

Table 9.6: The values of the MAP-MPE objective function as given in equation (6.34) along with the values of the weak-sense auxiliary function at different iteration of MAP-DLT estimation

It is worthwhile noting that an ML I-smoothing "prior" and a scale to the transform prior term were also used in the experiments. Though the weak-sense auxiliary function with the I-smoothing "prior" has been found to generally work for discriminative transforms estimation, the addition of a discriminative transform prior makes the scenario different. The I-smoothing term represents the *likelihood of certain observation points*, and its nature and dynamic range are similar to the numerator term in the discriminative objective function. On the other hand, the transform prior term represents the *likelihood of a transform* given the prior distribution, and its nature and dynamic range are different from those of the I-smoothing prior and other terms. This is specially true when the transform prior is very informative with small variances.

An alternative approach to the optimisation of the discriminative MAP objective function is based on a reverse-Jensen inequality as described in section 6.3.1.2. This was also investigated for estimating the discriminative MAP transforms for the utterance level adaptation. As noted earlier, this form of optimisation can be achieved by computing the smoothing factors using equation (6.44). The values of smoothing factors obtained using reverse-Jensen inequality is compared to that used in a a weak-sense auxiliary function in figure 9.2 for a typical utterance. The smoothing factor of the weak-sense auxiliary function is set as in equation (6.41), with $E = 0.8$.

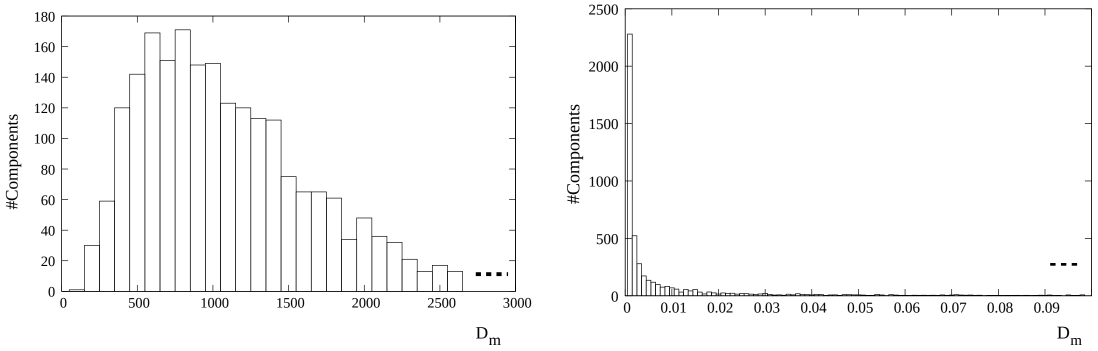


Figure 9.2: A histogram of smoothing factors D_m obtained with a reverse-Jensen inequality (left) and that used in a weak-sense auxiliary function (right).

It can be seen in the figure that a majority of the values of smoothing factors obtained by using reverse-Jensen inequality are very high compared to that used in a weak-sense auxiliary function. The majority of them turned out to be larger by a factor of 10^6 or more than the normally used values of smoothing factors in the weak-sense auxiliary function. This leads to minimal changes in the transform parameters, thus giving minimal or no change in the value of the discriminative MAP objective function, as shown in table 9.7. Consequently, the rank ordering of the hypotheses is not altered.

#Iteration	$\mathcal{F}(\mathbf{W})$	$\Delta\mathcal{Q}(\hat{\mathbf{W}};\mathbf{W})$	$\Delta\mathcal{F}(\mathbf{W})$
1	4540.199288	0.00000	0.00000
2	4540.199288	0.00000	0.00000
3	4540.199288	0.00000	0.00000
4	4540.199288	0.00000	–

Table 9.7: The values of the MAP-MPE objective function at different iterations obtained with a reverse-Jensen inequality based auxiliary function.

The high values of smoothing parameters with the reverse-Jensen inequality may be due to a very loose lower-bound obtained with it. It is known that the bounds obtained with reverse-Jensen's inequality are very loose [81]. It should be also noted that the reverse-Jensen inequality was not directly applied to the HMMs in this work, which may have further effects. Moreover, the transform estimation requires objective function maximisation involving computation of statistics summed over several components. In this case, the cumulative effect of the loose lower bounds may be even more severe than for acoustic model updates. It should be noted that in [1] further approximations were used so that the Jensen's reverse inequality based approach gave similar results to a weak-sense auxiliary function for model estimation. However, these approximations are not suitable for this work as a lower bound is desired. A strict lower-bound of the discriminative objective function obtained by tightly upper-bounding the whole denominator term can possibly improve the optimisation.

Instead of using above approaches, the standard gradient based optimisation schemes as described in section 6.3.1.3 can be used for discriminative MAP estimation. Computing the Hessian of the likelihood function for Newton's method is computationally expensive for large vocabulary speech recognition systems, and therefore is not examined here. Only the gradient based optimisation was investigated for estimating discriminative MAP transforms. The gradient of likelihood can be computed using equation (C.3), which can be applied to both numerator and denominator terms thus obtaining the gradient of discriminative objective function. The gradient based optimisation scheme was used to iteratively estimate the new value of MAP estimates of discriminative transforms, using equation (6.47). The discriminative MAP objective function against iteration is shown in figure 9.3 for a typical utterance from `eval03` testset. A learning rate of $\eta = 10^{-6}$ was selected manually for the utterance. As it can be seen, the MAP-MPE criteria increases smoothly and converges for the chosen learning rate, as more iterates are completed.

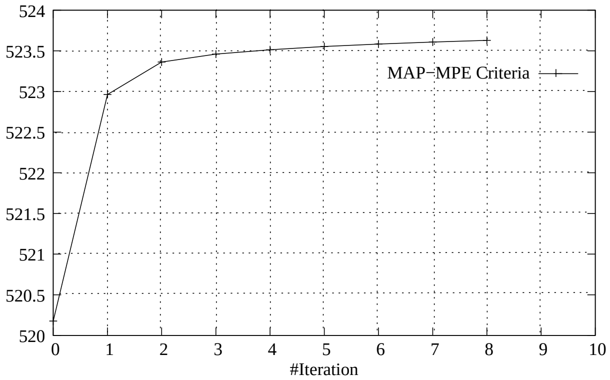


Figure 9.3: The change in MAP-MPE criteria using gradient ascent method with $\alpha^{\mathrm{p}} = 0.1$ and $\eta = 10^{-6}$ for a typical utterance

However, it should be noted that to use the gradient based optimisation, careful selection of learning parameter η is very important. Like the weak-sense auxiliary function, they are not guaranteed to converge and fine tuning of learning parameters is required. Furthermore, they are generally not elegant and efficient for a large speech recognition system with high dimensionality transform matrices. Therefore, the gradient based optimisation scheme was not used for ranking the hypotheses for the `eval03` testset for adaptive inference.

As seen above, it is difficult to obtain useful MAP estimates of discriminative transforms in an efficient manner. However, *robust* estimates of discriminative transforms are crucial for instantaneous unsupervised adaptation. Discriminative mapping transforms (DMTs) can be used for this purpose in a Bayesian framework as described in section 6.3.2 which is evaluated next.

9.3.2 DMT-based for Bayesian Adaptive Inference

This section examines the use of discriminative mapping transforms (DMTs) in a Bayesian framework as described in section 6.3.2. In this section, DMTs are used for utterance-level discriminative adaptation and decoding in the N-best rescoring framework. DMTs are applied over MAP estimates of linear transforms (MAPLR). The DMTs are estimated from training data, and have 1000 regression baseclasses. Other speaker-specific transforms had two baseclasses as described before. The experimental setup and baseline are already described at the beginning of this chapter.

The experimental results for the utterance-level N-best adaptive inference on `eval03` testset are given in table 9.8.

System	Adaptation		WER%	
	Training	Testing	ML	MPE
SI	–	–	32.8	29.2
SI	–	MLLR	35.5	32.4
		MAPLR	32.2	29.0
		MAPLR+DMT	30.8	28.4
SAT	MLLR	MLLR	35.2	32.3
		MAPLR	31.8	28.8
		MAPLR+DMT	30.9	28.6

Table 9.8: WER% performance for the utterance-level N-best adaptive inference on `eval03` testset

It can be seen from the table that the MAP estimates of linear transforms, MAPLR, has reduced the WER significantly. Using DMT with the MAPLR N-best adaptive inference gives a further improvement of 1.4% and 0.9% absolute on the ML SI and SAT systems, respectively, compared to using MAPLR alone. Similarly, for the MPE systems, the gains obtained with DMTs over MAPLR transforms for the utterance-level adaptation is 0.6% and 0.2% absolute on SI and SAT models, respectively. These gains are less than those obtained with DMTs for speaker level adaptation, as seen in chapter 7. The reduction in gains compared to the speaker-level adaptation is felt, in part, to be due to mismatch in applying a DMT estimated from speaker-level ML transforms to the utterance-level MAPLR. The SAT systems are more affected than the SI systems, as they are more sensitive to any mismatch in the training and the testing transforms than SI systems.

The effectiveness of N-best adaptive inference was also compared to the standard 1-best adaptation. A comparison of Bayesian N-best adaptation with the 1-best adaptation is given in table 9.9, typically for the ML-SAT system. As it can be observed, the N-best adaptive inference is giving better performance than the 1-best adaptation. Furthermore, the N-best MAPLR+DMT adaptation gives a gain of 0.9% absolute compared to using MAPLR alone, and a gain of 0.7% absolute compared to the 1-best adaptation using MAPLR+DMT.

Adaptation	Supervision	
	1-best	N-best
MAPLR	32.0	31.8
MAPLR+DMT	31.6	30.9

Table 9.9: A typical performance comparison for the 1-best and the N-best utterance-level adaptation on the ML-SAT system

9.4 Summary

This chapter has presented experimental results from investigation of Bayesian adaptation and inference. The Bayesian inference in adaptive system requires computation of marginal likelihoods. An approach based on expectation propagation was evaluated for approximation to the marginal likelihood, and then it was used for ranking N-best hypotheses for evaluation on the English CTS task. After that, discriminative Bayesian approaches were evaluated. The estimation of discriminative MAP transforms was investigated. This was followed by the investigation of using DMTs in a Bayesian framework. The DMTs were used with MAP estimates of ML transforms for N-best based adaptive inference. The results from the evaluation on the CTS task demonstrated a significant amount of performance gains through this approach compared to MAPLR.

Chapter 10
Conclusion

This thesis has investigated the problems of adaptation and adaptive training in large vocabulary speech recognition systems. In this chapter, the contributions of the thesis are summarised and possible directions for future work are outlined.

10.1 Summary of Work

In this thesis, the issues related to adaptation and adaptive training of acoustic models in state-of-the-art speech recognition systems have been addressed. As reviewed in chapters 3 and 4, state-of-the-art systems commonly use maximum-likelihood based linear transforms for the adaptation and adaptive training purpose. However, discriminative criteria such as minimum phone error are generally used to train the HMM parameters and have been found to significantly improve the performance. Therefore, the use of discriminative criteria for estimating transforms has been investigated in the past. However, they are not suitable for unsupervised adaptation. This is because they are biased towards the supervision hypothesis and thus highly sensitive to errors in it. Hence, they are not used in the adaptive training framework. Instead, ML transforms are used, and only canonical models are trained discriminatively. To deal with this problem, a discriminative mapping transform (DMT) based adaptive training scheme has been proposed in section 4.3. This is one of the contributions of the thesis. A consistent discriminative speaker adaptive training (DSAT) framework using DMTs both for training and testing has been investigated. The expressions for estimating transforms and canonical model parameters are derived, and the possible variants of the DMT-based DSAT scheme are discussed. The advantage of the DMT-based DSAT scheme is that it can be used for unsupervised adaptation as well, as DMTs do not directly depend upon the supervision hypothesis. The DMT-based DSAT scheme was evaluated on an English conversational speech recognition (CTS) task in chapter 7. In the experimental setup used, with one speaker-specific transform for speech and one for silence, the DMT-based DSAT scheme was found to yield a superior performance compared to the standard MLLR-based discriminative speaker adaptive training.

The adaptation of acoustic models requires some sample data from the target speaker. In many real-life applications of speech recognition systems, there is no separate adaptation data, and the model should be adapted online as soon as data becomes available, without delaying the response much. In this case, the data available for unsupervised adaption is small, and it is difficult to obtain robust estimates of transforms. This data sparsity problem in adaptation can be dealt with a Bayesian framework where the adaptation transform is regarded as a random variable with probability distributions. However, the Bayesian framework leads to intractable integrals for the marginal likelihood required for inference. Another contribution of this thesis is to approximate the intractable marginal likelihood in adaptive HMMs using expectation propagation (EP). In section 5.3, an expectation propagation based forward-backward algorithm is proposed to approximate the marginal likelihood. This can be used for Bayesian adaptive inference. The EP-based approximation was found to yield very accurate estimates of the marginal likelihood, compared to the lower-bound approaches. Using the EP-based approximation to marginal likelihood can thus give better rank-ordering of hypotheses in speech recognition. However, due to high computation cost and memory requirement involved with the EP-based approach, it was not used in its original form for rescoring the hypotheses on the CTS task. Rather, an a simplified approach was investigated for the speech recognition task. All the messages in the EP-based forward-backward algorithm were constrained to be diagonal. This simplified approach was found to yield poor approximations to the marginal likelihood, and did not give any improvement in performance over the variational Bayes lower-bound based reranking of the hypotheses.

Another contribution of the thesis is to extend the Bayesian framework for adaptive training and adaptation to discriminative criteria, as described in chapter 6. Discriminative adaptive training is first described from the Bayesian perspective in section 6.1. This is followed by the formulation of Bayesian inference in discriminative adaptive systems in section 6.2. This

Bayesian treatment leads to intractable integrals for marginal posteriors, which is required for inference. Various forms of approximations for discriminative adaptive inference have been proposed. This gives a discriminative way to rank possible hypotheses and select the best one, depending upon the discriminative inference evidence. A discriminative maximum-a-posterior approximation for inference has been also described in section 6.3.1. The estimation of discriminative MAP transforms requires optimisation of discriminative criteria along with the incorporated prior information. The forms of priors and their estimation have been detailed. The optimisation of the discriminative criteria was investigated using a weak-sense auxiliary function and other gradient based approaches in sections 6.3.1.1 and 6.3.1.3. A reverse-Jensen inequality based auxiliary function was also derived for optimising the discriminative objective function in section 6.3.1.2. The Bayesian approach has been further combined with discriminative mapping transforms in section 6.3.2, to obtain a framework for instantaneous discriminative adaptation. In the experiments, DMTs were applied over MAP estimates of the transforms. This was found to improve the performance of speech recognition systems for instantaneous adaptation compared to the other commonly used techniques used for online adaptation. A full Bayesian treatment is also possible for the DMT. The experimental results of Bayesian approaches were presented in chapters 8 and 9. The proposed methods were evaluated on a large vocabulary English conversational telephone speech (CTS) task. An utterance level adaptation was considered to simulate the scenario for instantaneous adaptation.

10.2 Future Work

There are a number of possible directions to further investigate and extend the work on adaptation and adaptive training presented in this dissertation. Some of them are described in this section.

The techniques proposed in this thesis have used model-based linear transformation of means. This is true for the discriminative mapping transforms based adaptive training scheme as well. However, feature-domain constrained transforms like CMLLR are efficient when implementing adaptive training. Therefore, the use of the DMT to map CMLLR transforms into discriminative ones would be useful to investigate. Similarly, it would be worthwhile to investigate the use of the constrained feature-domain transforms in the Bayesian framework. This requires finding an appropriate form of the prior for the constrained linear transforms. Also, as seen in section 6.1, the appropriate forms of the priors and their estimation is a major issue in formulating the Bayesian framework for discriminative criteria. It would be interesting to investigate the forms of priors for model parameters and transforms and their estimation for discriminative criteria.

As described in sections 5.2 and 6.1, approximations to intractable marginals are required for Bayesian adaptive inference. They are important for both maximum-likelihood and discriminative Bayesian adaptation frameworks. Though the proposed EP-based approach was found to give very accurate approximations to the marginal likelihood on a toy example, the computational and memory requirement makes it difficult to use for a large vocabulary speech recognition task. Therefore, it would be useful to investigate the techniques for making the approximation faster and efficient. Similarly, the approximation to the marginal posteriors would be worthwhile to investigate, as they are required for discriminative Bayesian inference.

As described in the thesis, state-of-the-art speech recognition systems generally use a weak-sense auxiliary function to optimise discriminative criteria. This is not necessarily a lower-bound to the discriminative objective function, as described in section 6.3.1.1. However, if a strict lower bound to the discriminative objective function can be obtained, it will give similar attributes as the ML auxiliary function. In section 6.3.1.2, the use of reverse-Jensen inequalities has been investigated, but this was not found to give a significant update to the estimates of the parameters. Therefore, other approaches of lower-bounding the discriminative objective function, which can give sufficient update with satisfactory convergence properties in the Bayesian framework, will be worthwhile to investigate. They can be used for the MAP estimation of discriminative transforms as well.

APPENDIX A
Expectation Propagation for Adaptive Inference

The inference in speech recognition with an adaptive system requires computation of the marginal likelihood given in equation (5.18), which is used as the acoustic score. In this section, the expectation propagation algorithm [67, 68, 119, 135] is applied to an adaptive HMM to approximate the intractable marginal likelihood.

A.1 EP-based Bayesian Adaptive Inference

A DBN for an adaptive HMM for one homogeneous block of T frames is shown in figure A.1. It should be noted that the transform is constrained to be constant for the homogeneous block by enforcing $\mathbf{W}_t = \mathbf{W}_{t-1}$. The goal is to approximate the marginal likelihood $p(\mathbf{O}|\mathcal{H})$ in equation (5.18) which is required for doing inference. In the DBN, the state sequence, the transform sequence and the observation sequence for T frames are represented by $\boldsymbol{\psi} = \{\psi_1, \ldots, \psi_T\}$, $\mathbb{W} = \{\mathbf{W}_1, \ldots, \mathbf{W}_T\}$ and $\mathbf{O} = \{\mathbf{o}_1, \ldots, \mathbf{o}_T\}$, respectively. The joint distribution of all variables in the DBN, using conditional independence relationships, is given by

$$p(\boldsymbol{\psi}, \mathbb{W}, \mathbf{O}) = P(\psi_1)p(\mathbf{W}_1)p(\mathbf{o}_1|\psi_1, \mathbf{W}_1)\prod_{t=2}^{T} P(\psi_t|\psi_{t-1})p(\mathbf{W}_t|\mathbf{W}_{t-1})p(\mathbf{o}_t|\psi_t, \mathbf{W}_t) \qquad \text{(A.1)}$$

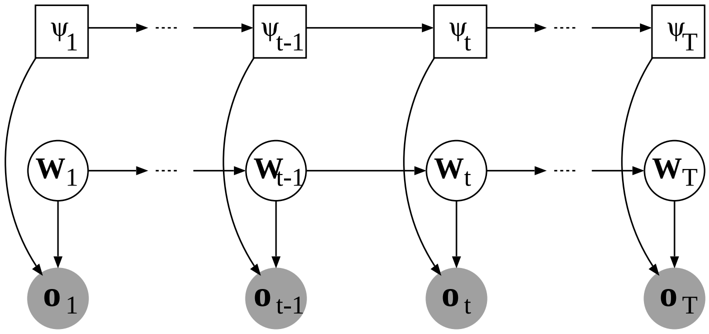


Figure A.1: A dynamic Bayesian network (DBN) for an adaptive HMM for one homogeneous block

By writing the prior $P(\psi_1)$ as $P(\psi_1|\psi_0)$ and $p(\mathbf{W}_1)$ as $p(\mathbf{W}_1|\mathbf{W}_0)$, this can be expressed as

$$p(\boldsymbol{\psi}, \mathbb{W}, \mathbf{O}) \quad = \quad \prod_{t=1}^{T} P(\psi_t|\psi_{t-1})p(\mathbf{W}_t|\mathbf{W}_{t-1})p(\mathbf{o}_t|\psi_t, \mathbf{W}_t) \tag{A.2}$$

Using Bayes' theorem leads to

$$p(\boldsymbol{\psi}, \mathbb{W}|\mathbf{O}) \quad = \quad \frac{1}{p(\mathbf{O})}\prod_{t=1}^{T} P(\psi_t|\psi_{t-1})p(\mathbf{W}_t|\mathbf{W}_{t-1})p(\mathbf{o}_t|\psi_t, \mathbf{W}_t) \tag{A.3}$$

Thus the likelihood $p(\mathbf{O})$ is the normalisation constant for the joint posteriors of state and transform sequences. The strategy adopted in this work is to compute the joint state and transform posteriors first, and then use their normalisation constants to obtain the marginal likelihood $p(\mathbf{O})$. In this way, both the posteriors and the likelihood can be obtained. In further derivations, a proportional sign, $\propto$, is used whenever a normalisation constant is dropped. In other words, the state and transform sequence joint posterior in the above equation is simply expressed as

$$p(\boldsymbol{\psi}, \mathbb{W}|\mathbf{O}) \quad \propto \quad \prod_{t=1}^{T} P(\psi_t|\psi_{t-1})p(\mathbf{W}_t|\mathbf{W}_{t-1})p(\mathbf{o}_t|\psi_t, \mathbf{W}_t) \tag{A.4}$$

Defining a potential function φ_t as

$$\varphi_t(\psi_{t-1}, \psi_t, \mathbf{W}_{t-1}, \mathbf{W}_t) = P(\psi_t|\psi_{t-1})p(\mathbf{W}_t|\mathbf{W}_{t-1})p(\mathbf{o}_t|\psi_t, \mathbf{W}_t) \tag{A.5}$$

the above joint posterior can be written as the product of potential functions

$$p(\boldsymbol{\psi}, \mathbb{W}|\mathbf{O}) \quad \propto \quad \prod_{t=1}^{T} \varphi_t(\psi_{t-1}, \psi_t, \mathbf{W}_{t-1}, \mathbf{W}_t) \tag{A.6}$$

These potential functions are shown in figure A.2 as in a standard graphical model. The potential function φ_t acts as a local function nodes between states $\{\psi_{t-1}, \mathbf{W}_{t-1}\}$ and $\{\psi_t, \mathbf{W}_t\}$

as shown in figure A.3. A state in the adaptive system at time t constitutes of both the speech state ψ_t and continuous transform state $\mathbf{W}_t$, and is also referred as 'supernode' $\{\psi_t, \mathbf{W}_t\}$ in this work.

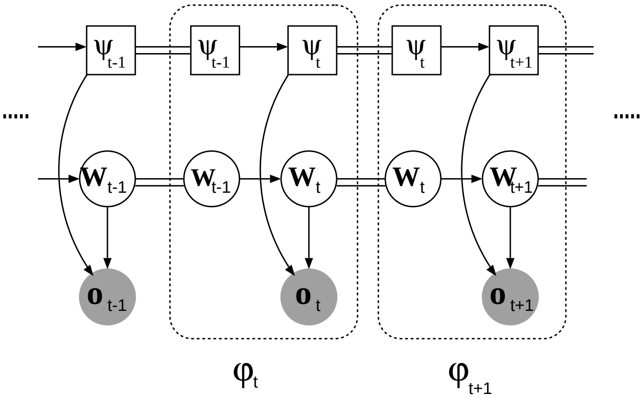


Figure A.2: The grouping of potential functions in the DBN for the adaptive HMM

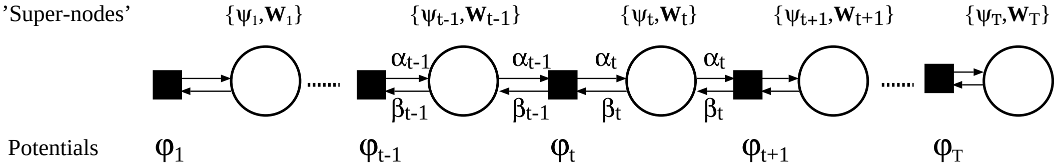


Figure A.3: The messages passing between 'supernodes' and potentials

The message passing between potential function and supernodes is also shown in the figure A.3. The dark box represents a potential node and circle represents a supernode. The message from potential φ_t forward to $\{\psi_t, \mathbf{W}_t\}$ is called the forward message, $\alpha_t(\psi_t, \mathbf{W}_t)$, and the message from φ_t back to $\{\psi_{t-1}, \mathbf{W}_{t-1}\}$ the backward message, $\beta_{t-1}(\psi_{t-1}, \mathbf{W}_{t-1})$. Both of these messages are chosen to have a form within the exponential family distribution. These forward and backward messages can be compared to the forward and backward probabilities in the standard forward-backward algorithm, which is also a form of a message-passing algorithm. However, the forward and backward messages in this case are a function of $\mathbf{W}_t$ as well, and are no more discrete forward and backward probabilities as in the standard HMM described in section 2.3.

In figure A.3, the potential function can be also defined as the product of outgoing messages at the potential node, by using the concept from graphical models, as

$$\varphi_t(\psi_{t-1}, \psi_t, \mathbf{W}_{t-1}, \mathbf{W}_t) = \alpha_t(\psi_t, \mathbf{W}_t)\beta_{t-1}(\psi_{t-1}, \mathbf{W}_{t-1}) \tag{A.7}$$

such that equation A.6 can be written in terms of forward and backward messages as

$$p(\boldsymbol{\psi}, \mathbb{W}|\mathbf{O}) \quad \propto \quad \prod_{t=1}^{T} \alpha_t(\psi_t, \mathbf{W}_t)\beta_{t-1}(\psi_{t-1}, \mathbf{W}_{t-1}) \tag{A.8}$$

This full-posterior $p(\boldsymbol{\psi}, \mathbb{W}|\mathbf{O})$ is approximated with an uncoupled distribution of exponential form

$$p(\boldsymbol{\psi}, \mathbb{W}|\mathbf{O}) \quad \approx \quad \prod_{t=1}^{T} p_t(\psi_t, \mathbf{W}_t) \tag{A.9}$$

where $p_t(\psi_t, \mathbf{W}_t)$ is the state-belief given as the product of all incoming message to the super-node given by

$$p_t(\psi_t, \mathbf{W}_t) \propto \alpha_t(\psi_t, \mathbf{W}_t)\beta_t(\psi_t, \mathbf{W}_t). \tag{A.10}$$

This allows factoring of the state and transform posterior sequence into the individual state and transform posteriors $p_t(\psi_t, \mathbf{W}_t)$ at time t, as in equation (A.9). The equation (A.9) can be thus re-expressed using (A.10) in (A.9) as

$$p(\boldsymbol{\psi}, \mathbb{W}|\mathbf{O}) \approx \hat{p}(\boldsymbol{\psi}, \mathbb{W}|\mathbf{O}) \propto \prod_{t=1}^{T} \alpha_t(\psi_t, \mathbf{W}_t)\beta_t(\psi_t, \mathbf{W}_t) \tag{A.11}$$

where $\hat{p}(\boldsymbol{\psi}, \mathbb{W}|\mathbf{O})$ is used for the approximate $p(\boldsymbol{\psi}, \mathbb{W}|\mathbf{O})$, arising from the above approximation. This equation can be rearranged as

$$\begin{aligned} &\hat{p}(\boldsymbol{\psi}, \mathbb{W}|\mathbf{O}) \\ &\propto \Big(\prod_{\tau<t-1} q_\tau(\psi_\tau, \mathbf{W}_\tau)\Big)\alpha_{t-1}(\psi_{t-1}, \mathbf{W}_{t-1})\beta_{t-1}(\psi_{t-1}, \mathbf{W}_{t-1})\alpha_t(\psi_t, \mathbf{W}_t)\beta_t(\psi_t, \mathbf{W}_t)\Big(\prod_{\tau>t} q_\tau(\psi_\tau, \mathbf{W}_\tau)\Big) \end{aligned} \tag{A.12}$$

Substituting the result from equation (A.7), it gives

$$\begin{aligned} &\hat{p}(\boldsymbol{\psi}, \mathbb{W}|\mathbf{O}) \\ &\propto \Big(\prod_{\tau<t-1} q_\tau(\psi_\tau, \mathbf{W}_\tau)\Big)\alpha_{t-1}(\psi_{t-1}, \mathbf{W}_{t-1})\varphi_t(\psi_t, \psi_{t-1}, \mathbf{W}_t, \mathbf{W}_{t-1})\beta_t(\psi_t, \mathbf{W}_t)\Big(\prod_{\tau>t} q_\tau(\psi_\tau, \mathbf{W}_\tau)\Big) \end{aligned} \tag{A.13}$$

Thus the current estimate of the two-slice marginal is given as

$$\begin{aligned} &\hat{p}_{t-1,t}(\psi_{t-1}, \psi_t, \mathbf{W}_{t-1}, \mathbf{W}_t) \\ &\quad = \frac{1}{k_t}\alpha_{t-1}(\psi_{t-1}, \mathbf{W}_{t-1})\varphi_t(\psi_t, \psi_{t-1}, \mathbf{W}_t, \mathbf{W}_{t-1})\beta_t(\psi_t, \mathbf{W}_t) \\ &\quad = \frac{1}{k_t}\alpha_{t-1}(\psi_{t-1}, \mathbf{W}_{t-1})\ P(\psi_t|\psi_{t-1})P(\mathbf{W}_t|\mathbf{W}_{t-1})p(\mathbf{o}_t|\psi_t, \mathbf{W}_t)\ \beta_t(\psi_t, \mathbf{W}_t) \end{aligned} \tag{A.14}$$

where k_t is a normalisation constant, and the value of the potential function from equation (A.5) has been substituted in the above equation. The normalisation constants can be used to compute the likelihood as

$$p(\mathbf{O}) = \prod_{t=1}^{T} k_t \tag{A.15}$$

This two-slice marginal in equation (A.14) is used for computing the state posteriors, and likelihood as above. It allows forward and backward messages to interact. The two-slice marginals are first computed with initialised values of forward and backward messages, and are then iteratively refined by using new estimates of the forward and backward messages. This gives iterative refinement of the posteriors as well as likelihood estimates.

To obtain the forward messages, the two-slice marginal $\hat{p}_{t-1,t}(\psi_{t-1}, \psi_t, \mathbf{W}_{t-1}, \mathbf{W}_t)$ in equation (A.14) is marginalised with respect to ψ_{t-1} and $\mathbf{W}_{t-1}$ to obtain the current estimate of state-belief $\hat{p}_t(\psi_t, \mathbf{W}_t)$ as

$$\hat{p}_t(\psi_t, \mathbf{W}_t) = \int_{d\mathbf{W}_{t-1}} \sum_{\psi_{t-1}} \hat{p}(\psi_{t-1}, \psi_t, \mathbf{W}_{t-1}, \mathbf{W}_t). \tag{A.16}$$

This will be a mixture of all components from all ψ_{t-1} as they sum over. This leads to the exponential growth of components in the overall posterior. Therefore, the state-belief is projected to (or approximated by) an exponential family distribution to check the growth in components. This is done by minimising the KL-divergence through moment matching (same expectation). The projected approximate belief $\bar{p}_t(\psi_t, \mathbf{W}_t)$ is given as

$$\bar{p}_t(\psi_t, \mathbf{W}_t) = \arg \min_{g(\psi_t, \mathbf{W}_t)} \mathtt{KL}\Big(\hat{p}(\psi_t, \mathbf{W}_t) || g(\psi_t, \mathbf{W}_t)\Big) \tag{A.17}$$

where $g(\psi_t, \mathbf{W}_t)$ is kept within an exponential family. Equation (A.10) tells that the state-belief is a product of the forward and backward messages at time t. Therefore, once the state-belief is projected, the new forward message is computed by dividing with the backward message as

$$\alpha_t(\psi_t, \mathbf{W}_t) = \frac{\bar{p}_t(\psi_t, \mathbf{W}_t)}{\beta_t(\psi_t, \mathbf{W}_t)} \tag{A.18}$$

In this way, the forward messages are computed successively for each time frame, keeping the backward messages $\beta_t(\psi_t, \mathbf{W}_t)$ constant during the forward pass. The current estimate of backward messages $\beta_t(\psi_t, \mathbf{W}_t)$ are used while computing the new values of the forward messages.

Similarly, the backward messages are estimated by running a backward pass, using the the current estimates of forward messages. The current estimate of the state-belief at $t-1$ is first obtained by marginalising the two-slice belief $\hat{p}_{t-1,t}(\psi_{t-1}, \psi_t, \mathbf{W}_{t-1}, \mathbf{W}_t)$ in equation (A.14) with respect to ψ_t and $\mathbf{W}_t$. The current estimate of the state-belief at time $t-1$, $\hat{p}_{t-1}(\psi_{t-1}, \mathbf{W}_{t-1})$, is given as

$$\hat{p}(\psi_{t-1}, \mathbf{W}_{t-1}) = \int_{d\mathbf{W}_t} \sum_{\psi_t} \hat{p}(\psi_{t-1}, \psi_t, \mathbf{W}_{t-1}, \mathbf{W}_t). \tag{A.19}$$

This is then projected/approximated by minimising the KL-divergence to give a projected state-belief $\bar{p}_{t-1}(\psi_{t-1}, \mathbf{W}_{t-1})$ as

$$\bar{p}_{t-1}(\psi_{t-1}, \mathbf{W}_{t-1}) = \underset{g(\psi_{t-1}, \mathbf{W}_{t-1})}{\operatorname{argmin}} \ \mathtt{KL}\Big(\hat{p}(\psi_{t-1}, \mathbf{W}_{t-1}) || g(\psi_{t-1}, \mathbf{W}_{t-1})\Big) \tag{A.20}$$

where $g(\psi_{t-1}, \mathbf{W}_{t-1})$ is again kept within an exponential family. Once the projected belief is obtained, the new backward message is computed by removing the forward message as

$$\beta_{t-1}(\psi_{t-1}, \mathbf{W}_{t-1}) = \frac{\bar{p}_{t-1}(\psi_{t-1}, \mathbf{W}_{t-1})}{\alpha_{t-1}(\psi_{t-1}, \mathbf{W}_{t-1})}. \tag{A.21}$$

The forward and backward passes are repeated until the messages are converged. Once converged, the value of the state-posteriors/state-beliefs can be obtained and the likelihood can be also estimated as given in equation (A.15). The EP-based forward-backward iterations can be summarised as given in algorithm 7.

```
Step 1: Initialise $\boldsymbol{\alpha}_t(\psi_t, \mathbf{W}_t)$ and $\boldsymbol{\beta}_t(\psi_t, \mathbf{W}_t)$,   $\forall t$ .
Step 2: Update $\boldsymbol{\alpha}_t(\psi_t, \mathbf{W}_t)$, for $t = 2$ to $T$.
    – marginalise the two-slice marginal $\hat{p}_{t-1,t}(\psi_{t-1}, \psi_t, \mathbf{W}_{t-1}, \mathbf{W}_t)$ with respect
      to $\psi_{t-1}$ and $\mathbf{W}_{t-1}$ to obtain the state-belief
    – project/approximate the state-belief distribution by moment matching
    – divide the projected-belief with $\beta_t(\psi_t, \mathbf{W}_t)$ to obtain $\boldsymbol{\alpha}_t(\psi_t, \mathbf{W}_t)$
Step 3: Update $\boldsymbol{\beta}_{t-1}(\psi_{t-1}, \mathbf{W}_{t-1})$, for $t = T$ to 2.
    – marginalise the two-slice marginal $\hat{p}_{t-1,t}(\psi_{t-1}, \psi_t, \mathbf{W}_{t-1}, \mathbf{W}_t)$ with respect
      to $\psi_t$ and $\mathbf{W}_t$ to obtain the state-belief
    – project/approximate the state-belief distribution by moment matching
    – divide the projected-belief with $\alpha_{t-1}(\psi_{t-1}, \mathbf{W}_{t-1})$ to obtain
      $\boldsymbol{\beta}_{t-1}(\psi_{t-1}, \mathbf{W}_{t-1})$
Step 4: Repeat steps (2) and (3), until the messages are converged.
Step 5: Compute the state posteriors and the likelihood.
```

Algorithm 7: *The EP-based forward-backward algorithm overview*

A.2 Combining Messages of Exponential Families

A.2.1 Product and Division of Messages

The messages in the expectation propagation algorithm are conveniently conditioned (multiplied/divided) by converting them to canonical forms. The Gaussian distribution of a parameter $\mathbf{w}$ in a normal moment form is represented as

$$p(\mathbf{w}) = \mathcal{N}(\mathbf{w}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) \tag{A.22}$$

where $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are the mean and covariance matrix for the parameter. This Gaussian distribution can be converted to a canonical form as

$$p(\mathbf{w}) = \exp\left(a + \boldsymbol{\eta}^{\mathrm{T}}\mathbf{w} - \frac{1}{2}\mathbf{w}^{\mathrm{T}}\boldsymbol{\Lambda}\mathbf{w}\right) \tag{A.23}$$

where $\{\boldsymbol{\eta}, \boldsymbol{\Lambda}\}$ are the natural parameters, and a is a normalisation constant given by

$$a = -\frac{1}{2}\left(D\log(2\pi) - \log|\boldsymbol{\Lambda}| + \boldsymbol{\eta}^{\mathrm{T}}\boldsymbol{\Lambda}^{-1}\boldsymbol{\eta}\right)$$

The parameters in one representation can be converted into another as

$$\begin{aligned}
\boldsymbol{\Lambda} &= \boldsymbol{\Sigma}^{-1} && \text{(A.24)}\\
\boldsymbol{\eta} &= \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} && \text{(A.25)}\\
\boldsymbol{\Sigma} &= \boldsymbol{\Lambda}^{-1} && \text{(A.26)}\\
\boldsymbol{\mu} &= \boldsymbol{\Lambda}^{-1}\boldsymbol{\eta} && \text{(A.27)}
\end{aligned}$$

The product or division of exponential family distributions is still in the exponential family, though the resulting distribution will require normalisation. The product of N Gaussian distributions $\mathcal{N}(\mathbf{w}; \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1), \ldots,\quad \mathcal{N}(\mathbf{w}; \boldsymbol{\mu}_N, \boldsymbol{\Sigma}_N)$ can be obtained by first converting them to canonical representations and then simply adding the natural parameters as

$$\begin{aligned}
\boldsymbol{\eta} &= \boldsymbol{\eta}_1 + \cdots + \boldsymbol{\eta}_N && \text{(A.28)}\\
\boldsymbol{\Lambda} &= \boldsymbol{\Lambda}_1 + \cdots + \boldsymbol{\Lambda}_N && \text{(A.29)}
\end{aligned}$$

Similarly, the parameters of distribution obtained by dividing $\mathcal{N}(\mathbf{w}; \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$ with $\mathcal{N}(\mathbf{w}; \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$ can be given in the canonical form as

$$\begin{aligned}
\boldsymbol{\eta} &= \boldsymbol{\eta}_1 - \boldsymbol{\eta}_2 && \text{(A.30)}\\
\boldsymbol{\Lambda} &= \boldsymbol{\Lambda}_1 - \boldsymbol{\Lambda}_2 && \text{(A.31)}
\end{aligned}$$

These messages after multiplication or division in the canonical form are converted back to the moment form using equations (A.26) and (A.27).

A.2.2 Combining Observation Message

The forward, backward and observation messages are combined together to find the two-slice marginal $\hat{p}(\psi_{t-1}, \psi_t, \mathbf{W}_{t-1}, \mathbf{W}_t)$ in equation (A.14). The forward and backward messages are in the form as in equations (5.43) and (5.44). They can be converted to the canonical representations as described above. The observation message should be also converted to an appropriate form in canonical representation to combine it with other messages. The observation message is given by

$$\mathcal{N}(\mathbf{o}_t; \mathbf{W}_t\boldsymbol{\xi}, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^D|\boldsymbol{\Sigma}|}} \exp\left(-\frac{1}{2}(\mathbf{o}_t - \mathbf{W}_t\boldsymbol{\xi})^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{o}_t - \mathbf{W}_t\boldsymbol{\xi})\right) \tag{A.32}$$

where $\boldsymbol{\xi}$ is the extended mean vector $\boldsymbol{\xi} = [\boldsymbol{\mu}^{\mathsf{T}} \ 1]^{\mathsf{T}}$, and $\boldsymbol{\Sigma}$ is the covariance matrix [1]. The covariance matrix $\boldsymbol{\Sigma}$ is assumed diagonal, and the rows of transforms as independent. Therefore, the above expression can be written in terms of each row of transform $\mathbf{w}_{td}$ as

$$\mathcal{N}(\mathbf{o}_t; \mathbf{W}_t\boldsymbol{\xi}, \boldsymbol{\Sigma}) = \prod_d p(o_{td}; \mathbf{w}_{td}^{\mathsf{T}}\boldsymbol{\xi}, \sigma_d^2) \tag{A.33}$$

where o_{td} is the dth element of observation vector $\mathbf{o}_t$, and σ_d^2 is the dth diagonal element of the covariance matrix $\boldsymbol{\Sigma}$. The above expression can be further expressed as

$$\begin{aligned}
\mathcal{N}(\mathbf{o}_t; \mathbf{W}_t\boldsymbol{\xi}, \boldsymbol{\Sigma}) &= \prod_d \frac{1}{\sqrt{2\pi\sigma_d^2}} \exp\left\{-\frac{1}{2\sigma_d^2}(o_{td} - \mathbf{w}_{td}^{\mathsf{T}}\boldsymbol{\xi})^2\right\} \\
&= \prod_d \frac{1}{\sqrt{2\pi\sigma_d^2}} \exp\left(-\frac{o_{td}^2}{2\sigma_d^2} + \frac{o_{td}}{\sigma_d^2}\boldsymbol{\xi}^{\mathsf{T}}\mathbf{w}_{td} - \frac{1}{2}\mathbf{w}_{td}^{\mathsf{T}}\frac{\boldsymbol{\xi}\boldsymbol{\xi}^{\mathsf{T}}}{\sigma_d^2}\mathbf{w}_{td}\right) \\
&= \prod_d \frac{1}{\sqrt{2\pi\sigma_d^2}} \exp\left(-\frac{o_{td}^2}{2\sigma_d^2}\right) \exp\left(\boldsymbol{\eta}_i^{\mathsf{T}}\mathbf{w}_{td} - \frac{1}{2}\mathbf{w}_{td}^{\mathsf{T}}\boldsymbol{\Lambda}_d\mathbf{w}_{td}\right) \\
&= \prod_d k_d \exp\left(\boldsymbol{\eta}_d^{\mathsf{T}}\mathbf{w}_{td} - \frac{1}{2}\mathbf{w}_{td}^{\mathsf{T}}\boldsymbol{\Lambda}_d\mathbf{w}_{td}\right)
\end{aligned} \tag{A.34}$$

where k_d is constant which does not necessarily normalise the resulting distribution in $\mathbf{w}_{td}$, and $\{\boldsymbol{\eta}_d, \boldsymbol{\Lambda}_d\}$ are the natural parameter for the distribution. They are given as

$$k_d = \frac{1}{\sqrt{2\pi\sigma_d^2}} \exp\left(-\frac{o_{td}^2}{2\sigma_d^2}\right) \tag{A.35}$$

$$\boldsymbol{\eta}_d = \frac{o_{td}}{\sigma_d^2}\boldsymbol{\xi} \tag{A.36}$$

$$\boldsymbol{\Lambda}_d = \frac{\boldsymbol{\xi}\boldsymbol{\xi}^{\mathsf{T}}}{\sigma_d^2} \tag{A.37}$$

The parameter of the observation message in this form is combined with other messages to obtain the two-slice marginal in equation (A.14).

[1] The index for the state and component has been dropped from the mean vector and the covariance matrix.

Appendix B
The Reverse-Jensen's Inequality and Parameter Estimation

A reverse-Jensen inequality based optimisation is used for the discriminative objective function in section 6.3.1.2. This section describes the form and parameters for the reverse-Jensen inequality, and shows its use in optimising discriminative objective function as well.

B.1 Reverse-Jensen's Inequality

A reverse-Jensen inequality reverses the usual Jensen's inequality to yield an upper bound to the log-summation or mixture likelihoods. One form described in [81, 82] finds this upper-bound by exploiting the convexity of the cumulant function of the components in log-summation, rather than using the concavity of the log function itself. This form of the reverse-Jensen inequality for mixtures of Gaussians has been used in this work. A GMM is represented in a canonical form as an exponential family of distributions with sufficient statistics X of observations, natural parameters Θ_m of the mixture component with weights c_m and cumulant function $\mathcal{K}(\Theta_m)$. The reverse Jensen inequality for the GMM is expressed as

$$\log \sum_{m} c_m \exp(\mathcal{A}(X) + X^{\mathtt{T}}\Theta_m - \mathcal{K}(\Theta_m)) \leq \sum_{m} -\tilde{\gamma}_m(\mathcal{A}(Y_m) + Y_m^{\mathtt{T}}\Theta_m - \mathcal{K}(\Theta_m)) + k \tag{B.1}$$

where $\tilde{\gamma}_m$ are positive weights, Y_m corresponds to component-dependent translated observations and k is a constant. The parameters for the upper bound that makes a tangential contact at $\tilde{\Theta}$ are given as [81]:

$$\begin{align}
k &= \log p(X|\tilde{\Theta}) + \sum_{m} \tilde{\gamma}_m \left(\mathcal{A}(Y_m) + Y_m^{\mathtt{T}}\Theta_m - \mathcal{K}(\tilde{\Theta}_m)\right) \tag{B.2}\\
Y_m &= \frac{\gamma_m}{\tilde{\gamma}_m}\left(\left.\frac{\partial \mathcal{K}(\Theta_m)}{\partial \Theta_m}\right|_{\tilde{\Theta}_m} - X\right) + \left.\frac{\partial \mathcal{K}(\Theta_m)}{\partial \Theta_m}\right|_{\tilde{\Theta}_m} \tag{B.3}\\
\tilde{\gamma}_m^{\mathtt{min}} &= \min \gamma \tag{B.4}\\
& \text{such that} \quad \frac{\gamma_m}{\gamma}\left(\left.\frac{\partial \mathcal{K}(\Theta_m)}{\partial \Theta_m}\right|_{\tilde{\Theta}_m} - X\right) + \left.\frac{\partial \mathcal{K}(\Theta_m)}{\partial \Theta_m}\right|_{\tilde{\Theta}_m} \in \frac{\partial \mathcal{K}(\Theta_m)}{\partial \Theta_m} \tag{B.5}\\
\tilde{\gamma}_m &= \tilde{\gamma}_m^{\mathtt{min}} + 4f(\gamma_m/2)\Big(X - \mathcal{K}'(\Theta_m)\Big)^{\mathtt{T}}\mathcal{K}''(\Theta_m)^{-1}\Big(X - \mathcal{K}'(\Theta_m)\Big) \tag{B.6}
\end{align}$$

The function $f(\gamma)$ is defined in equation (6.45).

B.2 Parameter Estimation Using Reverse-Jensen Inequality

The parameter estimation using a discriminative objective function is done by defining an auxiliary function, and then maximising it with respect to the parameter. The reverse-Jensen inequality has been used for conditional expectation maximisation in [81, 83]. It has been also investigated for model parameter estimation in HMMs in [1]. In this section, an auxiliary function based on the reverse-Jensen inequality is described assuming single-dimensional data for the sake of simplicity in the representation.

The auxiliary function for the numerator part of the discriminative objective function follows from the application of Jensen's inequality as in the ML estimation in section 2.3, and is given by

$$\mathcal{Q}^{\mathtt{num}}(\hat{\mathcal{M}};\mathcal{M}) = \tilde{K}^{\mathtt{num}} + \sum_{mt} \gamma_m^{\mathtt{num}}(t) \log \mathcal{N}(o_t; \hat{\mu}_m, \hat{\sigma}_m^2) \tag{B.7}$$

where $\gamma_m^{\mathtt{num}}(t)$ is the numerator occupation probability defined in equation (2.102) and computed using current model parameters $\mathcal{M}$, $\hat{\mu}_m$ and $\hat{\sigma}_m^2$ are the new mean and variance for the component m, o_t is the observation vector at time t, and $\tilde{K}^{\mathtt{num}}$ is a constant.

The auxiliary function for the denominator term is also obtained in the same form by using the reverse-Jensen inequality. However, as the application of the reverse-Jensen inequality to HMMs is highly complicated, it is not applied directly. Once the occupation probabilities for denominator components are computed using lattices as in the standard discriminative training, the process is regarded as training of GMMs. The reverse-Jensen's inequality is thus applied to denominator term treating it as GMMs however with the standard denominator occupations in equation (2.103) as component weights. After the application of the reverse-Jensen inequality, the denominator auxiliary function is given by

$$\mathcal{Q}^{\texttt{den}}(\hat{\mathcal{M}};\mathcal{M}) \quad = \quad \tilde{K}^{\texttt{den}} + \sum_{mt} \tilde{\gamma}_m^{\texttt{den}}(t) \log \mathcal{N}(y_{tm};\hat{\mu}_m,\hat{\sigma}_m^2) \tag{B.8}$$

where $\tilde{\gamma}_m^{\texttt{den}}(t)$ is the modified denominator occupation probability based on the current model parameters and observation, and is obtained using equation (B.6). It should be noted that the component occupation as well as the observations has been modified as a result of the application of reverse Jensen's inequality. The observation y_{tm} has now become dependent upon the component m, and is obtained through equation (B.3). The quantities required for finding $\tilde{\gamma}_m^{\texttt{den}}(t)$ and y_{tm} can be derived as follows. The probability of observation o_t for the component m is given by

$$\mathcal{N}(o_t;\mu_m,\sigma_m^2) = \frac{1}{\sqrt{2\pi\sigma_m^2}} \exp\left(-\frac{(o_t-\mu_m)^2}{2\sigma_m^2}\right). \tag{B.9}$$

This can be represented into an exponential form with parameters given as:

$$X \quad = \quad \begin{pmatrix} o_t & -\frac{1}{2}o_t^2 \end{pmatrix}^{\texttt{T}} \tag{B.10}$$

$$\mathcal{A}(X) \quad = \quad -\frac{1}{2}\log(2\pi) \tag{B.11}$$

$$\Theta_m \quad = \quad \begin{pmatrix} \frac{\mu_m}{\sigma_m^2} & \frac{1}{\sigma_m^2} \end{pmatrix}^{\texttt{T}} = (\vartheta_{m1} \quad \vartheta_{m2})^{\texttt{T}} \tag{B.12}$$

$$\mathcal{K}(\Theta) \quad = \quad \frac{1}{2}\left(\frac{\mu_m^2}{\sigma_m^2} - \log\left(\frac{1}{\sigma_m^2}\right)\right) \tag{B.13}$$

$$\mathcal{K}'(\Theta_m) \quad = \quad \begin{pmatrix} \frac{\vartheta_{m1}}{\vartheta_{m2}} & -\frac{\vartheta_{m1}^2}{2\vartheta_{m2}^2} - \frac{1}{2\vartheta_{m2}} \end{pmatrix}^{\texttt{T}} \tag{B.14}$$

$$\mathcal{K}''(\Theta_m) \quad = \quad \begin{pmatrix} 2\vartheta_{m1}^2 + \vartheta_{m2} & 2\vartheta_{m1}\vartheta_{m2} \\ 2\vartheta_{m1}\vartheta_{m2} & 2\vartheta_{m2}^2 \end{pmatrix} \tag{B.15}$$

These definitions can be used in equations (B.3) to (B.6), to obtain the values of $\tilde{\gamma}_m^{\texttt{den}}(t)$ and y_{tm} to obtain the denominator auxiliary function in equation (B.8).

Once both the numerator and denominator auxiliary function in equation (B.7) and (B.8) are defined, they can be combined together and maximised to estimate the model parameters. With some algebraic manipulations, the resulting auxiliary function can be expressed in the same form as the weak-sense auxiliary function, however with a smoothing factor of

$$D_m = \sum_t \gamma_m^{\texttt{den}}(t) + \sum_t \tilde{\gamma}_m^{\texttt{den}}(t) = \gamma_m^{\texttt{den}} + \tilde{\gamma}_m^{\texttt{den}} \tag{B.16}$$

where $\gamma_m^{\texttt{den}}(t)$ is the standard denominator occupancy for component m at time t given in equation (2.103) and $\tilde{\gamma}_m^{\texttt{den}}(t)$ can be computed using equation (B.6). Using the parameters for single-dimensional Gaussians from equation (B.10) to (B.15) in equation (B.6), the value of $\tilde{\gamma}_m^{\texttt{den}}(t)$ is obtained as

$$\begin{aligned}\tilde{\gamma}_m^{\texttt{den}} \quad = \quad & \sum_t \max\left(\gamma_m^{\texttt{den}}(t)\left(\frac{o_t^2}{\mu_m^2+\sigma_m^2}\right), 0\right) \\ & + 4\sum_t f\left(\gamma_m^{\texttt{den}}(t)/2\right)\frac{(o_t-\mu_m)^2}{\sigma_m^2} \\ & + 4\sum_t f\left(\gamma_m^{\texttt{den}}(t)/2\right)\left(\frac{(o_t-\mu_m)^2}{\sigma_m^2} - 1\right)^2 \end{aligned} \tag{B.17}$$

where μ_m and σ_m^2 are the current mean and variance for the component m. Similarly, the value of smoothing factor D_m for the multivariate Gaussian case can be obtained as given in equation (6.44). Therefore, the discriminative objective function can be optimised using the reverse-Jensen inequality simply by altering the smoothing factors as in equation (B.16).

Appendix C

The Gradient and Hessian of the Log-likelihood Function

The optimisation of the discriminative MAP objective function in equation (6.34) through Newton's method as described in section 6.3.1.3 requires gradient and Hessian of numerator and denominator likelihoods with respect to the transform. Therefore, the gradient and Hessian of the likelihood function is derived in this section.

The derivative of the log-likelihood with respect to the transform is given by

$$
\begin{aligned}
\frac{\partial \log p(\mathbf{O}|\mathcal{M},\mathbf{W})}{\partial {\mathbf{w}_d}^{\mathrm{T}}} &= \frac{1}{p(\mathbf{O}|\mathcal{M},\mathbf{W})} \frac{\partial}{\partial {\mathbf{w}_d}^{\mathrm{T}}} \left\{ \sum_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} \prod_{t=1}^{T} P(\theta_t|\theta_{t-1}) p(\mathbf{o}_t|\theta_t;\mathcal{M},\mathbf{W}) \right\} \\
&= \frac{1}{p(\mathbf{O}|\mathcal{M},\mathbf{W})} \sum_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} \sum_{t=1}^{T} \left\{ \frac{\prod_{\tau=1}^{T} P(\theta_\tau|\theta_{\tau-1}) p(\mathbf{o}_\tau|\theta_\tau;\mathcal{M},\mathbf{W})}{p(\mathbf{o}_t|\theta_t;\mathcal{M},\mathbf{W})} \right\} \frac{\partial p(\mathbf{o}_t|\theta_t;\mathcal{M},\mathbf{W})}{\partial {\mathbf{w}_d}^{\mathrm{T}}} \\
&= \sum_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} \sum_{t=1}^{T} \frac{p(\mathbf{O},\boldsymbol{\theta}|\mathcal{M},\mathbf{W})}{p(\mathbf{O}|\mathcal{M},\mathbf{W})} \quad \frac{1}{p(\mathbf{o}_t|\theta_t;\mathcal{M},\mathbf{W})} \frac{\partial p(\mathbf{o}_t|\theta_t;\mathcal{M},\mathbf{W})}{\partial {\mathbf{w}_d}^{\mathrm{T}}} \\
&= \sum_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} \sum_{t=1}^{T} P(\boldsymbol{\theta}|\mathbf{O},\mathcal{M},\mathbf{W}) \frac{\partial \log p(\mathbf{o}_t|\theta_t;\mathcal{M},\mathbf{W})}{\partial {\mathbf{w}_d}^{\mathrm{T}}} \qquad \text{(C.1)}
\end{aligned}
$$

This can be rearranged as

$$\frac{\partial \log p(\mathbf{O}|\mathcal{M},\mathbf{W})}{\partial \mathbf{w}_d{}^{\mathrm{T}}} = \sum_{jm}\sum_{t} P(\theta_t^{jm}|\mathbf{O},\mathcal{M},\mathbf{W})\frac{\partial \log p(\mathbf{o}_t|\theta_t;\mathcal{M},\mathbf{W})}{\partial \mathbf{w}_d{}^{\mathrm{T}}} \tag{C.2}$$

where θ_t^{jm} represents being in state j and mixture component m at time t, and is computed through the standard forward-backward algorithm described in section 2.3. Therefore, the gradient of the log-likelihood with respect to the dth row of the transform, $\mathbf{w}_d$, is

$$\frac{\partial \log p(\mathbf{O}|\mathcal{M},\mathbf{W})}{\partial \mathbf{w}_d{}^{\mathrm{T}}} = \sum_{jm}\sum_{t}\gamma_{jm}(t)\frac{\left(o_{td}-\mathbf{w}_d{}^{\mathrm{T}}\boldsymbol{\xi}_{jm}\right)\boldsymbol{\xi}_{jm}^{\mathrm{T}}}{\sigma_{jm,d}^2} \tag{C.3}$$

where $\boldsymbol{\xi}_{jm} = [\boldsymbol{\mu}_{jm}^{\mathrm{T}}\ \ 1]^{\mathrm{T}}$ is the extended mean vector, o_{td} represents the dth element of observation vector $\mathbf{o}_t$, and $\sigma_{jm,d}^2$ is the dth diagonal element of $\boldsymbol{\Sigma}_{jm}$. In the above equation, $\gamma_{jm}(t)$ is a state-component posterior given as

$$\gamma_{jm}(t) = P(\theta_t^{jm}|\mathbf{O},\mathcal{M},\mathbf{W}) \tag{C.4}$$

The second derivative of the log-likelihood function can be derived from equation (C.3) and is given as

$$\frac{\partial^2 \log p(\mathbf{O}|\mathcal{M},\mathbf{W})}{\partial \mathbf{w}_d \partial \mathbf{w}_d{}^{\mathrm{T}}} = \sum_{jm}\sum_{t}\frac{\partial\gamma_{jm}(t)}{\partial \mathbf{w}_d}\frac{(o_{td}-\mathbf{w}_d{}^{\mathrm{T}}\boldsymbol{\xi}_{jm})\boldsymbol{\xi}_{jm}^{\mathrm{T}}}{\sigma_{jm,d}^2} - \sum_{jm}\sum_{t}\gamma_{jm}(t)\frac{\boldsymbol{\xi}_{jm}\boldsymbol{\xi}_{jm}^{\mathrm{T}}}{\sigma_{jm,d}^2} \tag{C.5}$$

It should be noted that the derivative of the state-component posterior $\gamma_{jm}(t)$ is not zero, as it based on the current (not previous) value of the transform with respect to which the derivative is being computed. The derivative of the state-component posterior occupation is given by

$$\begin{aligned}
\frac{\partial\gamma_{jm}(t)}{\partial \mathbf{w}_d} &= \frac{\partial P(\theta_t^{jm}|\mathbf{O},\mathcal{M},\mathbf{W})}{\partial \mathbf{w}_d} \\
&= \frac{\partial}{\partial \mathbf{w}_d}\left(\frac{p(\theta_t^{jm},\mathbf{O}|\mathcal{M},\mathbf{W})}{p(\mathbf{O}|\mathcal{M},\mathbf{W})}\right) \\
&= \frac{1}{p(\mathbf{O}|\mathcal{M},\mathbf{W})}\frac{\partial p(\mathbf{O},\theta_t^{jm}|\mathcal{M},\mathbf{W})}{\partial \mathbf{w}_d} - \frac{p(\mathbf{O},\theta_t^{jm}|\mathcal{M},\mathbf{W})}{p(\mathbf{O}|\mathcal{M},\mathbf{W})^2}\frac{\partial p(\mathbf{O}|\mathcal{M},\mathbf{W})}{\partial \mathbf{w}_d}
\end{aligned}$$

Using the derivation for the gradient of the likelihood as in equation (C.1) once again for the first term in the above equation and rearranging gives

$$\begin{aligned}
\frac{\partial \gamma_{jm}(t)}{\partial \mathbf{w}_d} &= \sum_{in}\sum_{\tau} P(\theta_\tau^{in}, \theta_t^{jm}|\mathbf{O},\mathcal{M},\mathbf{W}) \frac{\partial \log p(\mathbf{o}_\tau|\theta_\tau^{in},\mathcal{M},\mathbf{W})}{\partial \mathbf{w}_d} - \gamma_{jm}(t)\frac{\partial \log p(\mathbf{O}|\mathcal{M},\mathbf{W})}{\partial \mathbf{w}_d} \\
&= \sum_{in}\sum_{\tau} P(\theta_\tau^{in}, \theta_t^{jm}|\mathbf{O},\mathcal{M},\mathbf{W}) \frac{(o_{\tau d} - \mathbf{w}_d{}^{\mathrm{T}}\boldsymbol{\xi}_{in})\boldsymbol{\xi}_{in}}{\sigma_{in,d}^2} \\
&\quad - \gamma_{jm}(t)\sum_{in}\sum_{\tau}\gamma_{in}(\tau)\frac{(o_{\tau d} - \mathbf{w}_d{}^{\mathrm{T}}\boldsymbol{\xi}_{in})\boldsymbol{\xi}_{in}}{\sigma_{in,d}^2} \\
&= \sum_{in}\sum_{\tau}\Big(P(\theta_\tau^{in}, \theta_t^{jm}|\mathbf{O},\mathcal{M},\mathbf{W}) - \gamma_{jm}(t)\gamma_{in}(\tau)\Big)\frac{(o_{\tau d} - \mathbf{w}_d{}^{\mathrm{T}}\boldsymbol{\xi}_{in})\boldsymbol{\xi}_{in}}{\sigma_{in,d}^2} \qquad \text{(C.6)}
\end{aligned}$$

Therefore, the Hessian of the log-likelihood can be obtained by substituting equation (C.6) in equation (C.5) as

$$\begin{aligned}
&\frac{\partial^2 \log p(\mathbf{O}|\mathcal{M},\mathbf{W})}{\partial \mathbf{w}_d \partial \mathbf{w}_d{}^{\mathrm{T}}} \\
&= \sum_{jm,t}\sum_{in,\tau}\Big(P(\theta_\tau^{in}, \theta_t^{jm}|\mathbf{O},\mathcal{M},\mathbf{W}) - \gamma_{jm}(t)\gamma_{in}(\tau)\Big)\frac{(o_{td} - \mathbf{w}_d{}^{\mathrm{T}}\boldsymbol{\xi}_{jm})(o_{\tau d} - \mathbf{w}_d{}^{\mathrm{T}}\boldsymbol{\xi}_{in})\boldsymbol{\xi}_{in}\boldsymbol{\xi}_{jm}^{\mathrm{T}}}{\sigma_{jm,d}^2\sigma_{in,d}^2} \\
&\quad - \sum_{jm,t}\gamma_{jm}(t)\frac{\boldsymbol{\xi}_{jm}\boldsymbol{\xi}_{jm}^{\mathrm{T}}}{\sigma_{jm,d}^2} \qquad \text{(C.7)}
\end{aligned}$$

The joint posterior $P(\theta_\tau^{in}, \theta_t^{jm}|\mathbf{O},\mathcal{M},\mathbf{W})$ required for computing the Hessian is given by

$$P(\theta_\tau^{in}, \theta_t^{jm}|\mathbf{O},\mathcal{M},\mathbf{W}) = \frac{\alpha_{jm}(t)\ p\big(\theta_{kn}, \mathbf{o}_{t+1,\ldots,}\mathbf{o}_\tau|\theta_{jm}(t),\mathcal{M},\mathbf{W}\big)\ \beta_{kn}(\tau)}{p(\mathbf{O}|\mathcal{M},\mathbf{W})} \qquad \text{(C.8)}$$

This joint posterior can be computed using a *double forward-backward algorithm* [99]. However, the computational cost of finding this joint posterior is quadratic in number of states, mixture components and the observation sequence length.

References

[1] M. Afify. Extended Baum-Welch reestimation of Gaussian mixture models based on reverse Jensen inequality. In *Proc. Interspeech*, pages 1113–1116, 2005. 6.3.1.2, 6.3.1.2, 9.3.1, B.2

[2] S. M. Ahadi and P. C. Woodland. Combined Bayesian and predictive techniques for rapid speaker adaptation of continuous density hidden Markov models. *Computer Speech and Language*, 11:187–206, 1997. 3.1.1

[3] S. M. Ahadi-Sarkani. *Bayesian and Predictive Techniques for Speaker Adaptation*. PhD thesis, Cambridge University, 1996. 3.1.1

[4] T. Anastasakos and S. V. Balakrishnan. The use of confidence measures in unsupervised adaptation of speech recognisers. In *Proc. ICSLP*, volume 6, pages 2303–2306, 1998. 3.1.5.1, 3.1.5.1

[5] T. Anastasakos, J. Mcdonough, R. Schwartz, and J. Makhoul. A compact model for speaker adaptive training. In *Proc. ICSLP*, pages 1137–1140, 1996. 1, 3, 3.2, 3.2.1, 3.2.1, 3.2.1.1, 3.2.1.1, 4.2

[6] B. S. Atal. Effectiveness of linear prediction characteristics of the speech wave for automatic speaker identification and verification. *Journal of the Acoustical Society of America*, 55(6):1304–1312, 1974. 2.2.3.1, 3.2

[7] H. Attias. A variational Bayesian framework for graphical models. In *NIPS 12*, 2000. 2.6.3.4, 2.6.3.4, 2.6.3.4

[8] S. Axelrod, R. Gopinath, and P. Olsen. Modeling with a subspace constraint on inverse covariance matrices. In *Proc. ICSLP*, 2002. 2.3.5

[9] A. Azzalini and A. W. Bowman. A look at some data on the Old Faithful geyser. *Applied Statistics*, 39:357–365, 1990. 9.2.1

[10] L. R. Bahl, P. F. Brown, P. V. de Souza, and R. L. Mercer. Maximum mutual information estimation of hidden Markov model parameters for speech recognition. In *Proc. ICASSP*, volume 1, pages 49–52, 1986. 2.3.4.1, 4.1

[11] L. R. Bahl, P. V. de Souza, P. S. Gopalakrishnan, D. Nahamoo, and M. A. Picheny. Context dependent modelling of phones in continuous speech using decision trees. In *Proc. DARPA Speech and Natural Language Processing Workshop*, pages 264–268, 1991. 2.3.5

[12] L. R. Bahl, P. V. de Souza, P. S. Gopalakrishnan, D. Nahamoo, and M. A. Picheny. Decision trees for phonological rules in continuous speech. In *Proc. ICASSP*, pages 185–188, 1991. 2.3.5

[13] L. E. Baum and J. A. Eagon. An inequality with applications to statistical estimation for probabilistic functions of Markov processes and to a model for ecology. *Bulletin of American Mathematical Society*, 73:360–363, 1967. 2.3.2

[14] M. J. Beal. *Variational Algorithms for Approximate Bayesian Inference*. PhD thesis, University College London, 2003. 2.6.3.4, 5.2.3.2

[15] M. J. Beal and Z. Ghahramani. The variational Bayesian EM algorithm for incomplete data: with application to scoring graphical model structures. *Bayesian Statistics*, 7: 1–10, 2003. 2.3.3, 2.6.3.4, 2.6.3.4

[16] H. Botterweck. Anisotropic MAP defined by eigenvoices for large vocabulary continuous speech recognition. In *Proc. ICASSP*, 2001. 3.2.2

[17] P. F. Brown, V. J. Della Pietra, P. V. de Souza, J. C. Lai, and R. L. Mercer. Class-based N-gram models of natural language. *Computational Linguistics*, 18(4):467–479, 1992. 2.5

[18] N. Campbell. Canonical variate analysis - a general formulation. *Australian Journal of Statistics*, 26:86–96, 1984. 2.3.6, 2.3.6, 2.3.6

[19] S. F. Chen and J. T. Goodman. An empirical study of smoothing techniques for language modeling. In *Proc. 34th Annual Meeting of the Association for Computational Linguistics*, pages 310–318, 1996. 2.5

[20] S. F. Chen and J. T. Goodman. An empirical study of smoothing techniques for language modeling. Technical Report TR-10-98, Computer Science Group, Harvard University, 1998. 2.5

[21] C. Chesta, O. Siohan, and C. Lee. Maximum a posteriori linear regression for hidden Markov model adaptation. *Proc. Eurospeech*, 1:211–214, 1999. 1, 3.1.2.4, 3.1.2.4

[22] J. Chien, C. Lee, and H. Wang. A hybrid algorithm for speaker adaptation using MAP transformation and adaptation. *IEEE signal processing letters*, 4(6):167–169, 1997. 1

[23] J. T. Chien. Linear regression based Bayesian predictive classification for speech recognition. *IEEE transactions on speech and audio processing*, 11:70–79, 2003. 5.2.2, 5.2.2

[24] W. Chou. Maximum a-posterior linear regression with elliptical symmetric matrix variate priors. In *Proc. ICASSP*, pages 1–4, 1999. 5.2.3.1

[25] W. Chou, C. H. Lee, and B. H. Juang. Minimum error rate training based on N-best string models. In *Proc. ICASSP*, pages 652–655, 1993. 2.3.4.1

[26] S. B. Davis and P. Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuous spoken sentences. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 28:357–366, 1980. 2.2.2

[27] M. DeGroot. *Optimal Statistical Decisions.* McGraw-Hill, New York, 1970. 2.3.4.2

[28] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39:1–38, 1977. 2.3.2, 5.2.3.1

[29] V. V. Digalakis, D. Rtischev, and L. G. Neumeyer. Speaker adaptation using constrained estimation of Gaussian mixtures. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 2:357–366, 1995. 3.1.2, 3.1.2.3

[30] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification.* John Wiley & Sons, Inc, 2001. 5.2.3.1, 6.2.3.2, 6.3.1.3

[31] G. Evermann and P. C. Woodland. Large vocabulary decoding and confidence estimation using word posterior probabilities. In *Proc. ICASSP*, volume 3, pages 1655–1659, 2000. 2.6.4, 3.1.5.1, 3.1.5.1

[32] G. Evermann, H. Y. Chan, M. J. F. Gales, T. Hain, X. Liu, D. Mrva, L. Wang, and P. C. Woodland. Development of the 2003 CU-HTK conversational telephone speech transcription system. In *Proc. ICASSP*, 2004. 2.2.3.3

[33] G. Evermann, H. Y. Chan, M. J. F. Gales, B. Jia, X. Liu, D. Mrva, K. C. Sim, L. Wang, P. C. Woodland, and K. Yu. Development of the 2004 CU-HTK English CTS systems using more than two thousand hours of data. In *Proc. Rich Transcription Workshop*, 2004. 3.1.5.1

[34] J. Fiscus. NIST speech recognition scoring toolkit, version 1.2, 2007. URL `http://www.nist.gov/speech/tools/`. 2.7, 2.7, 7.1

[35] J. G. Fiscus. A post-processing system to yield reduced word error rates: Recogniser output voting error reduction (ROVER). In *Proc. ASRU*, 1997. 2.6.4

[36] K. Fukunaga. *Introduction to statistical pattern recognition*. Academic Press, 1972. 2.3.6, 2.3.6, 2.3.6

[37] S. Furui. Unsupervised speaker adaptation method based on hierarchical spectral clustering. In *Proc. ICASSP*, volume 1, pages 286–289, 1989. 3.1.3

[38] S. Furui. Speaker independent isolated word recognition using dynamic features of speech spectrum. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 34: 52–59, 1986. 2.2.2

[39] M. Gales. Machine learning for speech and language processing. In *Foresight Cognitive Systems Workshop*, 2004. (document), 2.3

[40] M. Gales and S. Young. *The Application of Hidden Markov Models in Speech Recognition*. now Publishers Inc., 2007. 1, 2.3, 2.6.4, 3, 5

[41] M. J. F. Gales. Cluster adaptive training of hidden Markov models. *IEEE Transactions on Speech and Audio Processing*, 8:417–428, 2000. 3.1.3, 3.2.2, 3.2.2

[42] M. J. F. Gales. Acoustic factorisation. In *Proc. ASRU*, 2001. 5.2.2, 5.2.2, 5.2.2

[43] M. J. F. Gales. Adaptive training for robust ASR. In *Proc. ASRU*, 2001. 3.1, 3.2, 4.2, 5, 5.1, 5.1, 5.2, 6, 6.1

[44] M. J. F. Gales. Maximum likelihood linear transformations for HMM-based speech recognition. *Computer Speech and Language*, 12:75–98, 1998. 1, 3.1.2.2, 3.1.2.2, 3.1.2.2, 3.1.2.3, 3.1.2.3, 3.1.2.3, 3.2, 3.2.1, 3.2.1, 3.2.1.2, 3.2.1.2

[45] M. J. F. Gales. Transformation smoothing for speaker and environmental adaptation. In *Proc. Eurospeech*, 1997. 3.1.3

[46] M. J. F. Gales. Cluster adaptive training for speech recognition. In *Proc. ICSLP*, pages 1783–1786, 1998. 3.2.2

[47] M. J. F. Gales. Semi-tied covariance matrices for hidden Markov models. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 7:272–281, 1999. 2.3.5, 2.3.5

[48] M. J. F. Gales. *Model-Based Techniques for Noise Robust Speech Recognition*. PhD thesis, Cambridge University, 1995. 2.2.2

[49] M. J. F. Gales. The generation and use of regression class trees for MLLR adaptation. Technical Report CUED/F-INFENG/TR263, Cambridge University Engineering Department, 1996. 3.1.2, 3.1.4, 3.1.4

[50] M. J. F. Gales. Maximum likelihood linear transformations for HMM-based speech recognition. Technical Report CUED/F-INFENG/TR291, Cambridge University, 1997. (via anonymous) `ftp://svr-www.eng.cam.ac.uk`. 3.1.2

[51] M. J. F. Gales and P. C. Woodland. Mean and variance adaptation within the MLLR framework. *Computer Speech and Language*, 10:249–264, 1996. 3.1.2.2, 4.3, 4.3, 1

[52] M. J. F. Gales and P. C. Woodland. Recent advances in large vocabulary continuous speech recognition: An HTK perspective. In *ICASSP Tutorial Presentation*, 2006. (document), 2.8, 2.9

[53] M. J. F. Gales, B. Jia, X. Liu, K. C. Sim, P. C. Woodland, and K. Yu. Development of the CUHTK 2004 Mandarin conversational telephone speech transcription system. In *Proc. ICASSP*, pages 841–844, 2005. 2.2.3.2, 3.2

[54] M. J. F Gales, D. Y. Kim, P. C. Woodland, H. Y. Chan, D. Mrva, R. Sinha, and S. E. Tranter. Progress in the CU-HTK broadcast news transcription system. *IEEE Trans. on Speech and Audio Processing*, 14:1513–1525, 2006. 2.3.6, 2.6.4

[55] M. J. F. Gales, F. Diehl, C. K. Raut, M. Tomalin, P. C. Woodland, and K. Yu. Development of a phonetic system for large vocabulary arabic speech recognition. In *Proc. ASRU*, 2007. 2.6.4

[56] Y. Gao, B. Ramabhadran, and M. Picheny. New adaptation techniques for large vocabulary continuous speech recognition. In *Proc. ISCA ITRW*, 2000. 4.1.1

[57] J. L. Gauvain and C. H. Lee. Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains. *IEEE Transactions on Speech and Audio Processing*, 2:291–298, 1994. 1, 2.3.3, 2.3.3, 2.3.3, 2.6.3.3, 3.1.1, 3.1.1

[58] Z. Ghahramani and M. J. Beal. Propagation algorithms for variational Bayesian learning. In *NIPS 13*, pages 507–513, 2001. 2.3.3

[59] L. Gillick and S. J. Cox. Some statistical issues in the comparison of speech recognition. In *Proc. ICASSP*, pages 532–535, 1989. 2.7, 7.1

[60] V. Goel and W. Byrne. Minimum Bayes-risk automatic speech recognition. *Computer Speech and Language*, pages 115–135, 2000. 2.6.2

[61] I. J. Good. The population frequency of species and the estimation of population parameters. *Biometrika*, 40:237–264, 1953. 2.5

[62] P. S. Gopalakrishnan, D. Kanevsky, A. Nadas, and D. Nahamoo. A generalization of the Baum algorithm to rational objective functions. In *Proc. ICASSP*, 1989. 2.3.4.2, 6.3.1

[63] A. Gunawardana and W. Byrne. Discriminative speaker adaptation with conditional maximum likelihood linear regression. In *Proc. Eurospeech*, 2001. 1, 4, 4.1, 4.1.1

[64] T. Hain. *Hidden Model Sequence Models for Automatic Speech Recognition*. PhD thesis, Cambridge University, 2001. 2.3.5

[65] T. Hain, P. C. Woodland, G. Evermann, M. J. F. Gales, X. Liu, G. L. Moore, D. Povey, and L. Wang. Automatic transcription of conversational telephone speech. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 2005. 2.3.5

[66] H. Hermansky. Perceptual linear prediction of speech. *Journal of the acoustic society of America*, 87(4):1738–1752, 1990. 2.2.2, 2.2.2

[67] T. Heskes and O. Zoeter. Expectation propagation for approximate inference in dynamic Bayesian networks. In *Proc. Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 216–223, 2002. 5.2.3.2, 5.3, 5.3, A

[68] T. Heskes, M. Opper, W. Wiegerinck, O. Winther, and O. Zoeter. Approximate inference techniques with expectation constraints. *Journal of Statistical Mechanics: Theory and Experiment*, 2005. A

[69] A. J. Hewett. *Training and Speaker Adaptation in Template-Based Speech Recognition*. PhD thesis, Cambridge University, 1989. 3.1.2

[70] S. Homma, K. Aikawa, and S. Sagayama. Improved estimation of supervision in unsupervised speaker adaptation. In *Proc. ICASSP*, volume 2, pages 1023–1026, 1997. 3.1.5, 3.1.5.1

[71] H. W. Hon and K. F. Lee. Recent progress in robust vocabulary independent speech recognition. In *Proc. DARPA Speech and Natural Language Workshop*, pages 258–263, 1991. 2.3.5

[72] X. Huang and K. F. Lee. On speaker-independent, speaker-dependent and speaker-adaptive speech recognition. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 1(2):150–157, 1993. 1

[73] X. Huang, A. Acero, and H. Hon. *Spoken Language Processing*. Prentice Hall PTR, 2001. 1, 2.2.2, 2.2.2, 2.2.2, 2.3.2, 2.4, 2.5, 2.5, 2.6.1, 2.6.1, 3.1

[74] X. D. Huang. A study on speaker-adaptive speech recognition. In *Proc. Human Language Technology Workshop*, page 1991, 278 – 283. 3.1

[75] Q. Huo and C. H. Lee. On-line adaptive learning of the correlated continuous density hidden Markov models for speech recognition. In *Proc. ICSLP*, pages 985–988, 1996. 2.6.3.3

[76] Q. Huo and C. H. Lee. A Bayesian predictive classification approach to robust speech recognition. *IEEE transactions on speech and audio processing*, 8:200–204, 2000. 2.6.3.1, 2.6.3.1, 2.6.3.3

[77] Q. Huo and C. H. Lee. On-line adaptive learning of the continuous density hidden Markov model based on approximate recursive Bayes estimate. *IEEE transactions on speech and audio processing*, 5:161–172, 1997. 1, 2.3.3, 2.6.3.3

[78] Q. Huo, H. Jiang, and C. H. Lee. A Bayesian predictive classification approach to robust speech recognition. In *Proc. ICASSP*, volume 2, pages 1547–1550, 1997. 2.6.3, 2.6.3, 2.6.3.3

[79] T. S. Jaakkola and M. I. Jordan. Bayesian parameter estimation via variational methods. *Statistics and Computing*, 10(1):25–37, 2000. 2.3.3

[80] A. K. Jain. *Fundamentals of Digital Image Processing*. Prentice Hall, 1989. 1

[81] T. Jebara. *Discriminative, Generative and Imitative Learning*. PhD thesis, Massachusetts Institute of Technology, 2002. 6.3.1.2, 9.3.1, B.1, B.1, B.2

[82] T. Jebara and A. Pentland. On reversing Jensen's inequality. *Neural Information Processing Systems (NIPS)*, pages 231–237, 2000. 6.3.1.2, B.1

[83] T. Jebara and A. Pentland. Maximum conditional likelihood via bound maximization and the CEM algorithm, 1998. B.2

[84] F. Jelinek. A fast sequential decoding algorithm using a stack. *IBM Journal on Research and Development*, 13, 1969. 2.6.1

[85] F. Jelinek and R. L. Mercer. Interpolated estimation of Markov source parameters from sparse data. In *Proc. Workshop on Pattern Recognition in Practice*, 1980. 2.5

[86] J. L. W. V. Jensen. Sur les fonctions convexes et les inégalités entre les valeurs moyennes. *Acta Math*, 30:175–193, 1906. 2.3.2

[87] H. Jiang, K. Hirose, and Q. Huo. Robust speech recognition based on Viterbi Bayesian predictive classification. In *Proc. ICASSP*, 1997. 2.6.3.2, 2.6.3.2

[88] H. Jiang, K. Hirose, and Q. Huo. Robust speech recognition based on a Bayesian prediction approach. *IEEE transactions on speech and audio processing*, 7:426–440, 1999. 2.6.3.2, 2.6.3.2, 5.2.2

[89] M. I. Jordan. An introduction to variational methods for graphical models. In *Machine Learning*, pages 183–233. MIT Press, 1999. 2.3.3

[90] B. H. Juang, W. Chou, and C. H. Lee. Minimum classification error rate methods for speech recognition. *IEEE Transactions on Speech and Audio Processing*, 5:266–277, 1997. 2.3.4.1

[91] J. Kaiser, B. Horvat, and Z. Kacic. A novel loss function for the overall risk criterion based discriminative training of HMM models. In *Proc. ICSLP*, 2000. 2.3.4.1

[92] S. Kapadia. *Discriminative Training of Hidden Markov Models*. PhD thesis, Cambridge University, 1998. 2.3.4, 4

[93] S. M. Katz. Estimation of probabilities from spare data for the language model component of a speech recogniser. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 35(3):400–401, 1987. 2.5

[94] P. Kenny, M. Lenning, and P. Mermelstein. Speaker adaptation in a large vocabulary Gaussian HMM recogniser. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12:917–920, 1990. 3.1.2

[95] T. Kosaka and S. Sagayama. Tree structured speaker clustering for fast speaker adaptation. In *Proc. ICASSP*, volume 1, pages 245–248, 1994. 3.1.3

[96] F. Kubala and R. Schwartz. A new paradigm for speaker-independent training and speaker adaptation. In *Proc. Human Language Technology Workshop*, pages 306–310, 1990. 3.1

[97] R. Kuhn, P. Nguyen, J. C. Junqua, L. Goldwasser, N. Niedzielski, S. Fincke, K. Field, and M. Contolini. Eigenvoices for speaker adaptation. In *Proc. ICSLP*, pages 1771–1774, 1998. 3.1.3, 3.2.2

[98] N. Kumar. *Investigation of Silicon-Auditory Models and Generalization of Linear Discriminant Analysis for Improved Speech Recognition*. PhD thesis, John Hopkins University, 1997. 2.3.6, 2.3.6

[99] M. Layton. *Augmented Statistical Models for Classifying Sequence Data*. PhD thesis, Cambridge University, 2006. C

[100] K.-F. Lee, S. Hayamizu, H.-W. Hon, C. Huang, J. Schwartz, and R. Weide. Allophone clustering for continuous speech recognition. In *Proc. ICASSP*, pages 749–752, 1990. 2.3.5

[101] L. Lee and R. C. Rose. Speaker normalization using efficient frequency warping procedures. *Proc. ICASSP*, 1:353–356, 1996. 2.2.3.3, 2.2.3.3, 3.2

[102] C. J. Leggetter. *Improved acoustic modelling for HMMs using linear transformations*. PhD thesis, Cambridge University, 1995. 3.1.4

[103] C. J. Leggetter and P. C. Woodland. Maximum likelihood linear regression for speaker adaptation of continuous density HMMs. *Computer Speech and Language*, 9:171–186, 1995. 3.1.2, 3.1.2.1, 3.1.2.1, 3.2.1, 4.3, 4.3, 1

[104] C. J. Leggetter and P. C. Woodland. Flexible speaker adaptation using maximum likelihood linear regression. In *Proc. ARPA Spoken Language Technology Workshop*, pages 104–109, 1995. 3.1.2

[105] X. Liu, M. J. F. Gales, K. C. Sim, and K. Yu. Investigation of acoustic modelling techniques for LVCSR systems. In *Proc. ICASSP*, pages 849–852, 2005. 2.2.3.2, 3.2

[106] A. Ljolje. The AT&T LVCSR-2001 system. In *Proc. the NIST LVCSR Workshop*, 2001. 4, 4.2, 4.2.1, 4.2.1

[107] D. J. C MacKay. Ensemble learning for hidden Markov models, 1997. URL http://www.inference.phy.cam.ac.uk/mackay. Unpublished Report, Department of Physics, University of Cambridge. 2.3.3

[108] D. J. C. MacKay. *Information Theory, Inference and Learning Algorithms*. Cambridge University Press, 2007. 2.6.3.1

[109] D. J. C. MacKay. Introduction to Monte Carlo methods. In M. I. Jordan, editor, *Learning in Graphical Models*, NATO Science Series, pages 175–204. Kluwer Academic Press, 1998. 2.6.3.1

[110] L. Mangu, E. Brill, and A. Stolcke. Finding consensus among words: Lattice-based word error minimization. In *Proc. Eurospeech*, 1999. 2.6.2, 2.6.2

[111] L. Mangu, E. Brill, and A. Stolcke. Finding consensus in speech recognition: Word error minimization and other applications of confusion network. *Computer Speech and Language*, 14(3):373–400, 2000. 2.6.2, 6.2.3.2

[112] S. Martin, J. Liermann, and H. Ney. Algorithms for bigram and trigram word clustering. In *Proc. Eurospeech*, volume 2, pages 1253–1256, 1995. 2.5

[113] S. Matsoukas, R. Schwartz, H. Jin, and L. Nguyen. Practical implementations of speaker adaptive training. In *Proc. DARPA Speech Recognition Workshop*, 1997. 3.2.1.1

[114] T. Matsui and S. Furui. N-best-based unsupervised speaker adaptation for speech recognition. *Computer Speech and Language*, 12:41–50, 1998. 1

[115] T. Matsui and S. Furui. N-best-based instantaneous speaker adaptation method for speech recognition. In *Proc. ICSLP*, pages 973–976, 1996. 3.1.5.2

[116] T. Matsui, T. Matsuoka, and S. Furui. Smoothed N-best-based speaker adaptation for speech recognition. In *Proc. ICASSP*, volume 2, pages 1015 – 1018, 1997. 3.1.5.2, 3.1.5.2

[117] J. McDonough, W. Byrne, and X. Luo. Speaker compensation with all-pass transforms. In *Proc. ICSLP*, pages 869–872, 1998. 2.2.3.3

[118] J. McDonough, T. Schaaf, and A. Waibel. On maximum mutual information speaker-adapted training. In *Proc. ICASSP*, 2002. 1, 4, 4.1, 4.1.1, 4.2, 4.2.2

[119] T. Minka. *A Family of Algorithms for Approximate Bayesian Inference*. PhD thesis, MIT Media Lab, 2001. 2.6.3.4, 5.2.3.2, 5.3, 5.3, A

[120] T. Minka. Using lower bounds to approximate integrals. Technical report, MIT, 2001. URL http://research.microsoft.com/en-us/um/people/minka/papers/rem.html. 2.6.3.4

[121] L. Moisa and E. Giachin. Automatic clustering of words for probabilistic language models. In *Proc. Eurospeech*, pages 1249–1252, 1995. 2.5

[122] K. P. Murphy. *Dynamic Bayesian Networks: Representation, Inference and Learning*. PhD thesis, University of California Berkeley, 2002. 5.3

[123] K. Na, B. Jeon, D. Chang, S. Chae, and S. Ann. Discriminative training of hidden Markov models using overall risk criterion and reduced gradient method. In *Proc. Eurospeech*, 1995. 2.3.4.1

[124] A. Nadas. A decision theoretic formulations of a training problem in speech recognition and a comparison of training by unconditional versus conditional maximum likelihood. *IEEE Trans. on Acoustics, Speech, and Signal Processing*, 31:814–817, 1983. 2.3.2, 2.3.4.1

[125] A. Nadas. Optimal solution of a training problem in speech recognition. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 33(1):326–329, 1985. 2.3.4.1, 2.6.3

[126] R. M. Neal. Probabilistic inference using Markov chain Monte Carlo methods. Technical Report CRG-TR-93-1, Dept. of Computer Science, University of Toronto, 1993. 2.6.3.1

[127] H. Ney, U. Essen, and R. Kneser. On structuring the probabilistic dependencies in language modelling. *Computer Speech and Language*, 8:1–38, 1994. 2.5

[128] H. Ney, U. Essen, and R. Kneser. On the estimation of small probabilities by leaving one-out. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(12): 1202–1212, 1995. 2.5

[129] P. Nguyen, P. Gelin, J.-C. Junqua, and J.-T. Chien. N-best based supervised and unsupervised adaptation for native and non-native speakers in cars. In *Proc. ICASSP*, pages 173–176, 1999. 1, 3.1.5.2

[130] Y. Normandin. *Hidden Markov Models, Maximum Mutual Information Estimation, and the Speech Recognition Problem*. PhD thesis, McGill University, 1991. 2.3.4, 2.3.4.2, 4

[131] J. Odell. The use of decision trees with context sensitive phoneme modelling. Master's thesis, Cambridge University, 1992. 2.3.5

[132] J. Odell. *The Use of Context in Large Vocabulary Speech Recognition*. PhD thesis, Cambridge University, 1995. 2.3.5, 2.3.5

[133] J. Odell, V. Valtchev, P. C. Woodland, and S. Young. A one-pass decoder design for large vocabulary recognition. In *Proc. Human Language Technology Workshop*, pages 405–510, 1994. 2.6.1

[134] P. Olsen and R. Gopinath. Modelling inverse covariance matrices by basis expansion. In *Proc. ICSLP*, 2002. 2.3.5

[135] M. Opper and O. Winther. Expectation consistent approximate inference. *Journal of Machine Learning Research*, 6:2177–2204, 2005. A

[136] S. Ortmanns, H. Ney, and X. Aubert. A word graph algorithm for large vocabulary continuous speech recognition. *Computer Speech and Language*, 11(1):43–72, 1997. 2.6.1

[137] M. Padmanabhan, G. Saon, and G. Zweig. Lattice-based unsupervised MLLR for speaker adaptation. *Proc. ISCA ITRW ASR2000*, pages 128–131, 2000. 3.1.5.3, 4.1.1

[138] D. B. Paul. Algorithms for an optimal A* search and linearizing the search in the stack decoder. In *Proc. ICASSP*, 1991. 2.6.1

[139] M. Pitz, S. Molau, R. Schlüter, and H. Ney. Vocal tract normalization equals linear transformation in cepstral space. In *Proc. Eurospeech*, 2001. 2.2.3.3

[140] D. Povey. *Discriminative Training for Large Vocabulary Speech Recognition*. PhD thesis, Cambridge University, 2003. 1, 2.3.4, 2.3.4.1, 2.3.4.2, 2.3.4.2, 2.3.4.2, 2.3.4.2, 2.3.4.2, 2.3.4.2, 2.3.4.2, 4, 4.1, 4.1.1, 6.3.1

[141] D. Povey and P. C. Woodland. Improved discriminative training techniques for large vocabulary continuous speech recognition. In *Proc. ICASSP*, 2001. 2.3.4.1

[142] D. Povey and P. C. Woodland. Minimum phone error and I-smoothing for improved discriminative training. In *Proc. ICASSP*, 2002. 2.3.4.1, 2.3.4.2, 2.3.4.2, 2.3.4.2, 4.1.1, 4.2.1

[143] D. Povey, P. C. Woodland, and M. Gales. Discriminative MAP for acoustic model adaptation. In *Proc. ICASSP*, pages 312–315, 2003. 2.3.4.2

[144] D. Pye and P. C. Woodland. Experiments in speaker normalisation and adaptation for large vocabulary speech recognition. *Proc. ICASSP*, pages 1047–1050, 1997. 2.2.3.3

[145] L. Rabiner and B. H. Juang. *Fundamentals of Speech Recognition*. Prentice Hall PTR, 1993. 1, 2.3, 2.3, 2.3, 2.6.1

[146] C. K. Raut and M.J.F. Gales. Bayesian discriminative adaptation for speech recognition. In *Proc. ICASSP*, 2009. (document)

[147] C. K. Raut, K. Yu, and M.J.F. Gales. Adaptive training using discriminative mapping transforms. In *Proc. Interspeech*, 2008. (document), 4.1.1, 4.2.2, 6

[148] F. Richardson, M. Ostendorf, and J. R. Rohlicek. Lattice-based search strategies for large vocabulary recognition. In *Proc. ICASSP*, pages 576–579, 1995. 2.6.1

[149] H. Robbins. The empirical Bayes approach to statistical decision problems. *Ann. Math. Stastist.*, 35:1–20, 1964. 2.3.3

[150] H. Robbins. An empirical Bayes approach to statistics. In *Proc. Third Berkeley Symposium on Math. Statist. and Prob.*, pages 157–164, 1955. 2.3.3

[151] C. P. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer-Verlag, 1999. 2.6.3.1

[152] S. Sagayama, K. Shinoda, M. Nakai, and H. Shimodaira. Analytic methods for acoustic model adaptation: A review. In *Proc. ITRW on Adaptation Methods for Speech Recognition*, pages 67–76, 2001. 1

[153] G. Saon, S. Dharanipragada, and D. Povey. Feature space Gaussianization. In *Proc. ICASSP*, 2004. 2.2.3.2, 2.2.3.2, 3.2

[154] R. Schlüter. *Investigations on Discriminative Training Criteria*. PhD thesis, RWTH Aachen, Germany, 2000. 2.3.4, 4

[155] R. Schlüter and W. Macherey. Comparison of discriminative training criteria. In *Proc. ICASSP*, 1998. 2.3.4.1, 2.3.4.1, 2.3.4.2

[156] R. Schwartz and S. Austin. A comparison of several approximate algorithm for finding multiple (N-best) sentence hypotheses. In *Proc. ICASSP*, pages 701–704, 1991. 2.6.1

[157] R. Schwartz and Y. L. Chow. The N-Best algorithm: an efficient and exact procedure for finding the N most likely sentence hypotheses. In *Proc. ICASSP*, pages 81–84, 1990. 2.6.1

[158] R. Schwartz, Y. Chow, O. Kimball, S. Roucous, M. Krasner, and J. Makhoul. Context-dependent modelling for acoustic-phonetic recognition of continuous speech. In *Proc. ICASSP*, pages 1205–1208, 1985. 2.3.5

[159] M. Shannon. Sampling methods for instantaneous speaker adaptation. Master's thesis, Cambridge University, 2008. 5.2.1

[160] K. Shinoda and C. H. Lee. Structural MAP speaker adaptation using hierarchical priors. *Proc. ASRU*, pages 381–388, 1997. 3.1.1

[161] K. Shinoda and T. Watanabe. Speaker adaptation with autonomous control using tree structure. In *Proc. Eurospeech*, 1995. 3.1.4

[162] K. C. Sim. *Structured Precision Matrix Modelling for Speech Recognition*. PhD thesis, Cambridge University, 2006. 2.3.5

[163] R. Sinha, S. E. Tranter, M. J. F. Gales, and P. C. Woodland. The Cambridge University March 2005 speaker diarisation system. In *Proc. Interspeech*, 2005. 2.2.1

[164] O. Siohan, C. Chesta, and C. H. Lee. Joint maximum a posteriori adaptation of transformation and HMM parameters. *IEEE Transactions on Speech and Audio Processing*, 9(4):417–428, 2001. 3.1.2.4

[165] M. H. Siu, H. Gish, and F. Richardson. Improved estimation, evaluation and applications of confidence measures for speech recognition. In *Proc. Eurospeech*, volume 2, pages 831–834, 1997. 3.1.5.1

[166] A. Stolcke. Entropy-based pruning of backoff language models. In *Proc. DARPA Broadcast News Transcription and Understanding Workshop*, 1998. 2.5

[167] A. Stolcke, Y. Konig, and M. Weintraub. Explicit word error minimization in N-best list rescoring. In *Proc. Eurospeech*, 1997. 2.6.2, 2.6.2, 6.2.3.2

[168] A. C. Surendran and Chin-Hui Lee. Transformation based Bayesian prediction for adaptation of HMMs. *Speech Communication*, 34:159–174, 2001. 5.2.2

[169] L. Tierney and J. B. Kadane. Accurate approximations for posterior moments and marginal densities. *J. Ameri. Statist. Assoc.*, 81(393):82–86, 1986. 2.6.3.3

[170] S. Tsakalidis and S. Matsoukas. Bayesian adaptation in HMM training and decoding using a mixture of feature transforms. In *Proc. ASRU*, pages 329–334, 2007. 5, 5.2.1

[171] S. Tsakalidis, V. Doumpiotis, and W. Byrne. Discriminative linear transforms for feature normalisation and speaker adaptation in HMM estimation. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 13(3):367–376, 2005. 1, 4, 4.1, 4.1.1, 4.2, 4.2.2

[172] L. F. Uebel. *Speaker Normalisation and Adaptation in Large Vocabulary Speech Recognition.* PhD thesis, Cambridge University, 2002. 3.1.5.1

[173] L. F. Uebel and P. C. Woodland. An investigation into vocal tract length normalisation. *Proc. Eurospeech*, pages 911–914, 1999. 2.2.3.3

[174] L. F. Uebel and P. C. Woodland. Speaker adaptation using lattice-based MLLR. *Proc. ISCA ITR-Workshop on Adaptation Methods for Speech Recognition*, 2001. 3.1.5.1, 3.1.5.3, 4.1.1

[175] L. F. Uebel and P. C. Woodland. Discriminative linear transforms for speaker adaptation. *Proc. ISCA ITR-Workshop on Adaptation Methods for Speech Recognition*, 2001. 1, 4, 4.1, 4.1.1

[176] V. Valtchev, J. J. Odell, P. C. Woodland, and S. J. Young. MMIE training of large vocabulary recognition systems. *Speech Communication*, 22:303–314, 1997. 2.3.4.1

[177] V. Vanhoucke and A. Sankar. Mixtures of inverse covariances. In *Proc. ICASSP*, 2003. 2.3.5

[178] A. J. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13:260–269, 1967. 2.6.1

[179] F. Wallhoff, D. Willett, and G. Rigoll. Frame-discriminative and confidence-driven adaptation for LVCSR. In *Proc. ICASSP*, volume 3, pages 1835–1838, 2000. 4.1.1

[180] L. Wang. *Discriminative Linear Transforms for Adaptation and Adaptive Training.* PhD thesis, Cambridge University, 2006. 4, 4.1.1, 4.1.1, 4.1.1, 4.1.1, 4.1.2, 4.2, 4.2.1, 4.2.1, 4.2.1, 4.2.1, 4.2.2, 4.2.2, 6, 6.3.1.1

[181] L. Wang and P. C. Woodland. Discriminative adaptive training using the MPE criterion. In *Proc. ASRU*, 2003. 4.2, 4.2.1, 4.2.1

[182] L. Wang and P. C. Woodland. MPE-based discriminative linear transforms for speaker adaptation. *Computer Speech and Language*, 22(3):256–272, 2008. 1, 4, 4.1, 4.1.1, 4.1.1, 4.1.1

[183] S. Watanabe, Y. Minami, A. Nakamura, and N. Ueda. Application of variational Bayesian approach to speech recognition. In *NIPS 15*, 2003. 2.3.3, 2.3.3, 5.2.2

[184] T. Watanabe and K. Shinoda. Speech recognition using tree-structured probability density function. In *Proc. ICSLP*, 1995. 3.1.4

[185] A. Webb. *Statistical Pattern Recognition*. Oxford University Press, 1999. 2.3.6

[186] F. Wessel, R. Schlüter, and H. Ney. Using posterior word probabilities for improved speech recognition. In *Proc. ICASSP*, 2000. 2.6.2

[187] F. Wessel, R. Schlüter, and H. Ney. Explicit word error minimization using word hypothesis posterior probabilities. In *Proc. ICASSP*, 2001. 2.6.2

[188] I. H. Witten and T. C. Bell. The zero-frequency problem: estimating the probabilities of novel events in adaptive text compression. *IEEE Transactions on Information Theory*, 37(4):1085–1094, 1991. 2.5

[189] P. C. Woodland. Speaker adaptation for continuous density HMMs: A review. *Proc. ISCA ITR-Workshop on Adaptation Methods for Speech Recognition*, 2001. 1, 3, 3.1, 3.1, 3.1.3, 4

[190] P. C. Woodland and D. Povey. Large scale discriminative training of hidden Markov models for speech recognition. *Computer Speech and Language*, 16:25–48, 2002. 2.3.4.2, 2.3.4.2, 2.3.4.2, 3.1.5.3

[191] P. C. Woodland and D. Povey. Large scale discriminative training for speech recognition. In *Proc. ISCA ITR-Workshop on Adaptation Methods for Speech Recognition*, pages 7–16, 2000. 3.1.5.3

[192] P. C. Woodland, J. J. Odell, V. Valtchev, and S. J. Young. Large vocabulary continuous speech recognition using HTK. In *Proc. ICASSP*, 1994. 2.3.5

[193] P. C. Woodland, J. J. Odell, V. Valtchev, and S. J. Young. The development of the 1994 HTK large vocabulary speech recognition system. In *ARPA Workshop on Spoken Language Systems Technology*, pages 104–109, 1995. 3.2

[194] P. C. Woodland, D. Pye, and M. J. F. Gales. Iterative unsupervised adaptation using maximum likelihood linear regression. In *Proc. ICSLP*, pages 1133–1136, 1996. 3.1.5

[195] P. C. Woodland, M. J. F. Gales, D. Pye, and S. J. Young. The development of the 1996 HTK broadcast news transcription system. In *DARPA Speech Recognition Workshop*, pages 73–78, 1997. 2.2.2, 7.1

[196] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. Liu, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. Woodland. *The HTK Book (for HTK version 3.4)*. Cambridge University Engineering Department, 2006. 2.2.2, 2.3.4.1, 2.6.1, 2.7, 3.1.4, 7.1

[197] S. J. Young and P. C. Woodland. The use of state tying in continuous speech recognition. In *Proc. Eurospeech*, pages 2207–2210, 1993. 2.3.5, 2.3.5

[198] S. J. Young, J. Odell, and P. C. Woodland. Tree-based state tying for high accuracy acoustic modelling. In *ARPA Workshop on Human Language Technology*, pages 307–312, 1994. 2.3.5, 2.3.5, 2.3.5

[199] K. Yu. *Adaptive Training for Large Vocabulary Continuous Speech Recognition*. PhD thesis, Cambridge University, 2006. 3.1.5.2, 4.2.1, 5.1, 5.2.2, 5.2.2, 5.2.2, 5.2.3, 5.2.3.2, 5.2.3.2, 5.2.3.2, 6.1, 7.1, 9.1

[200] K. Yu and M. J. F. Gales. Bayesian adaptation and adaptively trained systems. In *Proc. ASRU*, 2005. 5, 5.1, 5.1, 5.1, 5.2

[201] K. Yu and M. J. F. Gales. Bayesian adaptive inference and adaptive training. *IEEE Transactions on Audio, Speech and Language Processing*, 15(6):1932–1943, 2007. 1

[202] K. Yu, M. J. F. Gales, and P. C. Woodland. Unsupervised discriminative adaptation using discriminative mapping transforms. In *Proc. ICASSP*, pages 4273–4276, 2008. 1, 4, 4.1, 4.1.1, 4.1.2, 4.2.2, 6, 7.2.2

[203] K. Yu, M. J. F. Gales, and P. C. Woodland. Unsupervised adaptation with discriminative mapping transforms. *IEEE Transactions on Audio, Speech and Language Processing*, 17(4):714–723, 2009. 4, 4.1, 4.1.1, 4.1.2

[204] G. Zavaliagkos, R. Schwartz, and J. Makhoul. Batch, incremental and instantaneous adaptation techniques for speech recognition. In *Proc. ICASSP*, volume 1, pages 676–679, 1995. 3.1, 3.1

[205] T. Zeppenfeld, M. Finke, K. Ries, M. Westphal, and A. Waibel. Recognition of conversational telephone speech using the JANUS speech engine. In *Proc. Eurospeech*, volume 3, pages 1815–1818, 1997. 3.1.5.1, 3.1.5.1

[206] Z. Zhou, J. Gao, F. K. Soong, and H. Meng. A comparative study of discriminative methods for reranking LVCSR N-best hypotheses in domain adaptation and generalization. In *Proc. ICASSP*, 2006. 6.2.3.2