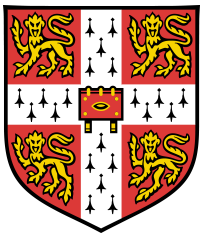
Adaptive Training for Large Vocabulary Continuous Speech Recognition

Kai Yu

Hughes Hall College
and
Cambridge University Engineering Department



July 2006

Dissertation submitted to the University of Cambridge
for the degree of Doctor of Philosophy

Summary

In recent years, there has been a trend towards training large vocabulary continuous speech recognition (LVCSR) systems on a large amount of *found* data. Found data is recorded from spontaneous speech without careful control of the recording acoustic conditions, for example, conversational telephone speech. Hence, it typically has greater variability in terms of speaker and acoustic conditions than specially collected data. Thus, in addition to the desired speech variability required to discriminate between words, it also includes various non-speech variabilities, for example, the change of speakers or acoustic environments. The standard approach to handle this type of data is to train hidden Markov models (HMMs) on the whole data set as if all data comes from a single acoustic condition. This is referred to as *multi-style* training, for example speaker-independent training. Effectively, the non-speech variabilities are ignored. Though good performance has been obtained with multi-style systems, these systems account for all variabilities. Improvement may be obtained if the two types of variabilities in the found data are modelled separately. *Adaptive training* has been proposed for this purpose. In contrast to multi-style training, a set of transforms is used to represent the non-speech variabilities. A canonical model that models the desired speech variability is trained given this set of transforms. In recognition, a test domain specific transform is estimated to adapt the canonical model to decode the test data. Various forms of transform may be used in adaptive training. Linear transforms, for example maximum likelihood linear regression (MLLR) and constrained MLLR (CMLLR), and interpolation weight vectors, for example cluster adaptive training (CAT) and eigenvoices are two widely used forms. Adaptive training and adaptation using the maximum likelihood (ML) criterion have been previously investigated. This thesis will propose two extensions in the area of adaptive training as described below.

Discriminative training is used in most state-of-the-art LVCSR systems. Discriminative criteria are more closely related to the recognition error rate than the ML criterion. They were first proposed for training multi-style systems. Recently, there has been research in applying these criteria to adaptive training. Linear transform based discriminative adaptive training has been previously studied. This work describes a new discriminative training technique for multiple-cluster systems, *discriminative cluster adaptive training*. This technique allows rapid adaptation to be performed on discriminatively trained systems. The minimum phone error (MPE) criterion is used as the specific discriminative criterion. To optimise the MPE criterion, a weak-sense auxiliary function is usually used. Due to the use of a multiple-cluster model in CAT, there are a number of differences in the weak-sense auxiliary function compared to the standard one. In particular, there are more choices of the prior in the I-smoothing distribution, which may significantly affect the recognition performance. In addition to the discriminative update of the multiple-cluster canonical model, MPE training for interpolation weights is also presented. The discriminative CAT technique can be extended to a more complex discriminative adaptive training with structured transforms (ST), in which CMLLR transforms are combined with CAT interpolation weights.

The second contribution of this thesis is to provide a *Bayesian framework* for adaptive training and adaptive inference. This framework addresses a number of limitations of the standard adaptive training. In particular, the issue that the canonical model can not be directly used for recognition due to the unavailability of target domain specific transform is addressed. From a Bayesian perspective, the two sets of parameters, the canonical model and the transform, are regarded as random variables and marginalised out to calculate the likelihood of the data. During training, two prior distributions, one for the canonical model and one for the transform, are trained. In this work, an ML estimate for the canonical model is used. This may be justified within the Bayesian framework by controlling the model complexity to reflect the amount of training data. In contrast, the transform prior distribution is a non-point distribution. Adaptive inference can then be performed directly using the adaptively trained model by integrating out over the transform prior distribution. However, as this marginalisation integral is intractable, approximations are required. Various marginalisation approximations, lower bound based or direct approximations, are discussed, including a new *variational Bayes* (VB) approach. By using an appropriate approximation approach, the issue of handling limited adaptation data is effectively addressed. Both batch and incremental modes of Bayesian adaptive inference are discussed in this work. These approaches are then applied to adaptively trained systems using either interpolation weights (CAT) or mean based linear transforms. The extension of the Bayesian framework to discriminative criteria is also discussed in this work.

The above two contributions were evaluated on a conversational telephone speech (CTS) task. For the experiments with discriminative adaptive systems, the form of discriminative adaptive training adopted in this work is to discriminatively update the canonical model given a set of ML estimated transforms. Experiments to investigate discriminative adaptive training showed that the MPE-CAT system outperformed the MPE gender-dependent system and the MPE-ST system outperformed all the other adaptively trained MPE systems. Experiments concerning Bayesian adaptive training and inference illustrated that adaptively trained systems can obtain significant gains over non-adaptively trained systems. By using a non-point transform distribution, the VB approximation significantly outperformed other approximation approaches with very limited adaptation data. As more data becomes available, the performance of the VB approach and the maximum a posteriori (MAP) approach became closer to each other. Both methods obtained significant gains over the standard ML approaches.

Keywords

Speech recognition, hidden Markov models, HMM, large vocabulary continuous speech recognition, adaptation, adaptive training, discriminative training, MPE, cluster adaptive training, eigenvoices, structured transforms, Bayesian adaptation, Bayesian adaptive training, Bayesian adaptive inference, incremental, maximum a posteriori, variational Bayes.

Declaration

This thesis is the result of my own work carried out at the Machine Intelligence Laboratory, Cambridge University Engineering Department. It includes nothing which is the outcome of any work done in collaboration except explicitly stated. Reference to the work of others is specifically indicated in the text where appropriate. Some of the materials have been presented at international conferences and workshops [136, 138, 141], published as technical reports [137, 139] and article in an academic journal [140]. To my best knowledge, the length of this thesis including footnotes and appendices is approximately 58900 words.

Acknowledgment

First of all, I would like to express utmost gratitude to my supervisor, Dr. Mark Gales, for his guidance, suggestion and criticism throughout my study in Cambridge. His responsibility to students is impressive, which has been invaluable to me. I learned a lot from his strictness in mathematics, strong motivation of concepts and clear logic flow during presentation and writing. The firm requirements and countless guidance on these aspects have given me the ability and confidence to carry out the research work of this thesis as well as works in the future. By initiating well-target questions, offering experienced suggestions and having constructive discussions, he is without doubt the most important person to have helped me make this work possible!

Special thanks go to Prof. Phil. Woodland for providing financial support for my study and attendance of many international conferences and workshops through the DARPA funded EARS and GALE programs. As the principal investigator, Prof. Woodland successfully leads the team and keeps the leading position of Cambridge in the competitive evaluations of these programs. This provided me with an excellent opportunity to carry out high-standard research and interact with other leading speech groups. His insight, experience and wide-range knowledge on speech and language processing have also benefited me a lot. It was a privilege to work with him.

I would also like to thank Prof. Steve Young for leading the Machine Intelligence Laboratory at Cambridge University Engineering Department. The work of this thesis would not have been possible without the wonderful hidden Markov model toolkit (HTK), which was first developed under the leadership of Prof. Young. Gratitude also goes to those who have contributed or are contributing to the development and maintenance of HTK.

I owe my thanks to my colleagues in the Machine Intelligence Lab for the help and encouragements they have given to me. Particular thanks must go to Khe Chai Sim, Xunying Liu, Martin Layton, Sarah Airey, Hank Liao and Catherine Breslin for proofreading various parts of this thesis. There are also many other individuals to acknowledge, but my thanks go to, in no particular order, Gunnar Evermann, Sue Tranter, Thomas Hain, Do Yeong Kim, Bin Jia, Ho Yin Chan, Lan Wang, Rohit Sinha, Marcus Tomalin and David Mrva. I must also thank Anna Langley and Patrick Gosling for their excellent work in maintaining the computing facilities. These intelligent and kind people make the group an interesting place to work in.

I can not imagine a life in Cambridge without the friends from my homeland China. They have shared my excitement, happiness as well as disappointment and sadness, just as I have shared theirs. I will miss the chats, jokes, encouragements, gatherings, sports, cookings, travels, and more importantly, the sincerity, optimism, loyalty, generosity and uprightness of them. The friendship I made in Cambridge will be a treasure forever.

Finally, the biggest thanks go to my parents to whom I always owe everything! For many years, they have offered everything possible to support me, despite my lack of going back home since I entered high-school.

Table of Notations

General Notation:

s	a scalar is denoted by a plain lowercase letter
$\boldsymbol{v}$	a column vector is denoted by a bold lowercase letter
$\boldsymbol{X}$	a matrix is denoted by a bold uppercase letter
$\mathcal{F}(\cdot)$	training criterion
$\mathcal{Q}(\cdot;\cdot)$	auxiliary function of the parameters to estimate given the current estimates

Mathematical Notation:

$p(\cdot)$	probability density function
$p(\cdot\|\cdot)$	conditional probability density
$P(\cdot)$	probability mass distribution
$P(\cdot\|\cdot)$	conditional probability mass distribution
$\{\cdot\}^T$	transpose of a vector or a matrix
$\|\cdot\|$	determinant of a square matrix
$\{\cdot\}^{-1}$	inverse of a square matrix
diag$(\cdot)$	diagonal vector of a square matrix
tr$(\cdot)$	trace of a square matrix
KL$(\cdot\|\|\cdot)$	Kullback Leibler (KL) distance between two distributions
H$(\cdot)$	Entropy of a distribution
$< f(x) >_{g(x)}$	expectation of $f(x)$ with respect to $g(x)$

Standard HMM Notation:

$\mathcal{M}$	parameter set of HMMs
$\mathcal{H}$	word (transcription or hypothesis) sequence
$\mathbf{O}$	observation sequence $\mathbf{O} = [\mathbf{o}_1, \cdots, \mathbf{o}_T]$, $\mathbf{o}_t$ is the observation vector at time t
$\boldsymbol{\omega}$	state sequence, $\boldsymbol{\omega} = [\omega_1, \cdots, \omega_T]$, ω_t is the state at time t
a_{ij}	discrete state transition probability from state i to j
$b_j(\mathbf{o})$	state output distribution given state j
$\boldsymbol{\theta}$	Gaussian component sequence $\boldsymbol{\theta} = [\theta_1, \cdots, \theta_T]$, θ_t is the Gaussian comp. at time t
m	index of a distinct Gaussian component
$\boldsymbol{\mu}^{(m)}$	mean vector of the m^{th} Gaussian component
$\boldsymbol{\Sigma}^{(m)}$	covariance matrix of the m^{th} Gaussian component
$\gamma_m(t)$	posterior probability of θ_t being m given observation and hypothesis sequence

Adaptive Training Notation:

s	index of a homogeneous data block associated with a distinct acoustic condition
$\mathcal{O}$	a set of observation sequences $\mathcal{O} = \{\mathbf{O}^{(1)}, \cdots, \mathbf{O}^{(S)}\}$, $\mathbf{O}^{(s)}$ is the observation sequence for homogeneous data block s
$\mathbf{H}$	a set of hypothesis sequences $\mathbf{H} = \{\mathcal{H}^{(1)}, \cdots, \mathcal{H}^{(S)}\}$, $\mathcal{H}^{(s)}$ is the hypothesis sequence for homogeneous data block s
$\boldsymbol{\mathcal{T}}$	a set of transforms $\boldsymbol{\mathcal{T}} = \{\mathcal{T}^{(1)}, \cdots, \mathcal{T}^{(S)}\}$, $\mathcal{T}^{(s)}$ is the transform for acoustic cond. s
$\hat{\boldsymbol{\mu}}^{(sm)}$	adapted mean vector of Gaussian component m for acoustic condition s
r_m	regression base class that Gaussian component m belongs to
$\mathbf{A}$	square linear transform
$\mathbf{b}$	bias vector
$\mathbf{W}$	extended linear transform, $\mathbf{W} = [\mathbf{A}\ \mathbf{b}]$
$\boldsymbol{\xi}^{(m)}$	extended mean vector of Gaussian component m, $\boldsymbol{\xi}^{(m)} = [\boldsymbol{\mu}^{(m)T}\ \ 1]^T$
$\boldsymbol{\zeta}_t$	extended observation vector at time t, $\boldsymbol{\zeta}_t = [\mathbf{o}_t^T\ \ 1]^T$
$\boldsymbol{\lambda}$	interpolation weight vector
$\mathbf{M}^{(m)}$	cluster means of Gaussian component m, $\mathbf{M}^{(m)} = [\boldsymbol{\mu}_1^{(m)},\ \ \ldots,\ \ \boldsymbol{\mu}_P^{(m)}]$

Discriminative Adaptive Training Notation:

$\boldsymbol{\Gamma}$	sufficient statistics
$\mathcal{G}(\cdot;\boldsymbol{\Gamma})$	auxiliary function expressed in terms of statistics
$\mathcal{S}(\cdot;\cdot)$	smoothing function of the parameters to update given the current estimate
D_m	component-specific smoothing constant to control convergence
$\hat{\boldsymbol{\mu}}_{\mathsf{c}}^{(m)}$	current estimate of mean vector of Gaussian component m
$\hat{\boldsymbol{\Sigma}}_{\mathsf{c}}^{(m)}$	current estimate of covariance matrix of Gaussian component m
τ^I	constant to control impact of the I-smoothing distribution
$\tilde{\boldsymbol{\mu}}^{(sm)}$	prior mean vector of Gaussian component m for acoustic condition s
$\tilde{\boldsymbol{\Sigma}}^{(sm)}$	prior covariance matrix of Gaussian component m for acoustic condition s

Bayesian Adaptive Training Notation:

$\mathcal{L}(\cdot)$	lower bound
$q(\cdot)$	variational distribution
$\bar{p}(\cdot)$	predictive distribution
$\tilde{p}(\cdot)$	pseudo-distribution
$\mathcal{Z}$	normalisation term
u	utterance index
$\mathbf{O}_{1:u}$	observation sequence from the 1^{st} to the u^{th} utterance
$\mathcal{H}_{1:u}$	hypothesis sequence from the 1^{st} to the u^{th} utterance

Acronyms

AF	Acoustic factorisation
ASR	Automatic speech recognition
BN	Broadcast news
CAT	Cluster adaptive training
CDF	Cumulative density function
CDHMM	Continuous density hidden Markov model
CML	Conditional maximum likelihood
CMLLR	Constrained maximum likelihood linear regression
CMN	Cepstral mean normalisation
CTS	Conversational telephone speech
CUHTK	Cambridge University hidden Markov model toolkit
CVN	Cepstral variance normalisation
DAT	Discriminative adaptive training
DBN	Dynamic Bayesian network
DCT	Discrete Cosine transform
DHMM	Discrete hidden Markov model
EBW	Extended Baum-Welch
EM	Expectation maximisation
FFT	Fast Fourier transform
FI	Frame independent
GD	Gender dependent
GI	Gender independent
GMM	Gaussian mixture model
HLDA	Heteroscedastic linear discriminant analysis
HMM	Hidden Markov model
HTK	Hidden Markov model toolkit

LDA	Linear discriminant analysis
LDC	Linguistic Data Corporation
LP	Linear prediction
LVCSR	Large vocabulary continuous speech recognition
MAP	Maximum a Posteriori
MBR	Minimum Bayesian risk
MCE	Minimum classification error
MFCC	Mel-frequency cepstral coefficients
ML	Maximum likelihood
MLLR	Maximum likelihood linear regression
MMI	Maximum mutual information
MPE	Minimum phone error
MWE	Minimum word error
NIST	National institution of standard technology
PCA	Principal component analysis
PLP	Perceptual linear prediction
RHS	Right hand side
SAT	Speaker adaptive training
SD	Speaker dependent
SI	Speaker independent
ST	Structured transforms
VB	Variational Bayes
VBEM	Variational Bayesian expectation maximisation
VTLN	Vocal tract length normalization
WER	Word error rate
WSJ	Wall Street Journal

Contents

List of Figures

List of Tables

1

Introduction

Speech is one of the most important ways for humans to communicate with each other and acquire information. Using machines to extend people's ability to process speech has been a research topic since the invention of the telephone in the late 19th century. Among speech processing problems, *automatic speech recognition* (ASR), the task of converting recorded speech waveforms automatically to text, is one of the most challenging tasks. Research on ASR progressed slowly in the early 20th century. In contrast to the development of the first speech synthesisor in 1936 by AT&T, the first automatic speech recogniser, a simple digit recogniser, appeared in 1952 [17]. In 1969, John Pierce of Bell Labs said that ASR will not be a reality for several decades. However, the 1970s witnessed a significant theoretical breakthrough in speech recognition: hidden Markov models (HMMs) [7, 61]. In the following decades, HMMs were extensively investigated and became the most successful technique for acoustic modelling in speech recognition. The fast development of computer hardware and algorithms of discrete signal processing in the 1960s and 1970s also greatly stimulated interest in building ASR systems.

With the introduction and further development of the theory of HMMs and dramatically increased computing power, significant progress for ASR has been achieved. Continuous speech recognition gradually became the main research interest after simple connected and isolated word recognition was dealt with well. From 1988, two U.S. government institutions, the National Institute of Standards and Technology (NIST) and the Defense Advanced Research Project Agency (DARPA), jointly organized a world-wide evaluation on continuous speech recognition every year. This evaluation proved significant in pushing ahead the research of ASR and setting many milestones in speech recognition[1]. The size of the recognition vocabulary increased from 900 words in the Resource Management task (1988-1992) to 20000 in the Wall Street Journal (WSJ) task (1993-1995, time-unlimited evaluation). A recognition system with a vocabulary size of the order of the WSJ task is often referred to as *large vocabulary continuous speech recognition* (LVCSR) system. However, now the recognition vocabulary size of a state-of-the-art LVCSR system has increased to over 65000 words. Besides the vocabulary size, the difficulty of the

[1] For more detailed history, refer to http://www.nist.gov/speech/history/index.htm.

evaluation task has also been increased in other aspects, which approximates a more realistic and practical recognition problem. For example, the acoustic environment of the evaluation data changed from a clean to a noisy environment. Found speech was used as the primary task instead of dictated speech from 1996 when the broadcast news (BN) task was set up. More natural and spontaneous speech with severe signal degradation, conversational telephone speech (CTS), has been introduced to the evaluation since 1998. Up to now, the state-of-the-art ASR systems are built for spontaneous natural continuous large vocabulary speech. This type of task is the target application domain of the techniques investigated in this thesis.

As the speech recognition tasks become more and more difficult, many challenging technical problems on acoustic modelling emerge. One of the main challenges is the diverse acoustic conditions of the recorded speech data. For example, different speakers exist and speech might be recorded in different acoustic environments or with different channel conditions. Though these acoustic conditions do not reflect what words people speak, the additional non-speech variations introduce more confusion during ASR and may significantly degrade the performance. This problem happened when using an acoustic model to recognise the test data with mismatched acoustic conditions from the training data. *Adaptation* techniques were proposed to solve this problem by normalising features of the test data or tuning the model parameters towards the testing domain. In recent years, building an ASR system on *found* training data has become more and more popular. This type of training data, for example BN or CTS, is normally non-homogeneous. To deal with the acoustic mismatch inside training data and build a compact system on non-homogeneous data, *adaptive training* techniques are widely used. The basic idea is to separately model the speech variabilities and the non-speech variabilities and to use adaptation in both the training and recognition processes. This thesis will investigate adaptive training techniques for LVCSR systems.

1.1 Speech Recognition System

The aim of a speech recognition system is to produce a word sequence (or possibly character sequence for languages like Mandarin) given a speech waveform. The basic structure of an ASR system is shown in figure 1.1.

The first stage of speech recognition is to compress the speech signals into streams of acoustic feature vectors, referred to as *observations*. The extracted observation vectors are assumed to contain sufficient information and be compact enough for efficient recognition. This process is known as the *front-end processing* or *feature extraction*. Given the observation sequence, generally three main sources of information are required to recognise, or infer, the most likely word sequence: the *lexicon*, *language model* and *acoustic model*. The lexicon, sometimes referred to as the *dictionary*, is normally used in LVCSR system to map sub-word units, from which the acoustic models are constructed, to the actual words present in the vocabulary and language model. The language model represents the local syntactic and semantic information of the uttered sentences. It contains information about the possibility of each word sequence. The acoustic model

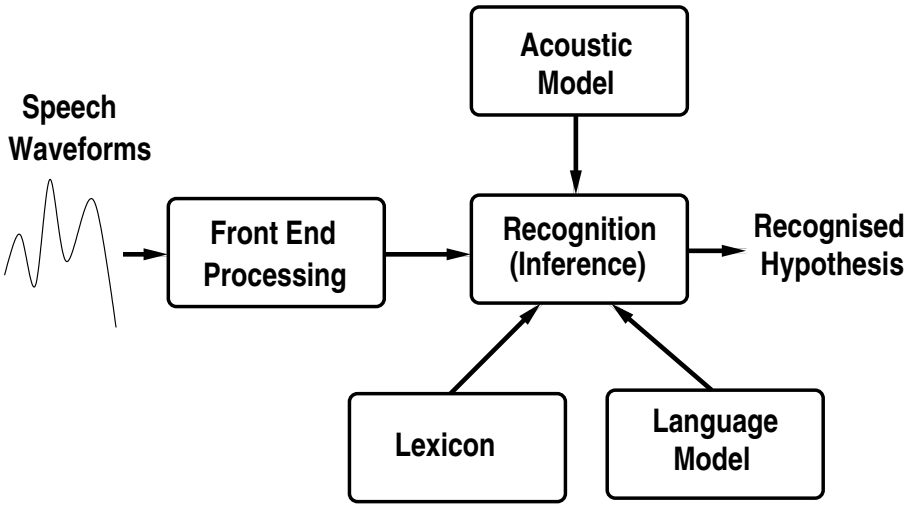


Figure 1.1 *General structure of an automatic speech recognition system*

maps the acoustic observations to the sub-word units. A detailed introduction to various sources in figure 1.1 will be given in chapter 2.

Statistical approaches are the most popular recognition algorithms to hypothesise the word sequence given the above information. Within a statistical framework, the general decision criterion to find the most likely word sequence $\hat{\mathcal{H}}$ for the observation sequence $\mathbf{O} = [\mathbf{o}_1, \cdots, \mathbf{o}_T]$ is the Bayesian decision rule

$$\hat{\mathcal{H}} = \arg\max_{\mathcal{H}} P(\mathcal{H}|\mathbf{O}) \tag{1.1}$$

By applying Bayes rule and considering that the most likely word sequence is independent of the marginal likelihood of the observations $p(\mathbf{O})$, the decision rule can be re-expressed as

$$\begin{aligned} \hat{\mathcal{H}} &= \arg\max_{\mathcal{H}} \left\{ \frac{p(\mathbf{O}|\mathcal{H})P(\mathcal{H})}{p(\mathbf{O})} \right\} \\ &= \arg\max_{\mathcal{H}} p(\mathbf{O}|\mathcal{H})P(\mathcal{H}) \end{aligned} \tag{1.2}$$

where $P(\mathcal{H})$ is the prior probability of a particular word sequence given by a language model. $p(\mathbf{O}|\mathcal{H})$ is calculated using the acoustic model. Hidden Markov models (HMMs) are the most popular and successful acoustic models to date. The adaptive training of systems using HMMs on non-homogeneous data are the focus of this thesis.

1.2 Adaptation and Adaptive Training

HMMs, used as the statistical models for the speech variabilities, have achieved great success in speech recognition. They are trained on observation sequences converted from speech waveforms. Ideally, the extracted features should represent only the *speech variability*, which is inherent to the acoustic realisation of the uttered words, so that HMMs independent of acoustic

conditions may be trained. However, the ideal feature extraction approach does not exist. Variabilities coming from the acoustic conditions during recording, such as speaker, acoustic environment or telephone channels, also affect the acoustic features extracted. As these variabilities do not reflect the inherent variability of the uttered words, they are referred to as *non-speech variabilities*. Therefore, HMMs trained on standard features reflect both speech variabilities and non-speech variabilities of the acoustic conditions. It has been found that improved recognition performance can be obtained on a particular test domain if the model used for recognition is trained on the training data with the same acoustic conditions. Speaker dependent (SD) models are a good illustration. SD models can reduce the average word error rate (WER) by a factor of two or three compared to a speaker independent (SI) system when both systems use the same amount of training data [126].

Unfortunately, in real world situations, it is often impossible to collect sufficient training data to build HMMs system with the same acoustic conditions as the test data. Consequently, the acoustic mismatch between the training and test data may significantly degrade the performance. *Adaptation* techniques have been proposed to solve this problem. The basic idea is to compensate for the mismatch between training and test conditions by modifying the model parameters based on some adaptation data or by enhancing the corrupted features of test data. Adaptation techniques were first used for reducing mismatch between speakers. These techniques have also been applied to deal with other acoustic factors, such as environmental noise.

Though adaptation techniques have achieved great success in robust speaker adaptation, the model that is adapted may limit the possible gains that adaptation can yield. Usually, the model is trained on carefully collected training data, where all data come from the same source. In recent years, to increase the amount of training data, there has been a trend towards using *found* training data to build models. Found data refers to the data recorded from natural speech without careful control of acoustic conditions. Hence, this type of data is normally non-homogeneous in terms of acoustic conditions and typically has more non-speech variabilities than carefully collected data. Though a model can be built on the whole dataset as if all data come from a single consistent source, the resultant model will account for various redundant non-speech variabilities and this may be less amenable to be adapted to a particular test domain. *Adaptive training* is a powerful approach for building a compact model on non-homogeneous training data. The basic concept is to train a set of transformations which represent all non-speech variabilities. Then a canonical model that represents "pure" speech variabilities is trained given the set of transformations. During recognition, adaptation is required to tune the canonical model to a particular test domain.

Depending on the form of transformations used, adaptation and adaptive training techniques can be split into two broad classes. *Feature normalisation* schemes normalise the corrupted features of both training and test dataset. Adaptive training simply requires estimating the canonical model given the normalised features. In recognition, the features of the test data are also normalised. The transforms used in feature normalisation are global for all model parameters. They are normally not dependent on the hypothesis of the test data and just dependent

on the observations, hence are easy to implement. Commonly used normalisation approaches include Cepstral mean normalisation (CMN), Cepstral variance normalisation (CVN) and vocal tract length normalisation (VTLN). *Model based transforms* transform the model parameters, normally the mean vectors and covariance matrices of the Gaussian components. During adaptive training, a canonical model is built given a set of transforms, in which each one represents a particular acoustic condition. This model is then adapted by the test domain specific transforms and used for recognition. Widely used techniques include maximum likelihood linear regression (MLLR), cluster adaptive training (CAT) and constrained MLLR (CMLLR).

Model based transforms are more powerful than feature normalisation because different transforms may be associated with different groups of Gaussians. The hypothesis of the test data is normally used when estimating these transforms. Model based transforms are the focus of this work. Traditionally, the transforms used in adaptive training and adaptation are estimated using the ML criterion. With the ML interpretation, there is a limitation of the adaptive training and adaptation framework that the canonical model can not be directly used in unsupervised adaptation. As the canonical model is the only output of the training stage and represents only speech variability, test domain specific transforms are required for recognition. The standard approach to handle this is to use another model, such as speaker-independent model, to generate initial hypothesis and then estimate the transforms given this hypothesis. Furthermore, ML estimate of transforms is not robust for limited adaptation data and may significantly degrade recognition performance. This work proposes a consistent Bayesian framework for adaptive training and adaptive inference to solve these problems [139]. Within this framework, a transform prior distribution is produced during adaptive training. This framework also allows the canonical model to be directly used in unsupervised adaptation by marginalising over the transform prior distribution. As the marginalisation integral in Bayesian adaptive inference is not tractable, approximations are required. Various Bayesian approximation approaches, including a new variational Bayes (VB) approximation, are discussed in detail. The robustness issue is effectively addressed by using full Bayesian approaches as shown in the experiments.

In most state-of-the-art systems, *discriminative training* is used to obtain the best performance [6, 64, 93]. It takes into account the competing incorrect hypothesis during training and aims at directly reducing the recognition error. Discriminative training has been investigated within the linear transform based adaptive training framework [110, 60, 119]. An alternative to linear transform based adaptive training is cluster adaptive training (CAT), in which the "transforms" used are the interpolation weight vectors. This work will propose a new discriminative adaptive training scheme based on CAT [140]. Due to the small number of parameters of interpolation weights, this discriminative CAT approach can be effectively used for rapid adaptation.

1.3 Thesis Structure

This thesis is structured as follows.

Chapter 2 reviews the fundamental issues of speech recognition including feature extraction,

HMMs training and recognition with HMMs. The standard maximum likelihood (ML) training of HMMs is discussed after an introduction to the concept of HMMs. Discriminative training and Bayesian training of HMMs are also reviewed. Finally, recognition issues, including language modelling and search algorithm are discussed.

Chapter 3 describes the standard adaptation and adaptive training techniques. The homogeneity assumption and the operating modes are described first. Widely used adaptation schemes, including maximum a posteriori (MAP) and maximum likelihood linear regression (MLLR) are then reviewed. Regression classes and extensions of standard schemes are discussed in the same section. The next section describes the framework of adaptive training and reviews feature normalisation schemes. As the most powerful adaptive training techniques, model based schemes are discussed in detail in section 3.3.2.2. These schemes include speaker adaptive training (SAT) with MLLR or constrained MLLR and cluster adaptive training (CAT). More complex adaptive training techniques, in which structured transforms are used to model multiple acoustic factors, are presented at the end of this chapter.

Chapter 4 addresses discriminative adaptive training techniques. The MLLR and constrained MLLR based discriminative adaptive training are reviewed first. An introduction to a simplified discriminative adaptive training procedure is included in the review. This chapter presents a new discriminative cluster adaptive training technique. Update formulae for both multiple-cluster canonical model and interpolation weights are given. Practical issues such as the selection of I-smoothing prior and smoothing constant are also discussed in detail. Then, discriminative adaptive training with structured transforms is discussed.

Chapter 5 describes a consistent Bayesian framework for adaptive training and adaptive inference. The framework is first discussed with likelihood criterion. Then it is extended to discriminative criteria. The next section introduces the approximation schemes that may be used in Bayesian adaptive inference. They include a sampling approach, frame-independent assumption, ML or MAP estimates and a new variational Bayesian approximation. The investigation of an incremental mode of Bayesian adaptive inference then follows. The proposed Bayesian adaptive inference techniques are applied to CAT and SAT with MLLR, which is detailed in section 5.5.

Chapter 6 and 7 present experimental results for discriminative adaptive training and Bayesian adaptive inference respectively. The development of discriminative cluster adaptive training and comparisons between various discriminative adaptive training techniques are given in chapter 6. Results of Bayesian adaptive inference on very limited adaptation data and incremental Bayesian adaptive inference are included in chapter 7. The conclusions and suggestions for the direction of future work of are summarised in the final chapter.

2

Acoustic Modelling in Speech Recognition

This chapter gives an introduction to speech recognition systems using hidden Markov models (HMMs) as the acoustic model. Various aspects of the system depicted in figure 1.1 are described in this chapter, including front-end processing or parameterisation of speech signals, the use of HMMs for acoustic modelling, the selection of recognition units, the search algorithms and the language model used in recognition. In particular, the training algorithms of HMMs are discussed in detail. Besides standard maximum likelihood (ML) training, discriminative training and Bayesian training of HMMs are also reviewed.

2.1 Front-end Processing of Speech Signals

The raw form of speech recorded is a continuous speech waveform. To effectively perform speech recognition, the speech waveform is normally converted into a sequence of time-discrete parametric vectors. These parametric vectors are assumed to give exact and compact representation of speech variabilities. These parametric vectors are often referred to as *feature* vectors or *observations*.

There are two widely used feature extraction schemes: *Mel-frequency Cepstral coefficients* (MFCC) [18] and *perceptual linear prediction* (PLP) [53]. Both schemes are based on Cepstral analysis. The initial frequency analysis of the two schemes are the same. First, the speech signal is split into discrete segments usually with 10ms shifting rate and 25ms window length. This reflects the short-term stationary property of speech signals [95]. These discrete segments are often referred to as *frames*. A feature vector will be extracted for each frame. A pre-emphasising technique is normally used during the feature extraction, where overlapping window functions, such as Hamming or Hanning windows are used to smooth the signals. Using a window function reduces the boundary effect in signal processing. A fast Fourier transform (FFT) is then performed on the time-domain speech signals of each frame, generating the complex frequency domains. Having obtained the frequency domain for each frame, different procedures are used to obtain MFCC or PLP features. The difference lies in the frequency warping methods and the Cepstral representation. Details are given below.

- **Mel-frequency Cepstral coefficients (MFCC)** [18]

  1. *Mel-frequency warping*

     The frequency is warped using the Mel-frequency scale, which means the frequency axis is scaled. As the magnitude of each FFT complex value will be used to extract MFCC features, this process results in a scaled magnitude-frequency domain.

  2. *Down-sampling using triangular filter bank*

     A bank of triangular filters is used to down-sample the scaled magnitude-frequency domain[1]. During this process, the magnitude coefficients are multiplied by corresponding filter gains and the results are accumulated as the amplitude value. Hence, each filter is associated with one amplitude value. Logarithm of the amplitude values are then calculated.

  3. *Discrete Cosine transform (DCT) to get Cepstral coefficients:*

     A DCT is then performed on the log-amplitude domain. This aims to reduce the spatial correlation between filter bank amplitudes. The resultant DCT coefficients are referred to as *Cepstral coefficients*, also *MFCC* coefficients. For the systems in this work, 12 coefficients plus a normalised log energy, or the zeroth order Cepstral coefficient are used. These coefficients form a 13-dimensional acoustic feature vector for each frame.

- **Perceptual linear prediction (PLP)** [53]

  1. *Bark-frequency warping*

     The frequency is warped using the Bark-frequency scale. As the power spectrum value, the square of the magnitude, will be used to extract PLP features, this process results in a scaled power spectrum.

  2. *Down-sampling and post-processing*

     The power spectrum is convolved with a number of critical band filters to get down-sampled values. These values are then scaled by using the curve of equal-loudness and intensity-loudness power law. The resultant down-sampled and post-processed spectrum is output to the next step.

     The above is the standard PLP feature extraction scheme. The PLP features can also be extracted based on Mel-frequency filter bank, referred to as *MF-PLP* [127]. The Mel filter bank coefficients are weighted by the equal-loudness curve and then compressed by taking the cubic root [133]. The resultant spectrum is output to the next step. This is the type of PLP features used in this work.

  3. *Linear prediction (LP) analysis*

[1] In practice, the warping is normally done by changing the center frequency and bandwidth of the triangular filters.

The above spectrum is converted to an auto-correlation sequence in the time domain. The LP coefficients are then calculated on that sequence. Normally, 12 coefficients are computed.

4. *Cepstral coefficients calculation*

   Given the LP coefficients, the Cepstral coefficients, i.e. the inverse FFT of the log magnitude of the spectrum of the LP coefficients, are calculated. These coefficients are referred to as *PLP* coefficients. Similar to MFCC, 12 PLP coefficients plus the normalised log energy, referred to as `C0`, are normally used forming a 13-dimensional vector.

When using hidden Markov models (HMMs) as the acoustic model, which will be introduced in the next section, there is a fundamental assumption that the observations are conditionally independent. This requires removal of the temporal correlation of speech signals. Dynamic coefficients may be incorporated into the feature vector to reduce the temporal correlation [28]. These dynamic coefficients represent the correlation between static feature vectors of different time instances. One common form is the *delta coefficient* , $\Delta \mathbf{o}_t$, which is calculated as a linear regression over a number of frames

$$\Delta \mathbf{o}_t = \frac{\sum_{k=1}^{K} k(\mathbf{o}_{t+k} - \mathbf{o}_{t-k})}{2\sum_{k=1}^{K} k^2} \tag{2.1}$$

where k is the regression parameter, K is the width over which the dynamic coefficients are calculated and the actual window size is $2K+1$ accordingly. The second-order dynamic coefficients, *delta-delta coefficient*, $\Delta^2 \mathbf{o}_t$, may also be calculated using a version of equation (2.1) in which the static parameters are replaced by the first-order delta coefficients. The first and second order dynamic coefficients are then appended to the standard static features, constructing a 39-dimensional acoustic feature vector, which is used in this work.

A disadvantage of introducing dynamic coefficients is that it leads to correlation between different dimensions of the feature vectors. Note, even for the static part, there still exists some correlation between low and high order Cepstral coefficients [77]. The correlation may decrease the discrimination ability of the features. To deal with this problem, *linear projection* schemes are normally used. In these schemes, feature vectors in both training and recognition are normalised using the same linear transforms so that the original acoustic space is projected to one or more uncorrelated sub-spaces. Linear discriminant analysis (LDA) [27, 12] and Heteroscedastic linear discriminant analysis (HLDA) [70] are widely used linear projection schemes. The HLDA transform is used as the linear projection scheme for the LVCSR systems in this work.

2.2 Hidden Markov Models

The previous section introduced how to extract an observation sequence, i.e. a series of feature vectors, from the raw speech waveforms. This section will discuss the *acoustic model*, which

gives a probabilistic mapping between an observation sequence and a given word sequence. Hidden Markov models (HMMs) are the most successful and popular statistical acoustic models in speech recognition. Since the technique was first introduced in 1970's, it has rapidly become the dominant form of acoustic modelling and has been applied to all kinds of speech recognition tasks [7, 61]. This is partly because of the existence of efficient parameters training and recognition algorithms for HMMs. This section will discuss the basic concept of HMMs.

2.2.1 HMMs as Acoustic Models

A *hidden Markov model* (HMM) is a statistical generative model. It is very popular and successful in speech recognition. In HMM based speech recognition, the speech observations of a particular acoustic unit, such as a word or a phone, are assumed to be generated by a finite state machine. The state changes at each time unit with a certain probability. At each time instance, when a state is entered, an observation is generated from some probability function. The use of HMMs is dependent on a number of assumptions:

- **Quasi-stationary**: Speech signals may be split into short segments corresponding to the hidden states, in which the waveform is considered to be stationary and feature vectors can be extracted. The transitions between these states are assumed to be instantaneous.
- **Conditional independence**: Each observation is assumed to be generated with a certain probability associated with a hidden state. This means the observation is only dependent on the current state and is conditionally independent of both the previous and the following observations, given the state.

Neither of the two assumptions is true for real speech. Much research has been carried out to compensate the effect of the poor assumptions or to find alternative models for speech. However, the standard HMM is still a successful acoustic modelling technique and is widely used in most speech recognition systems.

A *left-to-right* HMM with three emitting states is shown in figure 2.1. This is the typical topology of HMMs used in speech recognition, though the number of states may vary.

Let $\mathbf{O}$ be a sequence of observed speech feature vectors corresponding to the HMM of a particular acoustic unit, for example a word or a phone. It is defined as $\mathbf{O} = [\mathbf{o}_1, \cdots, \mathbf{o}_T]$ where $\mathbf{o}_t$, $1 \leq t \leq T$, is a D dimensional feature vector, T is the the length of the speech sequence. These observations are assumed to be generated one by one by the 3 emitting states in figure 2.1. The generation process starts from the first non-emitting state. At each time instance, the state transits with a certain probability to either itself or the contiguous right state. The transition probability is a discrete distribution denoted as a_{ij} for transition from state i to state j. When an emitting state is entered, an observation is generated at that time instance with a probability density $b_j(\mathbf{o}_t)$ for state j, which can be either discrete or continuous. Therefore, the observation sequence is associated with a state sequence, denoted as $\boldsymbol{\omega} = [\omega_1, \cdots, \omega_T]$. Note that the entry

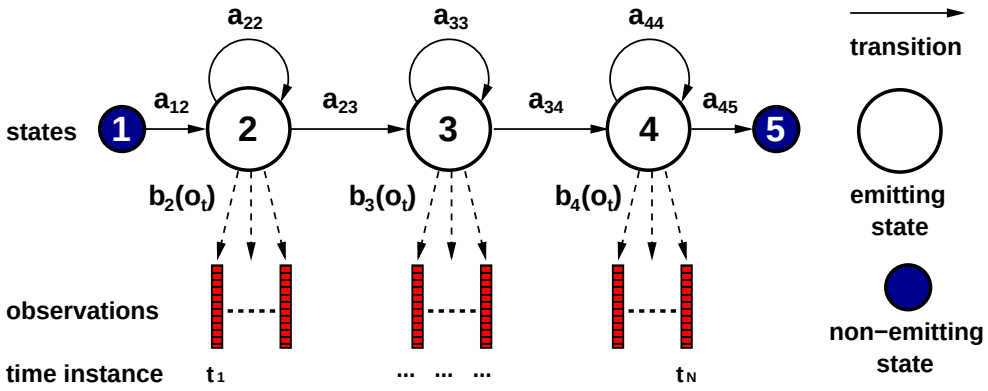


Figure 2.1 *A left-to-right HMM with three emitting states*

and exit states of the HMM in figure 2.1 are non-emitting, i.e. no observations are generated by the two states. This is for construction of composite HMMs and will be discussed later. In practice, only the observation sequence $\mathbf{O}$ can be observed and the underlying state sequence $\boldsymbol{\omega}$ is hidden. This is why the model is named hidden Markov model. The observation sequence and the hidden state sequence are sometimes put together as $\{\mathbf{O}, \boldsymbol{\omega}\}$ and referred to as *complete data set*. To sum up, the parameter set $\mathcal{M}$ of an HMM includes the following parameters:

- **π - Initial state distribution**

  Letting ω_t denote the state at time t, the initial state distribution of state i is expressed as

  $$\pi_i = P(\omega_1 = i) \tag{2.2}$$

  Being a distribution, it must satisfy

  $$\sum_{i=1}^{N} \pi_i = 1 \qquad \pi_i \geq 0 \tag{2.3}$$

  where N is the total number of states. From equation (2.3), by introducing the non-emitting entry state and having a standard left-to-right topology, the initial state distribution of the first state is always 1.

- **$\boldsymbol{A}$ - State transition probability matrix**

  The element of state transition probability matrix $\boldsymbol{A}$ is defined as

  $$a_{ij} = P(\omega_{t+1} = j | \omega_t = i) \tag{2.4}$$

  As the HMMs used in speech recognition are normally constrained to be left-to-right, the matrix is not necessarily full. Given the definition of $\boldsymbol{A}$, the elements must satisfy

  $$\sum_{j=1}^{N} a_{ij} = 1 \qquad a_{ij} \geq 0 \tag{2.5}$$

- **B - State output probability distributions**

  Each emitting state j is associated with one output probability distribution which generates an observation at each time instance. The distribution is defined as

$$b_j(\mathbf{o}_t) = p(\mathbf{o}_t|\omega_t = j) \tag{2.6}$$

  According to different types of the state output distribution, there are two kinds of HMM. If $b_j(\mathbf{o}_t)$ is a discrete distribution, the model is called discrete HMM (DHMM). Alternatively, if $b_j(\mathbf{o}_t)$ is a continuous density, the model is referred to as continuous density HMM (CDHMM). CDHMMs are the focus in this thesis.

In considering the use of HMMs in speech recognition, there are three main issues need to be discussed: likelihood calculation with HMMs, training HMM parameters and inference or recognition using HMMs. These issues will be addressed in the following sections.

2.2.2 Likelihood Calculation with HMMs

Likelihood calculation is a basic issue to be addressed when using HMMs. Its aim is to calculate the likelihood of a particular observation sequence given the hypothesis associated with it and a set of HMMs. As shown in figure 2.1, the observation sequence $\mathbf{O} = [\mathbf{o}_1, \cdots, \mathbf{o}_T]$ is generated by the HMM moving through a state sequence $\boldsymbol{\omega} = [\omega_1, \cdots, \omega_T]$. As $\boldsymbol{\omega}$ is hidden, the required likelihood is computed as an expectation over all possible state sequences $\boldsymbol{\omega}$ associated with the hypothesis $\mathcal{H}$

$$\begin{aligned} p(\mathbf{O}|\mathcal{H},\mathcal{M}) &= \sum_{\boldsymbol{\omega}} p(\mathbf{O},\boldsymbol{\omega}|\mathcal{H},\mathcal{M}) = \sum_{\boldsymbol{\omega}} P(\boldsymbol{\omega}|\mathcal{H},\mathcal{M}) p(\mathbf{O}|\boldsymbol{\omega},\mathcal{M}) \\ &= \sum_{\boldsymbol{\omega}} a_{\omega_0\omega_1} \prod_t a_{\omega_{t-1}\omega_t} b_{\omega_t}(\mathbf{o}_t) \end{aligned} \tag{2.7}$$

where $\mathcal{M} = \{\boldsymbol{\pi}, \boldsymbol{A}, \boldsymbol{B}\}$ is the set of model parameters including a_{ij} and the parameters of $b_j(\mathbf{o}_t)$. The initial transition probability from a non-emitting start state ω_0 to the first emitting state ω_1, $a_{\omega_0\omega_1}$, is always 1.

The above only considers a single HMM. For continuous speech recognition or where sub-word acoustic units are used, the observation sequence is associated with a model sequence. The exact time boundaries between individual words, or sub-words units, are not known. In this case, a series of HMMs are connected together to form the model sequence. This is an extension to a single HMM, which is done by linking individual HMMs to form a *composite* HMM, as shown in figure 2.2.

The non-emitting exit state of model A and the entry state of model B are removed and replaced by a connecting link. The transition probability of the connecting link is the same as the transition probability from the last emitting state to the former non-emitting exit state of model A. The entry state of the composite HMM is the entry state of model A and the exit state

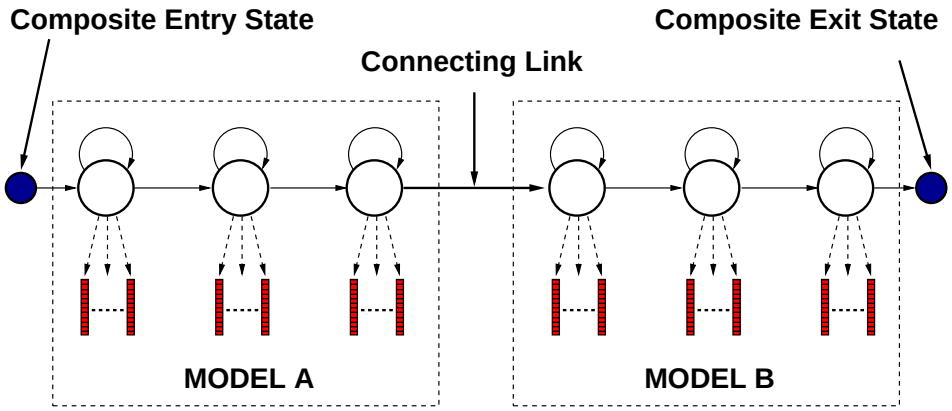


Figure 2.2 *A composite HMM constructed from two individual HMMs*

is that of model B. The set of possible state sequences is now the set of all state paths of this composite HMM AB.

To calculate the likelihood in equation (2.7), the form of the state output distribution $b_j(\mathbf{o}_t)$ needs to be known. In the majority of CDHMMs, a multivariate *Gaussian mixture model* (GMM) is used as the density function

$$b_j(\mathbf{o}_t) = \sum_{m=1}^{M_j} c_{jm} \mathcal{N}\left(\mathbf{o}_t; \boldsymbol{\mu}^{(jm)}, \boldsymbol{\Sigma}^{(jm)}\right) \tag{2.8}$$

where M_j is the number of mixture components for state j, c_{jm} is the weight of component m of state j, which satisfy a sum to one constraint to ensure $b_j(\mathbf{o}_t)$ is a valid distribution. Each individual component is a multivariate Gaussian distribution

$$\mathcal{N}(\mathbf{o}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = (2\pi)^{-\frac{D}{2}} |\boldsymbol{\Sigma}|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(\mathbf{o}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{o}-\boldsymbol{\mu})\right\} \tag{2.9}$$

where D is the dimension of feature vector, $\boldsymbol{\mu}$ is the mean vector, and the covariance matrix $\boldsymbol{\Sigma}$ is normally assumed to be diagonal. The use of a diagonal covariance matrix may give poor modelling of correlation between different dimensions. Hence, many complicated covariance modelling techniques have been investigated [35, 5]. However, diagonal covariance matrices are still widely used because of the low computational cost and their successful use in state-of-the-art LVCSR systems. In this thesis, only diagonal covariance matrices are considered[2].

2.3 Maximum Likelihood Training of HMMs

The second important issue of HMMs is the estimation of the HMM parameters. *Maximum likelihood* (ML) training is the most widely used approach of learning HMM parameters because an

[2]As HLDA transform is also used to de-correlate different dimensions of the raw features in this work, the diagonal Gaussian covariance assumption is reasonable.

efficient training algorithm can be derived in ML training. The aim is to find the model parameters that maximise the likelihood of the observation sequence given the correct transcriptions and the model parameters.

$$\hat{\mathcal{M}}_{\mathtt{ML}} = \arg\max_{\mathcal{M}} p(\mathbf{O}|\mathcal{H},\mathcal{M}) = \arg\max_{\mathcal{M}} \log p(\mathbf{O}|\mathcal{H},\mathcal{M}) \tag{2.10}$$

where $\mathcal{H}$ is the correct transcription and $\mathcal{M}$ is the HMM parameter set. Due to the existence of hidden variables in HMMs, direct optimisation of equation (2.10) with respect to $\mathcal{M}$ is non-trivial. One solution for this type of optimisation is the *expectation maximisation* (EM) algorithm [21].

2.3.1 Expectation Maximization (EM) Algorithm

The expectation maximisation (EM)algorithm is widely used for optimisation of statistical models with hidden variables [21]. The basic idea of the algorithm is to introduce a lower bound for the log-likelihood and iteratively update the model parameters to increase the bound and consequently increase the log-likelihood. For HMMs, the state is regarded as the hidden variable. A variational distribution of the hidden state sequence, $q(\boldsymbol{\omega})$, may be introduced to construct a lower bound. Applying Jensen's inequality to the log-likelihood of the training data yields

$$\begin{aligned} \log p(\mathbf{O}|\mathcal{H},\mathcal{M}) &= \log \sum_{\boldsymbol{\omega}} q(\boldsymbol{\omega}) \frac{p(\mathbf{O},\boldsymbol{\omega}|\mathcal{H},\mathcal{M})}{q(\boldsymbol{\omega})} && (2.11) \\ &\geq \langle \log p(\mathbf{O},\boldsymbol{\omega}|\mathcal{H},\mathcal{M}) \rangle_{q(\boldsymbol{\omega})} + \mathtt{H}(q(\boldsymbol{\omega})) && (2.12) \end{aligned}$$

where $< f(x) >_{g(x)}$ is the expectation of function $f(x)$ with respect to a distribution $g(x)$. If $g(x) = P(x)$ is a discrete distribution, the expectation is calculated by

$$< f(x) >_{P(x)} = \sum_{x} f(x) P(x) \tag{2.13}$$

while for continuous density $g(x) = p(x)$, it is defined as

$$< f(x) >_{p(x)} = \int_{x} f(x) p(x)\, dx \tag{2.14}$$

$\mathtt{H}(g(x))$ is the entropy of the distribution $g(x)$. For discrete distribution $g(x) = P(x)$,

$$\mathtt{H}\left(P(x)\right) = -\sum_{x} P(x) \log P(x) \tag{2.15}$$

while for continuous density $g(x) = p(x)$, the entropy is defined as

$$\mathtt{H}\left(p(x)\right) = -\int_{x} p(x) \log p(x)\, dx \tag{2.16}$$

It can be shown [21] that the inequality (2.12) becomes an equality when

$$q(\boldsymbol{\omega}) = P(\boldsymbol{\omega}|\mathbf{O},\mathcal{H},\mathcal{M}) \tag{2.17}$$

However, as $\mathcal{M}$ is the set of parameters to be optimised, it is not practical to directly calculate $P(\boldsymbol{\omega}|\mathbf{O},\mathcal{H},\mathcal{M})$. One solution is to use the current parameter estimate to calculate the state posterior. This yields a lower bound of the real log-likelihood. The new estimate of the model set can then be obtained by maximising the lower bound in equation (2.12). This is an iterative process. At the $(k+1)^{th}$ iteration, the lower bound can be expressed as

$$\log p(\mathbf{O}|\mathcal{H},\mathcal{M}_{k+1}) \geq \langle \log p(\mathbf{O},\boldsymbol{\omega}|\mathcal{H},\mathcal{M}_{k+1})\rangle_{P(\boldsymbol{\omega}|\mathbf{O},\mathcal{H},\hat{\mathcal{M}}_k)} + \mathtt{H}\left(P(\boldsymbol{\omega}|\mathbf{O},\mathcal{H},\hat{\mathcal{M}}_k)\right) \tag{2.18}$$

where $\mathcal{M}_{k+1}$ is the parameter set to estimate, $\hat{\mathcal{M}}_k$ is the current estimate of iteration k. The new estimate $\hat{\mathcal{M}}_{k+1}$ is then found by maximising the first term of right-hand-side (RHS) of equation (2.18), which is normally referred to as an *auxiliary function*[3]

$$\mathcal{Q}_{\mathtt{ML}}(\mathcal{M}_{k+1};\hat{\mathcal{M}}_k) = \langle \log p(\mathbf{O},\boldsymbol{\omega}|\mathcal{H},\mathcal{M}_{k+1})\rangle_{P(\boldsymbol{\omega}|\mathbf{O},\mathcal{H},\hat{\mathcal{M}}_k)} \tag{2.19}$$

By using equation (2.19) in equation (2.18) and considering equation (2.17), it is trivial to prove that the increase of the auxiliary function with respect to $\mathcal{M}_{k+1}$ guarantees that the log-likelihood with the new model estimate will not decrease, i.e.

$$\begin{aligned}
\mathcal{Q}_{\mathtt{ML}}(\hat{\mathcal{M}}_{k+1};\hat{\mathcal{M}}_k) &\geq \mathcal{Q}_{\mathtt{ML}}(\hat{\mathcal{M}}_k;\hat{\mathcal{M}}_k) \quad \Rightarrow \\
\mathcal{Q}_{\mathtt{ML}}(\hat{\mathcal{M}}_{k+1};\hat{\mathcal{M}}_k) + \mathtt{H}\left(P(\boldsymbol{\omega}|\mathbf{O},\mathcal{H},\hat{\mathcal{M}}_k)\right) &\geq \mathcal{Q}_{\mathtt{ML}}(\hat{\mathcal{M}}_k;\hat{\mathcal{M}}_k) + \mathtt{H}\left(P(\boldsymbol{\omega}|\mathbf{O},\mathcal{H},\hat{\mathcal{M}}_k)\right) \Rightarrow \\
\log p(\mathbf{O}|\mathcal{H},\hat{\mathcal{M}}_{k+1}) &\geq \log p(\mathbf{O}|\mathcal{H},\hat{\mathcal{M}}_k)
\end{aligned} \tag{2.20}$$

To find the optimal parameter estimate, there are two distinct steps:

- **Expectation:** Obtain the state posterior given the current model estimate to form the auxiliary function in equation (2.19).
- **Maximisation:** Maximise the auxiliary function to find a new model estimate.

This is why the algorithm is called *expectation maximisation* (EM) algorithm. It should be noted that the EM algorithm always requires an initialisation process. An existing model or a flat estimate can be used to start the iterative EM algorithm [133]. One limitation of the EM algorithm is that it is not a global optimisation approach. Due to the nature of the iterative update, it can only find a local optimum of the model parameters on convergence. The exact update formulae of model parameters depend on the form of the model used. Update formulae for HMM parameters will be discussed in the next section.

2.3.2 Forward-Backward Algorithm and Parameters Re-estimation

For HMMs, the likelihood of the complete data set is shown in equation (2.7). Using this, the auxiliary function in equation (2.19) can be re-arranged as

$$\mathcal{Q}_{\mathtt{ML}}(\mathcal{M}_{k+1};\hat{\mathcal{M}}_k) = \sum_{t,j}\gamma_j(t)\log b_j(\mathbf{o}_t) + \sum_{t,i,j}\xi_{ij}(t)\log a_{ij} \tag{2.21}$$

[3]The auxiliary function is used for deriving update formulae of the model parameters, hence $\mathcal{M}_{k+1}$ is the independent variable. The current estimate, $\hat{\mathcal{M}}_k$, is used to calculate the state posterior distribution. It is a condition rather than the independent variable of the auxiliary function, hence it is separated from $\mathcal{M}_{k+1}$ using a ";".

where t is the time index, $\gamma_j(t)$ is posterior probability of the state at time t being j given the whole training data and the model parameters of the previous iteration. It is often referred to as the *state posterior occupancy*. It is a statistic that may be accumulated from the the posterior distribution of the state sequences. Let ω_t be the hidden state at time t, the state posterior occupancy is defined as

$$\begin{aligned}\gamma_j(t) &= P(\omega_t = j|\mathbf{O}, \mathcal{H}, \hat{\mathcal{M}}_k) \\ &= \sum_{\omega_1,\cdots,\omega_{t-1},\omega_{t+1},\cdots,\omega_T} P(\omega_1,\cdots,\omega_{t-1},\omega_t = j,\omega_{t+1},\cdots,\omega_T|\mathbf{O}, \mathcal{H}, \hat{\mathcal{M}}_k) \end{aligned} \quad (2.22)$$

and ξ_{ij} is the *state pairwise posterior occupancy*, defined as

$$\begin{aligned}\xi_{ij} &= P(\omega_{t-1} = i, \omega_t = j|\mathbf{O}, \mathcal{H}, \hat{\mathcal{M}}_k) \\ &= \sum_{\omega_1,\cdots,\omega_{t-2},\omega_{t+1},\cdots,\omega_T} P(\omega_1,\cdots,\omega_{t-2},\omega_{t-1} = i,\omega_t = j,\omega_{t+1},\cdots,\omega_T|\mathbf{O}, \mathcal{H}, \hat{\mathcal{M}}_k) \end{aligned} \quad (2.23)$$

where T is the length of the observation sequence $\mathbf{O}$.

The calculation of the two state posterior distributions is a key stage in HMM parameter estimation. They can be efficiently computed using the *forward-backward* algorithm, also known as the *Baum-Welsh* algorithm [8]. This algorithm is an efficient re-arrangement of equation (2.22) and equation (2.23) by making use of two intermediate probabilities and the conditional independence assumption of HMMs. The *forward probability*, $\alpha_j(t)$, is defined as the joint likelihood of the partial observation sequence up to t and the hidden state at that time instance

$$\alpha_j(t) = p(\mathbf{o}_1,\cdots,\mathbf{o}_t,\omega_t = j|\mathcal{H}, \hat{\mathcal{M}}_k) \quad (2.24)$$

The forward probability can be efficiently calculated using a recursive formulae for $1 < j < N$ and $1 < t \leq T$,

$$\alpha_j(t) = \left(\sum_{i=2}^{N-1} \alpha_i(t-1)a_{ij}\right) b_j(\mathbf{o}_t) \quad (2.25)$$

where N is the total number of the states including non-emitting and emitting states. The initial and final conditions for the above recursion are

$$\alpha_j(t) = \begin{cases} 1 & j = 1 \quad t = 1 \\ a_{1j}b_j(\mathbf{o}_t) & 1 < j < N \quad t = 1 \\ \sum_{i=2}^{N-1} \alpha_i(T)a_{iN} & j = N \quad t = T \end{cases} \quad (2.26)$$

Similarly, the *backward probability*, $\beta_j(t)$, is introduced as the likelihood of the partial observation sequence from time instance $t+1$ to the end

$$\beta_j(t) = p(\mathbf{o}_{t+1},\cdots,\mathbf{o}_T|\omega_t = j, \mathcal{H}, \hat{\mathcal{M}}_k) \quad (2.27)$$

As in the forward case, this probability can also be computed using the following recursion for $1 < i < N$ and $1 \leq t < T$

$$\beta_j(t) = \sum_{i=2}^{N-1} a_{ji}b_i(\mathbf{o}_{t+1})\beta_i(t+1) \quad (2.28)$$

The initial and final conditions are

$$\beta_j(t) = \begin{cases} a_{jN} & 1 < j < N \quad t = T \\ \sum_{i=2}^{N-1} a_{1i} b_i(\mathbf{o}_1)\beta_i(1) & j = 1 \quad\quad t = 1 \end{cases} \tag{2.29}$$

Note that the parameters to calculate the forward probability and the backward probability are all from $\hat{\mathcal{M}}_k$. Given the two asymmetric probabilities, the state (pairwise) posterior occupancy can be efficiently calculated as below

$$\gamma_j(t) = \frac{\alpha_j(t)\beta_j(t)}{p(\mathbf{O}|\mathcal{H}, \hat{\mathcal{M}}_k)} \tag{2.30}$$

$$\xi_{ij}(t) = \frac{\alpha_i(t-1)a_{ij}b_j(\mathbf{o}_t)\beta_j(t)}{p(\mathbf{O}|\mathcal{H}, \hat{\mathcal{M}}_k)} \tag{2.31}$$

As this calculation requires both forward and backward probabilities, it is called the *forward-backward* algorithm. From the above recursive formulae, the likelihood of the whole observation sequence may also be calculated using either the forward or the backward algorithm

$$p(\mathbf{O}|\mathcal{H}, \hat{\mathcal{M}}_k) = \alpha_N(T) = \beta_1(1) \tag{2.32}$$

Given the sufficient statistics above, the update formulae of HMM parameters can be derived from the auxiliary function in equation (2.21). The transition probabilities between emitting states, i.e., $1 < i, j < N$ can be estimated by

$$\hat{a}_{ij} = \frac{\sum_{t=2}^{T} \xi_{ij}(t)}{\sum_{t=1}^{T} \gamma_i(t)} \tag{2.33}$$

The transition probabilities with non-emitting states, i.e. at the initial or final condition, can be estimated by

$$\hat{a}_{ij} = \begin{cases} \gamma_j(1) & i = 1 \quad\quad 1 < j < N \\ \frac{\gamma_i(T)}{\sum_{t=1}^{T} \gamma_i(t)} & 1 < i < N \quad j = N \end{cases} \tag{2.34}$$

Most CDHMM systems use a GMM as the state output distribution as shown in equation (2.8). In this case, the estimation formulae of the GMM parameters can not be directly derived from the auxiliary function in equation (2.21). To solve this problem, the Gaussian component index is regarded as a special hidden *sub-state*, in which the transition probability is the component weight times the state transition probability. Considering the distinct Gaussian component (sub-state) sequence as the hidden variable sequence, the Gaussian component posterior occupancy can be derived as [65]

$$\gamma_{jm}(t) = \frac{\sum_{i=2}^{N-1} \alpha_i(t-1)a_{ij}c_{jm}b_{jm}(\mathbf{o}_t)\beta_j(t)}{p(\mathbf{O}|\mathcal{H}, \hat{\mathcal{M}}_k)} \tag{2.35}$$

where jm denote the m^{th} Gaussian component of state j, $b_{jm}(\mathbf{o}_t)$ is a Gaussian distribution $\mathcal{N}(\mathbf{o}_t; \boldsymbol{\mu}^{(jm)}, \boldsymbol{\Sigma}^{(jm)})$ and c_{jm} is the weight for the component, which is also from the current

parameter set $\hat{\mathcal{M}}_k$. Given these statistics, the re-estimation formulae for the parameters of GMM at state j are given by

$$\hat{c}_{jm} = \frac{\sum_t \gamma_{jm}(t)}{\sum_{m,t} \gamma_{jm}(t)} \tag{2.36}$$

$$\hat{\boldsymbol{\mu}}^{(jm)} = \frac{\sum_t \gamma_{jm}(t)\mathbf{o}_t}{\sum_t \gamma_{jm}(t)} \tag{2.37}$$

$$\hat{\boldsymbol{\Sigma}}^{(jm)} = \text{diag}\left(\frac{\sum_t \gamma_{jm}(t)\left(\mathbf{o}_t - \hat{\boldsymbol{\mu}}^{(jm)}\right)\left(\mathbf{o}_t - \hat{\boldsymbol{\mu}}^{(jm)}\right)^T}{\sum_t \gamma_{jm}(t)}\right) \tag{2.38}$$

In equation (2.38), only the estimation of the diagonal elements of covariance matrices is given. This is because covariance update is considered for LVCSR systems in this work. For LVCSR systems, due to the large number of Gaussians used, the re-estimation of full covariance matrix $\hat{\boldsymbol{\Sigma}}^{(jm)}$ requires very high cost in both computation and storage of second order statistics. It is not practical given the limited computation resources. Therefore, normally a diagonal covariance is used for each Gaussian component as in equation (2.38).

Because only the mean and the diagonal covariance matrix of each distinct Gaussian component are of interest in this thesis, a new notation, $\boldsymbol{\theta} = [\theta_1, \cdots, \theta_T]$ is introduced to denote the distinct hidden Gaussian component sequence in contrast to $\boldsymbol{\omega}$ which denotes just the state sequence. θ_t is the combination of the state index and Gaussian component index, which denotes a unique Gaussian component at time t. This is required for deriving the update formulae for Gaussian component parameters. Using the component sequence, the likelihood calculation in equation (2.7) can be re-expressed as

$$\begin{aligned} p(\mathbf{O}|\mathcal{H},\mathcal{M}) &= \sum_{\boldsymbol{\theta}} p(\mathbf{O},\boldsymbol{\theta}|\mathcal{H},\mathcal{M}) \\ &= \sum_{\boldsymbol{\theta}} P(\boldsymbol{\theta}|\mathcal{H},\mathcal{M}) \prod_t p(\mathbf{o}_t|\mathcal{M},\theta_t) \end{aligned} \tag{2.39}$$

where $P(\boldsymbol{\theta}|\mathcal{H},\mathcal{M})$ is the probability of the component sequence $\boldsymbol{\theta}$ and $p(\mathbf{o}_t|\mathcal{M},\theta_t)$ is a Gaussian distribution with component index θ_t. A component level auxiliary function can also be obtained with a similar form as equation (2.19)

$$\mathcal{Q}_{\texttt{ML}}(\mathcal{M}_{k+1};\hat{\mathcal{M}}_k) = \langle \log p(\mathbf{O},\boldsymbol{\theta}|\mathcal{H},\mathcal{M}_{k+1})\rangle_{P(\boldsymbol{\theta}|\mathbf{O},\mathcal{H},\hat{\mathcal{M}}_k)} \tag{2.40}$$

For clarity, in the rest of this thesis, m will be used to denote the index of each distinct Gaussian component, which is the equivalent of the index jm in the equations (2.36) to (2.38). A notation $\texttt{ML}$ is also added to the posterior occupancy, $\gamma_m^{\texttt{ML}}(t)$, to denote that it is obtained for ML update. The iteration indices are also dropped for clarity wherever no confusion is introduced. Thus, $\mathcal{M}$ refers to the new parameter to estimate and $\hat{\mathcal{M}}$ refers to the current parameters which are used to calculate the component posterior occupancies in the auxiliary function. By ignoring the constant terms independent of Gaussian parameters, the auxiliary function in equation (2.40), for Gaussian parameter update can be re-written as

$$\mathcal{Q}_{\texttt{ML}}(\mathcal{M};\hat{\mathcal{M}}) = -\frac{1}{2}\sum_{t,m} \gamma_m^{\texttt{ML}}(t)\left\{\log|\boldsymbol{\Sigma}^{(m)}| + \left(\mathbf{o}_t - \boldsymbol{\mu}^{(m)}\right)^T \boldsymbol{\Sigma}^{(m)-1}\left(\mathbf{o}_t - \boldsymbol{\mu}^{(m)}\right)\right\} \tag{2.41}$$

where $\gamma_m^{\mathtt{ML}}(t)$ can be calculated using the component level forward-backward algorithm based on the current model parameters $\hat{\mathcal{M}}$. This auxiliary function will be frequently used in this thesis.

2.3.3 Acoustic Units and Parameters Tying

For speech recognition tasks with a small recognition vocabulary (<1K words), such as digit recognition, HMMs are often used to model individual words. However, for speech recognition with medium (1K-10K words) to large vocabularies (>10K words), it is not possible to obtain sufficient training data for each individual word in the vocabulary. One widely used solution to this problem is to use HMMs to model *sub-word* units, rather than the words themselves. *Phone* is a widely used sub-word unit. It is the smallest acoustic element of speech. One advantage of using phones as the sub-word unit is that there is a standard rule to map phones to words allowing words to be easily split into a sequence of phones. Models based on phones are phone models, which are also referred to as *phones* in this thesis when there is no confusion introduced. The number of phones is normally significantly smaller than the number of words in vocabulary. For example, in a state-of-the-art LVCSR system used in this work, there are only 46 distinct phones compared to typically 64K words in vocabulary. Hence it is usually possible to obtain enough training data to get robust parameter estimates. It should be noted that by using sub-word units, a *dictionary*, or *lexicon*, is required to map the word sequence to a sub-word sequence. The training and recognition are then performed at the sub-word units level[4]. At the end of recognition, the sub-word sequence is converted back to the word sequence.

There are two main types of phone model sets used, *mono-phones* which are context-independent phones, and *context-dependent phones*. The *mono-phone* set uses each individual phone as the sub-word unit and does not take into account the context information. Due to the co-articulatory[5] effect, the pronunciation of the current phone is highly dependent on the preceding and following phones. Thus, for many speech recognition tasks, the use of mono-phones do not yield good performance.

To model these variations, *context dependent* phones are used in most state-of-the-art speech recognition systems. One commonly used context-dependent phone set is the *tri-phone*, which takes into account the preceding and following phones of the current phone. For example, consider the phone **ah**, a possible tri-phone may be **w-ah+n**, where **w** is the preceding phone and **n** is the following phone, "**-**" denotes the preceding (left) context and "**+**" denotes the following (right) context. Therefore, for an isolated word "*one*" with silence at the start and end, the tri-phones are

$$\text{one}=\{\textbf{sil-w+ah w-ah+n ah-n+sil}\}$$

Although it is possible to build context-dependent phones which include more context information, for example, quin-phones [50] consider two phones on either side of the current phone, tri-phones are still the most popular phone models and are the ones used in this thesis. Depending

[4]Normally, the word history is also maintained during recognition.

[5]For example, in vowels, consonant neighbors can have a big effect on formant trajectories near the boundary.

on how word boundaries are considered, tri-phones may be further classified as *cross-word tri-phones* and *word-internal tri-phones*. Cross-word tri-phones allow the tri-phones to span across word boundaries, i.e., at the word boundary, the preceding or following phones of the current phone can be from the adjacent words. Word-internal tri-phones have the constraints that the tri-phone can only be spanned within the word boundaries. Hence, *bi-phones* have to be used to model the start and end phones at the word boundaries. In this work, cross-word tri-phones are considered as they yield good performance for LVCSR systems [128].

One issue with using tri-phones is that the number of possible acoustic units is significantly increased. For example, for a mono-phone set with 46 phones, the number of possible cross-word tri-phones is about 100,000. It is hard to collect sufficient training data to robustly train all tri-phones. To solve this problem, *parameter tying*, or clustering, techniques are often used [135, 134]. The basic idea of the technique is to consider a group of parameters as sharing the same set of values. In training, statistics of the whole group is used to estimate the shared parameter. Tying can be performed at various levels, such as phones, states, Gaussian components, or even mean vectors or covariance matrices of Gaussian components [49]. The most widely used approach is to do state level parameter tying, referred to as *state clustering* [135]. In state clustering, an output distribution is shared among a group of states as illustrated in figure 2.3.

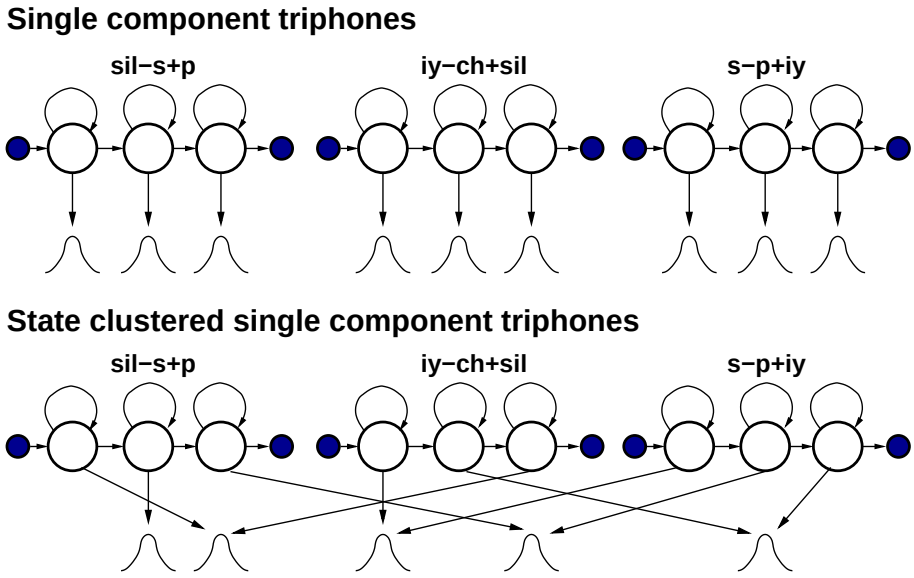


Figure 2.3 *State clustering for single Gaussian tri-phones*

To implement state tying, an appropriate scheme for determining which parameters to be tied together is required. One kind of approaches adopted are *data driven*. A standard data-drive approach is *bottom-up clustering*. A "distance" is calculated between each pair of tri-phones observed in the training data. Then, tri-phones with distances under a given threshold are clustered together. The main problem with data driven approaches is that it is not reliable for

contexts where there is insufficient training data. More seriously, it can not handle the contexts that do not appear in the training data.

The context coverage problem, both for rarely and unseen context, can be efficiently addressed by using the *phonetic decision tree* approach to perform state clustering [135, 134]. A phonetic decision tree is a binary tree with a set of "yes" or "no" questions about the left and right context of each phone. Clustering is performed in a top-down fashion. All states are grouped as one root node at the beginning. The states are then split into "children" nodes by answering these context questions. This split process stops when the amount of training data associated with the current node falls below a minimum threshold. The selection of phonetic questions is important. The questions selected at each split is the one that locally maximises the likelihood increase. Though decision tree clustering is a local optimal binary search, it can efficiently handle the problem of unseen tri-phones as all contexts are mapped to a leaf node. Hence, it is the most popular state clustering approach and it is also adopted in this work.

2.3.4 Limitation of ML Training

Although ML training of HMM parameters has achieved great success in speech recognition, it has two main limitations.

- **Modelling Error**

  ML training assumes that HMM is the "correct" generative model for speech, and hence that the parameters which maximise the likelihood of training data are likely to yield good recognition performance on unseen test data. However, as discussed in section 2.2, the HMM is an incorrect generative model for speech. Training HMMs with the ML criterion may not yield the most appropriate estimate for recognition. It is then preferable to use other training criterion to explicitly aim at reducing the recognition error rate. Discriminative criteria [86, 90] are widely investigated to achieve this goal, which will be discussed in detail in section 2.4.

- **Estimation Error**

  An assumption for ML training to be optimal is that there is sufficient training data given the complexity of the model. In this condition, a minimum variance or consistent estimate of model parameters can be obtained. However, this is not always true. In the case of insufficient training data for a given model complexity, the ML criterion may result in an unreliable estimate, i.e., the variance of the estimate may be large. To deal with this problem, Bayesian approaches may be used, where model parameters are treated as random variables. A prior distribution of parameters is used to represent the prior knowledge. By using the prior, a set of robust parameter estimates, for example maximum a posteriori (MAP) estimate, or the posterior distribution of the parameters may then be obtained. These can greatly reduce the estimation error of ML training due to insufficient data.

Possible solutions to overcome the two limitations of ML training of standard HMMs are reviewed in the next two sections. It should be noted that these limitations also exist in adaptive training and adaptation, which will be reviewed in chapter 3. The major concern of this work is to address the problems within an adaptive training framework. The main contributions are given in chapter 4 and chapter 5.

2.4 Discriminative Training of HMMs

The previous section introduced ML training where the HMM parameters are estimated by maximising the likelihood criterion, $p(\mathbf{O}|\mathcal{M},\mathcal{H})$. As discussed in section 2.3.4, parameters estimated using the ML criterion may not yield good performance because HMMs are not the correct generative model for speech. To overcome this problem, discriminative training criteria have been proposed as an alternative to the ML criterion. These criteria aim at increasing the discriminative ability of the estimate. They have been found to outperform the ML criterion and have been widely used in state-of-the-art speech recognition systems. This section will review some commonly used discriminative criteria and the optimisation schemes.

2.4.1 Discriminative Training Criteria

In the previous section, the ML criterion was described. This criterion may be expressed as:

$$\mathcal{F}_{\texttt{ML}}(\mathcal{M}) = p(\mathbf{O}|\mathcal{H},\mathcal{M}) \tag{2.42}$$

where $\mathbf{O}$ is the observation sequence, $\mathcal{H}$ is the corresponding correct transcription. In contrast to the ML criterion, discriminative criteria take into account competing incorrect hypotheses in training rather than only increasing the likelihood of the correct transcription. They are more related to the recognition error rate than the ML criterion. Commonly used criteria are discussed below.

2.4.1.1 Maximum Mutual Information (MMI)

Maximum mutual information (MMI) criterion is based on $P(\mathcal{H}_{\texttt{ref}}|\mathbf{O})$, the posterior probability of the correct transcription given the observation sequence [6]. By applying an empirically set probability scaling factor κ [6], the MMI criterion is expressed as[7]

$$\mathcal{F}_{\texttt{MMI}}(\mathcal{M}) = \frac{p^{\kappa}(\mathbf{O}|\mathcal{H}_{\texttt{ref}},\mathcal{M})P(\mathcal{H}_{\texttt{ref}})}{\sum_{\mathcal{H}} p^{\kappa}(\mathbf{O}|\mathcal{H},\mathcal{M})P(\mathcal{H})} \tag{2.43}$$

[6] It has been shown in [102] that the scaling factor κ is important for MMI training to lead to good test set recognition performance. It aims to make the less likely hypotheses contribute to the criterion and make the criterion more smoothly differentiable. It typically equals the inverse of the language model scale used in recognition.

[7] Here, only the criterion of a single utterance is discussed. For multiple utterances, the criterion is a product of the individual ones. This is equivalent to the formula in [6] where the logarithm of the posterior distribution is used as the MMI criterion.

where $\mathcal{H}_{\texttt{ref}}$ is the transcription corresponding to the observations, $\mathcal{H}$ denotes all possible hypothesis sequences of the observation sequence including both the correct transcription and the confusing hypotheses. The denominator hypotheses are normally stored as N-Best lists [16] or lattices [102] produced by a recogniser running on the training data. It should be noted that word-level lattices or N-Best lists are normally converted to phone-level ones as phones are the actual sub-word units to be trained. This process is referred to as *phone marking*. From the criterion in equation (2.43), MMI naturally reduces the sentence error rate of the training data as it is defined for the whole sentence. MMI is one of the earliest discriminative training criteria and has been successfully used in LVCSR [92].

2.4.1.2 Minimum Phone Error (MPE)

From the Bayesian perspective, a generic criterion to describe recognition error is the Bayesian risk, which is defined as an expected loss of an estimator as below

$$\mathcal{F}_{\texttt{MBR}}(\mathcal{M}) = \sum_{\mathcal{H}} P(\mathcal{H}|\mathbf{O},\mathcal{M}) l(\mathcal{H},\mathcal{H}_{\texttt{ref}}) \tag{2.44}$$

where $P(\mathcal{H}|\mathbf{O},\mathcal{M})$ is the posterior distribution of a hypothesis $\mathcal{H}$ given the observation sequence and the model parameters, $l(\mathcal{H},\mathcal{H}_{\texttt{ref}})$ is a loss function of $\mathcal{H}$ given the reference or correct transcription $\mathcal{H}_{\texttt{ref}}$. The *minimum Bayesian risk* (MBR) criterion was first used in decoding [44]. As the criterion is a good description of recognition error, it has also been adopted in discriminative training [23]. An MBR estimator finds the model parameters by minimising the Bayesian risk of the training data in equation (2.44)

$$\hat{\mathcal{M}}_{\texttt{MBR}} = \arg\max_{\mathcal{M}} \mathcal{F}_{\texttt{MBR}}(\mathcal{M}) \tag{2.45}$$

Different forms of the loss function $l(\mathcal{H},\mathcal{H}_{\texttt{ref}})$ may be used yielding different discriminative training criteria. One commonly used MBR criterion is the *minimum phone error* (MPE) [92] criterion, which has a loss function closely related to word error rate (WER) rather than sentence error rate.

$$l(\mathcal{H},\mathcal{H}_{\texttt{ref}}) = \mathcal{A}(\mathcal{H},\mathcal{H}_{\texttt{ref}}) \tag{2.46}$$

where $\mathcal{A}(\mathcal{H},\mathcal{H}_{\texttt{ref}})$ is the phone accuracy of the hypothesis $\mathcal{H}$ given the reference $\mathcal{H}_{\texttt{ref}}$. It equals the number of reference phones minus the number of errors[8]. Details about the calculation of $\mathcal{A}(\mathcal{H},\mathcal{H}_{\texttt{ref}})$ can be found in [90]. Using this loss function in equation (2.44) and again applying a scaling factor similar to the MMI criterion, the MPE criterion is expressed by [9]

$$\mathcal{F}_{\texttt{MPE}}(\mathcal{M}) = \sum_{\mathcal{H}} \frac{p^{\kappa}(\mathbf{O}|\mathcal{H},\mathcal{M})P(\mathcal{H})}{\sum_{\breve{\mathcal{H}}} p^{\kappa}(\mathbf{O}|\breve{\mathcal{H}},\mathcal{M})P(\breve{\mathcal{H}})} \mathcal{A}(\mathcal{H},\mathcal{H}_{\texttt{ref}}) \tag{2.47}$$

[8] This is equivalent to the number of correct phones minus the number of insertions.

[9] Note that here the criterion is a *maximisation* criterion during training because it uses the phone accuracy rather than phone error in the criterion.

where $\mathbf{O}$ is the observation sequence of the training utterance, and $\mathcal{H}$ and $\breve{\mathcal{H}}$ both denote possible hypotheses of the training data. Similar to the MMI case, they are obtained from a recogniser and phone-marked before training. Due to the nature of $\mathcal{A}(\mathcal{H}, \mathcal{H}_{\mathtt{ref}})$, the MPE criterion maximises the accuracy of phone transcriptions rather than word-level transcriptions. If the accuracy function $\mathcal{A}(\mathcal{H}, \mathcal{H}_{\mathtt{ref}})$ is defined on a word rather than phone basis, the criterion is referred to as *minimum word error* (MWE). Theoretically, MWE is a more effective criterion than MPE to maximise the word accuracy of the training data. However, it was shown to consistently give slightly poorer results on the test set [90].

Though both MMI and MPE criterion have been effectively used in LVCSR systems. The MPE criterion has been shown to give better performance [90]. Therefore, in this work, MPE is used as the specific discriminative criterion to derive estimation formulae of the parameters.

2.4.1.3 Minimum Classification Error (MCE)

Minimum classification error (MCE) criterion [64] was originally proposed for isolated word recognition, where there are a fixed number of candidate classes (words) $i = 1, \cdots, M$ in recognition. A mis-classification measure is defined for class i as

$$d_i(\mathbf{O}) = \log \left(\frac{1}{M-1} \sum_{j \neq i} p^{\eta}(\mathbf{O}|\mathcal{M}, j) \right)^{\frac{1}{\eta}} - \log p(\mathbf{O}|\mathcal{M}, i) \tag{2.48}$$

where j is the index of the classes other than i, M is the total number of classes, $\eta > 0$ is a constant. The misclassification measure tends to be positive if the system does not classify the word as class i, and negative if it is classified as class i. To transform the misclassification measure into a normalised differentiable function, a sigmoid function is used to embed $d_i(\mathbf{O})$ in a smooth zero-one function. The MCE criterion to be minimised is then a summation of the normalised misclassification measure over all the correct classes.

$$\mathcal{F}_{\mathtt{MCE}}(\mathcal{M}) = \sum_{r} \frac{1}{1 + \exp\left(-\gamma d_{i_r}(\mathbf{O}_r)\right)} \tag{2.49}$$

where r is the utterance index, $\mathbf{O}_r$ is the r^{th} observation sequence, i_r is the index of the correct classes of $\mathbf{O}_r$, and $\gamma > 0$ is also a constant. The MCE criterion is zero for each word that is correctly recognised and one for each incorrect word. The constants η and γ are used to control the differentiability of the criterion. The relationship between the MCE and the MMI criterion is discussed in [102]. The MCE criterion can be extended to continuous speech recognition by making use of N-Best lists [16] or lattices [102]. Recently, it was also used for LVCSR tasks and showed similar performance as the MWE criterion [80]. As the MCE criterion has not shown better performance than the MPE criterion and the optimisation scheme for LVCSR tasks is more complicated, it is not adopted in this work.

2.4.2 Weak-Sense Auxiliary Function and Parameter Re-estimation

The previous section discussed various discriminative criteria. This section will review the optimisation schemes that may be used to train models based on these criteria. MPE criterion is

used as the specific discriminative training criterion to be optimised.

The ML criterion can be optimised by the EM algorithm in which an auxiliary function is introduced as the lower bound of the likelihood. The auxiliary function is derived by applying Jensen's inequality to the log-likelihood. An important property is that the increase of the auxiliary function will guarantee not to decrease the log-likelihood. An auxiliary function with such property is sometimes referred to as a *strong-sense* auxiliary function [90]. However, in MPE training, it is hard to find a strong-sense auxiliary function due to the denominator term in the criterion. To allow the discriminative criterion to be optimised, an *extended Baum-Welch* algorithm (EBW) was proposed [46, 86, 131], which extends the Baum-Eagon inequality to rational functions by using an additional smoothing term to ensure the convexity of the auxiliary function. This allows discriminative training to be done in a similar fashion as the standard Baum-Welch algorithm for ML training. An alternative approach, which yields similar update formulae and is highly flexible, uses a *weak-sense auxiliary function* [90]. This is the approach adopted in this work. A *weak-sense auxiliary function* for a criterion $\mathcal{F}(\mathcal{M})$ is a smoothing function $\mathcal{Q}(\mathcal{M};\hat{\mathcal{M}})$ which has the same gradients at the current model parameters $\hat{\mathcal{M}}$, i.e.

$$\left.\frac{\partial \mathcal{Q}(\mathcal{M};\hat{\mathcal{M}})}{\partial \mathcal{M}}\right|_{\mathcal{M}=\hat{\mathcal{M}}} = \left.\frac{\partial \mathcal{F}(\mathcal{M})}{\partial \mathcal{M}}\right|_{\mathcal{M}=\hat{\mathcal{M}}} \tag{2.50}$$

Maximising the function $\mathcal{Q}(\mathcal{M};\hat{\mathcal{M}})$ with respect to $\mathcal{M}$ does not guarantee an increase of $\mathcal{F}(\mathcal{M})$. However, if $\mathcal{Q}(\mathcal{M};\hat{\mathcal{M}})$ reaches a local maximum at $\hat{\mathcal{M}}$, i.e., the gradient is 0 at that point, $\mathcal{F}(\mathcal{M})$ is also guaranteed to also be at a local maximum. An appropriate weak-sense auxiliary function of the MPE criterion in equation (2.47) can be defined as [90]

$$\mathcal{Q}_{\texttt{MPE}}(\mathcal{M};\hat{\mathcal{M}}) = \mathcal{Q}_{\texttt{n}}(\mathcal{M};\hat{\mathcal{M}}) - \mathcal{Q}_{\texttt{d}}(\mathcal{M};\hat{\mathcal{M}}) \tag{2.51}$$

where the numerator and denominator parts, $\mathcal{Q}_{\texttt{n}}(\mathcal{M};\hat{\mathcal{M}})$ and $\mathcal{Q}_{\texttt{d}}(\mathcal{M};\hat{\mathcal{M}})$ respectively, have the same form as the standard ML auxiliary function in equation (2.41). To clarify the relationship between different terms in the MPE weak-sense auxiliary function, the standard auxiliary function is re-expressed in terms of sufficient statistics in this work. For example, for the ML auxiliary function in equation (2.41), it may be re-expressed as

$$\mathcal{Q}_{\texttt{ML}}(\mathcal{M};\hat{\mathcal{M}}) = \mathcal{G}(\mathcal{M};\mathbf{\Gamma}_{\texttt{ML}}) \tag{2.52}$$

where

$$\begin{aligned}\mathcal{G}(\mathcal{M};\mathbf{\Gamma}_{\texttt{ML}}) \;=\; & -\frac{1}{2}\sum_m \Big\{ \gamma_m^{\texttt{ML}} \log|\mathbf{\Sigma}^{(m)}| + \text{tr}\left(\mathbf{L}_{\texttt{ML}}^{(m)}\mathbf{\Sigma}^{(m)-1}\right) \\ & - 2\boldsymbol{\mu}^{(m)T}\mathbf{\Sigma}^{(m)-1}\mathbf{k}_{\texttt{ML}}^{(m)} + \boldsymbol{\mu}^{(m)T}\mathbf{\Sigma}^{(m)-1}\boldsymbol{\mu}^{(m)} \Big\}\end{aligned} \tag{2.53}$$

where $\text{tr}(\cdot)$ is trace of a square matrix, and $\mathbf{\Gamma}_{\texttt{ML}}$ is the set of ML statistics of all Gaussian components

$$\mathbf{\Gamma}_{\texttt{ML}} = \left\{\gamma_m^{\texttt{ML}}, \mathbf{k}_{\texttt{ML}}^{(m)}, \mathbf{L}_{\texttt{ML}}^{(m)}\right\} \tag{2.54}$$

where m is the component index and

$$\gamma_m^{\mathtt{ML}} = \sum_t \gamma_m^{\mathtt{ML}}(t) \tag{2.55}$$

$$\mathbf{k}_{\mathtt{ML}}^{(m)} = \sum_t \gamma_m^{\mathtt{ML}}(t)\mathbf{o}_t \tag{2.56}$$

$$\mathbf{L}_{\mathtt{ML}}^{(m)} = \sum_t \gamma_m^{\mathtt{ML}}(t)\mathbf{o}_t\mathbf{o}_t^T \tag{2.57}$$

The only difference between the numerator, denominator associated with discriminative training and the standard auxiliary functions is that the "posterior occupancy" for Gaussian component m at time t, $\gamma_m^{\mathtt{n}}(t)$ and $\gamma_m^{\mathtt{d}}(t)$, is not calculated based on the correct transcription as it is in standard forward-backward algorithm. Instead, the forward-backward algorithm is first applied within each phone arc of the lattice [10]. It is then applied at word-level within the lattice to figure out the word arc posteriors. Phone accuracy is measured for each phone arc. The arcs with higher accuracy than the average are classified as numerator arcs, and those with lower accuracy as denominator arcs. $\gamma_m^{\mathtt{n}}(t)$ is then calculated based on the numerator arcs by multiplying together the within arc component posterior occupancy, the word posterior occupancy and the difference between the phone arc and average accuracy. Similarly, $\gamma_m^{\mathtt{d}}(t)$ is computed based on denominator arcs. Details about this lattice forward-backward algorithm and the calculation of arc-based phone accuracy can be found in [90]. With the new component posterior occupancy, the auxiliary function for the numerator can also be written as

$$\mathcal{Q}_{\mathtt{n}}(\mathcal{M};\hat{\mathcal{M}}) = \mathcal{G}(\mathcal{M};\mathbf{\Gamma}_{\mathtt{n}}) \tag{2.58}$$

where $\mathcal{G}(\cdot)$ is defined in equation (2.53), $\mathbf{\Gamma}_{\mathtt{n}}$ is similar to $\mathbf{\Gamma}_{\mathtt{ML}}$ except for using $\gamma_m^{\mathtt{n}}(t)$ instead of $\gamma_m^{\mathtt{ML}}(t)$. The denominator function is similar.

The weak-sense auxiliary function in equation (2.51) is not guaranteed to be a convex function, hence is unlikely to yield good convergence [90]. To ensure a convex weak-sense auxiliary function and consequently improve stability in optimisation, a smoothing function, $\mathcal{S}(\mathcal{M};\hat{\mathcal{M}})$, is added to the auxiliary function in equation (2.51). This smoothing function must satisfy the following constraint

$$\left.\frac{\partial\mathcal{S}(\mathcal{M};\hat{\mathcal{M}})}{\partial\mathcal{M}}\right|_{\hat{\mathcal{M}}} = 0 \tag{2.59}$$

to ensure the resulting auxiliary function is still a valid weak-sense auxiliary function. One form of smoothing function for Gaussian parameters was first introduced in [86] for discrete HMM and then extended to CDHMM later in [102]. It can be generally written as [102]

$$\mathcal{S}(\mathcal{M};\hat{\mathcal{M}}) = \sum_m D_m \int_{\mathbf{o}} p(\mathbf{o}|m,\hat{\mathcal{M}}) \log p(\mathbf{o}|m,\mathcal{M})\, d\mathbf{o} \tag{2.60}$$

[10] In common with the majority of discriminative training schemes, lattices are used to represent possible denominator paths. These lattices are phone-marked before training [93, 90].

where $p(\mathbf{o}|m,\mathcal{M})$ is the Gaussian distribution for component m given the model parameters $\mathcal{M}$, and D_m is a component-specific smoothing constant to ensure convergence. It is shown in appendix A that the above generic smoothing function satisfies the constraint equation (2.59). The exact form can then be derived by using the exact expression of $p(\mathbf{o}|m,\hat{\mathcal{M}})$ in equation (2.60). As shown in appendix A, for standard HMMs, it is expressed as

$$\begin{aligned}\mathcal{S}(\mathcal{M};\hat{\mathcal{M}}) &= \sum_m -\frac{D_m}{2}\Big\{\log|\boldsymbol{\Sigma}^{(m)}| + \mathrm{tr}\Big((\hat{\boldsymbol{\Sigma}}_{\mathsf{c}}^{(m)} + \hat{\boldsymbol{\mu}}_{\mathsf{c}}^{(m)}\hat{\boldsymbol{\mu}}_{\mathsf{c}}^{(m)T})\boldsymbol{\Sigma}^{(m)-1}\Big) \\ &\quad - 2\boldsymbol{\mu}^{(m)T}\boldsymbol{\Sigma}^{(m)-1}\hat{\boldsymbol{\mu}}_{\mathsf{c}}^{(m)} + \boldsymbol{\mu}^{(m)T}\boldsymbol{\Sigma}^{(m)-1}\boldsymbol{\mu}^{(m)}\Big\} \end{aligned} \tag{2.61}$$

where $\hat{\boldsymbol{\Sigma}}_{\mathsf{c}}^{(m)}$ and $\hat{\boldsymbol{\mu}}_{\mathsf{c}}^{(m)}$ are the covariance matrix and mean vector of Gaussian component m from the current model set $\hat{\mathcal{M}}$. This form was used in the standard MPE training of HMM parameters but described as a scalar version in [90].

The constant D_m is a critical value in MPE training in order to make the weak-sense auxiliary function convex and yield a rapid and stable update. A large value of D_m will guarantee that the MPE training does not go too aggressively to reach a stable update, but will result in a slow update. A small value may give a fast update but could affect the convexity of the weak-sense auxiliary function. There is no ideal approach for obtaining D_m satisfying both purposes. As with the EBW updates, the value of D_m for weak-sense auxiliary function is set using empirically derived heuristics. As suggested in [90], in this work D_m is determined by

$$D_m = \max\left(2\tilde{D}_m, E\gamma_m^{\mathsf{d}}\right) \tag{2.62}$$

where $\tilde{D}_m$ is the smallest value required to ensure the updated covariance matrix in equation (2.76) is positive-definite, E is a user-specified constant, normally 1 or 2 for LVCSR, and $\gamma_m^{\mathsf{d}} = \sum_t \gamma_m^{\mathsf{d}}(t)$ is the total denominator posterior occupancy for component m.

The definition of the smoothing function in equation (2.61) is closely related to the standard auxiliary function. By doing a little algebra, it can be expressed in the same form as equation (2.53),

$$\mathcal{S}(\mathcal{M};\hat{\mathcal{M}}) = \mathcal{G}(\mathcal{M};\boldsymbol{\Gamma}_{\mathsf{s}}) \tag{2.63}$$

where the smoothing function statistics are

$$\boldsymbol{\Gamma}_{\mathsf{s}} = \left\{D_m, D_m\mathbf{k}_{\mathsf{s}}^{(m)}, D_m\mathbf{L}_{\mathsf{s}}^{(m)}\right\} \tag{2.64}$$

and

$$\mathbf{k}_{\mathsf{s}}^{(m)} = \hat{\boldsymbol{\mu}}_{\mathsf{c}}^{(m)} \tag{2.65}$$

$$\mathbf{L}_{\mathsf{s}}^{(m)} = \hat{\boldsymbol{\Sigma}}_{\mathsf{c}}^{(m)} + \hat{\boldsymbol{\mu}}_{\mathsf{c}}^{(m)}\hat{\boldsymbol{\mu}}_{\mathsf{c}}^{(m)T} \tag{2.66}$$

For MPE training, model parameters tend to be over-trained for LVCSR system [90]. Therefore, it is essential to perform some additional smoothing, referred to as "I-smoothing" to get

robust parameter estimates [93]. To achieve this, a prior distribution over the model parameters, $p(\mathcal{M}|\Phi)$, is added to the MPE criterion. Then the complete MPE criterion becomes

$$\mathcal{F}_{\texttt{MPE}}(\mathcal{M}) = \sum_{\mathcal{H}} \frac{p^{\kappa}(\mathbf{O}|\mathcal{H},\mathcal{M})P(\mathcal{H})}{\sum_{\breve{\mathcal{H}}} p^{\kappa}(\mathbf{O}|\breve{\mathcal{H}},\mathcal{M})P(\breve{\mathcal{H}})}\mathcal{A}(\mathcal{H},\mathcal{H}_{\texttt{ref}}) + \log p(\mathcal{M}|\Phi) \tag{2.67}$$

Note that $\log p(\mathcal{M}|\Phi)$ may be viewed as a weak-sense auxiliary function for itself because it naturally satisfies equation (2.50). As the summation of weak-sense auxiliary functions is also a weak-sense auxiliary function for the summation of the corresponding criteria, a valid weak-sense auxiliary function for the complete MPE criterion in equation (2.67) may be constructed by directly adding $\log p(\mathcal{M}|\Phi)$ to the original weak-sense auxiliary function. Hence, the complete weak-sense auxiliary function for equation (2.67) becomes

$$\mathcal{Q}_{\texttt{MPE}}(\mathcal{M};\hat{\mathcal{M}}) = \mathcal{Q}_{\texttt{n}}(\mathcal{M};\hat{\mathcal{M}}) - \mathcal{Q}_{\texttt{d}}(\mathcal{M};\hat{\mathcal{M}}) + \mathcal{S}(\mathcal{M};\hat{\mathcal{M}}) + \log p(\mathcal{M}|\Phi) \tag{2.68}$$

One commonly used distribution for model parameters is the Normal-Wishart distribution [20] which was also used for maximum a posteriori (MAP) training in [42]. This prior distribution has a similar form as the standard auxiliary function. Ignoring the constants independent of the parameters, the logarithm of the distribution is expressed as

$$\begin{aligned}\log p(\mathcal{M}|\Phi) \quad = \quad & -\frac{\tau^I}{2}\sum_m \Big\{ \log|\boldsymbol{\Sigma}^{(m)}| + \mathrm{tr}\Big(\tilde{\boldsymbol{\Sigma}}^{(m)}\boldsymbol{\Sigma}^{(m)-1}\Big) \\ & + \Big(\boldsymbol{\mu}^{(m)} - \tilde{\boldsymbol{\mu}}^{(m)}\Big)^T \boldsymbol{\Sigma}^{(m)-1}\Big(\boldsymbol{\mu}^{(m)} - \tilde{\boldsymbol{\mu}}^{(m)}\Big)\Big\}\end{aligned} \tag{2.69}$$

where $\Phi = \{\tau^I, \tilde{\boldsymbol{\mu}}^{(m)}, \tilde{\boldsymbol{\Sigma}}^{(m)}\}$ is the set of hyper-parameters of the I-smoothing distribution. τ^I is the specified parameter of the Normal-Wishart distribution which controls the impact of the prior and is normally tuned to tasks. The recognition performance was found to be insensitive to the precise value of τ^I used (within a reasonable range). $\tilde{\boldsymbol{\mu}}^{(m)}$ and $\tilde{\boldsymbol{\Sigma}}^{(m)}$ are the prior hyper-parameters of the distribution. This form of I-smoothing prior may also be re-expressed as

$$\log p(\mathcal{M}|\Phi) = \mathcal{G}(\mathcal{M};\boldsymbol{\Gamma}_{\texttt{p}}) \tag{2.70}$$

where $\mathcal{G}(\cdot)$ is defined in equation (2.53)

$$\boldsymbol{\Gamma}_{\texttt{p}} = \Big\{\tau^I, \tau^I\mathbf{k}_{\texttt{p}}^{(m)}, \tau^I\mathbf{L}_{\texttt{p}}^{(m)}\Big\} \tag{2.71}$$

and

$$\mathbf{k}_{\texttt{p}}^{(m)} = \tilde{\boldsymbol{\mu}}^{(m)} \tag{2.72}$$

$$\mathbf{L}_{\texttt{p}}^{(m)} = \tilde{\boldsymbol{\Sigma}}^{(m)} + \tilde{\boldsymbol{\mu}}^{(m)}\tilde{\boldsymbol{\mu}}^{(m)T} \tag{2.73}$$

Though the log prior in equation (2.69), and the smoothing function in equation (2.61), have similar forms, they are kept separate as they have different functions. The smoothing function, $\mathcal{S}(\mathcal{M};\hat{\mathcal{M}})$, is used to stabilise the optimisation and control the update rate of MPE training. It is based on the *current model parameters* resulting in a similar form to the EBW re-estimation

formulae [131, 86]. On the other hand, the prior distribution, $\log(p(\mathcal{M}|\Phi))$, is used to avoid over-training in a similar way to MAP training [42]. It ensures that for parameters with little data, the estimates are robust, hence are likely to have good generalisation on unseen data. A range of possible priors can be used. The hyper-parameters $\tilde{\boldsymbol{\mu}}^{(m)}$ and $\tilde{\boldsymbol{\Sigma}}^{(m)}$ for I-smoothing distribution was first introduced based on the ML statistics [93], i.e., $\tilde{\boldsymbol{\mu}}^{(m)}$ and $\tilde{\boldsymbol{\Sigma}}^{(m)}$ are ML estimates obtained using $\boldsymbol{\Gamma}_{\texttt{ML}}$, which has been given in equation (2.37) and equation (2.38). Though the standard form of I-smoothing has obtained good performance, using a modified I-smoothing prior, such as dynamic MMI prior [101], may obtain further gains. Though in theory, $\log(p(\mathcal{M}|\Phi))$ with a large τ^I can also lead to stable optimisation without the smoothing term $\mathcal{S}(\mathcal{M};\hat{\mathcal{M}})$, it may lead to very slow update and the updated parameters may be dominated by the prior. Hence, in practice, the two smoothing functions equation (2.61) and (2.69) are both needed to achieve efficient, stable and robust MPE updates.

Given the above sufficient statistics expression of each individual element of the MPE weak-sense auxiliary function, the whole weak-sense auxiliary function in equation (2.68) may be written as

$$\mathcal{Q}_{\texttt{MPE}}(\mathcal{M};\hat{\mathcal{M}}) = \mathcal{G}(\mathcal{M};\boldsymbol{\Gamma}_{\texttt{n}}) - \mathcal{G}(\mathcal{M};\boldsymbol{\Gamma}_{\texttt{d}}) + \mathcal{G}(\mathcal{M};\boldsymbol{\Gamma}_{\texttt{s}}) + \mathcal{G}(\mathcal{M};\boldsymbol{\Gamma}_{\texttt{p}}) \tag{2.74}$$

By differentiating the weak-sense auxiliary function in equation (2.74) with respect to the model parameters and setting it to zero, a closed-form solution can be derived [90]

$$\hat{\boldsymbol{\mu}}^{(m)} = \frac{\mathbf{k}_{\texttt{MPE}}^{(m)}}{\gamma_m^{\texttt{MPE}}} \tag{2.75}$$

$$\hat{\boldsymbol{\Sigma}}^{(m)} = \text{diag}\left(\frac{\mathbf{L}_{\texttt{MPE}}^{(m)}}{\gamma_m^{\texttt{MPE}}} - \hat{\boldsymbol{\mu}}^{(m)}\hat{\boldsymbol{\mu}}^{(m)T}\right) \tag{2.76}$$

where the MPE sufficient statistics is a combination of the individual sufficient statistics

$$\boldsymbol{\Gamma}_{\texttt{MPE}} = \left\{\gamma_m^{\texttt{MPE}}, \mathbf{k}_{\texttt{MPE}}^{(m)}, \mathbf{L}_{\texttt{MPE}}^{(m)}\right\} \tag{2.77}$$

and

$$\gamma_m^{\texttt{MPE}} = \gamma_m^{\texttt{n}} - \gamma_m^{\texttt{d}} + D_m + \tau^I \tag{2.78}$$

$$\mathbf{k}_{\texttt{MPE}}^{(m)} = \mathbf{k}_{\texttt{n}}^{(m)} - \mathbf{k}_{\texttt{d}}^{(m)} + D_m\mathbf{k}_{\texttt{s}}^{(m)} + \tau^I\mathbf{k}_{\texttt{p}}^{(m)} \tag{2.79}$$

$$\mathbf{L}_{\texttt{MPE}}^{(m)} = \mathbf{L}_{\texttt{n}}^{(m)} - \mathbf{L}_{\texttt{d}}^{(m)} + D_m\mathbf{L}_{\texttt{s}}^{(m)} + \tau^I\mathbf{L}_{\texttt{p}}^{(m)} \tag{2.80}$$

The formulae have the same form as the extended Baum-Welch algorithm [86] but are derived from a different perspective.

2.5 Bayesian Training of HMMs

Standard ML training gives a point estimate of the HMM parameters by maximising the likelihood criterion over the training data. In recognition, the ML estimate of the HMM parameters is

used to calculate the likelihood of all test data. One assumption is that there is sufficient training data to yield a reliable point estimate of the HMM parameters. However, this assumption is not always met. For those cases where the training data associated with a particular state or model is limited, the point estimate will not be reliable. To deal with the uncertainty coming from limited training data, Bayesian approaches may be used [120]. In these approaches, the model parameters are assumed to be random variables and associated with a distribution. Hence, the marginal likelihood of the training data is expressed as

$$p(\mathbf{O}|\mathcal{H}) = \int_{\mathcal{M}} p(\mathbf{O}|\mathcal{H},\mathcal{M})p(\mathcal{M}|\Phi)\, d\mathcal{M} \tag{2.81}$$

with

$$p(\mathbf{O}|\mathcal{H},\mathcal{M}) = \sum_{\boldsymbol{\omega}} P(\boldsymbol{\omega}|\mathcal{H},\mathcal{M}) \prod_{t} b_{\omega_t}(\mathbf{o}_t) \tag{2.82}$$

where $\mathbf{O}$ is the observation sequence of all training data, $\mathcal{H}$ is the associated transcription, $\mathcal{M}$ is the HMM parameter set, $p(\mathcal{M}|\Phi)$ is the prior distribution with hyper-parameters Φ, $P(\boldsymbol{\omega}|\mathcal{H},\mathcal{M})$ is the distribution of a particular state sequence $\boldsymbol{\omega}$ given the observation, transcription and model parameters, and $b_{\omega_t}(\mathbf{o}_t)$ is the state output distribution for the state at time t, which is a GMM as defined in equation (2.8).

Given the above Bayesian descriptions, Bayesian training of HMMs aims to update the parameter prior distribution to yield a posterior distribution based on training data which is believed to be similar to the data to be recognised. This parameter posterior distribution represents the uncertainty of HMM parameters given the training data and the prior distribution. During recognition, the likelihood of the test observation sequence is calculated using the posterior distribution. This recognition process is different from the standard one and will be discussed in detail in section 2.6.3.

Normally, the form of the prior distribution is determined in advance. In the Bayesian community, a *conjugate prior* to the likelihood is a common choice [9]. This is because when a conjugate prior is used, the posterior distribution of the parameters given the observations will have the same functional form as the prior. The estimation of the posterior distribution is equivalent to updating the hyper-parameters of the prior distribution. Unfortunately, for HMM parameters, a conjugate prior to the likelihood of observation sequence does not exist due to the hidden variables, states or Gaussian components, in the likelihood calculation [42]. However, a conjugate prior of HMM parameters to the likelihood of the *complete* data set, i.e. the joint data set of hidden variables and observations, may be found. For example, for the Gaussian distributions in standard HMMs [42], a Normal-Wishart distribution is a conjugate prior of mean vectors and covariance matrices. Given the form of the prior distribution, the hyper-parameters need to be estimated. In most research on Bayesian training of HMMs, the hyper-parameters of the prior distribution $p(\mathcal{M}|\Phi)$ are often assumed to be known beforehand [42, 120]. They are usually obtained by prior knowledge, or some ad-hoc methods, such as in MAP training [42]. In this context, Bayesian training of HMMs updates the hyper-parameters of the prior distribution using

the training data so that a posterior parameter distribution is obtained. However, if there is no prior information available, the hyper-parameters of the prior distribution must be estimated from the training data, i.e., to maximise the marginal likelihood in equation (2.81), with respect to the hyper-parameters Φ. This is the basic idea of the *empirical Bayesian* approach [97, 98]. In this case, it can be shown that the empirically estimated prior distribution must have the same form and hyper-parameters as the posterior distribution $p(\mathcal{M}|\mathbf{O},\mathcal{H})$ which is estimated on the training data given a non-informative prior. A detailed discussion will be given in section 5.1.1.

The fundamental problem in Bayesian training of HMMs is the estimation of the hyper-parameter Φ of the parameter posterior distribution. Unfortunately, due to the existence of hidden variables in likelihood calculation, direct estimation of the hyper-parameters is hard. Various approximations have been investigated to solve the problem. For example, a variational Bayesian approach has been used to calculate a variational posterior as the approximation of the true posterior [120]. Another algorithm, the Quasi-Bayesian algorithm, has also been investigated for a similar purpose. The posterior distribution is assumed to be proportional to the exponentiation of the standard auxiliary function [59]. Once the posterior distribution is estimated, it is used to calculate the marginal likelihood of the test data for recognition.

The above discussions have assumed that the training data is limited, hence, a non-point estimate posterior is required to represent the parameter uncertainties. As the quantity of training data increases, the variance of the distribution of the HMM parameters will decrease. Given sufficient training data, the posterior distribution will tend to a Dirac delta function. In this case, the use of a point estimate of the parameters is justified. However, it is worth noting that there is another underlying assumption that all the training data and test data come from the same acoustic condition. Unfortunately, this is not true when building systems on non-homogeneous training data. In this case, simply training HMMs on the whole data set as if all data comes from a single homogeneous block may not yield the best classifier. It is therefore preferable to use the adaptive training technique. A detailed review will be given in chapter 3.

2.6 Recognition of Speech Using HMMs

The previous sections discussed how to train HMM parameters. This section will investigate the use of HMMs for *inference*, also known as *recognition* or *decoding*. As a statistical model, the general inference of HMMs follow the Bayesian rule. The recognised word sequence is the one that gives the highest likelihood given the observation sequence and the HMMs

$$\hat{\mathcal{H}} = \arg\max_{\mathcal{H}} P(\mathcal{H}|\mathbf{O},\mathcal{M}) \tag{2.83}$$

where $\mathcal{H}$ is a hypothesis word sequence, $\hat{\mathcal{H}}$ is the recognised word sequence, and $\mathbf{O}$ is the observation sequence to recognise. As the marginal likelihood $p(\mathbf{O})$ is irrelevant to the recognised hypothesis, by using Bayesian rule, equation (2.83) can be rewritten as

$$\hat{\mathcal{H}} = \arg\max_{\mathcal{H}} p(\mathbf{O}|\mathcal{H},\mathcal{M})P(\mathcal{H}) \tag{2.84}$$

where $p(\mathbf{O}|\mathcal{H},\mathcal{M})$ is the likelihood of the observation sequence given HMM parameters and a possible word sequence $\mathcal{H}$, referred to as *acoustic score*, which is calculated using the acoustic model $\mathcal{M}$. $P(\mathcal{H})$ is the prior probability of a word sequence, calculated using a *language model* and the value is referred to as *language score*. Therefore, the overall inference evidence is a combination of the two terms. The calculation of the inference evidence will be discussed in the following sections.

2.6.1 Language Modelling

The prior probability of a hypothesised sequence of K words, $\mathcal{H} = \{\mathcal{W}_1, \cdots, \mathcal{W}_K\}$, is given by the language model. The language model is a discrete probability and can be factorised into a product of conditional probabilities

$$P(\mathcal{H}) = \prod_{k=1}^{K} P(\mathcal{W}_k|\mathcal{W}_{k-1}, \cdots, \mathcal{W}_1) \tag{2.85}$$

where $\mathcal{W}_k$ is the k^{th} word of the sequence. The calculation of the probability of any word sequence using equation (2.85) requires calculating the probability of its full history. However, for LVCSR, the number of possible history word sequences is too large to explore due to the vocabulary size. It is hard to get a robust estimate of each possible word sequence. One possible solution is to restrict the length of the history required to calculate the conditional probability. *N-gram* language models use such a strategy and are the most widely used statistical language model in speech recognition. The assumption here is that it is enough to use a history of N words to calculate the probability, i.e.,

$$P(\mathcal{W}_k|\mathcal{W}_{k-1}, \cdots, \mathcal{W}_1) \approx P(\mathcal{W}_k|\mathcal{W}_{k-1}, \cdots, \mathcal{W}_{k-N+1}) \tag{2.86}$$

where N is the pre-determined size of word history. N is normally small, for example 3, which is referred to as a *tri-gram* language model. With this approximation, it is easy to get the ML estimate for N-gram by using the counts of word sequence with length N

$$P(\mathcal{W}_k|\mathcal{W}_{k-1}, \cdots, \mathcal{W}_{k-N+1}) = \frac{f(\mathcal{W}_k, \mathcal{W}_{k-1}, \cdots, \mathcal{W}_{k-N+1})}{\sum_{\mathcal{W}} f(\mathcal{W}, \mathcal{W}_{k-1}, \cdots, \mathcal{W}_{k-N+1})} \tag{2.87}$$

where $f(\mathcal{W}_k, \mathcal{W}_{k-1}, \cdots, \mathcal{W}_{k-N+1})$ denotes the frequency counts of the N-gram word sequence observed in the training data. Coverage of all possible N-grams with sufficient counts is required to get a robust estimate. However, this is still not practical for LVCSR, even if N is very small. Therefore, further smoothing approaches are used to obtain robust estimates. There are three main categories of the smoothing schemes:

- **Discounting**

  To handle unobserved N-grams, a certain amount of the overall probability mass is taken from the seen N-grams and allocated to the unseen N-grams. The portion of the allocated probability mass is controlled by a discounting factor. Commonly used discounting

approaches include Good-Turing discounting [45, 66], Witten-Bell discounting [125] and absolute discounting [84].

- **Back-off**

  The basic idea of back-off is to make use of shorter histories which can be estimated more robustly, rather than assigning probability mass to those unlikely N-grams in discounting approaches. The distributions with shorter history are referred to as back-off distributions. The probabilities of unseen N-grams are then taken from the back-off distributions with appropriate normalisation. The back-off strategy is often applied hierarchically in practice. For example, a 4-gram distribution may be backed-off to tri-gram, bi-gram and finally uni-gram distributions.

- **Deleted Interpolation**

  If an N-gram language model is not robust enough, it may be interpolated with the language models of shorter N-grams to construct a smoothed model. For example, uni-gram, bi-gram and tri-gram distributions may be interpolated to construct a more robust language model. The interpolation weights are tuned on some held-out data set. A similar interpolation strategy can also be used to combine several N-gram language models from different sources to build corpus or topic specific language models.

With the above smoothing techniques, language model probabilities $P(\mathcal{H})$ can be obtained and used in inference. Some implementation issues will be discussed in the below section 2.6.2.2.

2.6.2 Search and Decoding

The previous section discussed calculation of language score $P(\mathcal{H})$. This section will discuss how to calculate the acoustic score $p(\mathbf{O}|\mathcal{H})$ and will review some practical issues to be considered in recognition.

2.6.2.1 Forward-Backward Likelihood Calculation and Viterbi Decoding

In recognition, the acoustic score is calculated using the same formula as equation (2.7) in section 2.2.2. It is re-written here as

$$p(\mathbf{O}|\mathcal{H},\mathcal{M}) = \sum_{\boldsymbol{\omega}} p(\mathbf{O},\boldsymbol{\omega}|\mathcal{H},\mathcal{M}) \tag{2.88}$$

where $\boldsymbol{\omega}$ is the hidden state sequence. As discussed in section 2.3.2, this likelihood can be efficiently calculated using the forward-backward algorithm. However, this algorithm does not provide the best path (state sequence). In many applications, especially continuous speech recognition, it is desirable to find the best path. A widely used approach for LVCSR is to find the state sequence that has the highest probability to generate the observation sequences. This is the Viterbi algorithm [116]. Here the maximum likelihood of the observation sequence given

one hidden state sequence is used to approximate the marginal likelihood over all possible state sequences

$$p(\mathbf{O}|\mathcal{H},\mathcal{M}) \approx \max_{\boldsymbol{\omega}} p(\mathbf{O},\boldsymbol{\omega}|\mathcal{H},\mathcal{M}) \tag{2.89}$$

With this approximation and the conditional independence assumption of HMM, a recursion can be derived to calculate the maximum likelihood of the partial observation sequence, which is referred to as the Viterbi algorithm [116]. $\phi_j(t)$ is introduced to denote the maximum likelihood of the partial observation sequence from $\mathbf{o}_1$ to $\mathbf{o}_t$ and being in state j at time t. It can be recursively calculated by

$$\phi_j(t) = \max_i \{\phi_i(t-1)a_{ij}\}\, b_j(\mathbf{o}_t) \tag{2.90}$$

where a_{ij} is the transition probability from state i to state j, and $b_j(\mathbf{o}_t)$ is the state output distribution at state j. For $1 < j < N$, where N is the total number of states of an HMM,

$$\phi_1(1) = 1 \tag{2.91}$$

$$\phi_j(1) = a_{1j}b_j(\mathbf{o}_1) \tag{2.92}$$

Then, the maximum likelihood of the whole observation sequence is given by

$$p(\mathbf{O}|\mathcal{H},\mathcal{M}) \approx \phi_N(T) = \max_i \{(\phi_i(T)a_{iN})\} \tag{2.93}$$

where T is the length of the observation sequence.

The above Viterbi algorithm is designed for isolated word recognition. For LVCSR, because there is normally a large number of possible word sequences, it is not practical to construct one single composite HMM for each word sequence. In this case a *token passing* algorithm [133] is often used as an extension to the standard Viterbi algorithm. Each state has one or more tokens associated with each time instance. The *token* contains the likelihood of the partial path,$\phi_j(t)$, and a pointer to the history of the HMM sequence. At each time instance, these tokens are updated and propagated forward for each state within the models. The most likely token calculated at the exit state of each HMM is propagated to all connected HMMs and the history of the HMM sequence for that token is updated. At the word boundaries, the language model probability is added during the propagation. At the end of the whole observation sequence, the token with the highest value of $\phi_j(t)$ is traced back to give the most likely sequence of HMMs.

Even with the token passing algorithm, the searching cost is still high for most LVCSR systems if all possible tokens are propagated. To further reduce the computational cost, a *pruning* technique is widely used. In this scheme, the tokens with the value of $\phi_j(t)$ falling below a given threshold are pruned, or removed. The most commonly used threshold is a fixed likelihood value below the current most likely path. This value is referred to as *beam width* as it sets the minimum "width" between the current most likely path and the paths to be deleted. Pruning can also be performed at the end of words when the language model is applied. Though the pruning technique can dramatically reduce the computational cost, it introduces *search errors*.

Part of the real most likely path could be pruned at the early stage of searching if a tight beam width is used. Therefore, the choice of beam width is a trade off between saving computational cost and reducing search errors.

2.6.2.2 Practical Issues in Recognition

In speech recognition, there is often a significant mismatch between the dynamic range of the language scores and the acoustic scores. The dynamic range of the acoustic likelihood can be excessively high, which makes the effect of language model relatively small. To handle this problem, the language scores are often scaled. The scaling factor balances the information of the language model and acoustic model and may be experimentally set. Another issue is the use of a word insertion penalty. Word errors in recognition are of three types: substitutions, deletions and insertions. To minimise the total number of errors, a fixed word insertion penalty is commonly used to balance the insertions to the deletions. The value of insertion penalty is often fixed for a particular task. With these techniques, the recognised word sequence is practically determined by

$$\hat{\mathcal{H}} = \arg\max_{\mathcal{H}} \{\log p(\mathbf{O}|\mathcal{H}, \mathcal{M}) + \alpha \log P(\mathcal{H}) + \beta L_{\mathcal{H}}\} \tag{2.94}$$

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

2.6.3 Bayesian Inference with Parameter Distribution

The Viterbi decoding algorithm described in section 2.6.2.1 is based on the conditional independence assumption of HMMs given the point estimate of the model parameters. This section will discuss the inference algorithm with a parameter distribution, where the conditional independence assumption is not valid. As described in the Bayesian training section 2.5, a posterior distribution of model parameters may be obtained from Bayesian training. With the posterior distribution, *Bayesian inference* need to be used. The Bayesian inference criterion may be written as

$$\hat{\mathcal{H}} = \arg\max_{\mathcal{H}} p(\mathbf{O}|\mathcal{H}) P(\mathcal{H}) \tag{2.95}$$

where $\mathbf{O}$ is the observation sequence of the test data, $\mathcal{H}$ is a possible hypothesis sequence associated with $\mathbf{O}$, $p(\mathbf{O}|\mathcal{H})$ is the marginal likelihood over all possible model parameters

$$\begin{aligned} p(\mathbf{O}|\mathcal{H}) &= \int_{\mathcal{M}} p(\mathbf{O}|\mathcal{H}, \mathcal{M}) p(\mathcal{M}|\mathbf{O}_{\mathtt{trn}}, \mathcal{H}_{\mathtt{trn}})\, d\mathcal{M} \\ &= \int_{\mathcal{M}} \left(\sum_{\boldsymbol{\omega}} P(\boldsymbol{\omega}|\mathcal{H}, \mathcal{M}) \prod_{t} b_{\omega_t}(\mathbf{o}_t) \right) p(\mathcal{M}|\mathbf{O}_{\mathtt{trn}}, \mathcal{H}_{\mathtt{trn}})\, d\mathcal{M} \end{aligned} \tag{2.96}$$

where $p(\mathcal{M}|\mathbf{O}_{\mathtt{trn}}, \mathcal{H}_{\mathtt{trn}})$ is the posterior distribution of the model parameters with the hyperparameter estimated on training data $\mathbf{O}_{\mathtt{trn}}$ and $\mathcal{H}_{\mathtt{trn}}$. This inference criterion is also referred to as *Bayesian predictive classification* [57].

It is worth noting that the integral in equation (2.96) is over the *whole* observation sequence, which means the model parameters are constrained to be fixed over the whole utterance. With this model parameter constraint, and treating model parameters as random variables, the state output distribution at each time instance is no longer conditionally independent given the current state. Hence, the Viterbi and token passing algorithm are not applicable unless some additional approximations are used. Instead, the marginal likelihood $p(\mathbf{O}|\mathcal{H})$ of *every* possible hypothesis sequence needs to be explicitly calculated. For LVCSR, as the possible number of hypothesis sequences is very large, it is necessary to select a small set of "reasonable" candidate hypothesis for use in the inference process. One approach to generate this small set of possible sequences is to run a standard recognition system and take the top N hypotheses, referred to as *N-Best list*. The best hypothesis is then selected from the N-Best list by comparing the inference evidence. This process is known as *N-Best rescoring* [103]. In this process, the calculation of the marginal likelihood $p(\mathbf{O}|\mathcal{H})$ is required. As the Bayesian integral in equation (2.96) is intractable due to the coupling of model parameters and state sequence, approximations are required to calculate the marginal likelihood $p(\mathbf{O}|\mathcal{H})$.

One approximation that can be used is to relax the constraint that the HMM parameters are constant over all frames of the utterance. Instead, the parameters are allowed to vary from one time instance to another time. This approximation is referred to as a *frame-independent* assumption in this thesis [37, 138]. Mathematically, this approximation means the integral in equation (2.96) can be performed at each time instance rather than over the whole observation sequence, i.e.

$$p(\mathbf{O}|\mathcal{H}) \approx \sum_{\boldsymbol{\omega}} P(\boldsymbol{\omega}|\mathcal{H},\mathcal{M}) \prod_{t} \bar{b}_{\omega_t}(\mathbf{o}_t) \tag{2.97}$$

The distribution

$$\bar{b}_j(\mathbf{o}) = \int_{\mathcal{M}} b_j(\mathbf{o}) p(\mathcal{M}|\mathbf{O}_{\mathtt{trn}}, \mathcal{H}_{\mathtt{trn}})\, d\mathcal{M} \tag{2.98}$$

is sometimes referred to as *Bayesian predictive density* [62], which is the averaged state output distribution at state j with the parameters from $\mathcal{M}$. With the appropriate form of the parameter posterior distribution, this frame-level integral is tractable and the Bayesian predictive density has a closed-form solution. A constrained uniform distribution was used to model the uncertainty of the mean vector in each Gaussian component and led to a tractable integral in [62]. Normal-Wishart distribution on Gaussian mean and covariance matrices was also used to yield a tractable Bayesian predictive density for recognition in [120]. With the frame-independent assumption, the *predictive distribution* is used instead of the original state output distribution in inference and the conditional independence assumption of HMM does not change. Hence, the Viterbi algorithm described in section 2.6.2.1 can still be used, which is the advantage of this approximation[11]. However, with this assumption, it is hard to state how close the approximation

[11] A modified frame-synchronous Viterbi algorithm with the predictive density was also reported in [62]. In this algorithm, at each time instance, the likelihoods of all previous partial hypothesis paths are re-calculated with some further approximations.

is to the real likelihood.

Another approach to approximate the Bayesian integral in marginal likelihood calculation is to use a *Laplacian approximation* or *normal approximation* [108, 57]. In this approach, the integral in equation (2.96) is approximated by

$$p(\mathbf{O}|\mathcal{H}) \approx p(\mathbf{O}|\hat{\mathcal{M}}_{\texttt{MAP}}, \mathcal{H}) p(\hat{\mathcal{M}}_{\texttt{MAP}}|\mathbf{O}_{\texttt{trn}}, \mathcal{H}_{\texttt{trn}}) (2\pi)^{\frac{D}{2}} |\boldsymbol{\Sigma}_{\texttt{MAP}}|^{\frac{1}{2}} \tag{2.99}$$

where D is the total number of parameters of $\mathcal{M}$, $\hat{\mathcal{M}}_{\texttt{MAP}}$ is the estimate which maximises the MAP criterion

$$\hat{\mathcal{M}}_{\texttt{MAP}} = \arg\max_{\mathcal{M}} p(\mathbf{O}|\mathcal{M}, \mathcal{H}) p(\mathcal{M}|\mathbf{O}_{\texttt{trn}}, \mathcal{H}_{\texttt{trn}}) \tag{2.100}$$

$\boldsymbol{\Sigma}_{\texttt{MAP}} = (-\mathbf{V})^{-1}$ and $\mathbf{V}$ is the Hessian matrix, or second derivatives, of $\log\left(p(\mathbf{O}|\mathcal{M}, \mathcal{H}) p(\mathcal{M}|\mathbf{O}_{\texttt{trn}}, \mathcal{H}_{\texttt{trn}})\right)$ evaluated at $\hat{\mathcal{M}}_{\texttt{MAP}}$. From equation (2.99), the basic idea of Laplacian approximation is to make a local Gaussian approximation of the likelihood with respect to the model parameters. The Gaussian mean is the MAP estimate $\hat{\mathcal{M}}_{\texttt{MAP}}$, which can be estimated using the EM algorithm [42, 21]. The Gaussian covariance matrix is related to the Hessian matrix $\mathbf{V}$ evaluated at $\hat{\mathcal{M}}_{\texttt{MAP}}$. The most difficult part in Laplacian approximation is the calculation of the Hessian matrix $\mathbf{V}$. Although the Hessian matrix may be approximated using a Quasi-Bayesian (QB) algorithm [59, 58], it is computationally expensive. This is why this approximation was only reported for isolated words recognition [57] rather than LVCSR tasks. Due to the computation issue, the scheme is not further discussed in this thesis.

In addition to the above, a broad class of approaches are based on lower bound approximating the intractable Bayesian integral in equation (2.96). One simple and widely used scheme of this class is to just use a point estimate, for example the MAP estimate $\hat{\mathcal{M}}_{\texttt{MAP}}$ or ML estimate $\hat{\mathcal{M}}_{\texttt{ML}}$, to approximate the parameter posterior distribution in recognition. In this case, the recognition process is the same as the standard process.

Though the above approximations have been used in various Bayesian inference problems, they have not been investigated in a consistent framework with Bayesian training. In this work, such a consistent Bayesian framework is proposed and applied to adaptive training and adaptive inference. A more detailed discussion will be given in chapter 5.

2.7 Summary

This chapter reviews the fundamental of speech recognition with hidden Markov models (HMMs). Feature extraction approaches, specifically MFCC and PLP, are discussed first. Hidden Markov models (HMMs), the most successful acoustic models, are then introduced. The maximum likelihood (ML) training of HMM parameters and the expectation maximisation (EM) algorithm are presented in detail. The acoustic units used in practical speech recognition and the parameter tying technique are also discussed. To overcome the limitation of ML training, discriminative training and Bayesian training of HMM parameters are proposed. Discriminative training aims

to find parameters directly reducing the recognition error, which addresses the incorrect modelling problem of HMMs when using ML training. This chapter reviews the approach of using weak-sense auxiliary function to optimise discriminative training criteria. Another limitation of the ML criterion is the sufficient training data assumption. To solve this problem, Bayesian training of HMMs assumes that the parameters are random variables and use a prior distribution of the parameters. This is to yield a robust parameter estimate or a parameter posterior distribution when the training data is limited. As the Bayesian integral is intractable, approximations are required during training. After the discussion of HMMs training, a review of the recognition process is given. The use of N-gram language models is presented first. With an acoustic model and a language model, the recognition process is a search problem, which is to find the appropriate word sequence with the maximum inference evidence. An efficient and widely used searching algorithm, Viterbi decoding, is discussed in detail. The scaling of language model probability and the introduction of word insertion penalty is also discussed. Finally, recognition with a parameter posterior distribution rather than a point estimate is discussed.

3

Adaptation and Adaptive Training

The training approaches described in chapter 2 use an assumption that both training data and test data come from the same acoustic condition. However, this is not always true. The acoustic mismatch may significantly degrade the recognition performance compared to systems built in matched conditions. To compensate the mismatch of acoustic conditions between test and training data, *adaptation* techniques are often used. Adaptation aims to improve performance either by normalising the features or by tailoring the model toward a particular test acoustic condition. This is found to significantly improve the performance of speech recognition systems on test data with diverse acoustic conditions. Recently, there has been interest in building systems on *found* data where various different acoustic conditions exist. To deal with this acoustic non-homogeneity in training data, *adaptive training* has been proposed. The basic idea is to use adaptation transformation in the acoustic model training. Two sets of parameters are estimated during training: a *canonical* model to represent the speech variability and a set of transforms to represent different acoustic conditions. With adaptive training techniques, compact systems can be effectively built on non-homogeneous data and further improve the recognition performance after adaptation. This chapter will review the framework and standard maximum likelihood (ML) schemes for adaptation and adaptive training.

3.1 Adaptation in Speech Recognition

Although speaker-independent (SI) systems trained using the approach described in chapter 2 can achieve good performance, speaker dependent (SD) systems can obtain an average WER that is a factor of two or three lower than speaker-independent (SI) systems if both systems use the same amount of data [126]. This shows the importance of reducing the speaker mismatch between training and test data. Speaker *adaptation* was originally motivated to compensate for the speaker mismatch between test and training data [56]. It aims to normalise the features or to modify HMM parameters using a small amount of test speaker-specific data so that the resulting system has SD-like performance. It has therefore attracted much attention in the speech community. Speaker adaptation techniques have been extended to deal with other non-

speech variabilities as well. This section will review the homogeneity assumption and modes of adaptation commonly used in speech recognition.

3.1.1 Non-speech Variabilities and Homogeneity

In addition to changes in speakers, there are a number of other acoustic conditions that alter the acoustic signals and result in corrupted features, such as environmental noise or channel difference. The variabilities of observations introduced by these acoustic conditions (including speaker changes) are independent of the inherent variability of the uttered words, hence are generally referred to as *non-speech variabilities*. Though specific approaches may be used to handle the non-speech variabilities caused by environmental noise etc., speaker adaptation techniques have also been used to deal with this problem [37]. When using speaker adaptation schemes for generic acoustic conditions, the algorithm design and implementation is the same. Therefore, in this thesis, unless explicitly stated, the term *acoustic condition* is used instead of the term *speaker* when describing adaptation techniques.

One important assumption in adaptation is that the non-speech variabilities of a particular acoustic condition have the same statistical property. This is referred to as *homogeneity* of the acoustic condition. Due to the homogeneity constraint, adaptation is performed for each separate *homogeneous* data block respectively. This homogeneity constraint may be formulated using a *Dynamic Bayesian Network* (DBN)[1]. The DBN in figure 3.1 shows the statistical dependencies in adaptation on HMMs.

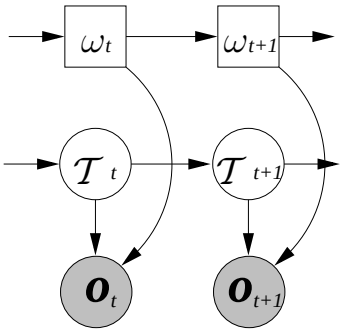


Figure 3.1 *Dynamic Bayesian network for adaptation on HMMs*

Here, ω_t represents the hidden state at time t, $\mathbf{o}_t$ is the observation vector, and $\mathcal{T}_t$ is the transform used for adaptation. With the homogeneity assumption, the transform in figure 3.1 is constrained to be constant over all frames within one homogeneous data block, i.e., $\mathcal{T}_t = \mathcal{T}_{t+1}$. Hence, the observation at each time instance is not only dependent on the state at that time instance, but also the adaptation transform associated with the homogeneous block. Though

[1] A DBN is a graph that shows statistical dependencies [19]. In DBNs, a circle represents a continuous variable, a square represents a discrete variable, blank ones represent observable variables, and shaded ones represent unobservable variables. Lack of an arrow from A to B indicates that B is conditionally independent of A.

standard adaptation and adaptive training approaches all adopt this block homogeneity assumption, there are few publications discussing adaptation using the strict Bayesian interpretation as in figure 3.1 in detail. For some previous discussions, refer to [38]. This work will give a more detailed discussion of the Bayesian interpretation of the homogeneity constraint and its effects on adaptive training and adaptive inference in chapter 5.

In practice, the test data is normally non-homogeneous. Hence, to do adaptation, it is necessary to first partition the incoming audio data stream into homogeneous blocks. For many tasks, such as broadcast news recognition, no information about homogeneous blocks is available beforehand. Therefore, automatic partitioning of the test data is normally required before adaptation and recognition. This is generally done in two separate stages First, the audio data is split into small homogeneous *segments*. This is referred to as *segmentation*. In this stage, noise (including music if any) and long silence are removed and the length of the segments may also be constrained to reflect any limitation in the recognition. The current widely used segmentation approach is to construct several broad class models, such as speech, silence, music and speech with music, and then to use them to classify the audio data into small segments [109, 105]. Second, the segments are clustered together into homogeneous *blocks* where all segments are assumed to have the same acoustic condition. These blocks are sufficiently large for robust adaptation. This is referred to as the *clustering* stage. These homogeneous blocks are then used for adaptation. Then agglomerative clustering is normally performed using a specific distance measure and stopping criterion, such as likelihood ratio with a penalised likelihood (Bayesian information criterion, BIC) [105]. In this work, the task of interest is conversational telephone speech (CTS). In CTS, the speech has been split into speaker sides, hence, there is no need to do clustering. However, segmentation is still required to partition the speech of each side into short utterances. As there is no music in CTS, only speech and silence models are built to classify the audio data into small segments [109].

Once the test data is split into homogeneous data blocks, a two-step process is usually used for recognition within the adaptation framework. First, features are normalised or HMMs are transformed on a *block basis*, which is often referred to as *adaptation*. This means that one distinct set of modified HMMs or normalised features is constructed for *each* homogeneous block. Second, the adapted model or normalised features are used for recognition on the particular data block. This is the standard framework, which is reviewed in this chapter. Though the two-step adaptation/recognition framework has been widely used, it may be viewed from a general viewpoint of adaptive inference. The integrated adaptive inference process will be discussed within a Bayesian framework in chapter 5.

3.1.2 Modes of Adaptation

Adaptation operates in a number of *modes*. In terms of availability of transcribed supervision data, it operates in either supervised or unsupervised mode. In terms of when the supervision data becomes available, it operates in either batch or incremental mode. It is worth noting that

the selection of adaptation mode is highly dependent on the application.

3.1.2.1 Supervised and Unsupervised Modes

The process of adaptation is to normalise features or to modify model parameters with a small amount of test domain specific data, referred to as *supervision data* or *adaptation data*[2]. In addition to the observations of the supervision data, many adaptation techniques require the corresponding hypothesis as well. There are two *modes* of adaptation in terms of the availability of the supervision data.

If both the observations and associated (word level) transcriptions of the supervision data are known, adaptation operates in a *supervised* mode. In this case, the supervision data can be regarded as a small amount of additional training data. The quality of adaptation depends on the amount of supervision data.

If the correct transcriptions of the supervision data are not available, adaptation operates in *unsupervised* mode. The solution to unsupervised adaptation is to recognise the supervision data multiple times and use the final recognised hypothesis instead of the real correct transcriptions to do adaptation. In this case, the quality of the adaptation is not only dependent on the amount of data, but also dependent on the quality of the recognised hypothesis. The adapted model parameters or normalised features are likely to be tuned to the error-full hypothesis, which may degrade the final recognition performance. In practice, there is a special scenario where there is no supervision data available at all. Unsupervised adaptation has to be performed using the test data to be recognised, which is sometimes referred to as *self adaptation* [38]. In this scenario, the hypothesis bias problem is even more serious as the adaptation data is the test data itself. In this thesis, self adaptation is the main concern. Unless otherwise specified, the term "*unsupervised adaptation*" in this thesis only refers to "*self adaptation*".

3.1.2.2 Batch and Incremental Modes

Adaptation can also operate in either *batch* or *incremental* modes. *Batch* adaptation, also referred to as *static* adaptation, requires all the adaptation data being available before adaptation starts. Batch adaptation is widely used in off-line speech recognition. In some other scenarios, for example on-line adaptation, the adaptation data does not come in one block. It is not possible to perform adaptation after all data become available. Adaptation has to operate when only part of the adaptation data becomes available and continue as more data is received. The recognition results are also produced causally. This mode is referred to as *incremental* mode. In incremental mode, different strategies can be used to propagate adaptation information. This will be discussed in detail in section 5.4.

[2] Some simple adaptation techniques do not require supervision, for example, Cepstral mean normalisation (CMN). Those techniques always run in the unsupervised mode as discussed in later sections.

3.2 Adaptation Schemes

As mentioned before, adaptation can be performed either on model parameters or on features[3]. The procedure of model based adaptation/recognition is to tune a well-trained acoustic model to a particular test domain using a small amount of test domain specific data and then to recognise the test data associated with that particular domain using the modified acoustic model. This section will discuss standard schemes on how to *adapt*, or modify, HMMs toward a specific test acoustic condition. Although feature normalisation may also be used to compensate the acoustic mismatch, it is normally done for both training and test data. Hence, it is discussed in the adaptive training section 3.3.2.

3.2.1 Maximum a Posteriori (MAP)

Given the adaptation data, a straightforward method to modify the acoustic model is to perform more ML re-training iterations. However, as the amount of supervision data is often very small, the ML criterion easily leads to over-training of the HMM parameters. To overcome this problem, The *maximum a posteriori* (MAP) criterion was proposed [42]. Here, rather than optimising the likelihood criterion, the posterior distribution of HMM parameters is maximised. The adapted model parameters are obtained by[4]

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

where $\mathbf{O}$ and $\mathcal{H}$ are the observation sequence and associated transcription of the supervision data from one homogeneous block, $p(\mathcal{M}|\Phi)$ is the prior distribution over HMM parameters. The incorporation of the prior $p(\mathcal{M}|\Phi)$ means that HMM parameters are less likely to be overtrained even if there is only very limited adaptation data. To obtain the model parameter estimate using the MAP criterion, an iterative EM algorithm similar to ML training is used. It can be shown that the MAP auxiliary function is the ML auxiliary function in equation (2.40) plus the prior term [21, 42]. By ignoring the constant terms independent of Gaussian parameters, the MAP auxiliary function for Gaussian components update can be written as

$$\mathcal{Q}_{\texttt{MAP}}(\mathcal{M};\hat{\mathcal{M}}) = \log p(\mathcal{M}|\Phi) - \frac{1}{2}\sum_{t,m}\gamma_m^{\texttt{ML}}(t)\left\{\log|\boldsymbol{\Sigma}^{(m)}| + \left(\mathbf{o}_t - \boldsymbol{\mu}^{(m)}\right)^T\boldsymbol{\Sigma}^{(m)-1}\left(\mathbf{o}_t - \boldsymbol{\mu}^{(m)}\right)\right\} \tag{3.2}$$

where $\hat{\mathcal{M}}$ is the current estimate of the HMMs, and $\gamma_m^{\texttt{ML}}(t)$ is the ML posterior occupancy of component m, calculated using the forward backward algorithm with $\hat{\mathcal{M}}$.

In mathematics, the above MAP estimate is a point estimate version of Bayesian training of HMMs described in section 2.5. However, it is worth noting that in adaptation, a distinct MAP estimate of HMMs is obtained for each homogeneous block respectively. This is different

[3] In the rest of this chapter, the term "normalisation" refers to adaptation on features, while the term "adaptation" usually refers to model based adaptation.

[4] As adaptation is always performed within one homogeneous block, the index s to denote homogeneous block is omitted in section 3.2.

from the concept of standard Bayesian training, where all training data are regarded as a single block. As discussed in section 2.5, the MAP criterion will yield mathematically simple forms if the prior distribution $p(\mathcal{M}|\Phi)$ is a *conjugate prior* to the likelihood of the observations. However, for GMMs, the state output distribution normally used in HMMs, a finite dimensional conjugate prior does not exist. An alternative approach is to assume independence between the component weights and the parameters of individual components. Then conjugate priors to the likelihood of the complete data set may be separately defined for component weights and parameters of the individual Gaussians [42][5]. With these conjugate priors, closed-form re-estimation formulae for MAP estimates can be derived from the auxiliary function in equation (3.2). For a Gaussian component m, with $p(\mathcal{M}|\Phi)$ being a Normal-Wishart distribution in equation (2.69), the MAP estimate of mean vector can be shown to be[6]

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

where $\tilde{\boldsymbol{\mu}}^{(m)}$ is the prior mean vector from $p(\mathcal{M}|\Phi)$. It is normally set as the corresponding mean vector of a robustly trained model, such as a speaker-independent model set. τ is a meta-parameter which balances the ML estimate and the prior.

From equation (3.3), as the amount of adaptation data increases towards infinity, the estimate converges to the ML estimate. While for limited supervision data case, the prior mean vectors reduce sensitivity to the supervision data and lead to robust estimates. This is a major advantage of MAP adaptation. One limitation of MAP is that it can only update those Gaussian components that are observed in supervision data. The other components are not altered. As the number of Gaussian components in LVCSR system is normally very large, standard MAP adaptation will require a considerable amount of supervision data to update all parameters. Hence it is very slow.

To solve this problem, various extensions to MAP have been proposed. For example, the Regression based Model Prediction (RMP) [1] finds linear regression relationships between HMM parameters and uses the relationships to update rarely observed or unobserved parameters based on well-adapted parameters. Structured MAP (SMAP) [104] organises all Gaussian components into a tree structure and applies MAP adaptation using a top-down strategy from the root node (containing all components). In addition to extensions within the MAP framework, alternative approaches have been also proposed to get rapid adaptation of all Gaussian parameters, which will be discussed in the next section.

3.2.2 Linear Transform Based Adaptation

Linear transform based adaptation is a widely used alternative to MAP when there is limited adaptation data. The idea is to estimate a test-domain specific linear transform for the means

[5]The conjugate prior of Gaussian parameters in HMMs is a Normal-Wishart distribution [20] as defined in equation (2.69). The conjugate prior of component weights is a Dirichlet distribution [63].

[6]For details on MAP update formulae of other HMM parameters refer to [42].

and/or covariance matrices of the Gaussian components. This scheme has the advantage that the same transform can be shared between a large number of Gaussians. Although in this case, this kind of adaptation does not guarantee to give the ML estimate of Gaussian parameters with a large amount of supervision data, the sharing allows adaptation of all Gaussians components with a small amount of supervision data. In standard schemes, the transforms are estimated using the ML criterion given a set of HMMs. Hence, similar to equation (2.40), the generic auxiliary function for updating transform $\mathcal{T}$ can be written as

$$\mathcal{Q}_{\texttt{ML}}(\mathcal{T};\hat{\mathcal{T}},\hat{\mathcal{M}}) = \left\langle \log p(\mathbf{O},\boldsymbol{\theta}|\mathcal{H},\hat{\mathcal{M}},\mathcal{T}) \right\rangle_{P(\boldsymbol{\theta}|\mathbf{O},\mathcal{H},\hat{\mathcal{M}},\hat{\mathcal{T}})} \tag{3.4}$$

where $\mathbf{O}$ is the observation sequence of a homogeneous data block, $\mathcal{H}$ is the associated transcription, $\boldsymbol{\theta}$ is the hidden component sequence, $\hat{\mathcal{M}}$ is the model that the transform is based on, $\hat{\mathcal{T}}$ is the current estimate of the transform parameters. The exact form of auxiliary function depends on the form of linear transform. This section will discuss some common forms including unconstrained maximum likelihood linear regression (MLLR), variance MLLR and constrained MLLR (CMLLR). All Gaussian components are assumed to share one global transform in the discussion. The use of multiple transforms and regression base class will be discussed in section 3.2.4.

3.2.2.1 Maximum Likelihood Linear Regression (MLLR)

Maximum likelihood linear regression (MLLR) uses the ML criterion to estimate a linear transform to adapt Gaussian parameters of HMMs. It was originally proposed to adapt mean vectors [74] and extended to variance adaptation later [41, 34]. To avoid confusion, the term "MLLR" will only refer to mean based linear transforms in this work. In MLLR, the mean of Gaussian component m is adapted to a particular acoustic condition by

$$\hat{\boldsymbol{\mu}}^{(m)} = \mathbf{A}\boldsymbol{\mu}^{(m)} + \mathbf{b} = \mathbf{W}\boldsymbol{\xi}^{(m)} \tag{3.5}$$

where $\hat{\boldsymbol{\mu}}^{(m)}$ is the adapted mean of component m for the target acoustic condition, $\boldsymbol{\xi}^{(m)} = [\boldsymbol{\mu}^{(m)T} \;\; 1]^T$ is the extended mean vector, and $\mathbf{W} = [\mathbf{A} \;\; \mathbf{b}]$ is the extended linear transform. In a linear transform, $\mathbf{A}$ can be either a full or a block-diagonal matrix according to the amount of supervision data available [83]. It is worth emphasising that the transform $\mathbf{W}$ is associated with a particular test acoustic condition and is distinct for each homogeneous data block.

The estimation of the transform is based on a set of pre-trained HMMs $\hat{\mathcal{M}}$ and the current transform estimate $\hat{\mathcal{T}}$. Ignoring the constants independent of the transform, the generic auxiliary function in equation (3.4) for MLLR transform updates can be explicitly written as [74]

$$\mathcal{Q}_{\texttt{ML}}(\mathcal{T};\hat{\mathcal{T}},\hat{\mathcal{M}}) = -\frac{1}{2}\sum_{t,m}\gamma_m^{\texttt{ML}}(t)\left(\mathbf{o}_t - \mathbf{W}\boldsymbol{\xi}^{(m)}\right)^T \boldsymbol{\Sigma}^{(m)-1}\left(\mathbf{o}_t - \mathbf{W}\boldsymbol{\xi}^{(m)}\right) \tag{3.6}$$

where $\mathcal{T}$ is now the MLLR transform $\mathbf{W}$, $\gamma_m^{\texttt{ML}}(t)$ is the posterior occupancy of component m calculated using the forward-backward algorithm with HMMs adapted by the current transform estimate. In this work, the Gaussian covariance matrices are assumed to be diagonal. This will greatly simplify the estimation of MLLR transforms. Differentiating the auxiliary function

in equation (3.6), with respect to $\mathbf{W}$ and equating to zero yields the ML estimate. Let $\mathbf{W} = [\mathbf{w}_1, \cdots, \mathbf{w}_D]^T$, where $\mathbf{w}_d^T$ is the d^{th} row of $\mathbf{W}$, D is the dimension number, the ML estimate of the d^{th} row is given by

$$\hat{\mathbf{w}}_d = \mathbf{G}_{\mathtt{ML},d}^{-1}\mathbf{k}_{\mathtt{ML},d} \tag{3.7}$$

where the sufficient statistics for the d^{th} row are given by

$$\mathbf{G}_{\mathtt{ML},d} = \sum_{t,m} \frac{\gamma_m^{\mathtt{ML}}(t)}{\sigma_{dd}^{(m)}} \boldsymbol{\xi}^{(m)} \boldsymbol{\xi}^{(m)T} \tag{3.8}$$

$$\mathbf{k}_{\mathtt{ML},d} = \sum_{t,m} \frac{\gamma_m^{\mathtt{ML}}(t) o_{t,d}}{\sigma_{dd}^{(m)}} \boldsymbol{\xi}^{(m)} \tag{3.9}$$

where $o_{t,d}$ is the d^{th} element of observation vector $\mathbf{o}_t$, $\sigma_{dd}^{(m)}$ is the d^{th} diagonal element of $\boldsymbol{\Sigma}^{(m)}$.

The MLLR transform estimation is also an iterative process. A new transform is estimated by making use of the current transform until convergence. The final transform $\mathbf{W}$ is then used to adapt the canonical model for recognition. This iterative estimation process requires initialisation. An identity transform is normally used as the initial transform. This also applies to other linear transform based adaptation schemes.

3.2.2.2 Variance MLLR

The previous section describes the use of linear transforms to adapt mean vectors. The covariance matrix of each component can also be adapted using linear transforms. This is referred to as *variance MLLR*. The covariance matrix may be adapted by [41, 34]

$$\hat{\boldsymbol{\Sigma}}^{(m)} = \mathbf{L}^{(m)T}\mathbf{H}\mathbf{L}^{(m)} \tag{3.10}$$

where $\mathbf{H}$ is the linear transform to adapt variance matrices, $\mathbf{L}^{(m)}$ is the inverse of the Choleski factor of $\boldsymbol{\Sigma}^{(m)-1}$, i.e. $\mathbf{L}^{(m)} = \mathbf{C}^{(m)-1}$, where

$$\boldsymbol{\Sigma}^{(m)-1} = \mathbf{C}^{(m)}\mathbf{C}^{(m)T} \tag{3.11}$$

A closed-form solution for estimating transform $\mathbf{H}$ using the EM algorithm can then be derived. For exact formulae, refer to [41].

A disadvantage of the above form of variance adaptation is the high computational cost. From equation (3.10), the covariance matrices after adaptation will be full rank matrices which results in increased computational load in likelihood calculation. To solve this problem, an alternative form of variance MLLR was proposed as [34]:

$$\hat{\boldsymbol{\Sigma}}^{(m)} = \mathbf{H}\boldsymbol{\Sigma}^{(m)}\mathbf{H}^T \tag{3.12}$$

where $\mathbf{H}$ is again the variance transform. A closed-form solution for estimating $\mathbf{H}$ using EM algorithm has also been derived in [34]. This form has the advantage that the likelihood can

be efficiently calculated. It can be shown that the log-likelihood of an observation $\mathbf{o}_t$ given a Gaussian component with the adapted covariance matrix is

$$\log \mathcal{N}(\mathbf{o}_t; \boldsymbol{\mu}^{(m)}, \hat{\boldsymbol{\Sigma}}^{(m)}) = \log \left(\mathcal{N} \left(\mathbf{H}^{-1}\mathbf{o}_t; \mathbf{H}^{-1}\boldsymbol{\mu}^{(m)}, \boldsymbol{\Sigma}^{(m)} \right) \right) - \log |\mathbf{H}| \tag{3.13}$$

When the original covariance matrix $\boldsymbol{\Sigma}^{(m)}$ is diagonal, with equation (3.13), the likelihood can be calculated by appropriately modifying the mean and observations, which is much more efficient than the calculation using a full covariance matrix. An EM algorithm has also been derived for this type of variance MLLR in [34].

3.2.2.3 Constrained MLLR

The mean and variance MLLR transforms described in the previous two sections can be simultaneously applied to both mean vectors and covariance matrices [34]. As the two types of transforms are estimated separately, the computation cost is high. An alternative scheme to adapt both mean vectors and covariance matrices is to use constrained linear transforms [22, 34]. Here the linear transform applied to the covariance matrix must correspond to the transform applied to the mean vector. This is referred to as *constrained MLLR*

$$\hat{\boldsymbol{\mu}}^{(m)} = \mathbf{A}'\boldsymbol{\mu}^{(m)} - \mathbf{b}' \tag{3.14}$$

$$\hat{\boldsymbol{\Sigma}}^{(m)} = \mathbf{A}'\boldsymbol{\Sigma}^{(m)}\mathbf{A}'^{T} \tag{3.15}$$

where $\mathbf{A}'$ is the constrained linear transform, $\mathbf{b}'$ is the bias on the mean vector, and $\boldsymbol{\mu}^{(m)}$ and $\boldsymbol{\Sigma}^{(m)}$ are the original Gaussian parameters. It can be shown that the above constrained MLLR is equivalent to a feature transform with an additional normalisation term in likelihood calculation [34]. The log likelihood of an observation $\mathbf{o}_t$ given an adapted Gaussian component m can be calculated by

$$\log \mathcal{N}(\mathbf{o}_t; \hat{\boldsymbol{\mu}}^{(m)}, \hat{\boldsymbol{\Sigma}}^{(m)}) = \log \left(\mathcal{N} \left(\hat{\mathbf{o}}_t; \boldsymbol{\mu}^{(m)}, \boldsymbol{\Sigma}^{(m)} \right) \right) + \log |\mathbf{A}| \tag{3.16}$$

where

$$\hat{\mathbf{o}}_t = \mathbf{A}'^{-1}\mathbf{o}_t + \mathbf{A}'^{-1}\mathbf{b}' = \mathbf{A}\mathbf{o}_t + \mathbf{b} = \mathbf{W}\boldsymbol{\zeta}_t \tag{3.17}$$

and $\boldsymbol{\zeta}_t = [\mathbf{o}_t^T \;\; 1]^T$ is the extended observation, $|\mathbf{A}|$ is the determinant of $\mathbf{A}$, and $\mathbf{W} = [\mathbf{A} \;\; \mathbf{b}] = [\mathbf{A}'^{-1} \;\; \mathbf{A}'^{-1}\mathbf{b}']$ is the extended linear transform for features associated with one particular homogeneous data block. From equation (3.16), the linear transform $\mathbf{A}$ does not impact the form of covariance matrices after adaptation. Hence, the likelihood calculation is efficient if the original covariance matrix is diagonal because there is no need to calculate the likelihood with a full covariance matrix. Furthermore, the mean vector does not need to be adapted either, which saves more computational resources.

Since it is more efficient to implement constrained MLLR as a feature transform, it is preferable to estimate the feature transform $\mathbf{W} = [\mathbf{A} \;\; \mathbf{b}]$ rather than the original transform $[\mathbf{A}' \;\; \mathbf{b}']$. Again, the EM algorithm is used to iteratively update $\mathbf{W}$. The diagonal transform case was

solved in [22] and the full transform case was solved in [34]. The auxiliary function for full constrained MLLR is defined as [34]

$$\mathcal{Q}_{\texttt{ML}}(\mathcal{T};\hat{\mathcal{T}},\hat{\mathcal{M}}) = -\frac{1}{2}\sum_{t,m}\gamma_m^{\texttt{ML}}(t)\left\{\left(\mathbf{W}\boldsymbol{\zeta}_t-\boldsymbol{\mu}^{(m)}\right)^T\boldsymbol{\Sigma}^{(m)-1}\left(\mathbf{W}\boldsymbol{\zeta}_t-\boldsymbol{\mu}^{(m)}\right)-\log(|\mathbf{A}|^2)\right\} \quad (3.18)$$

where the exact form of $\mathcal{T}$ is the feature transform $\mathbf{W}$, $\boldsymbol{\zeta}_t$ are the extended observations, $\gamma_m^{\texttt{ML}}(t)$ is the posterior occupancy of component m at time t calculated with the current constrained transform. Optimising the auxiliary function, equation (3.18), with respect to $\mathbf{W}$ leads to the update formulae. It has been shown in [34], given sufficient statistics

$$\mathbf{G}_{\texttt{ML},d} = \sum_m \frac{1}{\sigma_{dd}^{(m)}}\sum_t \gamma_m^{\texttt{ML}}(t)\boldsymbol{\zeta}_t\boldsymbol{\zeta}_t^T \quad (3.19)$$

$$\mathbf{k}_{\texttt{ML},d} = \sum_m \frac{\mu_d^{(m)}}{\sigma_{dd}^{(m)}}\sum_t \gamma_m^{\texttt{ML}}(t)\boldsymbol{\zeta}_t \quad (3.20)$$

where $\sigma_{dd}^{(m)}$ is the d^{th} diagonal element of covariance matrix $\boldsymbol{\Sigma}^{(m)}$, $\mu_d^{(m)}$ is the d^{th} element of $\boldsymbol{\mu}^{(m)}$. The d^{th} row of the transform,$\mathbf{w}_d^T$, can be estimated by

$$\hat{\mathbf{w}}_d = \mathbf{G}_{\texttt{ML},d}^{-1}\left(\alpha\mathbf{p}_d+\mathbf{k}_{\texttt{ML},d}\right) \quad (3.21)$$

where $\mathbf{p}_d$ is the extended cofactor vector $[0\ c_{d1}\ \cdots\ c_{dD}]^T$, D is the dimension number, $c_{ij} = cof(\mathbf{A}_{ij})$ is the cofactor. The coefficient α satisfies a quadratic expression given the total occupancy $\beta = \sum_{m,t}\gamma_m^{\texttt{ML}}(t)$

$$\alpha^2\mathbf{p}_d^T\mathbf{G}_{\texttt{ML},d}^{-1}\mathbf{p}_d+\alpha\mathbf{p}_d^T\mathbf{G}_{\texttt{ML},d}^{-1}\mathbf{k}_{\texttt{ML},d}-\beta = 0 \quad (3.22)$$

From the above formulae, the update is an iterative solution over the rows since the rows of the transform are dependent on one another via the extended cofactor vector. Once the final transform $\mathbf{W}$ is obtained, it is used to transform the observations to the specific test acoustic condition before recognition.

3.2.3 Cluster Based Adaptation

The model based adaptation in the previous section is based on a standard set of HMMs. An alternative is to perform adaptation on a series of sets of HMMs, referred to as *cluster based adaptation*. One traditional cluster adaptation scheme is to build several cluster-dependent models, for example speaker-dependent (SD) HMMs and choose an appropriate one for a particular test acoustic condition [29]. In this approach, the adaptation, or the selection of the appropriate model for recognition is a "hard" choice. Alternatively, some researchers used a linear combination of a set of cluster-dependent models [52, 51]. The final "adapted" model is not necessarily one of the reference models but a new interpolated one. Hence, this choice is a "soft" choice. The transform smoothing [32], cluster adaptation [33] and eigenvoices [69] are all based on this soft choice of HMMs. In the cluster based adaptation technique, the "transform" to adapt

the model is the interpolation weights or selection indicator. Rather than using standard HMMs, cluster adaptation techniques require a set of HMM clusters. Therefore, this adaptation approach is closely related to training multiple-cluster HMMs and will be discussed in detail in the cluster based adaptive training section 3.4.2.

3.2.4 Regression Classes and Transform Tying

The transform based schemes described above use a *global* transform to adapt all Gaussian components. This is likely to give a reliable transform estimate even if there is only a small amount of data available. However, as more data becomes available, the adaptation performance may be improved by increasing the number of transforms. Each transform is then specific and applied to a certain group of Gaussian components[7]. These groups are commonly referred to as *base classes*. This achieves a more flexible adaptation depending on the amount of data available.

Two approaches can be used to group Gaussian components into base classes: phonetic characteristics and regression class trees [72]. The phonetic characteristics approach is a static knowledge-based decision tree method. Gaussian components are grouped into the broad phone classes: silence, vowels, stops, glides, nasals, fricatives, etc. Phone class specific transforms may then be generated in adaptation and applied to these groups during recognition. The disadvantage of this approach is that the tying of parameters is performed at the phone level and independent of the adaptation data, which is not flexible.

Rather than specifying static classes, a dynamic scheme is often used to construct additional transforms as more adaptation data become available. A *regression class tree* [72, 31] is used to group Gaussian components so that the number of the transforms to be estimated can be dynamically chosen according to the amount of available adaptation data. The regression class tree is constructed before adaptation by automatically clustering Gaussian components which are close in acoustic space. For example, a centroid splitting algorithm with Euclidean distance can be used as a specific clustering scheme. The basic assumption here is that similar components should be transformed, or adapted, in a similar way [133].

Having constructed a regression class tree, it is used in adaptation to dynamically determine the Gaussian groups on which a transform is based. Figure 3.2 illustrates the use of a binary regression class tree with four terminal nodes. The shaded terminal nodes specify the final Gaussian component groups, or *base classes*. During adaptation, the occupancy counts are accumulated for each regression base class. If there is sufficient data for a transform to be estimated for a node, the node is a solid circle, otherwise, it is dotted. The solid line also indicates sufficient data and dotted line for insufficient data. As the figure shows, neither node 6 or 7 has sufficient data; however when pooled at node 3, there is sufficient adaptation data for a transform to be generated. The threshold to determine the sufficiency of adaptation data should be set in advance. The global adaptation case is equivalent to using a regression tree with just

[7] When constrained MLLR transforms in section 3.2.2.3 are used, the multiple transforms will be applied to features and result in multiple feature streams, each one is associated with a certain group of Gaussian components.

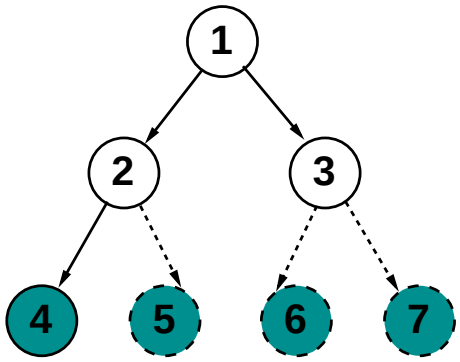


Figure 3.2 *A binary regression tree with four terminal nodes*

the root node. As the threshold is used, this global adaptation may result in no transforms if there is not sufficient data for generating a global transform.

When using a regression class tree, the estimation of the transforms described in section 3.2.2 and the interpolation weights in the following section 3.4.2.2 need to be slightly modified. One transform is estimated for each base class. Rather than accumulating the sufficient statistics over all Gaussian components, only the Gaussian components within the base class that a particular transform belongs to need to be accumulated over. Given the sufficient statistics accumulated on the appropriate group of Gaussian components, the final transform re-estimation formulae remain unchanged.

3.2.5 Extensions of Standard Schemes

The previous sections reviewed the standard adaptation schemes. Though these schemes have achieved good performance, there are some limitations when using them in unsupervised adaptation. Moreover, the use of the ML criterion limits the gains that can be obtained from adaptation. This section will describe a number of extensions which address various limitations of the standard techniques. The extensions introduced here are for transform and cluster based adaptation, not for MAP adaptation of HMMs.

3.2.5.1 Confidence Score Based Adaptation

In unsupervised adaptation, the supervision transcription used to estimate transforms is recognised hypothesis sequences. As indicated before, if the error rate of the hypothesis is high, it can reduce the effectiveness the adaptation process due to incorporating incorrect statistics in transform estimation. With partially incorrect statistics, it is hard to robustly estimate transforms. Consequently, the recognition performance may be degraded.

To solve this problem, confidence score based adaptation was proposed [55, 106, 142, 2, 111]. The basic idea is to calculate a confidence score for each word of the recognised supervision hypothesis. During adaptation, only those words with a high confidence score are used to accumulate statistics for transform estimation. Words with a confidence score below a certain

threshold are ignored[122]. Word posterior probabilities from the recogniser are widely used to calculate the confidence scores for each word in the hypothesis [123, 124].

This technique is effective when the word error rate of the hypothesis supervision is high. Since confidence score based adaptation "eliminates" incorrect words from the hypothesis, in an ideal case, it is similar to obtaining a correct transcription and then performing supervised adaptation. It has been reported that confidence score based adaptation may obtain significant improvements over standard transform based adaptation [142, 117, 2, 111]. However, due to the elimination process, confidence score based adaptation has the disadvantage of reducing the amount of adaptation data and consequently limiting the number of transforms that can be estimated.

3.2.5.2 Lattice Based Adaptation

Though confidence score based adaptation can improve the recognition performance, it only uses a single hypothesis. Further improvements may be obtained by also considering alternative hypotheses. Lattice based adaptation has been proposed for this purpose [89, 115]. The idea is to perform adaptation by accumulating statistics against a lattice rather than a 1-Best recognised hypothesis. As the oracle error rate of lattice is usually much lower than that of the 1-Best hypothesis, the performance degradation due to error-full hypothesis may be significantly reduced.

In lattice based adaptation, an extended forward-backward algorithm is performed through the recognised lattice, or alternative hypotheses [130, 131]. This lattice forward-backward algorithm gives the posterior probability of each Gaussian component given all possible hypotheses in the lattice. Using this posterior probability to accumulate statistics for updating transforms automatically weights different hypotheses in the lattice. Hence, there is no need to discard complete frames of data as in confidence score based adaptation. As no adaptation data is eliminated, more transforms can be robustly generated than for confidence based adaptation.

Lattice based adaptation was first introduced for MLLR adaptation [89, 115] and later for other adaptation techniques, such as constrained MLLR [111]. Due to its improvement in unsupervised adaptation, it has been used in state-of-the-art multi-pass speech recognition systems [25].

3.2.5.3 Extensions to Overcome ML Limitations

The transform based adaptation schemes introduced above use the ML criterion to estimate the transform parameters. However, as indicated in section 2.3.4, the ML criterion has some limitations, which may result in degraded performance. To address these limitations, several extended adaptation schemes are proposed.

One limitation of the ML criterion is that it does not directly aim at reducing the word error rate of the adaptation data. Section 2.4 discussed *discriminative training*, which was proposed to address this issue. With the recent interest in discriminative training of standard HMMs, there is also a trend to discriminatively estimate transforms during adaptation. A frame discrimination

criterion was used in [117] to estimate linear transform parameters. Another study used the H-criterion to estimate discriminative linear transforms [113]. The objective function was an interpolation of ML and MMI criteria. Conditional MLLR uses a conditional maximum likelihood auxiliary function [47, 48]. It is an alternative approach to derive discriminative linear transforms. The MMI criterion [110] and the MPE criterion [119] were also investigated for a similar purpose. Besides using discriminative criteria for linear transform based adaptation, recently, the application to cluster based adaptation was also investigated [137, 79]. It has been shown that discriminative adaptation can provide significant reduction in word error rate for supervised adaptation [114, 113]. Application of discriminative adaptation in unsupervised mode has also been investigated [47, 119]. As the correct transcription is not known, discriminatively estimating the test set transformation is problematic. To maintain a consistent criterion in transform training and in testing adaptation, a simplified discriminative adaptive training scheme is normally used where only the canonical model parameters are discriminatively updated with the ML estimated transforms fixed [78]. The model can then be used together with ML-estimated transforms in testing adaptation. This is the approach adopted in this work. A detailed discussion of discriminative adaptive training will be given in chapter 4.

The ML criterion has another limitation related to robust estimation of the parameters. When estimating transforms using very little adaptation data, the ML estimate of transform may be unreliable. A general solution to this problem is to use a Bayesian framework in adaptation [138]. One commonly used implementation of Bayesian approach is to perform a MAP-style estimation of the transform parameters. This is referred to as *maximum a posteriori linear regression* (MAPLR) for linear transforms [15]. The MAP criterion was also investigated for cluster based adaptation [33]. A general Bayesian framework for adaptive training and adaptive inference will be given in chapter 5.

3.3 Adaptive Training on Non-homogeneous Training Data

A pre-requisite of adaptation is well trained HMMs. The traditional approach to get the model is to train an HMM model set on specifically collected data which comes from a single source. The model is then adapted to the test domain during recognition using the techniques in the previous section. Recently, there has been a trend towards building systems on *found*, or *non-homogeneous* data, which does not come from the same source, hence acoustic mismatch exists within the training data. The basic idea of adaptive training is to use the adaptation fashion in the training process to build compact systems on non-homogeneous data. This section will discuss the general issues regarding adaptive training.

3.3.1 Multi-style Training and Adaptive Training

A simple approach for dealing with non-homogeneous training data is to build a system on the whole training data set as if all data comes from a single homogeneous data block. The assump-

tion here is that the extracted features do not contain any unwanted non-speech variabilities. Current front-ends for speech recognition, such as MFCCs and PLPs, can not completely remove the unwanted variabilities. Consequently, acoustic models trained on normal features not only represent the speech variability, but also model some acoustic condition variabilities. For this reason, systems directly built on *all* non-homogeneous training data with the normal features are referred to as *multi-style* trained systems [38], such as speaker-independent model. As a general model, a multi-style system can be used in recognition without adaptation. The adaptation on top of the multi-style system fine tunes it to a particular test acoustic condition. Though good performance has been obtained with multi-style systems, the acoustic mismatch between different parts of the training data is not well addressed. It would be preferable to use other training schemes that are more powerful to handle the non-speech variabilities in training data.

Adaptive training is a powerful solution for building systems on non-homogeneous training data [3]. Rather than dealing with all the data as a single block, the training data is split into several *homogeneous* blocks, for example speaker side or data block with the same acoustic environment. Thus, the training data is written as $\mathcal{O} = \{\mathbf{O}^{(1)}, \ldots, \mathbf{O}^{(S)}\}$ and $\mathbf{H} = \{\mathcal{H}^{(1)}, \ldots, \mathcal{H}^{(S)}\}$, where $\mathbf{O}^{(s)}$ is the observation sequence of a homogeneous block associated with a particular acoustic condition s, $\mathcal{H}^{(s)}$ is the corresponding transcription sequence. Given this split, two distinct sets of parameters are introduced to separately model the speech and the non-speech variabilities

1. **Canonical model $\mathcal{M}$**

   The ideal canonical model represents the desired *speech variability* of the whole training data. Hence, $\mathcal{M}$ is independent of acoustic conditions. It is estimated given the whole data set $\mathcal{O}$ and $\mathbf{H}$ and a set of adaptation transforms. The nature of canonical model depends on the form of the adaptation transform. For linear transforms, the form of the canonical model is standard HMMs. However, multiple-cluster HMMs can also be used as a special type of canonical model for interpolation weights [39].

2. **A set of transforms $\boldsymbol{\mathcal{T}} = \{\mathcal{T}^{(1)}, \ldots, \mathcal{T}^{(S)}\}$**

   A set of transforms is used to represent the unwanted *non-speech variabilities*. One *distinct* transform $\mathcal{T}^{(s)}$ represents the acoustic condition of a particular homogeneous block s and is estimated using $\mathbf{O}^{(s)}$ and $\mathcal{H}^{(s)}$. The transform acts as a counterpart of the canonical model to normalise features or adapt the canonical model to that particular acoustic condition.

It is worth emphasising that the canonical model is always estimated *given* the set of transforms accounting for non-speech variabilities. Hence, in recognition, the canonical model can not be directly used. It must be adapted by an appropriate transform to represent both speech and specific non-speech variabilities of a particular test acoustic condition. Due to the modelling of desired speech variability, the canonical model is more compact than the multi-style model. Hence, it should be more adaptable to a new test acoustic condition than a multi-style trained model. The sections below will review some schemes for adaptive training.

3.3.2 Adaptive Training Schemes

In terms of the targets that transforms work on, adaptive training can be classified into two broad classes: feature normalisation and model based transforms. This section will review the schemes for the two classes of adaptive training.

3.3.2.1 Feature Normalisation

Features used in speech recognition systems are supposed to be sensitive to speech variabilities but inherently robust to non-speech variabilities. If such an ideal set of features could be obtained, adaptation and adaptive training would yield no performance gain. However, as indicated in chapter 2.1, no such feature extraction approach actually exists, hence adaptation and adaptive training are useful. Although features that are completely robust to acoustic condition changes do not exist, it is possible to normalise the features so that they are less sensitive to acoustic condition changes. This approach is referred to as *feature normalisation*. It is assumed that the HMMs trained on normalised features have less mismatch if used on test data normalised using the same approach. Therefore, feature normalisation is normally applied in both training and testing [38]. It is worth emphasising that the feature normalisation is performed for each homogeneous data block respectively rather than for the whole data set.

An advantage of using feature normalisation in adaptive training is that the re-estimation formulae of the canonical model are almost unchanged. The only difference from standard HMM training is that the normalised features are used instead of the original ones. This significantly reduces the computational cost which may be very high for adaptive training with model based transforms. According to whether the normalisation is dependent on HMMs or not, this approach can be further classified into two categories.

1. Model Independent Feature Normalisation

The model independent schemes do not explicitly use any model and transcription information. The normalisation transform is directly estimated from and applied to features so that general statistic attributes of acoustic conditions can be removed. As the transform estimation does not require any transcription/hypothesis information of the test data, it is simple and can be effectively used in both supervised and unsupervised adaptation. However, as the transform is global from the viewpoint of HMMs, the ability to compensate acoustic mismatch is limited. Hence, the model independent normalisation is often used together with other adaptive training techniques. The most commonly used schemes include:

- **Cepstral Mean Normalisation (CMN) and Cepstral Variance Normalisation (CVN)**

  One standard normalisation transform is to *sphere* the data, i.e., transform the data so that it has zero mean and unit variance at each dimension of the feature. Cepstral mean normalisation (CMN) [4, 129] and Cepstral variance normalisation (CVN) are simple techniques to achieve this goal.

  The idea is to normalise the mean and variance of each dimension of the observations.

Therefore, the CMN is performed by

$$\hat{\mathbf{o}}_t^{\texttt{CMN(s)}} = \mathbf{o}_t^{(s)} - \bar{\mathbf{o}}^{(s)} = \mathbf{o}_t^{(s)} - \frac{1}{T}\sum_{i=1}^{T}\mathbf{o}_i^{(s)} \tag{3.23}$$

where $\mathbf{o}_t^{(s)}$ is the observation vector associated with homogeneous block s at time t, $\hat{\mathbf{o}}_t^{\texttt{CMN(s)}}$ is the transformed observation vector after CMN, $\bar{\mathbf{o}}^{(s)}$ is the mean value of the observation sequence $\mathbf{O}^{(s)} = [\mathbf{o}_1^{(s)}, \cdots, \mathbf{o}_T^{(s)}]$, and T is the length of the observation sequence. From the signal processing point of view, CMN is similar to the RASTA approach, where a high-pass filter is applied to a log-spectral representation of speech, such as Cepstral coefficients [54]. Equation (3.23) is also a high-pass filter in the Cepstral domain. This filtering will suppress the constant spectral components, which reflect the effect of convolutive noise factors in the input speech signal, hence, yields features robust to slowly varying convolutive noise.

After CMN, the mean value of the observations from 1 to T is zero. The CVN further normalises the variance of each dimension of the observations to be 1

$$\hat{o}_{t,d}^{\texttt{CVN(s)}} = \hat{o}_{t,d}^{\texttt{CMN(s)}} / \sqrt{\sigma_{dd}^{(s)}} \tag{3.24}$$

where $\hat{\mathbf{o}}_{t,d}^{\texttt{CVN(s)}}$ is the d^{th} dimension of the normalised observation after CVN on top of CMN, $\sqrt{\sigma_{dd}^{(s)}}$ is the square root of the variance of observations at the d^{th} dimension

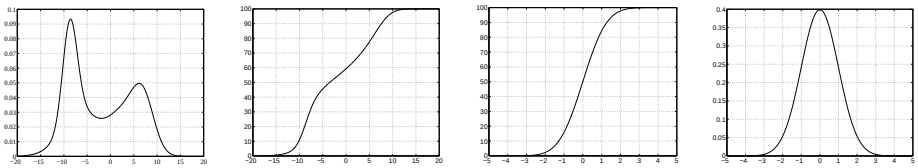
$$\sigma_{dd}^{(s)} = \frac{1}{T}\sum_{i=1}^{T}\left(\hat{o}_{i,d}^{\texttt{CMN(s)}}\right)^2 \tag{3.25}$$

The advantage of CMN and CVN is that they are simple to calculate and apply since no transcription information is required. After normalisation of both training and test data, the non-speech acoustic mismatch between different homogeneous data block is effectively reduced. Hence, they are widely used in state-of-the-art speech recognition system [25].

- **Gaussianisation**

  Though CMN and CVN results in sphered data if the distribution of each dimension of the observations is Gaussian, the simple linear normalization scheme may not work well for highly non-homogeneous speech data. For such data, the distribution of observations can not be assumed to be a single Gaussian. A non-linear feature normalization scheme, *Gaussianisation*, is then proposed to solve this problem [100]. The basic idea is to normalise the cumulative density function (CDF) of the observations (the source CDF) to a CDF of a standard Gaussian (the target CDF). Hence, it may be viewed as higher order version of CMN and CVN. An illustration of Gaussianisation is given in figure 3.1.

  Gaussianisation was first implemented using a complex iterative scheme based on histogram matching [100]. An alternative approach, which provides a more compact and smooth representation of the distribution of the original observations [40, 76] is based on



Source PDF Source CDF Target CDF Target PDF

Table 3.1 *Illustration of Gaussianisation*

GMM. In this approach, each dimension of the original observations is represented by a GMM. Let o_d denote the d^{th} dimension of a D dimensional acoustic feature vector $\mathbf{o}$. Then the Gaussianised feature on the d^{th} dimension associated with homogeneous block s is given by

$$\hat{o}_d^{(s)} = \phi^{-1}\left(\int_{-\infty}^{o_d^{(s)}} \sum_{n=1}^{N_d^{(s)}} c_{dns}\mathcal{N}\left(x;\ \mu^{(dns)}, \sigma^{(dns)}\right)\ dx\right) \tag{3.26}$$

where $\hat{o}_d^{(s)}$ is the normalised observation element, $\phi^{-1}(\cdot)$ denotes the Gaussian inverse CDF. The GMM component means, variances and component weights are denoted by $\mu^{(dns)}$, $\sigma^{(dns)}$ and c_{dns} respectively, where n is the component index. They are specific for each dimension d and each homogeneous block s. Therefore, for each homogeneous block s, a total of D single dimension $N_d^{(s)}$ component GMMs need to be trained using the ML criterion. As Gaussianisation gives a better representation of the original observations, it has been found to yield considerable gains on highly non-homogeneous data, such as Mandarin conversational telephone speech (CTS) [40].

Although the model-independent schemes are simple and efficient to implement, they have some limitations. One limitation is that the adaptation data (observations only) are assumed to be sufficient to give robust estimates of the moments. For short data blocks, the estimated normalisation transform is sensitive to outliers of the observations, which may result in poor performance. Another limitation is that the normalisation transform is independent of the model parameters. This makes the model-independent normalisation always "global" in terms of the effect on HMMs. This inflexibility may limit the possible gains of adaptation. Hence, model dependent techniques are proposed to get further gains.

2. Model Dependent Feature Normalisation

This approach also calculates a feature normalisation transform dependent on acoustic condition. However, the transform is estimated by making use of a model set and transcription or hypothesis of the supervision data. Generally speaking, the normalisation transform can be applied either on the extracted features or some intermediate steps of the feature extraction process. The form of the transform can be either linear or non-linear. The standard techniques

normally use the ML criterion to estimate the transforms. Widely used model dependent normalisation schemes include

- **Constrained Maximum Likelihood Linear Regression (CMLLR)**

  Constrained MLLR (CMLLR) [34] has been introduced in section 3.2.2.3. Though it is a constrained linear transform on both the mean vector and covariance matrix of a Gaussian component, it can be equivalently written as a feature normalisation transform as indicated in section 3.2.2.3. As CMLLR is closely related to model based transform techniques, it is discussed in detail in the following section 3.4.1.2.

- **Vocal Tract Length Normalisation (VTLN)**

  One major non-speech variability that affects the performance of a speech recognition system is the variability of the human voice among different speakers. *Vocal tract length normalisation* (VTLN) [71] is a technique that can reduce the mismatch between speakers. The basic idea is to map the actual speech signal to a normalised signal with less variability due to different vocal tract lengths of different speakers. A warping factor α is used to perform the mapping. It compresses or expands the frequency domain before the features are extracted from the speech signals. Therefore, it may be viewed as a non-linear transform on the feature domain. It is worth noting that VTLN requires estimation of a distinct warping factor for each speaker (homogeneous block) in the training and test dataset. Hence, it is introduced as an adaptive training technique.

  The estimation of the optimal warping factor often uses a grid search scheme [94, 71]. A sequence of discrete values of α are used as candidates. The optimal one is selected to maximise the likelihood of the data. As VTLN is typically a non-linear feature transform, exact calculation of the *Jacobian* of the warping transform, which is required in likelihood calculations, is highly complex. Approximation is normally required, for example, linear Cepstral transform was used to approximate the VTLN process [112]. As HMMs are required to calculate the likelihood, VTLN is regarded as a model-dependent normalisation. However, rather than using a complex model, simple HMMs and GMMs may be used to rapidly select the optimal warping factor [121]. VTLN is an effective technique to normalise features and has been shown to give additive gains when combined with other adaptation techniques, such as MLLR [94]. It has therefore been widely used in state-of-the-art speech recognition systems [25].

3.3.2.2 Model Based Transformation

Although feature normalisation can compensate the acoustic mismatch between training data, it is global from the viewpoint of modifying HMM parameters (except for CMLLR). Hence, it is not flexible in terms of modifying the acoustic model and the gains it may obtain are limited. Model based transformation is an alternative and more powerful adaptive training approach. In this approach, the HMM parameters, normally means and possibly covariances of Gaussians, are

adapted by a transform. This is the main focus of this thesis. It should be noted that, model based transforms are often combined with feature normalisation to obtain the best performance. For example, in a state-of-the-art CTS recognition system, model based transforms are estimated on top of the features normalised by CMN, CVN and VTLN [25].

3.4 Model Based Adaptive Training Schemes

Model based adaptive training is a flexible and powerful scheme for building systems on non-homogeneous data. Two sets of parameters, a canonical model and a set of transforms, are used to separately model the speech variability and non-speech variability respectively. Here, the transforms are used to adapt parameters of the canonical model to different acoustic conditions. In contrast to feature normalisation, model based schemes are more powerful due to the flexible modification of HMMs. This section will review ML adaptive training, which is the standard scheme.

Given the canonical model, homogeneous blocks are assumed to be independent of each others. The likelihood of the whole observation sequence $\mathcal{O} = \{\mathbf{O}^{(1)}, \ldots, \mathbf{O}^{(S)}\}$ given both sets of parameters is then expressed as

$$p(\mathcal{O}|\mathbf{H}, \mathcal{M}, \boldsymbol{\mathcal{T}}) = \prod_{s=1}^{S} p(\mathbf{O}^{(s)}|\mathcal{H}^{(s)}, \mathcal{M}, \mathcal{T}^{(s)}) \tag{3.27}$$

where s is the index of the acoustic condition or homogeneous data block, and $\boldsymbol{\mathcal{T}} = \{\mathcal{T}^{(1)}, \ldots, \mathcal{T}^{(S)}\}$ is a set of transforms, one for each homogeneous block. $\mathbf{H} = \{\mathcal{H}^{(1)}, \ldots, \mathcal{H}^{(S)}\}$ is the corresponding transcription sequence. Given the model parameters $\mathcal{M}$ and transform $\mathcal{T}^{(s)}$, the observation likelihood of each acoustic condition can be written as[8]

$$p(\mathbf{O}^{(s)}|\mathcal{H}^{(s)}, \mathcal{M}, \mathcal{T}^{(s)}) = \sum_{\boldsymbol{\theta}} P(\boldsymbol{\theta}|\mathcal{H}^{(s)}, \mathcal{M}) \prod_{t} p(\mathbf{o}_t|\mathcal{M}, \mathcal{T}^{(s)}, \theta_t) \tag{3.28}$$

where $\boldsymbol{\theta}$ is the hidden component sequence, $P(\boldsymbol{\theta}|\mathcal{H}^{(s)}, \mathcal{M})$ is the probability of a particular component sequence $\boldsymbol{\theta}$, and $p(\mathbf{o}_t|\mathcal{M}, \mathcal{T}^{(s)}, \theta_t)$ is the adapted Gaussian distribution at component θ_t for acoustic condition s. ML adaptive training seeks to estimate the canonical model parameters that maximise equation (3.27)

$$\hat{\mathcal{M}}_{\texttt{ML}} = \arg\max_{\mathcal{M}} p(\mathcal{O}|\mathbf{H}, \mathcal{M}, \hat{\boldsymbol{\mathcal{T}}}) \tag{3.29}$$

where $\hat{\boldsymbol{\mathcal{T}}}$ is a set of ML transform estimates given the canonical model estimate. The transform estimate for homogeneous block s is given by

$$\hat{\mathcal{T}}^{(s)}_{\texttt{ML}} = \arg\max_{\mathcal{T}} p(\mathbf{O}^{(s)}|\mathcal{H}^{(s)}, \hat{\mathcal{M}}, \mathcal{T}) \tag{3.30}$$

[8] As in the standard HMMs case, using the component sequence as the hidden variable sequence is convenient for deriving formulae for updating the Gaussian component parameters. This is a natural extension to the use of the state sequence.

As it is hard to directly estimate $\hat{\mathcal{M}}_{\texttt{ML}}$ from equation (3.29) due to the hidden component sequence $\boldsymbol{\theta}$, the expectation maximisation (EM) algorithm is again used to iteratively learn the parameter estimates [21]. Similar to the standard HMM case in section 2.3, an auxiliary function $\mathcal{Q}_{\texttt{ML}}(\mathcal{M}_{k+1};\hat{\mathcal{M}}_k,\hat{\boldsymbol{\mathcal{T}}})$, is introduced to find the estimate at iteration $k+1$, $\hat{\mathcal{M}}_{k+1}$, given the transform estimate $\hat{\boldsymbol{\mathcal{T}}}$ and the previous iteration's estimate, $\hat{\mathcal{M}}_k$.

$$\mathcal{Q}_{\texttt{ML}}(\mathcal{M}_{k+1};\hat{\mathcal{M}}_k,\hat{\boldsymbol{\mathcal{T}}}) = \Big\langle \log p(\mathcal{O},\boldsymbol{\theta}|\mathcal{M}_{k+1},\hat{\boldsymbol{\mathcal{T}}},\mathbf{H}) \Big\rangle_{P(\boldsymbol{\theta}|\mathcal{O},\mathbf{H},\hat{\mathcal{M}}_k,\hat{\boldsymbol{\mathcal{T}}})} \tag{3.31}$$

where $< f(x) >_{q(x)}$ denotes the expectation of function $f(x)$ with respect to the distribution of $q(x)$ defined in equation (2.14), $P(\boldsymbol{\theta}|\mathcal{O},\mathbf{H},\hat{\mathcal{M}}_k,\hat{\boldsymbol{\mathcal{T}}})$ is the component sequence posterior for the whole training data calculated using $\hat{\mathcal{M}}_k$ and $\hat{\boldsymbol{\mathcal{T}}}$. Equation (3.31) is a strict lower bound of $\log p(\mathcal{O}|\mathbf{H},\mathcal{M},\hat{\boldsymbol{\mathcal{T}}})$. Each iteration is guaranteed not to decrease the auxiliary function. Consequently the likelihood $p(\mathcal{O}|\mathbf{H},\mathcal{M},\hat{\boldsymbol{\mathcal{T}}})$ is not decreased. Eventually, the estimate of the parameters tends to be a local maximum. A similar auxiliary function can also be introduced to individually obtain the transform estimate for each particular homogeneous block

$$\mathcal{Q}_{\texttt{ML}}\left(\mathcal{T}^{(s)}_{k+1};\hat{\mathcal{T}}^{(s)}_{k},\hat{\mathcal{M}}_k\right) = \Big\langle \log p(\mathbf{O}^{(s)},\boldsymbol{\theta}|\hat{\mathcal{M}}_k,\mathcal{T}^{(s)}_{k+1},\mathcal{H}^{(s)}) \Big\rangle_{P(\boldsymbol{\theta}|\mathbf{O}^{(s)},\mathcal{H}^{(s)},\hat{\mathcal{M}}_k,\hat{\mathcal{T}}^{(s)}_{k})} \tag{3.32}$$

where $\mathbf{O}^{(s)}$ and $\mathcal{H}^{(s)}$ are observation sequence and transcription of the homogeneous data block s respectively, and $\mathcal{T}^{(s)}_{k+1}$ is the transform to be estimated at iteration $k+1$. As the canonical model and transforms are dependent on each other, an interleaving procedure is often used to refine both sets of parameters as below:

1. Initialise parameters sets $\hat{\mathcal{M}}_0$ and $\hat{\boldsymbol{\mathcal{T}}}_0$, set $k=0$.
2. Estimate $\hat{\boldsymbol{\mathcal{T}}}_{k+1}$ given $\hat{\mathcal{M}}_k$ and $\hat{\boldsymbol{\mathcal{T}}}_k$ using equation (3.32)
3. Estimate $\hat{\mathcal{M}}_{k+1}$ given $\hat{\mathcal{M}}_k$ and $\hat{\boldsymbol{\mathcal{T}}}_{k+1}$ using equation (3.31)
4. $k=k+1$. Goto 2 until convergence.

ML adaptive training requires sufficient training data at both the homogeneous block level and the global level to ensure robust estimates of both sets of parameters. The canonical model estimate is then used in adaptation/recognition. As the canonical model can not be directly used in recognition, testset transforms are usually estimated given some supervision data. The estimation process is similar to the training step 2. Then, the adapted model is used for final recognition.

Depending on the form of the transforms and the canonical model, model based adaptive training can be classified into two main categories: linear transform based schemes [3, 34] and cluster based schemes [33, 69]. This section will review them in detail.

3.4.1 Linear Transform Based Adaptive Training

As described in section 3.2.2, linear transforms are widely used to represent non-speech variabilities. A canonical model with the form of standard HMMs is adapted by those transforms

to construct new HMMs for each homogeneous data block associated with a particular acoustic condition. This process is illustrated in figure 3.3.

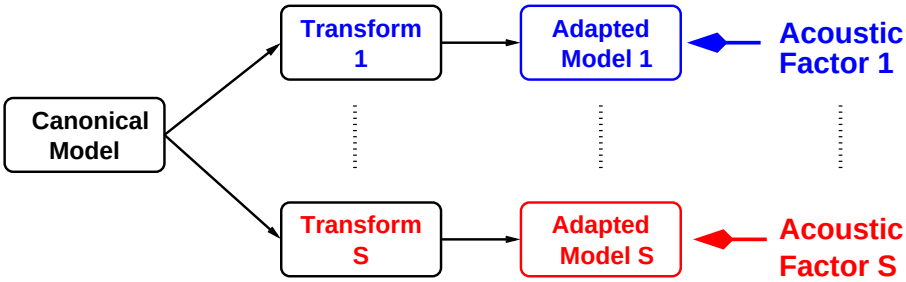


Figure 3.3 *Illustration of linear transform based adaptive training*

The transform update formulae in adaptive training is similar to that in adaptation, which has been detailed in 3.2.2. In this section, the estimation of the canonical model given the set of transforms is discussed in detail. The most popular forms of linear transforms used in adaptive training are MLLR and CMLLR. As variance MLLR is not normally used in linear transform based adaptive training, it is not considered here, but can be considered within a similar framework. The linear transform based adaptive training is sometimes referred to as *speaker adaptive training* (SAT)[3] because they were first introduced to handle the speaker variability. In the rest of this thesis, "*SAT*" is used to denote linear transform based adaptive training only.

3.4.1.1 SAT with MLLR

In SAT with MLLR, the *mean* vectors of the canonical model are adapted to each homogeneous data block using a distinct MLLR transform. As the canonical model training involves all training data, to make the derivation clear, the MLLR adaptation formula in equation (3.5), is rewritten here using an explicit notation that denotes the dependency on acoustic conditions and the regression base class used

$$\hat{\boldsymbol{\mu}}^{(sm)} = \mathbf{A}^{(sr_m)}\boldsymbol{\mu}^{(m)} + \mathbf{b}^{(sr_m)} = \mathbf{W}^{(sr_m)}\boldsymbol{\xi}^{(m)} \tag{3.33}$$

where r_m is the regression base class, or Gaussian group, that the Gaussian component m belongs to, $\hat{\boldsymbol{\mu}}^{(sm)}$ is the adapted mean of component m to acoustic condition s, $\boldsymbol{\xi}^{(m)} = [\boldsymbol{\mu}^{(m)T} \;\; 1]^T$ is the extended mean vector, and $\mathbf{W}^{(sr)} = [\mathbf{A}^{(sr)} \;\; \mathbf{b}^{(sr)}]$ is the extended linear transform associated with acoustic condition s and regression base class r. With the ML criterion, the general auxiliary function in equation (3.31) can be explicitly written for SAT with MLLR as below[9]

$$\begin{aligned}\mathcal{Q}_{\mathtt{ML}}(\mathcal{M};\hat{\mathcal{M}},\hat{\boldsymbol{\mathcal{T}}}) \;\; = \;\; & -\frac{1}{2}\sum_{s,m,t}\gamma_m^{\mathtt{ML}}(t)\Big\{\log\left|\boldsymbol{\Sigma}^{(m)}\right| \\ & + \left(\mathbf{o}_t^{(s)} - \mathbf{W}^{(sr_m)}\boldsymbol{\xi}^{(m)}\right)^T \boldsymbol{\Sigma}^{(m)-1}\left(\mathbf{o}_t^{(s)} - \mathbf{W}^{(sr_m)}\boldsymbol{\xi}^{(m)}\right)\Big\}\end{aligned} \tag{3.34}$$

[9] Iteration index k is omitted for clarity.

where $\hat{\mathcal{M}}$ is the current canonical model estimate, diagonal covariance is again used here, $\hat{\boldsymbol{\mathcal{T}}}$ is the MLLR transform set consisting of $\mathbf{W}^{(sr)}$ for acoustic condition s, $\mathbf{o}_t^{(s)}$ is the observation vector of homogeneous block s at time t, and $\gamma_m^{\mathtt{ML}}(t)$ is the posterior occupancy of component m at time t calculated using the forward-backward algorithm based on the current canonical model $\hat{\mathcal{M}}$ and the transform estimate $\hat{\boldsymbol{\mathcal{T}}}$. It can be shown [3] that given the sufficient statistics

$$\mathbf{G}_{\mathtt{ML}}^{(m)} = \sum_{s,t} \gamma_m^{\mathtt{ML}}(t) \mathbf{A}^{(sr_m)T} \boldsymbol{\Sigma}^{(m)-1} \mathbf{A}^{(sr_m)} \tag{3.35}$$

$$\mathbf{k}_{\mathtt{ML}}^{(m)} = \sum_{s,t} \gamma_m^{\mathtt{ML}}(t) \mathbf{A}^{(sr_m)T} \boldsymbol{\Sigma}^{(m)-1} \left(\mathbf{o}_t^{(s)} - \mathbf{b}^{(sr_m)} \right) \tag{3.36}$$

the new mean is estimated by

$$\hat{\boldsymbol{\mu}}^{(m)} = \mathbf{G}_{\mathtt{ML}}^{(m)^{-1}} \mathbf{k}_{\mathtt{ML}}^{(m)} \tag{3.37}$$

It is hard to simultaneously update mean and covariance using equation (3.34). Hence, The covariance re-estimation is performed after the mean update, and is similar to the standard covariance update [3]:

$$\hat{\boldsymbol{\Sigma}}^{(m)} = \text{diag} \left(\frac{\sum_{s,t} \gamma_m^{\mathtt{ML}}(t) \left(\mathbf{o}_t^{(s)} - \mathbf{W}^{(sr_m)} \hat{\boldsymbol{\xi}}^{(m)} \right) \left(\mathbf{o}_t^{(s)} - \mathbf{W}^{(sr_m)} \hat{\boldsymbol{\xi}}^{(m)} \right)^T}{\sum_{s,t} \gamma_m^{\mathtt{ML}}(t)} \right) \tag{3.38}$$

where $\hat{\boldsymbol{\xi}}^{(m)}$ is the extended mean vector with the new estimate of the canonical model mean $\hat{\boldsymbol{\mu}}^{(m)}$ in equation (3.37).

The MLLR transform update is similar to section 3.2.2.1 and is not rewritten here. It is worth emphasising that the transform update is based on each homogeneous data block rather than the whole training dataset [10]. From the update formulae in equations (3.37) and (3.38), the re-estimation of mean vectors requires considerable memory and computational load because it needs to store a full or block-diagonal matrix for each Gaussian component [81]. This becomes impractical if the number of Gaussian components in the system is increased. Furthermore, covariances can not be updated in the same pass as the mean vector. These disadvantages limit the use of SAT with MLLR for systems with a high complexity.

3.4.1.2 SAT with Constrained MLLR

The computation issue of SAT with MLLR can be avoided by using constrained MLLR to build SAT systems [34]. As mentioned before, although CMLLR is a model parameter transform which uses the same transform for both mean and variance, it is equivalent to a feature transform. The formula for transformed observations in equation (3.17) is also rewritten here with explicit notation of acoustic condition and regression base class

$$\hat{\mathbf{o}}_t^{(sr_m)} = \mathbf{A}^{(sr_m)} \mathbf{o}_t^{(s)} + \mathbf{b}^{(sr_m)} = \mathbf{W}^{(sr_m)} \boldsymbol{\zeta}_t^{(s)} \tag{3.39}$$

[10] As the transform update does not involve different acoustic conditions, the acoustic condition index s is omitted in the previous adaptation section.

where r_m is the regression base class for the Gaussian component m, $\mathbf{W}^{(sr)} = [\mathbf{A}^{(sr)} \quad \mathbf{b}^{(sr)}]$ is the extended constrained transform associated with acoustic condition s and regression base class r, $\boldsymbol{\zeta}_t^{(s)} = [\mathbf{o}_t^{(s)T} \quad 1]^T$ is the extended observation, and $\hat{\mathbf{o}}_t^{(sr_m)}$ is the transformed observation. The transformed observation is now dependent on Gaussian component groups. This means one distinct transformed observation needs to be stored for each regression base class when accumulating statistics during training.

The characteristics of being a feature transform not only saves computational resources when calculating the likelihood, but also significantly simplifies the re-estimation formula for model parameters. It can be shown that the auxiliary function for SAT with CMLLR is expressed by [34]

$$\mathcal{Q}_{\texttt{ML}}(\mathcal{M};\hat{\mathcal{M}},\hat{\boldsymbol{\mathcal{T}}}) = -\frac{1}{2}\sum_{s,m,t}\gamma_m^{\texttt{ML}}(t)\Big\{\log\left|\boldsymbol{\Sigma}^{(m)}\right| + \left(\hat{\mathbf{o}}_t^{(sr_m)} - \boldsymbol{\mu}^{(m)}\right)^T\boldsymbol{\Sigma}^{(m)-1}\left(\hat{\mathbf{o}}_t^{(sr_m)} - \boldsymbol{\mu}^{(m)}\right)\Big\} \quad (3.40)$$

where $\mathcal{T}$ is now the constrained linear transform, $\hat{\mathbf{o}}_t^{(sr_m)}$ is the transformed observation vector using equation (3.39), and $\gamma_m^{\texttt{ML}}(t)$ is the posterior occupancy calculated using the transformed observations. Comparing this to the auxiliary function for standard HMMs in equation (2.41), the main difference is that the transformed observations $\hat{\mathbf{o}}_t^{(sr_m)}$ is used here instead of the original observations. The resultant update formulae of mean and covariance are thus given by

$$\begin{aligned}\hat{\boldsymbol{\mu}}^{(m)} &= \frac{\sum_{s,t}\gamma_m^{\texttt{ML}}(t)\hat{\mathbf{o}}_t^{(sr_m)}}{\sum_{s,t}\gamma_m^{\texttt{ML}}(t)} \\ \hat{\boldsymbol{\Sigma}}^{(m)} &= \text{diag}\left(\frac{\sum_{s,t}\gamma_m^{\texttt{ML}}(t)(\hat{\mathbf{o}}_t^{(sr_m)} - \hat{\boldsymbol{\mu}}^{(m)})(\hat{\mathbf{o}}_t^{(sr_m)} - \hat{\boldsymbol{\mu}}^{(m)})^T}{\sum_{s,t}\gamma_m^{\texttt{ML}}(t)}\right)\end{aligned}$$

The above formulae are also similar to equations (2.37) and (2.38). This means that the computational load and memory requirements of SAT with CMLLR are similar to that of the standard HMM system and greatly reduced compared to SAT with MLLR. Again, the estimation of the CMLLR transform is the same as in section 3.2.2.3 and is not rewritten here.

3.4.2 Cluster Based Adaptive Training

Linear transform based adaptive training described before uses standard HMMs as the canonical model. However, the form of the canonical model may vary depending on the nature of the transform used. An alternative transform is a set of interpolation weights to combine multiple sets of HMMs. In this case the canonical model is a multiple-cluster model.

As shown in figure 3.4, multiple sets of HMMs, one for each cluster, are used. The adapted model is generated by interpolating the cluster parameters to form a standard set of HMMs. Though this approach was originally motivated for rapid speaker adaptation, it can be effectively extended for adaptation with respect to other non-speech variabilities in LVCSR [39]. One special type of the interpolation weights are $0/1$ indicators, referred to as *hard* weights. These result in the traditional cluster dependent model. As a generalisation, arbitrary values may be used as the interpolation weights, referred to as *soft* weights. Cluster adaptive training [33]

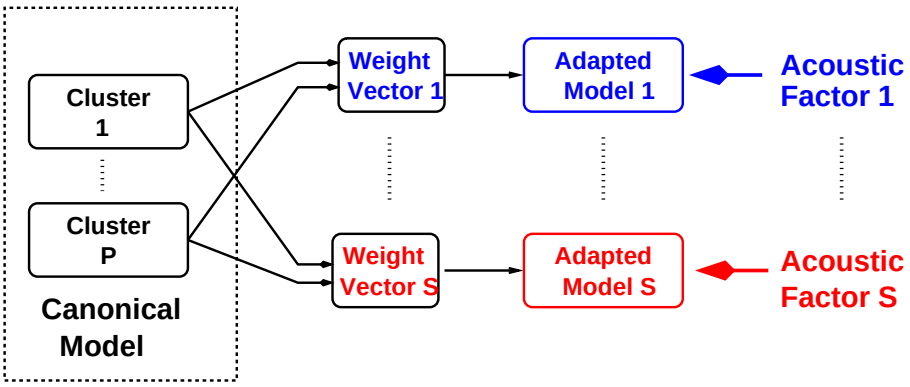


Figure 3.4 *Illustration of cluster based adaptive training*

and eigenvoices [69] belong to this type. This section will review the two types of cluster based adaptive training, in particular cluster adaptive training.

3.4.2.1 Cluster Dependent Modelling

Traditional cluster dependent modelling may be viewed from an adaptive training point of view. The non-homogeneous training data is split into homogeneous data blocks in terms of acoustic clusters, such as gender or speaker. A standard set of HMMs is then trained using each cluster data block. During recognition, a cluster selection process, such as gender detection, is performed for each test homogeneous block. This may be regarded as the *adaptation* process. For cluster dependent modelling, the adaptation of model parameters may be written in a generic form. For instance, the mean vector is adapted by

$$\hat{\boldsymbol{\mu}}^{(sm)} = \sum_{c} \delta(s,c)\boldsymbol{\mu}_c^{(m)} \tag{3.41}$$

where $\hat{\boldsymbol{\mu}}^{(sm)}$ is the adapted mean for the test domain s, c is the index of the cluster dependent models, and $\delta(s,c)$ is a *Kronecker delta* function which has value 1 if two arguments match and value 0 otherwise, i.e.

$$\delta(s,c) = \begin{cases} 1 & s = c \\ 0 & otherwise \end{cases} \tag{3.42}$$

The use of Kronecker delta function can be viewed as the selection of the candidate system c which is closest to speaker s. The adapted, or selected, model is then used to recognise the corresponding test data. Gender dependent or speaker dependent modelling are good examples and have been widely used [67]. The main problem with this approach is that, by taking a hard decision, only a limited number of HMM systems representing the training acoustic conditions can be chosen from. If the test acoustic condition does not appear in training data and is very different from any of the training acoustic conditions, any choice is likely to give poor performance. Namely, the hard decision has limited generalization ability and consequently limits its use. Cluster based approaches with *soft* decisions are proposed to address this problem.

3.4.2.2 Cluster Adaptive Training and Eigenvoices

Cluster adaptive training (CAT) [33], or *eigenvoices* [88], is a multiple-cluster HMM training approach with *soft* weights. The basic idea is to build a target-domain-specific model by using a weighted sum of multiple sets of HMMs. An interpolation weight vector, acting as the *transform* in adaptive training, is computed for each distinct acoustic condition during training. Since the weight used can take arbitrary values instead of just the $1/0$ indicator, this approach is a generalization of traditional cluster dependent modelling.

In order to simplify training, it is often assumed that different clusters have the same covariance matrices, transition matrices and mixture weights and that only the mean values differ between the clusters. Therefore, in the multiple-cluster canonical model, also referred to as a *CAT model*, each component m consists of a mixture weight, $c^{(m)}$, a covariance matrix (usually diagonal), $\mathbf{\Sigma}^{(m)}$, and a set of P means, one for each of the P clusters, normally arranged into a matrix $\mathbf{M}^{(m)}$,

$$\mathbf{M}^{(m)} = \left[\begin{array}{ccc} \boldsymbol{\mu}_1^{(m)}, & \ldots & ,\boldsymbol{\mu}_P^{(m)} \end{array} \right]$$

The complete canonical model $\mathcal{M}$ consists of[11]

$$\mathcal{M} = \left\{ \{\mathbf{M}^{(1)}, \cdots, \mathbf{M}^{(M)}\}, \{\mathbf{\Sigma}^{(1)}, \cdots, \mathbf{\Sigma}^{(M)}\} \right\}$$

where M is the total number of components. The transform parameters in CAT are the interpolation weights $\boldsymbol{\lambda}^{(sr)}$, associated with the regression base class r and each acoustic condition s

$$\boldsymbol{\lambda}^{(sr)} = \left[\begin{array}{ccc} \lambda_1^{(sr)}, & \ldots & ,\lambda_P^{(sr)} \end{array} \right]^T \tag{3.43}$$

where $\lambda_p^{(sr)}$ is the interpolation weight for cluster p. In some systems a bias cluster is used where $\lambda_P^{(sr)} = 1$ for all acoustic conditions [36]. The adapted mean for a particular acoustic condition s, can then be written as

$$\hat{\boldsymbol{\mu}}^{(sm)} = \mathbf{M}^{(m)} \boldsymbol{\lambda}^{(sr_m)} \tag{3.44}$$

where r_m is the regression base class that component m belongs to. There are two kinds of CAT models [36], which differ in the representation of the cluster means. In *model based CAT*, the clusters are represented as a distinct set of mean vectors, which is the focus of this work. An alternative is *transform based CAT*, where the clusters are represented by a set of cluster-specific transforms and a single standard HMM set. This yields a more compact cluster representation. Details of transform based CAT can be found in [36].

ML update of CAT model parameters and interpolation weights

[11] Gaussian component weights and transition matrices are not included as their estimation is similar to the standard estimation schemes.

Maximum likelihood (ML) CAT training is used to find both sets of parameters maximising the likelihood criterion. As in linear transform based adaptive training, the two updates are interleaved. Again, the EM algorithm is used to iteratively estimate the parameters. The auxiliary function of the canonical model parameters for CAT can be written as

$$\begin{aligned}\mathcal{Q}_{\texttt{ML}}(\mathcal{M};\hat{\mathcal{M}},\hat{\boldsymbol{\mathcal{T}}}) &= -\frac{1}{2}\sum_{s,m,t}\gamma_m^{\texttt{ML}}(t)\Big\{\log\left|\boldsymbol{\Sigma}^{(m)}\right| \\ &\quad + \left(\mathbf{o}_t^{(s)}-\mathbf{M}^{(m)}\boldsymbol{\lambda}^{(sr_m)}\right)^T\boldsymbol{\Sigma}^{(m)-1}\left(\mathbf{o}_t^{(s)}-\mathbf{M}^{(m)}\boldsymbol{\lambda}^{(sr_m)}\right)\Big\} \end{aligned} \quad (3.45)$$

where $\hat{\boldsymbol{\mathcal{T}}}$ is the set of interpolation weights consisting of $\boldsymbol{\lambda}^{(sr)}$, and $\gamma_m^{\texttt{ML}}(t)$ is the posterior occupancy of Gaussian component m calculated given the current CAT model and weights estimate. By differentiating equation (3.45) with respect to CAT model parameters and equating it to zero, ML re-estimation formulae can be derived [36]. Given the sufficient statistics

$$\gamma_m^{\texttt{ML}} = \sum_{s,t}\gamma_m^{\texttt{ML}}(t) \quad (3.46)$$

$$\mathbf{G}_{\texttt{ML}}^{(m)} = \sum_{s,t}\gamma_m^{\texttt{ML}}(t)\boldsymbol{\lambda}^{(sr_m)}\boldsymbol{\lambda}^{(sr_m)T} \quad (3.47)$$

$$\mathbf{K}_{\texttt{ML}}^{(m)} = \sum_{s,t}\gamma_m^{\texttt{ML}}(t)\boldsymbol{\lambda}^{(sr_m)}\mathbf{o}_t^{(s)T} \quad (3.48)$$

$$\mathbf{L}_{\texttt{ML}}^{(m)} = \sum_{s,t}\gamma_m^{\texttt{ML}}(t)\mathbf{o}_t^{(s)}\mathbf{o}_t^{(s)T} \quad (3.49)$$

the mean and covariance matrix can be updated by

$$\hat{\mathbf{M}}^{(m)T} = \mathbf{G}_{\texttt{ML}}^{(m)-1}\mathbf{K}_{\texttt{ML}}^{(m)} \quad (3.50)$$

$$\hat{\boldsymbol{\Sigma}}^{(m)} = \text{diag}\Big(\frac{\mathbf{L}_{\texttt{ML}}^{(m)}-\hat{\mathbf{M}}^{(m)}\mathbf{K}_{\texttt{ML}}^{(m)}}{\gamma_m^{\texttt{ML}}}\Big) \quad (3.51)$$

The auxiliary function for the weights update is defined for each homogeneous data block by [36]

$$\mathcal{Q}_{\texttt{ML}}(\mathcal{T}^{(s)};\hat{\mathcal{T}}^{(s)},\hat{\mathcal{M}}) = -\frac{1}{2}\sum_{m,t}\gamma_m^{\texttt{ML}}(t)\left(\mathbf{o}_t^{(s)}-\mathbf{M}^{(m)}\boldsymbol{\lambda}^{(sr_m)}\right)^T\boldsymbol{\Sigma}^{(m)-1}\left(\mathbf{o}_t^{(s)}-\mathbf{M}^{(m)}\boldsymbol{\lambda}^{(sr_m)}\right) \quad (3.52)$$

where the transform $\mathcal{T}^{(s)}$ is the interpolation weight vector. From the auxiliary function, the weight vector associated with regression class r and acoustic condition s can be estimated as

$$\hat{\boldsymbol{\lambda}}^{(sr)} = \mathbf{G}_{\texttt{ML}}^{(sr)-1}\mathbf{k}_{\texttt{ML}}^{(sr)} \quad (3.53)$$

where the sufficient statistics are given by

$$\mathbf{G}_{\texttt{ML}}^{(sr)} = \sum_{m\in M_r}\left(\sum_t\gamma_m^{\texttt{ML}}(t)\right)\mathbf{M}^{(m)T}\boldsymbol{\Sigma}^{(m)-1}\mathbf{M}^{(m)} \quad (3.54)$$

$$\mathbf{k}_{\texttt{ML}}^{(sr)} = \sum_{m\in M_r}\mathbf{M}^{(m)T}\boldsymbol{\Sigma}^{(m)-1}\left(\sum_t\gamma_m^{\texttt{ML}}(t)\mathbf{o}_t^{(s)}\right) \quad (3.55)$$

where M_r is the Gaussian component group of regression base class r. Comparing the interpolate weight update to the linear transform update in SAT, the number of parameters to be estimated is usually significantly smaller. This results in a lower cost of computation and memory. Hence, CAT is more suitable for rapid adaptation.

Initialisation in CAT training

When training a model using the EM algorithm, initialisation is always an important issue. For CAT, it is possible to either initialise the interpolation weights first or to initialise the multiple-cluster model first [36]. In both schemes, the posterior probability of each Gaussian component at time t, $\gamma_m^{\mathtt{ML}}(t)$, is required. This is typically obtained from a multi-style trained model, such as a speaker-independent (SI) model.

For state-of-the-art systems, there are often a large number of model parameters. Directly constructing multiple-cluster models is expensive. Thus in practice, interpolation weights are often initialised first. From the sufficient statistics in equations (3.46) to (3.49), given a set of initial weights and $\gamma_m^{\mathtt{ML}}(t)$ obtained from a standard SI model, an initial CAT model can be constructed and the iterative CAT training may continue. The procedure is illustrated in Figure 3.5.

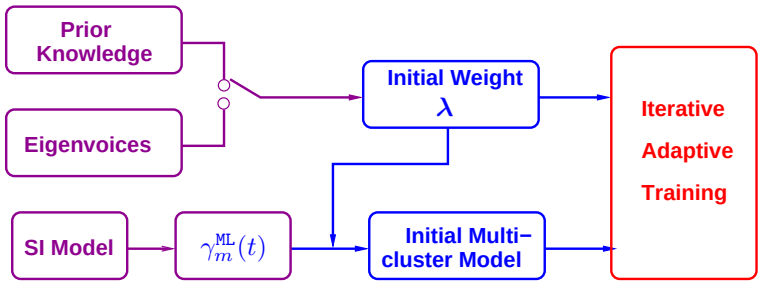


Figure 3.5 *Training initialisation of a CAT system*

There are two schemes for the weights initialisation.

- **Clustering based approach**.

  If some prior knowledge related to acoustic condition clustering is available, the training data of each homogeneous block can be assigned an initial weight vector based on this knowledge. Each element of the vector is a $0/1$ value, according to which cluster it belongs to. For example, with gender information, an initial 2-cluster weight vector, $[0, 1]$ or $[1, 0]$ can be assigned to male and female speakers respectively. If there is no prior cluster knowledge available, standard automatic clustering approaches may be used to obtain the initial assignment of each cluster [67].

- **Eigen-decomposition approach**.

Eigen-decomposition approach was first proposed for speaker adaptation [69] and can be used as an initialisation scheme for the interpolation weights. Similar to speaker clustering, a number of simple speaker-dependent models may be trained first. The means of each model are then concatenated to form a single *meta-vector*, i.e., one meta-vector for each speaker. Principal component analysis (PCA) or linear discriminant analysis (LDA) is then performed on the set of meta-vectors. Several orthogonal eigen-meta-vectors, or *eigenvoices*, with the largest eigenvalues are selected as a basis set. The number of the selected eigenvoices will give the number of clusters. Given the basis eigenvoices and the meta-vectors for each speaker, initial weights for each speaker can be obtained using either a projection scheme or a ML based approach [36, 68]. This approach will naturally output a bias cluster, i.e. a cluster with a weight value of 1.0, which is the mean of all meta-vectors. During CAT training, the weight of this bias cluster can either be fixed as 1.0 or updated similarly to the other weights [36]. An advantage here is, by using the eigenvoices approach, an arbitrary number of clusters can be initialised without knowing prior knowledge.

Eigenvoices systems, a multiple-cluster scheme similar to CAT, are constructed by further updating the basis initialised by the eigen-decomposition approach [69, 68]. It is interesting to briefly contrast the two. Both systems use a set of distinct mean vectors. The "eigenvoices" correspond to the cluster mean matrices in the CAT model. An eigenvoices system always uses an eigen-decomposition initialisation approach based on PCA or LDA. But CAT systems may also use prior knowledge for initialisation. For some eigenvoices systems, the initialised basis eigenvoices are not updated but directly used in adaptation and decoding [68], while CAT systems always update the multiple-cluster model [36]. If eigenvoices are updated using the maximum likelihood eigenspace (MLES) approach [88], it is equivalent to updating CAT cluster mean matrices and leaving covariance matrices unchanged [36][12]. Due to this close-relationship, the discriminative and Bayesian training techniques for CAT described in later chapters can also be used in eigenvoices systems.

Recognition using a CAT model

During recognition, the initialisation of test interpolation weights is also required. From the sufficient statistics $\mathbf{G}^{(s)}$ and $\mathbf{k}^{(s)}$ in the weights update formulae in equation (3.53), to initialise weights, the only requirement is the multiple-cluster model and $\gamma_m^{\mathtt{ML}}(t)$, which may be obtained from a multi-style trained standard model, e.g. SI model. Then further weights estimation can be performed in a similar iterative fashion as in training. This adaptation procedure is illustrated in Figure 3.6.

Comparing this to the training initialisation, no prior knowledge or eigenvoices approach is

[12]The MLES approach in [88] was used as a simple extension of the PCA approach. The update of eigenvoices and interpolation weights were not interleaved during training. To the author's knowledge, eigenvoices have not been discussed in an adaptive training framework before.

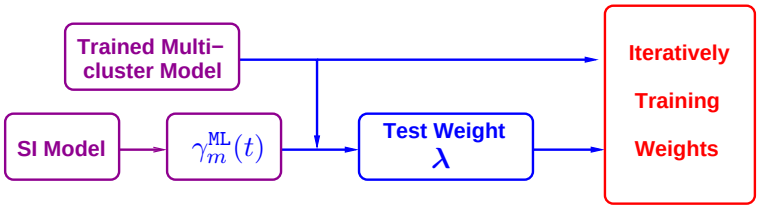


Figure 3.6 *Testing adaptation initialisation of a CAT system*

required in adaptation. The test weights can be initialised naturally. This makes the use of CAT systems straightforward. It is also interesting to compare the number of parameters required for CAT adaptation to SAT (linear transform based) adaptation. The number of interpolation weights is normally significantly less than the number of linear transforms. For instance, for very little adaptation data, only a global transform can be generated. Using CAT weights as the transform, the number of parameters is only P, which is the cluster number, normally less than 10. When using a full MLLR transform as the transform, the number of parameters is $D\times(D+1)$, where D is the number of dimensions, typically 39. It is obvious that the number of CAT weight parameters is considerably less than the number of MLLR parameters. This characteristic makes CAT very useful for *rapid* and *robust* adaptation. However, as fewer parameters are used to tune the canonical model, the adaptation power of CAT is limited compared to SAT.

3.4.3 Multiple Acoustic Factors and Structured Transforms

The adaptive training described before uses only one set of transforms to represent all kinds of non-speech variabilities. However, for highly non-homogeneous data, there may be multiple acoustic factors affecting the speech signal. For instance, speaker variations and environmental noise may exist at the same time. Using one set of transforms to model the whole unwanted speech variability may not be powerful enough. More recent schemes use multiple sets of transforms, denoted here as *structured transforms* (ST), to represent complex non-speech variabilities within an adaptive training framework [136, 37]. When two acoustic factors exist, the DBN for adaptive training and adaptation with ST is shown in figure 3.7.

Where ω_t is the hidden state and $\mathbf{o}_t$ is the observation vector. The structured transform consists of two types of transforms: $\boldsymbol{\lambda}$ and $\mathbf{W}$. From figure 3.7, $\boldsymbol{\lambda}$ and $\mathbf{W}$ are associated with the same homogeneous block, and are used to represent distinct acoustic factors, such as environmental noise and speaker. Therefore, the unwanted acoustic variabilities are modelled in a structural way. This type of structural modelling is also referred to as *acoustic factorisation* [37]. Various structured transforms have been examined. For example, the *parallel model combination* (PMC) technique was combined with spectral subtraction as a ST and obtained improvements on a small vocabulary task [30]. CAT was combined with MLLR as a model based ST and outperformed adaptive training with one set of transforms [37]. In this thesis, only model based ST is considered.

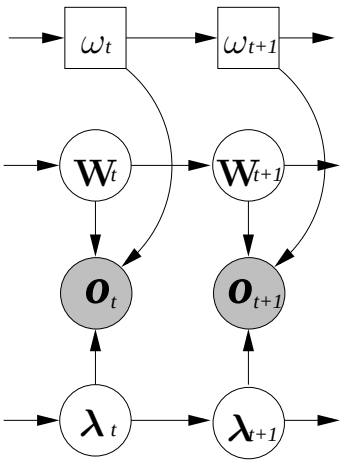


Figure 3.7 *Dynamic Bayesian network for adaptive HMM with structured transforms*

There are three main forms of model based transforms: MLLR transforms, CMLLR transforms and CAT weights. Although the ST of MLLR combined with CAT may yield good performance [37], it requires a considerably large memory during training due to the nature of the model update [39]. It is therefore not used for adaptive training in this work. Instead, another kind of ST, which combines CAT weights with CMLLR transforms [136], is investigated for adaptive training. When using this form of ST, the canonical model is a multiple-cluster model. However, in addition to CAT, a transform on the feature space is also applied. Then the multiple-cluster model is trained in a transformed feature space, where the observations are transformed using CMLLR as in equation (3.39). The auxiliary function for the canonical model update can be expressed as [137]

$$\begin{aligned}\mathcal{Q}_{\mathtt{ML}}(\mathcal{M};\hat{\mathcal{M}},\hat{\boldsymbol{\mathcal{T}}}) &= -\frac{1}{2}\sum_{s,m,t}\gamma_m^{\mathtt{ML}}(t)\Big\{\log\left|\boldsymbol{\Sigma}^{(m)}\right| \\ &\quad + \left(\hat{\mathbf{o}}_t^{(sr_m^W)} - \mathbf{M}^{(m)}\boldsymbol{\lambda}^{(sr_m^\lambda)}\right)^T \boldsymbol{\Sigma}^{(m)-1}\left(\hat{\mathbf{o}}_t^{(sr_m^W)} - \mathbf{M}^{(m)}\boldsymbol{\lambda}^{(sr_m^\lambda)}\right)\Big\} \quad (3.56)\end{aligned}$$

where $\boldsymbol{\mathcal{T}}$ is now the structured transform, s denotes the index of the homogeneous block, r_m^W is the CMLLR regression class which component m belongs to, r_m^λ is the CAT weight regression base class for component m, $\gamma_m^{\mathtt{ML}}(t)$ is the posterior occupancy calculated using the current model and ST estimates and $\hat{\mathbf{o}}_t^{(sr_m^W)}$ is the transformed observation for acoustic condition s using the CMLLR transform associated with regression base class r_m^W, which is obtained using equation (3.39).

As the auxiliary function suggests, the estimation of the multiple-cluster canonical model is a simple extension to the model based CAT estimation approach. Re-estimation formulae are the same as equations (3.50) and (3.51) except that the transformed observation $\hat{\mathbf{o}}_t^{(sr_m^W)}$ is used in the sufficient statistics. Transform estimation also requires little modification, where given the interpolation weights, the adapted mean, $\hat{\boldsymbol{\mu}}^{(sm)} = \mathbf{M}^{(m)}\boldsymbol{\lambda}^{(sr_m^\lambda)}$ is used instead of $\boldsymbol{\mu}^{(m)}$ in the statistics in equation (3.20) to estimate the CMLLR transform. Then the interpolation weights

are estimated using equation (3.53) with the transformed features $\hat{\mathbf{o}}_t^{(sr_m^W)}$.

The overall process of adaptive training with ST is also an interleaving update process similar to standard adaptive training except that the transform update step consists of two sub-steps for updating CAT weights and CMLLR respectively. The procedure in this work is described as below:

1. Initialise canonical model $\hat{\mathcal{M}}_0$, constrained MLLR $\hat{\mathbf{W}}_0$ and CAT weights $\hat{\boldsymbol{\lambda}}_0$, set $k = 0$.
2. Estimate $\hat{\mathbf{W}}_{k+1}$ given $\hat{\mathcal{M}}_k$, $\hat{\mathbf{W}}_k$ and $\hat{\boldsymbol{\lambda}}_k$
3. Estimate $\hat{\boldsymbol{\lambda}}_{k+1}$ given $\hat{\mathcal{M}}_k$, $\hat{\mathbf{W}}_{k+1}$ and $\hat{\boldsymbol{\lambda}}_k$
4. Estimate $\hat{\mathcal{M}}_{k+1}$ given $\hat{\mathcal{M}}_k$, $\hat{\mathbf{W}}_{k+1}$ and $\hat{\boldsymbol{\lambda}}_{k+1}$
5. $k = k + 1$. Goto 2 until convergence.

The adaptation process also uses the two sub-steps of estimating constrained MLLR and CAT interpolation weights. The final constrained MLLR is used to transform the features and the CAT interpolation weights are used to construct the adapted model for recognition.

3.5 Summary

This section has reviewed the basic concepts and standard schemes of adaptation and adaptive training. Adaptation is proposed to compensate for the acoustic mismatch between training data and test data. A well trained model is modified according to some supervision data to match each target acoustic domain. Adaptation may be run in either supervised or unsupervised modes depending on the availability of correct transcriptions. From another point of view, it may be run in either batch or incremental modes according to when adaptation data becomes available. Standard adaptation schemes include MAP, MLLR and CMLLR. To overcome the incorrect hypothesis problem in unsupervised adaptation, extended adaptation techniques based on confidence scores or lattices may be used. Adaptation may also be used during training to build systems on non-homogeneous data, referred to as adaptive training. Here, a set of transforms is used to represent the unwanted acoustic variabilities. A canonical model is constructed which represents only the underlying speech variability. This canonical model is then adapted to a particular test speaker or environment using the transforms. Feature normalisation may be viewed as an adaptive training technique. Widely used approaches include CMN, CVN, Gaussianisation and VTLN. They are all simple techniques and may be combined with complex model based adaptive training techniques. There are two main categories of model based schemes: linear transforms and cluster based adaptive training. MLLR and CMLLR are commonly used linear transforms to build adaptively trained systems. Cluster dependent models can be viewed as a special case of cluster adaptive training (CAT), which uses a multiple-cluster canonical model and uses interpolation weights to represent non-speech variabilities. For highly non-homogeneous training data, where multiple acoustic factors affect the speech signals, structured transforms may be used to model each acoustic factor respectively. Maximum likelihood (ML) estimation of canonical model/transform parameters is discussed in detail in this chapter.

4

Discriminative Adaptive Training

The previous chapter has described adaptive training and adaptation schemes with the maximum likelihood (ML) criterion. For ASR systems, standard ML training suffers from the incorrect modelling problem of HMMs as discussed in section 2.3.4. Discriminative criteria have been proposed to solve this problem. This chapter investigates discriminative training within the framework of adaptive training. A simplified discriminative adaptive training strategy is adopted in this thesis, in which only the canonical model is discriminatively updated given a set of ML estimated transforms. This will be discussed in section 4.1. Depending on the type of transform used, discriminative adaptive training can also be classified as two broad categories: linear transform based or cluster based schemes. Linear transform based one has been previously proposed. In this work, a new discriminative cluster adaptive training is discussed in detail. As the minimum phone error (MPE) criterion has been shown to give good performance improvements for LVCSR tasks, it is used to derive the discriminative update formulae of Gaussian mean and covariance parameters [1].

4.1 Simplified Discriminative Adaptive Training

An ideal discriminative adaptive training paradigm is to discriminatively update both the canonical model and the transform parameters during training. In adaptation, given the discriminative canonical model and some supervision data, a transform is discriminatively estimated for each test homogeneous data block. The adapted discriminative model is then used in recognition. In this paradigm, the use of discriminative criterion to estimate the transform parameters in both training and adaptation gives good discrimination ability of the final adapted model.

However, this paradigm is impractical when the canonical model is used in unsupervised adaptation. As indicated in section 2.4, discriminative training requires the correct transcription. Unfortunately, in unsupervised adaptation, there is no correct transcription available. Though it is possible to generate multiple hypothesis candidates and regard the best one as the "correct

[1] Refer to [90] for the MPE update of transition probability and Gaussian component weight as they are similar to the standard discriminative estimation schemes.

transcription", this errorful "transcription" is not likely to give correct discrimination information for testing transform estimation. It has been found that discriminative training is more sensitive to the quality of the supervision than ML training. Though discriminatively trained transforms can lead to good gain in supervised adaptation [113], the gain in unsupervised adaptation is found to be greatly reduced [119]. Unsupervised discriminative adaptation is still an open problem. A more detailed discussion can be found in [118].

A simple way to avoid the unsupervised discriminative adaptation problem is to use the ML criterion to update transforms during adaptation. However, there is a criterion mismatch if the transforms are discriminatively updated during training. To keep transform update criterion consistent in both training and unsupervised testing adaptation, an alternative training paradigm is often used [78]. This paradigm is sometimes referred to as the *simplified discriminative adaptive training* [119]. It has been found to yield about the same performance as discriminatively training both sets of parameters. However it is simpler and more consistent when dealing with unsupervised adaptation tasks. The basic training procedures are as follows.

1. Perform standard ML adaptive training resulting in a canonical model estimate $\hat{\mathcal{M}}_{\texttt{ML}}$ and a set of ML transform estimates $\hat{\boldsymbol{\mathcal{T}}}_{\texttt{ML}} = \{\hat{\mathcal{T}}^{(1)}_{\texttt{ML}}, \cdots, \hat{\mathcal{T}}^{(S)}_{\texttt{ML}}\}$.

2. Fixing the ML transform estimates, the canonical model is discriminatively updated. After several iterations, a discriminative canonical model, $\hat{\mathcal{M}}_{\texttt{MPE}}$, is produced.

In adaptation, given the discriminatively trained model $\hat{\mathcal{M}}_{\texttt{MPE}}$, ML criterion is again used to estimate transform parameters for each homogeneous block of the test data.

As unsupervised adaptation is the focus of this work, the simplified discriminative adaptive training strategy is adopted. Therefore, in the below sections, the discriminative update of the canonical model is the main concern. Discriminative transform estimation is only briefly reviewed.

4.2 Linear Transform Based Discriminative Adaptive Training

The linear transform based discriminative adaptive training has been previously investigated [60, 47]. The most commonly used linear transforms are mean transforms [74] and constrained transforms [34]. Discriminative adaptive training with the two types of transforms are reviewed in this section. A more detailed review can be found in [118].

4.2.1 DAT with Mean Transform

ML adaptive training with mean transform is the form of speaker adaptive training (SAT) discussed in section 3.4.1.1. This section will review discriminatively training the canonical model. As discussed previously, due to the use of the simplified discriminative adaptive training strategy, only the discriminative update of the canonical model is investigated. For the discriminative

estimation of mean transforms, an MMI implementation was first introduced in [114] and an MPE implementation can be found in [118].

In early research, the MMI criterion was used to update the canonical model parameters given the mean transforms [60, 118]. This section presents the MPE update of the canonical model parameters. Similar to ML SAT, the mean vector and the covariance matrix of the canonical model are updated separately. The mean update is discussed first, followed by the covariance update.

The ML auxiliary function for SAT with mean transform was given in equation (3.34), which is rewritten as below

$$\begin{aligned}\mathcal{Q}_{\mathtt{ML}}(\mathcal{M};\hat{\mathcal{M}},\hat{\boldsymbol{\mathcal{T}}}) &= -\frac{1}{2}\sum_{s,m,t}\gamma_m^{\mathtt{ML}}(t)\Big\{\log\left|\boldsymbol{\Sigma}^{(m)}\right| \\ &\quad + \left(\mathbf{o}_t^{(s)} - \mathbf{W}^{(sr_m)}\boldsymbol{\xi}^{(m)}\right)^T \boldsymbol{\Sigma}^{(m)-1}\left(\mathbf{o}_t^{(s)} - \mathbf{W}^{(sr_m)}\boldsymbol{\xi}^{(m)}\right)\Big\} \end{aligned} \tag{4.1}$$

It can also be re-expressed in terms of a set of sufficient statistics as in discriminative training of standard HMMs. Considering only the mean update

$$\mathcal{Q}_{\mathtt{ML}}(\mathcal{M};\hat{\mathcal{M}},\hat{\boldsymbol{\mathcal{T}}}) = \mathcal{G}_{\mathtt{SAT}}(\mathcal{M};\boldsymbol{\Gamma}_{\mathtt{ML}}) = -\frac{1}{2}\sum_m\left\{\boldsymbol{\mu}^{(m)T}\mathbf{G}_{\mathtt{ML}}^{(m)}\boldsymbol{\mu}^{(m)} - 2\boldsymbol{\mu}^{(m)T}\mathbf{k}_{\mathtt{ML}}^{(m)}\right\} \tag{4.2}$$

where the sufficient statistics is

$$\boldsymbol{\Gamma}_{\mathtt{ML}} = \left\{\mathbf{G}_{\mathtt{ML}}^{(m)}, \mathbf{k}_{\mathtt{ML}}^{(m)}\right\} \tag{4.3}$$

are defined in equation (3.35) and equation (3.36).

To get an MPE estimate of the parameters, a weak-sense auxiliary function similar to equation (2.68) may be used. Compared to the ML SAT auxiliary function in equation (4.2), the numerator part of the weak-sense auxiliary function uses numerator posterior occupancy $\gamma_m^{\mathtt{n}}(t)$ instead of $\gamma_m^{\mathtt{ML}}(t)$ in the sufficient statistics $\boldsymbol{\Gamma}_{\mathtt{n}}$. Similarly, the denominator part yields the statistics $\boldsymbol{\Gamma}_{\mathtt{d}}$ by using $\gamma_m^{\mathtt{d}}(t)$.

As in standard discriminative training, a smoothing function is also required and it must yield the current model parameters, $\hat{\mathcal{M}}$, as a maximum to meet the constraint in equation (2.59). The generic smoothing function for standard HMMs was introduced in [86] and [102]. Since in adaptive training the adapted model is dependent on acoustic conditions, one approach is to define a smoothing function at acoustic condition level [60]. Then, the generic form of the smoothing function for both sets of parameters in adaptive training may be written as

$$\mathcal{S}(\mathcal{M},\boldsymbol{\mathcal{T}};\hat{\mathcal{M}},\hat{\boldsymbol{\mathcal{T}}}) = \sum_{s,m} D_m\nu_m^{(s)}\int_{\mathbf{o}} p(\mathbf{o}|m,\hat{\mathcal{M}},\hat{\mathcal{T}}^{(s)})\log p(\mathbf{o}|m,\mathcal{M},\mathcal{T}^{(s)})\,d\mathbf{o} \tag{4.4}$$

where s is the index of acoustic condition, $\nu_m^{(s)}$ is a weight for different acoustic conditions and will be defined in equation (4.6) later. As shown in appendix A, this smoothing function also satisfies the constraint in equation (2.59). The exact form for mean update is then expressed as

$$\mathcal{S}(\mathcal{M};\hat{\mathcal{M}},\hat{\boldsymbol{\mathcal{T}}}) = -\frac{1}{2}\sum_{s,m} D_m\nu_m^{(s)}\left(\boldsymbol{\xi}^{(m)} - \hat{\boldsymbol{\xi}}_{\mathtt{c}}^{(m)}\right)^T\mathbf{W}^{(sr_m)T}\boldsymbol{\Sigma}^{(m)-1}\mathbf{W}^{(sr_m)}\left(\boldsymbol{\xi}^{(m)} - \hat{\boldsymbol{\xi}}_{\mathtt{c}}^{(m)}\right) \tag{4.5}$$

where $\mathbf{W}^{(sr)} = [\mathbf{A}^{(sr)} \ \mathbf{b}^{(sr)}]$ is the extended mean transform associated with regression base class r and acoustic condition s, r_m is the regression base class that component m belongs to, $\boldsymbol{\xi} = [\boldsymbol{\mu}^T 1]^T$ is the extended mean vector and $\hat{\boldsymbol{\xi}}_{\mathtt{c}}$ is the extended mean of the current model set. D_m is a positive smoothing constant for component m to control the impact of the smoothing function and make the optimisation stable. This smoothing function is a valid smoothing function for all values of $\nu_m^{(s)}$. However, rather than using a constant value for all acoustic conditions, it is more appropriate to use this value to reflect the proportions of data for the particular component of an acoustic condition. In this work it is set as

$$\nu_m^{(s)} = \frac{\sum_t \gamma_m^{\mathtt{n}}(t)}{\sum_{s,t} \gamma_m^{\mathtt{n}}(t)} \tag{4.6}$$

where the summation in the numerator only involves data associated with a particular acoustic condition s. It is usual to re-express equation (4.5) in terms of sufficient statistics as the same form as equation (4.2)

$$\mathcal{S}(\mathcal{M}; \hat{\mathcal{M}}, \hat{\boldsymbol{\mathcal{T}}}) = \mathcal{G}_{\mathtt{SAT}}(\mathcal{M}; \boldsymbol{\Gamma}_{\mathtt{s}}) \tag{4.7}$$

where

$$\boldsymbol{\Gamma}_{\mathtt{s}} = \left\{ D_m \mathbf{G}_{\mathtt{s}}^{(m)}, D_m \mathbf{k}_{\mathtt{s}}^{(m)} \right\} \tag{4.8}$$

and

$$\mathbf{G}_{\mathtt{s}}^{(m)} = \sum_s \nu_m^{(s)} \mathbf{A}^{(sr_m)T} \boldsymbol{\Sigma}^{(m)-1} \mathbf{A}^{(sr_m)} \tag{4.9}$$

$$\mathbf{k}_{\mathtt{s}}^{(m)} = \mathbf{G}_{\mathtt{s}}^{(m)} \hat{\boldsymbol{\xi}}_{\mathtt{c}}^{(m)} \tag{4.10}$$

As only the mean update is concerned here, D_m only needs to be selected to ensure a positive MPE occupancy count. It may be simply set as

$$D_m = E\gamma_m^{\mathtt{d}} \tag{4.11}$$

where $\gamma_m^{\mathtt{d}}$ is the denominator occupancy for component m and E is 1 or 2 as suggested in [90].

The I-smoothing distribution used for the mean update is a Normal distribution, which is part of the Normal-Wishart distribution in equation (2.69) in standard discriminative training. Here it is defined at acoustic condition level as well. Ignoring constants, it is expressed as

$$\log p(\mathcal{M}|\Phi) = -\frac{\tau^I}{2} \sum_{s,m} \tilde{\nu}_m^{(s)} \left(\mathbf{W}^{(sr_m)} \boldsymbol{\xi}^{(m)} - \tilde{\boldsymbol{\mu}}^{(sm)} \right)^T \boldsymbol{\Sigma}^{(m)-1} \left(\mathbf{W}^{(sr_m)} \boldsymbol{\xi}^{(m)} - \tilde{\boldsymbol{\mu}}^{(sm)} \right) \tag{4.12}$$

where τ^I is the parameter to control impact of the prior, $\tilde{\boldsymbol{\mu}}^{(sm)}$ is the prior mean of the m^{th} component for acoustic condition s, $\tilde{\nu}_m^{(s)}$ is a slightly modified version of equation (4.6). Since in standard MPE training, ML estimates are often used as the priors, the ML posterior occupancy, $\gamma_m^{\mathtt{ML}}(t)$, are therefore used here to define $\tilde{\nu}_m^{(s)}$. The logarithm of the I-smoothing distribution may be re-expressed using the sufficient statistics as

$$\log p(\mathcal{M}|\Phi) = \mathcal{G}_{\mathtt{SAT}}(\mathcal{M}; \boldsymbol{\Gamma}_{\mathtt{p}}) \tag{4.13}$$

where

$$\mathbf{\Gamma}_{\mathtt{p}} = \left\{ \tau^I \mathbf{G}_{\mathtt{p}}^{(m)}, \tau^I \mathbf{K}_{\mathtt{p}}^{(m)} \right\} \tag{4.14}$$

and

$$\mathbf{G}_{\mathtt{p}}^{(m)} = \sum_s \tilde{\nu}_m^{(s)} \mathbf{A}^{(sr_m)T} \mathbf{\Sigma}^{(m)-1} \mathbf{A}^{(sr_m)} \tag{4.15}$$

$$\mathbf{k}_{\mathtt{p}}^{(m)} = \sum_s \tilde{\nu}_m^{(s)} \mathbf{A}^{(sr_m)T} \mathbf{\Sigma}^{(m)-1} \left(\tilde{\boldsymbol{\mu}}^{(sm)} - \mathbf{b}^{(sr_m)} \right) \tag{4.16}$$

The prior mean $\tilde{\boldsymbol{\mu}}^{(sm)}$ may have various forms. One choice is to use the ML estimate as in standard discriminative training, i.e.,

$$\tilde{\boldsymbol{\mu}}^{(sm)} = \mathbf{A}^{(sr_m)} \hat{\boldsymbol{\mu}}_{\mathtt{ML}}^{(m)} + \mathbf{b}^{(sr_m)} \tag{4.17}$$

where $\hat{\boldsymbol{\mu}}_{\mathtt{ML}}^{(m)}$ is the ML estimate obtained using equation (3.37). In this case, it is trivial to prove that $\mathbf{G}_{\mathtt{p}}^{(m)}$ and $\mathbf{k}_{\mathtt{p}}^{(m)}$ are the ML statistics given in equation (3.35) and equation (3.36), normalised by the ML posterior occupancy $\gamma_m^{\mathtt{ML}}$. This is the form considered in this thesis. MMI estimate or static model parameters may also be used as the prior, though to the author's knowledge, there is no report on using the two priors for mean transform based DAT. A more detailed discussion about the I-smoothing prior will be given in section 4.3.2.2.

Given the above definitions of statistics, the overall weak-sense auxiliary function may be expressed as

$$\mathcal{Q}_{\mathtt{MPE}}(\mathcal{M}; \hat{\mathcal{M}}, \hat{\boldsymbol{\mathcal{T}}}) = \mathcal{G}_{\mathtt{SAT}}(\mathcal{M}; \mathbf{\Gamma}_{\mathtt{n}}) - \mathcal{G}_{\mathtt{SAT}}(\mathcal{M}; \mathbf{\Gamma}_{\mathtt{d}}) + \mathcal{G}_{\mathtt{SAT}}(\mathcal{M}; \mathbf{\Gamma}_{\mathtt{s}}) + \mathcal{G}_{\mathtt{SAT}}(\mathcal{M}; \mathbf{\Gamma}_{\mathtt{p}}) \tag{4.18}$$

where $\mathcal{G}_{\mathtt{SAT}}(\cdot)$ is defined for mean update only, as in equation (4.2). It is then trivial to show that the mean vector can be updated by

$$\hat{\boldsymbol{\mu}}^{(m)} = \mathbf{G}_{\mathtt{MPE}}^{(m)^{-1}} \mathbf{k}_{\mathtt{MPE}}^{(m)} \tag{4.19}$$

where

$$\mathbf{G}_{\mathtt{MPE}}^{(m)} = \mathbf{G}_{\mathtt{n}}^{(m)} - \mathbf{G}_{\mathtt{d}}^{(m)} + D_m \mathbf{G}_{\mathtt{s}}^{(m)} + \tau^I \mathbf{G}_{\mathtt{p}}^{(m)} \tag{4.20}$$

$$\mathbf{k}_{\mathtt{MPE}}^{(m)} = \mathbf{k}_{\mathtt{n}}^{(m)} - \mathbf{k}_{\mathtt{d}}^{(m)} + D_m \mathbf{k}_{\mathtt{s}}^{(m)} + \tau^I \mathbf{k}_{\mathtt{p}}^{(m)} \tag{4.21}$$

The covariance matrices are discriminatively estimated after the mean update. During this estimation, each part of the weak-sense auxiliary function is re-arranged for covariance matrices update. The ML auxiliary function in equation (4.1) may be re-expressed for covariance matrices update as following

$$\mathcal{Q}_{\mathtt{ML}}(\mathcal{M}; \hat{\mathcal{M}}, \hat{\boldsymbol{\mathcal{T}}}) = \mathcal{G}_{\mathtt{SAT}}(\mathcal{M}; \mathbf{\Gamma}_{\mathtt{ML}}) = -\frac{1}{2} \sum_m \left\{ \gamma_m^{\mathtt{ML}} \log \left| \mathbf{\Sigma}^{(m)} \right| + \mathrm{tr} \left(\mathbf{L}_{\mathtt{ML}}^{(m)} \mathbf{\Sigma}^{(m)-1} \right) \right\} \tag{4.22}$$

where

$$\mathbf{\Gamma}_{\mathtt{ML}} = \left\{ \gamma_m^{\mathtt{ML}}, \mathbf{L}_{\mathtt{ML}}^{(m)} \right\} \tag{4.23}$$

and

$$\gamma_m^{\mathtt{ML}} = \sum_{s,t} \gamma_m^{\mathtt{ML}}(t) \tag{4.24}$$

$$\mathbf{L}_{\mathtt{ML}}^{(m)} = \sum_{s,t} \gamma_m^{\mathtt{ML}}(t) \left(\mathbf{o}_t^{(s)} - \mathbf{W}^{(sr_m)}\boldsymbol{\xi}^{(m)}\right)\left(\mathbf{o}_t^{(s)} - \mathbf{W}^{(sr_m)}\boldsymbol{\xi}^{(m)}\right)^T \tag{4.25}$$

The numerator part can be defined similarly as the ML auxiliary function except for using the numerator occupancy $\gamma_m^{\mathtt{n}}(t)$ instead of $\gamma_m^{\mathtt{ML}}(t)$. So does the denominator part. As shown in appendix A, the smoothing function for covariance matrices can also be derived from the general form in equation (4.4),

$$\mathcal{S}(\mathcal{M};\hat{\mathcal{M}},\hat{\boldsymbol{\mathcal{T}}}) = \mathcal{G}_{\mathtt{SAT}}(\mathcal{M};\boldsymbol{\Gamma}_{\mathtt{s}}) = -\frac{1}{2}\sum_m D_m \left\{\log\left|\boldsymbol{\Sigma}^{(m)}\right| + \mathrm{tr}\left(\hat{\boldsymbol{\Sigma}}_{\mathtt{c}}^{(m)}\boldsymbol{\Sigma}^{(m)-1}\right)\right\} \tag{4.26}$$

The sufficient statistics are given by

$$\boldsymbol{\Gamma}_{\mathtt{s}} = \left\{D_m, D_m\hat{\boldsymbol{\Sigma}}_{\mathtt{c}}^{(m)}\right\} \tag{4.27}$$

where $\hat{\boldsymbol{\Sigma}}_{\mathtt{c}}^{(m)}$ is the covariance matrix of the m^{th} component of the current canonical model, D_m is the smoothing constant to control convergence, which is set using the standard way as in equation (2.62). The I-smoothing distribution for covariances is defined similarly to the smoothing function, which is also part of the Normal-Wishart distribution in equation (2.69)[2]

$$\log p(\mathcal{M}|\Phi) = \mathcal{G}_{\mathtt{SAT}}(\mathcal{M};\boldsymbol{\Gamma}_{\mathtt{p}}) = -\frac{\tau^I}{2}\sum_m \left\{\log\left|\boldsymbol{\Sigma}^{(m)}\right| + \mathrm{tr}\left(\tilde{\boldsymbol{\Sigma}}^{(m)}\boldsymbol{\Sigma}^{(m)-1}\right)\right\} \tag{4.28}$$

where the sufficient statistics are

$$\boldsymbol{\Gamma}_{\mathtt{p}} = \left\{\tau^I, \tau^I\tilde{\boldsymbol{\Sigma}}^{(m)}\right\} \tag{4.29}$$

and $\tilde{\boldsymbol{\Sigma}}^{(m)}$ is the prior covariance matrix of the m^{th} component. Again, as in standard discriminative training, the ML estimate $\hat{\boldsymbol{\Sigma}}_{\mathtt{ML}}^{(m)}$ in equation (3.38) is used as the prior in this work.

Given the above statistics, the overall weak-sense auxiliary function in equation (4.18) may be obtained. Differentiating the weak-sense auxiliary function with respect to covariances (assumed to be diagonal) parameters and equating to zero, closed-form formula is given by [118]

$$\hat{\boldsymbol{\Sigma}}^{(m)} = \mathrm{diag}\left(\frac{\mathbf{L}_{\mathtt{n}}^{(m)} - \mathbf{L}_{\mathtt{d}}^{(m)} + D_m\hat{\boldsymbol{\Sigma}}_{\mathtt{c}}^{(m)} + \tau^I\tilde{\boldsymbol{\Sigma}}^{(m)}}{\gamma_m^{\mathtt{n}} - \gamma_m^{\mathtt{d}} + D_m + \tau^I}\right) \tag{4.30}$$

As discussed in section 3.4.1.1, SAT with mean transforms demands a severe memory requirement. In DAT with mean transforms, due to the additional sufficient statistics to be accumulated, this memory load problem is even more severe. For example, for a state-of-the-art LVCSR system considered in this work which has about 6K state and 28 components per state, the memory requirement for MPE mean update is about 1.6G (39 dimensional feature is used). This requirement is too high to be fulfilled by our current computational resources. Therefore, DAT with mean transform can only be used for systems of small complexity, i.e., the total number of Gaussian components should be small enough to fit the memory limitation.

[2]As only covariance update is concerned, the prior mean vector is the same as the current mean vector. Hence equation (2.69) becomes equation (4.28).

4.2.2 DAT with Constrained Linear Transform

Constrained linear transforms are widely used as an alternative to unconstrained transforms as discussed in section 3.4.1.2. A constrained linear transform can be viewed as a feature space transform. The form of this transform is re-written here which is the same as equation (3.39)

$$\hat{\mathbf{o}}_t^{(sr_m)} = \mathbf{A}^{(sr_m)}\mathbf{o}_t^{(s)} + \mathbf{b}^{(sr_m)} = \mathbf{W}^{(sr_m)}\boldsymbol{\zeta}_t^{(s)} \tag{4.31}$$

where r_m is the regression base class of Gaussian component m, $\mathbf{W}^{(sr)} = [\mathbf{A}^{(sr)} \quad \mathbf{b}^{(sr)}]$ is the extended constrained transform associated with acoustic condition s and regression base class r, $\boldsymbol{\zeta}_t^{(s)} = [\mathbf{o}_t^{(s)T} \quad 1]^T$ is the extended observation, $\hat{\mathbf{o}}_t^{(sr_m)}$ is the transformed observation.

A discriminative update of constrained linear transforms was first proposed in [47], where the MMI criterion is used. MPE criterion has also been used for updating constrained linear transforms and achieved good performance [119]. Refer to [118] for details of discriminative constrained linear transforms. This work concentrates on the canonical model update.

Since constrained linear transforms may be implemented as feature transforms as shown in equation (4.31), the MPE update of the canonical model is fairly straightforward. The form of the auxiliary function is similar to those in section 2.4.1 except that the transformed observation $\hat{\mathbf{o}}_t^{(sr_m)}$ is used instead of the original observation $\mathbf{o}_t^{(s)}$. Hence, the update formulae for means and covariances are the same as equation (2.75) and equation (2.76). The only change introduced is to use the adapted features $\hat{\mathbf{o}}_t^{(sr_m)}$ in the statistics. There is no memory load problem for the canonical model update. Therefore, constrained transforms are widely used in discriminative adaptive training for LVCSR systems [25, 40, 85].

4.3 Cluster Based Discriminative Adaptive Training

The previous section reviews linear transform based discriminative adaptive training schemes. Cluster based adaptive training is an alternative scheme. In this scheme, multiple sets of HMMs are used as the canonical model. The adapted model is constructed by interpolating the parameters of different sets. Cluster dependent modelling can be viewed as a special type of cluster based adaptive training. Cluster adaptive training or eigenvoices, where soft interpolation weights are used, is a more generic form of cluster based adaptive training. The ML training scheme has been described in section 3.4.2. In this work, discriminative approaches to estimate both the multiple-cluster model and interpolation weights are proposed.

4.3.1 Discriminative Cluster-Dependent Model Training

One straightforward approach to train a cluster-dependent discriminative model is to simply split the whole training dataset into cluster-specific data blocks and train a set of HMMs for each cluster-specific data block with the standard discriminative training technique in section 2.4. This is a simple extension of the ML cluster-dependent model training. However, as the discriminative criteria take into account all competing hypotheses rather than the correct transcription

only, they require more data to give a good coverage of those hypotheses than ML training. Hence, the discriminatively trained model is more likely to be overtrained especially for LVCSR systems. As the number of parameters to be trained for cluster-dependent models is much larger than the standard model, this overtraining problem is even more severe. It has been found that, with the MPE criterion [91, 140], the above straightforward cluster-dependent training gave no gain, or even degradation, compared to a discriminative cluster-independent model trained on all data. Therefore, new techniques are required to build effective discriminative cluster-dependent models.

A solution to overcome the overtraining problem is to use more robust parameters in the I-smoothing distribution of the MPE criterion. The form of I-smoothing distribution has been given in equation (2.69) in section 2.4.2. It is a Normal-Wishart distribution for Gaussian parameters, where $\tilde{\boldsymbol{\mu}}^{(m)}$ and $\tilde{\boldsymbol{\Sigma}}^{(m)}$ are the priors. In contrast to using the ML estimates $\boldsymbol{\mu}_{\texttt{ML}}^{(m)}$ and $\boldsymbol{\Sigma}_{\texttt{ML}}^{(m)}$ as the priors, more robust MAP estimates [42] are used in [91] to avoid overtraining. In this case, the priors of the I-smoothing distribution are set as

$$\tilde{\boldsymbol{\mu}}^{(m)} = \frac{\tau^{\texttt{MAP}}\tilde{\boldsymbol{\mu}}_{\texttt{MAP}}^{(m)} + \sum_t \gamma_m^{\texttt{ML}}(t)\mathbf{o}_t}{\tau^{\texttt{MAP}} + \sum_t \gamma_m^{\texttt{ML}}(t)} \tag{4.32}$$

$$\tilde{\boldsymbol{\Sigma}}^{(m)} = \frac{\tau^{\texttt{MAP}}\left(\tilde{\boldsymbol{\mu}}_{\texttt{MAP}}^{(m)}\tilde{\boldsymbol{\mu}}_{\texttt{MAP}}^{(m)T} + \tilde{\boldsymbol{\Sigma}}_{\texttt{MAP}}^{(m)}\right) + \sum_t \gamma_m^{\texttt{ML}}(t)\mathbf{o}_t\mathbf{o}_t^T}{\tau^{\texttt{MAP}} + \sum_t \gamma_m^{\texttt{ML}}(t)} - \tilde{\boldsymbol{\mu}}^{(m)}\tilde{\boldsymbol{\mu}}^{(m)T} \tag{4.33}$$

where $\gamma_m^{\texttt{ML}}(t)$ is the ML posterior occupancy of component m, the MAP prior $\tilde{\boldsymbol{\mu}}_{\texttt{MAP}}^{(m)}$ and $\tilde{\boldsymbol{\Sigma}}_{\texttt{MAP}}^{(m)}$ are robust parameter estimate, such as the parameters of a MPE trained cluster-independent model. $\tau^{\texttt{MAP}}$ is the MAP parameter which controls the impact of the MAP prior. The MAP prior is a trade-off between the dynamic ML statistics and the static prior. Larger value of $\tau^{\texttt{MAP}}$ will lead to a more robust prior, which reduces the risk of overtraining. This MAP prior scheme was first investigated for gender-dependent (GD) model training with MMI and MPE criteria [91]. The basic procedure is to first train a gender-independent (GI) MPE model and then perform MPE-MAP training on the gender-specific training data by using the MPE GI model to construct the MAP prior. It has been shown that MPE GD models trained using this scheme effectively and consistently outperformed the MPE GI model [91, 140]. In other work [26], the MPE GI model parameters were used directly as the I-smoothing prior instead of the MAP estimate and only means and Gaussian component weights were updated during the MPE GD training.

$$\tilde{\boldsymbol{\mu}}^{(m)} = \hat{\boldsymbol{\mu}}_{\texttt{MPE-GI}}^{(m)} \tag{4.34}$$

This has been found to yield slight gains over the standard MPE-MAP scheme. A more detailed discussion about the prior used in the I-smoothing distribution is presented in the section below.

4.3.2 Discriminative Cluster Adaptive Training

Cluster adaptive training (CAT) [33], or eigenvoices [69, 88], is an extension to cluster-dependent modelling, where arbitrary interpolation weights are used to combine multiple sets of parameters. The ML based cluster adaptive training has been reviewed in section 3.4.2.2. In this thesis,

discriminative criterion is applied to cluster adaptive training to derive a new discriminative training technique for rapid adaptation. The specific discriminative criterion used is the MPE criterion. In additional to the canonical model update, the update of interpolation weights is also detailed to form a complete discriminative CAT framework.

4.3.2.1 MPE Training of Multiple-cluster Model Parameters

The canonical model for CAT is a multiple-cluster model, where each Gaussian component has multiple mean vectors and shared covariance matrices and component weights, as discussed in section 3.4.2.2. Given the interpolation weight vectors, the MPE training for the multiple-cluster canonical model also requires an appropriate definition of the weak-sense auxiliary function.

The numerator and denominator parts of the weak-sense auxiliary functions for estimating the multiple-cluster HMM parameters also have the same form as the ML CAT auxiliary function in equation (3.45). The ML auxiliary function for the canonical model is rewritten as

$$\begin{aligned}\mathcal{Q}_{\mathtt{ML}}(\mathcal{M};\hat{\mathcal{M}},\hat{\boldsymbol{\mathcal{T}}}) &= -\frac{1}{2}\sum_{s,m,t}\gamma_m^{\mathtt{ML}}(t)\Big\{\log\left|\boldsymbol{\Sigma}^{(m)}\right| \\ &\quad + \left(\mathbf{o}_t^{(s)}-\mathbf{M}^{(m)}\boldsymbol{\lambda}^{(sr_m)}\right)^T\boldsymbol{\Sigma}^{(m)-1}\left(\mathbf{o}_t^{(s)}-\mathbf{M}^{(m)}\boldsymbol{\lambda}^{(sr_m)}\right)\Big\} \qquad (4.35)\end{aligned}$$

where $\boldsymbol{\mathcal{T}}$ is now the set of interpolation weights $\boldsymbol{\lambda}^{(sr)}$ associated with acoustic condition s and regression base class r, $\gamma_m^{\mathtt{ML}}(t)$ is the standard ML posterior occupancy for Gaussian component m. This auxiliary function can also be re-expressed in terms of sufficient statistics as

$$\begin{aligned}\mathcal{G}_{\mathtt{CAT}}(\mathcal{M};\boldsymbol{\Gamma}_{\mathtt{ML}}) &= -\frac{1}{2}\sum_m\Big\{\gamma_m^{\mathtt{ML}}\log|\boldsymbol{\Sigma}^{(m)}| + \mathrm{tr}\left(\mathbf{L}_{\mathtt{ML}}^{(m)}\boldsymbol{\Sigma}^{(m)-1}\right) \\ &\quad -2\mathrm{tr}\left(\mathbf{K}_{\mathtt{ML}}^{(m)}\boldsymbol{\Sigma}^{(m)-1}\mathbf{M}^{(m)}\right)+\mathrm{tr}\left(\mathbf{G}_{\mathtt{ML}}^{(m)}\mathbf{M}^{(m)T}\boldsymbol{\Sigma}^{(m)-1}\mathbf{M}^{(m)}\right)\Big\} \qquad (4.36)\end{aligned}$$

and

$$\boldsymbol{\Gamma}_{\mathtt{ML}}=\left\{\gamma_m^{\mathtt{ML}},\mathbf{G}_{\mathtt{ML}}^{(m)},\mathbf{K}_{\mathtt{ML}}^{(m)},\mathbf{L}_{\mathtt{ML}}^{(m)}\right\}$$

The elements of $\boldsymbol{\Gamma}_{\mathtt{ML}}$ have been defined in equations (3.46) to (3.49). To obtain the numerator and denominator parts, it is sufficient to replace the ML posterior, $\gamma_m^{\mathtt{ML}}(t)$, in equations (3.46) to (3.49) with the appropriate numerator and denominator posteriors, $\gamma_m^{\mathtt{n}}(t)$ and $\gamma_m^{\mathtt{d}}(t)$ respectively. This yields numerator and denominator statistics, $\boldsymbol{\Gamma}_{\mathtt{n}}$ and $\boldsymbol{\Gamma}_{\mathtt{d}}$. The numerator part can then be written in the generic form as equation (4.36)

$$\mathcal{Q}_{\mathtt{n}}(\mathcal{M};\hat{\mathcal{M}})=\mathcal{G}_{\mathtt{CAT}}(\mathcal{M};\boldsymbol{\Gamma}_{\mathtt{n}}) \qquad (4.37)$$

and similarly for the denominator part.

As in linear transform based discriminative adaptive training, the smoothing function for multiple-cluster model can also be derived from the generic form in equation (4.4) (shown in

appendix A)

$$\begin{aligned}\mathcal{S}(\mathcal{M};\hat{\mathcal{M}},\hat{\boldsymbol{\mathcal{T}}}) &= -\sum_{m,s}\frac{D_m\nu_m^{(s)}}{2}\Big\{\log|\boldsymbol{\Sigma}^{(m)}|+\mathrm{tr}(\hat{\boldsymbol{\Sigma}}_{\mathtt{c}}^{(m)}\boldsymbol{\Sigma}^{(m)-1}) \\ &\quad +\boldsymbol{\lambda}^{(sr_m)T}\Big(\mathbf{M}^{(m)}-\hat{\mathbf{M}}_{\mathtt{c}}^{(m)}\Big)^T\boldsymbol{\Sigma}^{(m)-1}\Big(\mathbf{M}^{(m)}-\hat{\mathbf{M}}_{\mathtt{c}}^{(m)}\Big)\boldsymbol{\lambda}^{(sr_m)}\Big\} \qquad (4.38)\end{aligned}$$

where $\hat{\mathbf{M}}_{\mathtt{c}}^{(m)}$ and $\hat{\boldsymbol{\Sigma}}_{\mathtt{c}}^{(m)}$ are the current model parameters. The constant D_m is a positive smoothing constant for component m to control the impact of the smoothing function and make the optimisation stable, $\nu_m^{(s)}$ is the same term as defined in equation (4.6) to reflect the proportions of data for the particular component of a acoustic condition [3]. The smoothing function can also be expressed as the general auxiliary function in terms of sufficient statistics

$$\mathcal{S}(\mathcal{M};\hat{\mathcal{M}},\hat{\boldsymbol{\mathcal{T}}}) = \mathcal{G}_{\mathtt{CAT}}(\mathcal{M};\boldsymbol{\Gamma}_{\mathtt{s}}) \qquad (4.39)$$

where

$$\boldsymbol{\Gamma}_{\mathtt{s}} = \Big\{D_m, D_m\mathbf{G}_{\mathtt{s}}^{(m)}, D_m\mathbf{K}_{\mathtt{s}}^{(m)}, D_m\mathbf{L}_{\mathtt{s}}^{(m)}\Big\}$$

and

$$\mathbf{G}_{\mathtt{s}}^{(m)} = \sum_s \nu_m^{(s)}\boldsymbol{\lambda}^{(sr_m)}\boldsymbol{\lambda}^{(sr_m)T} \qquad (4.40)$$

$$\mathbf{K}_{\mathtt{s}}^{(m)} = \mathbf{G}_{\mathtt{s}}^{(m)}\hat{\mathbf{M}}_{\mathtt{c}}^{(m)T} \qquad (4.41)$$

$$\mathbf{L}_{\mathtt{s}}^{(m)} = \hat{\boldsymbol{\Sigma}}_{\mathtt{c}}^{(m)} + \hat{\mathbf{M}}_{\mathtt{c}}^{(m)}\mathbf{G}_{\mathtt{s}}^{(m)}\hat{\mathbf{M}}_{\mathtt{c}}^{(m)T} \qquad (4.42)$$

For multiple-cluster MPE training, the selection of D_m is different from the single-cluster MPE training. This will be discussed later.

The I-smoothing distribution used for a multiple-cluster HMM is again a Normal-Wishart distribution defined at acoustic condition level. Ignoring the constant term, the general form of the logarithm of the I-smoothing distribution for multiple-cluster model parameters may be written as

$$\begin{aligned}\log p(\mathcal{M}|\Phi) &= -\frac{\tau^I}{2}\sum_{s,m}\tilde{\nu}_m^{(s)}\Big\{\log|\boldsymbol{\Sigma}^{(m)}|+\mathrm{tr}(\tilde{\boldsymbol{\Sigma}}^{(m)}\boldsymbol{\Sigma}^{(m)-1}) \\ &\quad +\Big(\mathbf{M}^{(m)}\boldsymbol{\lambda}^{(sr_m)}-\tilde{\boldsymbol{\mu}}^{(sm)}\Big)^T\boldsymbol{\Sigma}^{(m)-1}\Big(\mathbf{M}^{(m)}\boldsymbol{\lambda}^{(sr_m)}-\tilde{\boldsymbol{\mu}}^{(sm)}\Big)\Big\} \qquad (4.43)\end{aligned}$$

where τ^I is the parameter which controls the impact of the prior. $\tilde{\boldsymbol{\mu}}^{(sm)}$ and $\tilde{\boldsymbol{\Sigma}}^{(m)}$ are the prior parameters for the mean and covariance matrix of the m^{th} component for acoustic condition s. $\tilde{\nu}_m^{(s)}$ is also a slightly modified version of $\nu_m^{(s)}$ where the ML posterior occupancy is used instead of the numerator one. The sufficient statistics required for this I-smoothing distribution is

$$\log p(\mathcal{M}|\Phi) = \mathcal{G}_{\mathtt{CAT}}(\mathcal{M};\boldsymbol{\Gamma}_{\mathtt{p}}) \qquad (4.44)$$

[3] When using a constant value for $\nu_m^{(s)}$, the WER of MPE-CAT was about 0.1% worse compared to using equation (4.6) (with dynamic multiple-cluster ML prior and the same configurations as the 16 component development systems in chapter 6).

where $\mathcal{G}_{\texttt{CAT}}(\cdot)$ is the general form defined in equation (4.36)

$$\mathbf{\Gamma}_{\mathtt{p}} = \left\{\tau^I, \tau^I \mathbf{G}_{\mathtt{p}}^{(m)}, \tau^I \mathbf{K}_{\mathtt{p}}^{(m)}, \tau^I \mathbf{L}_{\mathtt{p}}^{(m)}\right\}$$

and

$$\mathbf{G}_{\mathtt{p}}^{(m)} = \sum_s \tilde{\nu}_m^{(s)} \boldsymbol{\lambda}^{(sr_m)} \boldsymbol{\lambda}^{(sr_m)T} \tag{4.45}$$

$$\mathbf{K}_{\mathtt{p}}^{(m)} = \sum_s \tilde{\nu}_m^{(s)} \boldsymbol{\lambda}^{(sr_m)} \tilde{\boldsymbol{\mu}}^{(sm)T} \tag{4.46}$$

$$\mathbf{L}_{\mathtt{p}}^{(m)} = \tilde{\mathbf{\Sigma}}^{(m)} + \left(\sum_s \tilde{\nu}_m^{(s)} \tilde{\boldsymbol{\mu}}^{(sm)} \tilde{\boldsymbol{\mu}}^{(sm)T}\right) \tag{4.47}$$

As the number of parameters for a multiple-cluster model dramatically increases, the priors in I-smoothing distribution, $\tilde{\boldsymbol{\mu}}^{(sm)}$ and $\tilde{\mathbf{\Sigma}}^{(m)}$, are increasingly important. The options are more than the standard single cluster model update and will be discussed in section 4.3.2.2.

Having introduced the form of the smoothing function and I-smoothing distribution, the weak-sense auxiliary function for MPE training may be obtained by combining all individual elements as in equation (4.18). Here, the sufficient statistics based auxiliary function $\mathcal{G}_{\texttt{CAT}}(\cdot)$ in equation (4.36) is used instead of $\mathcal{G}_{\texttt{SAT}}(\cdot)$. Differentiating the weak-sense auxiliary function with respect to multiple-cluster model parameters and equating it to zero yields simple closed-form parameter estimate formulae

$$\hat{\mathbf{M}}^{(m)T} = \mathbf{G}_{\texttt{MPE}}^{(m)-1} \mathbf{K}_{\texttt{MPE}}^{(m)} \tag{4.48}$$

$$\hat{\mathbf{\Sigma}}^{(m)} = \text{diag}\left(\frac{\mathbf{L}_{\texttt{MPE}}^{(m)} - \hat{\mathbf{M}}^{(m)} \mathbf{K}_{\texttt{MPE}}^{(m)}}{\gamma_m^{\texttt{MPE}}}\right) \tag{4.49}$$

where $\mathbf{\Gamma}_{\texttt{MPE}} = \{\gamma_m^{\texttt{MPE}}, \mathbf{G}_{\texttt{MPE}}^{(m)}, \mathbf{K}_{\texttt{MPE}}^{(m)}, \mathbf{L}_{\texttt{MPE}}^{(m)}\}$, and

$$\gamma_m^{\texttt{MPE}} = \gamma_m^{\mathtt{n}} - \gamma_m^{\mathtt{d}} + D_m + \tau^I \tag{4.50}$$

$$\mathbf{G}_{\texttt{MPE}}^{(m)} = \mathbf{G}_{\mathtt{n}}^{(m)} - \mathbf{G}_{\mathtt{d}}^{(m)} + D_m \mathbf{G}_{\mathtt{s}}^{(m)} + \tau^I \mathbf{G}_{\mathtt{p}}^{(m)} \tag{4.51}$$

$$\mathbf{K}_{\texttt{MPE}}^{(m)} = \mathbf{K}_{\mathtt{n}}^{(m)} - \mathbf{K}_{\mathtt{d}}^{(m)} + D_m \mathbf{K}_{\mathtt{s}}^{(m)} + \tau^I \mathbf{K}_{\mathtt{p}}^{(m)} \tag{4.52}$$

$$\mathbf{L}_{\texttt{MPE}}^{(m)} = \mathbf{L}_{\mathtt{n}}^{(m)} - \mathbf{L}_{\mathtt{d}}^{(m)} + D_m \mathbf{L}_{\mathtt{s}}^{(m)} + \tau^I \mathbf{L}_{\mathtt{p}}^{(m)} \tag{4.53}$$

Comparing the sufficient statistics of MPE CAT to the MPE statistics of of standard HMMs in equations (2.78) to (2.80), MPE CAT requires more memory than the standard HMMs. Due to the statistics in equations (4.51) and (4.52), over P times memory is required compared to the standard $\mathbf{k}$ statistics in equation (2.79), where P is the number of clusters. Hence, in practice, when building MPE CAT systems, the number of clusters has to be controlled according to the memory limitation.

4.3.2.2 Priors in I-smoothing Distribution

One key issue in defining the I-smoothing distribution (4.43) is to choose an appropriate form of the prior parameters $\tilde{\boldsymbol{\mu}}^{(sm)}$ and $\tilde{\mathbf{\Sigma}}^{(m)}$ [140]. The prior forms include *multiple-cluster* and *single-cluster*. In both cases, the prior covariance matrix is independent of the acoustic conditions.

When using a multiple-cluster prior, a global multiple-cluster mean matrix is used as the prior and $\tilde{\boldsymbol{\mu}}^{(sm)}$ is the adapted mean vector, hence it is acoustic condition specific. Alternatively, a *single-cluster* prior may also be selected, then the prior mean vector is independent of acoustic conditions. In additional to the above form of the prior, the selected prior, either multiple-cluster or single-cluster, can be parameters of an existing model set (*static*), or model estimates in terms of sufficient statistics on-the-fly (*dynamic*). Finally, the prior may be generated using different criteria, such as ML, MMI or MPE.

There are many possible combinations of the above properties, which leads to quite a large number of possible priors to choose. Some of the forms considered in this work are discussed below.

1. Multiple-cluster prior

Here the prior mean vector is made to be acoustic condition specific using the interpolation weights. Thus the following substitution is applied to equation (4.43),

$$\tilde{\boldsymbol{\mu}}^{(sm)} = \tilde{\mathbf{M}}^{(m)} \boldsymbol{\lambda}^{(sr_m)} \tag{4.54}$$

where $\tilde{\mathbf{M}}^{(m)}$ is the prior cluster mean matrix, $\boldsymbol{\lambda}^{(sr)}$ is the interpolation weight associated with regression class r and acoustic condition s. The sufficient statistics in equation (4.45) to equation (4.47) become

$$\mathbf{G}_{\mathrm{p}}^{(m)} = \sum_{s} \tilde{\nu}_m^{(s)} \boldsymbol{\lambda}^{(sr_m)} \boldsymbol{\lambda}^{(sr_m)T} \tag{4.55}$$

$$\mathbf{K}_{\mathrm{p}}^{(m)} = \left(\sum_{s} \tilde{\nu}_m^{(s)} \boldsymbol{\lambda}^{(sr_m)} \boldsymbol{\lambda}^{(sr_m)T} \right) \tilde{\mathbf{M}}^{(m)T} = \mathbf{G}_{\mathrm{p}}^{(m)} \tilde{\mathbf{M}}^{(m)T} \tag{4.56}$$

$$\mathbf{L}_{\mathrm{p}}^{(m)} = \tilde{\boldsymbol{\Sigma}}^{(m)} + \tilde{\mathbf{M}}^{(m)} \mathbf{G}_{\mathrm{p}}^{(m)} \tilde{\mathbf{M}}^{(m)} \tag{4.57}$$

Any appropriate CAT model may be used as the prior parameters $\tilde{\mathbf{M}}^{(m)}$ and $\tilde{\boldsymbol{\Sigma}}^{(m)}$. One form is to use the ML estimates for $\tilde{\mathbf{M}}^{(m)}$ and $\tilde{\boldsymbol{\Sigma}}^{(m)}$ as dynamic priors. Substituting the ML estimates for $\tilde{\mathbf{M}}^{(m)}$ and $\tilde{\boldsymbol{\Sigma}}^{(m)}$, equations (4.55) to (4.57) become

$$\mathbf{G}_{\mathrm{p}}^{(m)} = \frac{1}{\gamma_m^{\mathtt{ML}}} \mathbf{G}_{\mathtt{ML}}^{(m)} = \frac{1}{\gamma_m^{\mathtt{ML}}} \sum_{s,t} \gamma_m^{\mathtt{ML}}(t) \boldsymbol{\lambda}^{(sr_m)} \boldsymbol{\lambda}^{(sr_m)T} \tag{4.58}$$

$$\mathbf{K}_{\mathrm{p}}^{(m)} = \frac{1}{\gamma_m^{\mathtt{ML}}} \mathbf{K}_{\mathtt{ML}}^{(m)} = \frac{1}{\gamma_m^{\mathtt{ML}}} \sum_{s,t} \gamma_m^{\mathtt{ML}}(t) \boldsymbol{\lambda}^{(sr_m)} \mathbf{o}_t^{(s)T} \tag{4.59}$$

$$\mathbf{L}_{\mathrm{p}}^{(m)} = \frac{1}{\gamma_m^{\mathtt{ML}}} \mathbf{L}_{\mathtt{ML}}^{(m)} = \frac{1}{\gamma_m^{\mathtt{ML}}} \sum_{s,t} \gamma_m^{\mathtt{ML}}(t) \mathbf{o}_t^{(s)} \mathbf{o}_t^{(s)T} \tag{4.60}$$

Comparing the above statistics with the ML statistics in equations (3.47) to (3.49), they are all normalised by $\gamma_m^{\mathtt{ML}}$ to yield "unit" counts, as the "occupancy" of the I-smoothing part is represented by τ^I. This is a natural extension to standard I-smoothing in MPE training, in which the ML estimates are also used as priors of the I-smoothing distribution [90].

Considering the MPE sufficient statistics in equations (4.50) to (4.53), when $\tau^I \rightarrow \infty$, sufficient statistics of I-smoothing distribution $\mathbf{\Gamma}_{\mathtt{p}}$ will dominate $\mathbf{\Gamma}_{\mathtt{MPE}}$. In this case, if a multiple-cluster prior is used, it can be shown that the MPE estimates of model parameters will back off to the multiple-cluster prior. Furthermore, if multiple-cluster ML dynamic prior is used, the MPE estimates will back off to the ML CAT estimates in equation (3.50) and equation (3.51).

In addition to the ML estimates, multiple-cluster MAP estimates may also be used as dynamic multiple-cluster priors. In this case, the ML statistics from equations (4.58) to (4.60) are replaced by the MAP statistics. It is shown in appendix B, that the sufficient statistics for a valid multiple-cluster MAP estimate are

$$\mathbf{\Gamma}_{\mathtt{MAP}} = \left\{ \gamma_m^{\mathtt{MAP}}, \mathbf{G}_{\mathtt{MAP}}^{(m)}, \mathbf{K}_{\mathtt{MAP}}^{(m)}, \mathbf{L}_{\mathtt{MAP}}^{(m)} \right\}$$

and

$$\gamma_m^{\mathtt{MAP}} = \gamma_m^{\mathtt{ML}} + \tau^{\mathtt{MAP}} \tag{4.61}$$

$$\mathbf{G}_{\mathtt{MAP}}^{(m)} = \mathbf{G}_{\mathtt{ML}}^{(m)} + \tau^{\mathtt{MAP}} \sum_s \tilde{\nu}_m^{(s)} \boldsymbol{\lambda}^{(sr_m)} \boldsymbol{\lambda}^{(sr_m)T} \tag{4.62}$$

$$\mathbf{K}_{\mathtt{MAP}}^{(m)} = \mathbf{K}_{\mathtt{ML}}^{(m)} + \tau^{\mathtt{MAP}} \Big(\sum_s \tilde{\nu}_m^{(s)} \boldsymbol{\lambda}^{(sr_m)} \Big) \tilde{\boldsymbol{\mu}}_{\mathtt{MAP}}^{(m)T} \tag{4.63}$$

$$\mathbf{L}_{\mathtt{MAP}}^{(m)} = \mathbf{L}_{\mathtt{ML}}^{(m)} + \tau^{\mathtt{MAP}} \Big(\tilde{\boldsymbol{\mu}}_{\mathtt{MAP}}^{(m)} \tilde{\boldsymbol{\mu}}_{\mathtt{MAP}}^{(m)T} + \tilde{\mathbf{\Sigma}}_{\mathtt{MAP}}^{(m)} \Big) \tag{4.64}$$

where $\tau^{\mathtt{MAP}}$ is the tunable parameter for the new Normal-Wishart distribution for MAP estimate, $\gamma_m^{\mathtt{ML}}$, $\mathbf{G}_{\mathtt{ML}}^{(m)}$, $\mathbf{K}_{\mathtt{ML}}^{(m)}$ and $\mathbf{L}_{\mathtt{ML}}^{(m)}$ are the ML statistics, $\tilde{\boldsymbol{\mu}}_{\mathtt{MAP}}^{(m)}$ and $\tilde{\mathbf{\Sigma}}_{\mathtt{MAP}}^{(m)}$ are parameters from a robust single-cluster model. It is obvious that this multiple-cluster MAP prior is a trade-off between the multiple-cluster ML prior and the static single-cluster prior [4]. If $\tau^{\mathtt{MAP}} \rightarrow 0$, the MAP statistics will lead to multiple-cluster ML statistics. If $\tau^{\mathtt{MAP}} \rightarrow \infty$, the MAP statistics will back off to a standard single-cluster prior, which will be discussed below. This MAP prior may be regarded as an extension to the standard MPE-MAP prior for gender-dependent model in [91].

2. Single-cluster prior

When using a single-cluster prior, it is possible to use a standard single-cluster HMM as the prior. The following substitution is applied to equation (4.43)

$$\tilde{\boldsymbol{\mu}}^{(sm)} = \tilde{\boldsymbol{\mu}}^{(m)} \tag{4.65}$$

The sufficient statistics for I-smoothing distribution in equations (4.45) to (4.47) become

$$\mathbf{G}_{\mathtt{p}}^{(m)} = \sum_s \tilde{\nu}_m^{(s)} \boldsymbol{\lambda}^{(sr_m)} \boldsymbol{\lambda}^{(sr_m)T} \tag{4.66}$$

$$\mathbf{K}_{\mathtt{p}}^{(m)} = \Big(\sum_s \tilde{\nu}_m^{(s)} \boldsymbol{\lambda}^{(sr_m)} \Big) \tilde{\boldsymbol{\mu}}^{(m)T} \tag{4.67}$$

$$\mathbf{L}_{\mathtt{p}}^{(m)} = \tilde{\boldsymbol{\mu}}^{(m)} \tilde{\boldsymbol{\mu}}^{(m)T} + \tilde{\mathbf{\Sigma}}^{(m)} \tag{4.68}$$

[4] Note that in the MAP prior case, $\tilde{\nu}_m^{(s)}$ need to be re-defined using the MAP occupancy.

Similarly to the multiple-cluster priors, either *static* or *dynamic* single-cluster priors can be used. When using static priors, parameters from an appropriate standard HMM are selected. The topology of the prior HMM should be the same as the model to update, otherwise, there is no unique component mapping and it is hard to choose prior parameters for a particular component. However, even with the same topology, certain risk still remains. The initialisation for the prior HMM may be different from the intialisation for the current model to update. Hence, for a particular state, the order of the components in the prior HMM is not guaranteed to be the same as the current model. In this case, there is a risk of selecting a wrong prior component due to order difference. However, if the prior HMM has the same "seed model", or starting model, with the current model to update, e.g., the prior MPE-SI model and the current MPE-CAT model are both generated from the same ML-SI model, it is a reasonable assumption that the component order issue would not affect the training much.

When using dynamic priors, as the statistics are accumulated at component level on-the-fly, the component order is not an issue. ML statistics can be accumulated as in the standard formulae. However, to accumulate single-cluster MPE statistics, the "current single-cluster mean vector" is required for the smoothing function in equation (2.61), which does not exist because the current model is a multiple-cluster model. One solution for this is to first read in an MPE-SI model as the "current single-cluster model" for the first iteration allowing the single-cluster MPE statistics to be obtained. Then a new standard HMM estimated from the MPE statistics can be output as a "current single-cluster model" for the second iteration and so on. Therefore, for each iteration, two model sets are trained, one is the multiple-cluster model to update, the other is a "current single-cluster model" for accumulating standard MPE statistics in the next iteration.

Considering the whole MPE sufficient statistics in equation (4.50) to equation (4.53), though the sufficient statistics of I-smoothing distribution $\mathbf{\Gamma}_{\mathrm{p}}$ will dominate the whole statistics when $\tau^I \rightarrow \infty$, the MPE estimates can not back off to the single-cluster prior $\tilde{\boldsymbol{\mu}}^{(m)}$ and $\tilde{\boldsymbol{\Sigma}}^{(m)}$ due to the interpolation weights $\boldsymbol{\lambda}$. This is different from the multiple-cluster prior case.

4.3.2.3 Selection of Smoothing Constant

As discussed in section 2.4.2, the smoothing constant D_m is a critical value in MPE training to get a rapid and stable update. It is suggested to be set as the maximum of either twice of the smallest value required to ensure the updated covariance matrix is positive-definite, denoted as $\tilde{D}_m$, or E times the component denominator occupancy $\gamma_m^{\mathtt{d}}$, where E is typically 1 or 2 [90]. To find $\tilde{D}_m$, the equation $\boldsymbol{\Sigma}^{(m)} = \mathbf{0}$ must be solved from equation (4.49), i.e.

$$\mathbf{L}_{\mathtt{MPE}}^{(m)} = \mathbf{K}_{\mathtt{MPE}}^{(m)T} \mathbf{G}_{\mathtt{MPE}}^{(m)-1} \mathbf{K}_{\mathtt{MPE}}^{(m)} \tag{4.69}$$

As the covariance matrix is assumed to be diagonal, the equation can be re-written for each dimension with respect to the smoothing constant $\tilde{D}_m$ as

$$(\mathbf{k}_1 + \tilde{D}_m \mathbf{k}_2)^T (\mathbf{G}_1 + \tilde{D}_m \mathbf{G}_2)^{-1} (\mathbf{k}_1 + \tilde{D}_m \mathbf{k}_2) = l_1 + \tilde{D}_m l_2 \tag{4.70}$$

Where $\mathbf{k}_1$ and $\mathbf{k}_2$ are $P \times 1$ vectors, $\mathbf{G}_1$ and $\mathbf{G}_2$ are $P \times P$ matrices, l_1 and l_2 are scalars, P is the number of clusters, which are all constants given appropriate accumulated statistics.

For a single-cluster system, the equation for each dimension is a quadratic equation and can be easily solved as in MPE training of standard HMMs [90]. However, for a multiple-cluster model, equation (4.70) is a high order polynomial equation. As the inversion of a matrix can be represented as its adjoint matrix divided by the determinant, it can be easily seen that the order of $\mathbf{G}_1 + \tilde{D}_m \mathbf{G}_2$ with respect to $\tilde{D}_m$ is P. Hence, the order of the polynomial equation (4.70) is $P+1$. As there is no closed-form solution for general polynomial equations with an order greater than 4, equation (4.70) has no closed-form solutions for $P > 3$.

For some special cases, such as $P = 2$ or $P = 3$, cubic or quartic polynomial equation, a closed-form solution of the above function can be derived and then the largest real root can be found directly. For larger number of clusters, numerical approaches may be used to find the largest real root. Alternatively, as an approximated estimate, $E\gamma_m^{\mathtt{d}}$ may be used directly as D_m together with appropriate variance floors to ensure positive definite of covariance matrices [5].

4.3.2.4 MPE Training of Interpolation Weights

The previous sections have discussed various issues of MPE training of multiple-cluster HMMs. This section will give details of the MPE training of interpolation weights. The MPE training scheme in this section was first proposed in [137]. After that another implementation using the MMI criterion was also introduced in [79] within the equivalent eigenvoices framework.

The ML auxiliary function for weights update are the same form as equation (4.35) except that the parameters to be updated is the interpolation weights rather than the canonical model parameters. Therefore, the sufficient statistics form of the ML auxiliary function can be written as

$$\mathcal{Q}_{\mathtt{ML}}(\mathcal{T}^{(s)}; \hat{\mathcal{T}}^{(s)}, \hat{\mathcal{M}}) = \mathcal{G}_{\mathtt{CAT}}(\mathcal{T}^{(s)}; \mathbf{\Gamma}_{\mathtt{ML}}) = -\frac{1}{2} \sum_r \left\{ \boldsymbol{\lambda}^{(sr)T} \mathbf{G}_{\mathtt{ML}}^{(sr)} \boldsymbol{\lambda}^{(sr)} - 2 \boldsymbol{\lambda}^{(sr)T} \mathbf{k}_{\mathtt{ML}}^{(sr)} \right\} \tag{4.71}$$

where $\mathcal{T}^{(s)}$ is now the interpolation weight vector $\boldsymbol{\lambda}^{(sr)}$, which is associated with regression base class r and acoustic condition s, the sufficient statistics is

$$\mathbf{\Gamma}_{\mathtt{ML}} = \left\{ \mathbf{G}_{\mathtt{ML}}^{(sr)}, \mathbf{k}_{\mathtt{ML}}^{(sr)} \right\}$$

defined in equations (3.54) and (3.55). The sufficient statistics based numerator and denominator parts of the weak-sense auxiliary function have the same form as equation (4.71) except for using the numerator occupancy $\gamma_m^{\mathtt{n}}(t)$ or the denominator occupancy $\gamma_m^{\mathtt{d}}(t)$ instead of the ML occupancy $\gamma_m^{\mathtt{ML}}(t)$ in $\mathbf{\Gamma}_{\mathtt{ML}}$.

The smoothing function for weights update may also be derived from the general form in equation (4.4), as shown in appendix A

$$\begin{aligned} \mathcal{S}(\mathcal{T}^{(s)}; \hat{\mathcal{T}}^{(s)}, \hat{\mathcal{M}}) \quad = \quad & -\sum_m \frac{D_m}{2} \Big\{ \boldsymbol{\lambda}^{(sr_m)T} \mathbf{M}^{(m)T} \mathbf{\Sigma}^{(m)-1} \mathbf{M}^{(m)} \boldsymbol{\lambda}^{(sr_m)} \\ & -2 \boldsymbol{\lambda}^{(sr_m)T} \mathbf{M}^{(m)T} \mathbf{\Sigma}^{(m)-1} \mathbf{M}^{(m)} \hat{\boldsymbol{\lambda}}_{\mathtt{c}}^{(sr_m)} \Big\} \end{aligned} \tag{4.72}$$

[5] The approximate selection has been shown in experiments to give only marginal difference from the strict selection. In this work, strict selection of D_m is used for 2-cluster systems. For more clusters, the approximate selection is used.

where D_m is the constant to ensure stable optimisation. It can be set as $E\gamma_m^{\mathtt{d}}$ as an approximation since no covariance update is involved. For acoustic condition s, $\hat{\boldsymbol{\lambda}}_{\mathtt{c}}^{(sr_m)}$ is the current estimate of the interpolation weight vector for regression base class r_m that component m belongs to. The smoothing function may also be expressed in the form of sufficient statistics in equation (4.71)

$$\mathcal{S}(\mathcal{T}^{(s)};\hat{\mathcal{T}}^{(s)},\hat{\mathcal{M}}) = \mathcal{G}_{\mathtt{CAT}}(\mathcal{T}^{(s)};\boldsymbol{\Gamma}_{\mathtt{s}}) \tag{4.73}$$

where

$$\boldsymbol{\Gamma}_{\mathtt{s}} = \left\{\mathbf{G}_{\mathtt{s}}^{(sr)}, \mathbf{k}_{\mathtt{s}}^{(sr)}\right\} \tag{4.74}$$

and

$$\mathbf{G}_{\mathtt{s}}^{(sr)} = \sum_{m\in M_r} D_m \mathbf{M}^{(m)T}\boldsymbol{\Sigma}^{(m)-1}\mathbf{M}^{(m)} \tag{4.75}$$

$$\mathbf{k}_{\mathtt{s}}^{(sr)} = \left(\sum_{m\in M_r} D_m \mathbf{M}^{(m)T}\boldsymbol{\Sigma}^{(m)-1}\mathbf{M}^{(m)}\right)\hat{\boldsymbol{\lambda}}_{\mathtt{c}}^{(sr)} \tag{4.76}$$

where M_r is the Gaussian component group of regression base class r.

The Normal distribution around mean vector is used here as the I-smoothing distribution for weights update. Ignoring constant terms, the log distribution can be written as

$$\log p(\mathcal{T}^{(s)}|\phi) = -\frac{\tau^I}{2}\sum_m \left(\mathbf{M}^{(m)}\boldsymbol{\lambda}^{(sr_m)} - \tilde{\boldsymbol{\mu}}^{(sm)}\right)^T \boldsymbol{\Sigma}^{(m)-1}\left(\mathbf{M}^{(m)}\boldsymbol{\lambda}^{(sr_m)} - \tilde{\boldsymbol{\mu}}^{(sm)}\right) \tag{4.77}$$

where $\phi = \{\tau^I, \tilde{\boldsymbol{\mu}}^{(sm)}\}$ are the hyper-parameters of the I-smoothing distribution, τ^I is the parameter to control the impact of the I-smoothing distribution and can be empirically set, $\tilde{\boldsymbol{\mu}}^{(sm)}$ is the prior for the I-smoothing distribution and can have various forms as discussed in section 4.3.2.2. The sufficient statistics for the I-smoothing distribution are

$$\boldsymbol{\Gamma}_{\mathtt{p}} = \left\{\mathbf{G}_{\mathtt{p}}^{(sr)}, \mathbf{k}_{\mathtt{p}}^{(sr)}\right\} \tag{4.78}$$

where

$$\mathbf{G}_{\mathtt{p}}^{(sr)} = \tau^I \sum_{m\in M_r} \mathbf{M}^{(m)T}\boldsymbol{\Sigma}^{(m)-1}\mathbf{M}^{(m)} \tag{4.79}$$

$$\mathbf{k}_{\mathtt{p}}^{(sr)} = \tau^I \sum_{m\in M_r} \mathbf{M}^{(m)T}\boldsymbol{\Sigma}^{(m)-1}\tilde{\boldsymbol{\mu}}^{(sm)} \tag{4.80}$$

Given the above definition of the elements of the weak-sense auxiliary function and relevant statistics, the interpolation weight vector of regression class r for acoustic condition s can be estimated by

$$\hat{\boldsymbol{\lambda}}^{(sr)} = \mathbf{G}_{\mathtt{MPE}}^{(sr)-1}\mathbf{k}_{\mathtt{MPE}}^{(sr)} \tag{4.81}$$

where

$$\mathbf{G}_{\mathtt{MPE}}^{(sr)} = \mathbf{G}_{\mathtt{n}}^{(sr)} - \mathbf{G}_{\mathtt{d}}^{(sr)} + \mathbf{G}_{\mathtt{s}}^{(sr)} + \mathbf{G}_{\mathtt{p}}^{(sr)} \tag{4.82}$$

$$\mathbf{k}_{\mathtt{MPE}}^{(sr)} = \mathbf{k}_{\mathtt{n}}^{(sr)} - \mathbf{k}_{\mathtt{d}}^{(sr)} + \mathbf{k}_{\mathtt{s}}^{(sr)} + \mathbf{k}_{\mathtt{p}}^{(sr)} \tag{4.83}$$

4.4 Discriminative Adaptive Training with Structured Transforms

Standard discriminative adaptive training often uses a single set of transforms to represent non-speech variabilities, such as CMLLR in SAT. However, for highly non-homogeneous data, there are commonly multiple acoustic factors affecting the speech signal. This motivates the use of *structured transforms* (ST) [136, 37]. ST based adaptive training can also be extended to a discriminative paradigm. Due to computation and memory load, as in ML case, only one particular form of ST is investigated within discriminative adaptive training framework in this work. CAT interpolation weights [36] and CMLLR transforms [34] are combined together to form the structured transform used for both adaptive training and testing adaptation. Hence, the canonical model is a multiple-cluster model. The ML auxiliary function for ST based adaptive training has been given in equation (3.56).

As CMLLR can be implemented as a feature transform, discriminatively training the multiple-cluster model parameters is a straight-forward extension of the model update in discriminative CAT. The only modification to MPE training is that the model is estimated in the transformed feature space, where the transformed feature vectors $\hat{\mathbf{o}}_t^{(sr_m^{\mathbf{W}})}$ are used instead of standard feature vectors $\mathbf{o}_t^{(s)}$, where s denotes the index of the homogeneous block, r_m^W is the CMLLR regression class which component m belongs to. Since a simplified MPE adaptive training scheme is used here, discriminative estimation of transformation parameters are not required. The simplified training procedure has been described in section 4.1.

4.5 Summary

This chapter has described different forms of discriminative adaptive training using the minimum phone error (MPE) training criterion. First the procedure of discriminative adaptive training is discussed. A simplified training procedure is motivated to keep consistent transform update criterion in both training and unsupervised adaptation. In this procedure, only the canonical model is updated with the discriminative criterion given a set of ML estimated transforms. Then, in testing adaptation, the ML criterion is used to estimate transforms. As this procedure is adopted in the work, the MPE update of the canonical model is the focus in this chapter.

Three forms of MPE adaptive training of the canonical model are described. The first is the MPE adaptive training with mean transforms. It requires redefinition of the weak-sense auxiliary function. An appropriate smoothing function and an I-smoothing distribution can be defined at acoustic condition level. Using the weak-sense auxiliary function, closed-form re-estimation formulae for means and covariances can be derived. In this case, they have to be updated separately rather than simultaneously. MPE adaptive training with constrained transforms is the second form. It is a straightforward extension of the standard discriminative training because the constrained transform can be implemented as a feature transform. The only modification is to use the transformed observations instead of the original observations to accumulate suffi-

cient statistics. The third is a new form of MPE adaptive training, discriminative cluster adaptive training (CAT). By using an appropriately defined smoothing function and I-smoothing distribution, closed-form update formulae of both multiple-cluster canonical model and interpolation weights can be obtained. Due to the increase of the number of model parameters, discriminative training for multiple-cluster models is more prone to overtraining. Hence, the selection of the prior in I-smoothing distribution is crucial. The prior in I-smoothing distribution may have a variety of forms. It can be either multiple-cluster or single-cluster, either dynamic or static. The setting of the smoothing constant is also discussed. Discriminative CAT can be easily extended to discriminative adaptive training with structured transforms (ST), where the ST is constrained MLLR combined with CAT weights. In this case, the canonical model is still a multiple-cluster model. However, it is trained using the observations transformed by the constrained MLLR.

5

Bayesian Adaptive Training and Adaptive Inference

Chapter 3 presented an ML framework for adaptive training. Chapter 4 discussed adaptive training using discriminative criteria. Both chapters consider deterministic point estimates of parameters during adaptive training and use the standard two-step adaptation/recognition in inference. In contrast to the standard adaptive training and adaptation framework investigated in the previous chapters, this chapter will present a consistent Bayesian framework for both adaptive training and adaptive inference. Particularly, an integrated Bayesian adaptive inference process is described in detail. This framework allows the canonical model to be directly used in inference. The Bayesian framework is discussed for both likelihood and discriminative criteria. As the Bayesian marginalisation integral over the transform distribution is intractable, approximations are required. By using an appropriate Bayesian approximation approach, the issue of handling limited adaptation data is effectively addressed. This chapter describes approximations to Bayesian adaptive inference in both batch and incremental modes. Various marginalisation approximations, including a new variational Bayes (VB) approximation, are discussed. The application of these approaches to adaptively trained systems, such as SAT with MLLR or CAT, are then given.

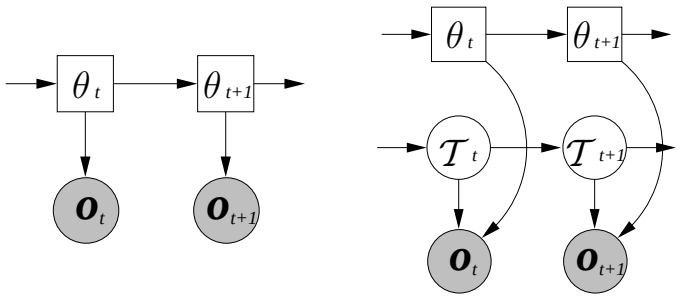
5.1 A Bayesian Framework for Adaptive Training and Inference

Adaptive training has become a popular technique to build systems on non-homogeneous training data [38]. A two-step *adaptation*/recognition process is usually used in inference on adaptively trained systems. Adaptive training and adaptation are normally described in a ML framework as discussed in chapter 3. This section will view adaptive training and inference from a Bayesian perspective.

Adaptive training and adaptive inference may be viewed as modifying the dynamic Bayesian network (DBN) associated with the acoustic condition. Figure 5.1 shows the comparison between an HMM and an adaptive HMM, where the transform is applied.

Here, θ_t represents the hidden Gaussian component at time t[1], $\mathbf{o}_t$ is the observation vector,

[1]Here Gaussian component θ is used instead of state ω to denote the hidden variable, which is slightly different



(a) HMM **(b) Adaptive HMM**

Figure 5.1 *Dynamic Bayesian network comparison between HMM and adaptive HMM*

and $\mathcal{T}_t$ is the transform representing non-speech variabilities, which is used to normalise features or modify HMM parameters for a particular acoustic condition. Figure 5.1(a) shows the DBN for an HMM. The observations are conditionally independent given the hidden variables. In contrast, figure 5.1(b) shows the DBN for an adaptive HMM, where an additional level of dependency is employed. The observations are additionally dependent on the transforms. As discussed in section 3.1.1, within a homogeneous data block, the transform is constrained to be unchanged.

The DBNs given in figure 5.1 can be used in various ways for training and inference. Standard speaker-independent (SI) training and decoding is an example of using the HMM DBN in both stages. It is also possible to use the HMM DBN in training and the adaptive HMM DBN in inference. This corresponds to the idea of performing adaptation on multi-style trained models. If the adaptive HMM DBN is used in training, a canonical model representing the desired speech variability is estimated given a set of transforms. In this case, it is not possible to use the HMM DBN for inference due to the lack of test domain specific transforms. The adaptive HMM DBN is required during inference to adapt the canonical model to the test domain. The effect of different ways of using the DBNs for training and inference will be shown in chapter 7.

5.1.1 Bayesian Adaptive Training

ML adaptive training has been reviewed in section 3.4. This section investigates Bayesian adaptive training, where the adaptive HMM DBN in figure 5.1(b) is employed. In adaptive training, the training data is split into homogeneous data blocks, $\mathcal{O} = \{\mathbf{O}^{(1)}, \ldots, \mathbf{O}^{(S)}\}$, where $\mathbf{O}^{(s)}$ is the observation sequence of a homogeneous block associated with a particular acoustic condition s. From the Bayesian perspective, two sets of parameters, the canonical model and transforms, are treated as random variables. The marginal likelihood of the training data can be expressed as

$$p(\mathcal{O}|\mathbf{H}) = \int_{\mathcal{M}} p(\mathcal{O}|\mathbf{H}, \mathcal{M}) p(\mathcal{M}|\Phi)\, d\mathcal{M} \tag{5.1}$$

from figure 3.1. This aims to relate the DBN to the likelihood calculation formulae in this chapter. The change does not affect the discussion of the general framework.

where $p(\mathcal{M}|\Phi)$ is the prior distribution for the canonical model parameters, Φ denotes the hyper-parameters of the model parameter prior[2] When using the adaptive HMM DBN in figure 5.1(b), the transform transition can be viewed as a first-order Markov process. As the transform is assumed to be the same within each homogeneous block, the transition probability of the Markov process is

$$p(\mathcal{T}_{t+1}|\mathcal{T}_t) = \delta(\mathcal{T}_{t+1} - \mathcal{T}_t) \tag{5.2}$$

where $\delta(\mathcal{T}_{t+1} - \mathcal{T}_t)$ is a *Dirac delta* function defined as

$$\delta(\mathcal{T} - \hat{\mathcal{T}}) = \begin{cases} \infty & \mathcal{T} = \hat{\mathcal{T}} \\ 0 & otherwise \end{cases} \tag{5.3}$$

and

$$\int_{\mathcal{T}} \delta(\mathcal{T} - \hat{\mathcal{T}})\, d\mathcal{T} = 1 \tag{5.4}$$

This means the transform is constant over all frames within *each* homogeneous block respectively. Considering that different homogeneous blocks are conditionally independent to each other, the homogeneity constraint results in

$$p(\mathcal{O}|\mathbf{H}, \mathcal{M}) = \prod_{s=1}^{S} \int_{\mathcal{T}} p(\mathbf{O}^{(s)}|\mathcal{H}^{(s)}, \mathcal{M}, \mathcal{T}) p(\mathcal{T}|\phi)\, d\mathcal{T} \tag{5.5}$$

where $\mathbf{H} = \{\mathcal{H}^{(1)}, \ldots, \mathcal{H}^{(S)}\}$ is the set of transcriptions, $p(\mathcal{T}|\phi)$ is the prior distributions for the transform parameters [3], ϕ denotes the hyper-parameters. The likelihood of the observation sequence of each acoustic condition given the parameters is expressed by

$$p(\mathbf{O}^{(s)}|\mathcal{H}^{(s)}, \mathcal{M}, \mathcal{T}) = \sum_{\boldsymbol{\theta}} P(\boldsymbol{\theta}|\mathcal{H}^{(s)}, \mathcal{M}) \prod_{t} p(\mathbf{o}_t|\mathcal{M}, \mathcal{T}, \theta_t) \tag{5.6}$$

where $P(\boldsymbol{\theta}|\mathcal{H}^{(s)}, \mathcal{M})$ is the probability of a particular component sequence $\boldsymbol{\theta}$, $p(\mathbf{o}_t|\mathcal{M}, \mathcal{T}, \theta_t)$ is the Gaussian distribution for component θ_t, and θ_t is the Gaussian component at time t.

Within the Bayesian framework, the aim of adaptive training is to update the prior distributions of the two sets of parameters to posterior distributions given the training data. Usually no prior information (except that the form of the prior is normally assumed to be known beforehand) is available before training, hence the estimation of the posterior distributions is equivalent to the empirical Bayesian estimation of the prior distributions. In this case, the aim of *Bayesian adaptive training* is to obtain the *prior distributions* of the two sets of parameters from the training data. The form of the prior distributions and the estimation of the hyper-parameters are two main issues to be considered.

[2] As the prior parameters are normally assumed to be independent of hypothesis, a generic prior $p(\mathcal{M}|\Phi)$ may be used instead of the strict expression $p(\mathcal{M}|\mathbf{H}, \Phi)$.

[3] Although the distribution of the transform parameters is dependent on the model set, for clarity of notation, this dependence has also been dropped. Hence the transform prior is expressed as $p(\mathcal{T}|\phi)$.

To obtain tractable mathematical update formulae, conjugate priors to the likelihood of complete data set are the preferable forms for the two prior distributions. When using a conjugate prior, the posterior distribution will have the same functional form as the prior and lead to simple mathematical formulae. When mean based transforms such as linear transform [74] or interpolation weights [36] are used, a Gaussian distribution over the transform parameters is a conjugate prior to the complete data set. In some cases, a single-component prior may not be able to account for the parameters variations. For example in the case of cluster adaptive training (CAT) initialised with gender information, the distribution of the interpolation weights may be highly bimodal, each mode representing one gender [36]. For these instances it makes sense to use a mixture model as the prior distribution for the transform parameters, or interpolation weights. For example, an N-component GMM may be used

$$p(\mathcal{T}|\phi) = \sum_{n=1}^{N} c_n \mathcal{N}(\mathcal{T}; \boldsymbol{\mu}_{\mathcal{T}}^{(n)}, \boldsymbol{\Sigma}_{\mathcal{T}}^{(n)}) \tag{5.7}$$

where c_n is the component weight. Unfortunately, this kind of mixture model is no longer a conjugate prior to the likelihood of the complete data set. Using a mixture prior distribution further complicates the training and inference. The effect will be discussed in detail in section 5.3.

Once the form of the prior distributions is determined, it is possible to estimate the hyper-parameters of the two priors using an *empirical Bayes* approach [97, 98]. The marginal likelihood in equation (5.1) is to be maximised with respect to the *hyper-parameters* of the priors[4]. Directly optimising the marginal likelihood in equation (5.1), is highly complex. One solution to this problem is to use a lower bound in order to make the optimisation feasible in a similar fashion to the EM algorithm. Initially the hyper-parameters estimation of the canonical model is considered. Introducing a variational distribution $q(\mathcal{M})$ of the canonical model parameters and applying Jensen's inequality yields a lower bound of the marginal likelihood as below

$$\begin{aligned} \log p(\mathcal{O}|\mathbf{H}) &\geq \left\langle \log \frac{p(\mathcal{O}|\mathbf{H},\mathcal{M})p(\mathcal{M}|\Phi)}{q(\mathcal{M})} \right\rangle_{q(\mathcal{M})} \\ &= \langle \log p(\mathcal{O}|\mathbf{H},\mathcal{M}) \rangle_{q(\mathcal{M})} - \mathtt{KL}\left(q(\mathcal{M})||p(\mathcal{M}|\Phi)\right) \end{aligned} \tag{5.8}$$

where $< f(x) >_{g(x)}$ is the expectation of $f(x)$ with respect to $g(x)$ and $\mathtt{KL}(\cdot||\cdot)$ is the *Kullback Leibler* (KL) distance of the two distributions. The KL distance between two continuous distributions is defined as

$$\mathtt{KL}\left(p_1(x)||p_2(x)\right) = \int_x p_1(x) \log \frac{p_1(x)}{p_2(x)}\, dx = \left\langle \log \frac{p_1(x)}{p_2(x)} \right\rangle_{p_1(x)} \tag{5.9}$$

and for two discrete distributions as

$$\mathtt{KL}\left(P_1(x)||P_2(x)\right) = \sum_x P_1(x) \log \frac{P_1(x)}{P_2(x)} = \left\langle \log \frac{P_1(x)}{P_2(x)} \right\rangle_{P_1(x)} \tag{5.10}$$

The KL distance is always positive unless the two distributions are the same, in which case the distance is zero.

[4] Hyper-parameters estimated using this approach are also referred to as an ML-II estimate [10, 96].

From the properties of Jensen's inequality, the inequality in equation (5.8) becomes an equality when

$$q(\mathcal{M}) = p(\mathcal{M}|\mathcal{O}, \mathbf{H}) \tag{5.11}$$

Maximising the lower bound in equation (5.8) with respect to the prior hyper-parameters Φ is equivalent to minimising the KL distance between the prior distribution $p(\mathcal{M}|\Phi)$ and $q(\mathcal{M})$. The optimal value for $q(\mathcal{M})$ is obtained from equation (5.11). Thus the empirical Bayesian estimate of the canonical model prior is just the posterior distribution

$$p(\mathcal{M}|\Phi) = q(\mathcal{M}) = p(\mathcal{M}|\mathcal{O}, \mathbf{H}) \tag{5.12}$$

Though the prior distribution can be estimated using the above equation, the posterior distribution $p(\mathcal{M}|\mathcal{O}, \mathbf{H})$ must be estimated. This requires maximising the first term in RHS of equation (5.8). This is equivalent to optimise $\log p(\mathcal{O}|\mathbf{H})$ with respect to the posterior distribution $p(\mathcal{M}|\mathcal{O}, \mathbf{H})$ given an uninformative prior. Again, the optimisation is complicated. However, with sufficient training data, an efficient optimisation scheme may be obtained, as discussed below.

The optimisation of the hyper-parameters of the transform prior is more complex than the canonical model prior due to the separate transforms being considered for different homogeneous block. As the homogeneous data blocks are assumed to be conditionally independent, one distinct transform posterior distribution $p(\mathcal{T}|\mathbf{O}^{(s)}, \mathcal{H}^{(s)}, \mathcal{M})$ is associated with each particular acoustic condition s. Applying Jensen's inequality to equation (5.5) yields

$$\begin{aligned} \log p(\mathcal{O}|\mathbf{H}, \mathcal{M}) &\geq \sum_{s=1}^{S} \left\langle \log \frac{p(\mathbf{O}^{(s)}|\mathcal{H}^{(s)}, \mathcal{M}, \mathcal{T}) p(\mathcal{T}|\phi)}{q^{(s)}(\mathcal{T})} \right\rangle_{q^{(s)}(\mathcal{T})} \\ &= \sum_{s=1}^{S} \left\langle \log p(\mathbf{O}^{(s)}|\mathcal{H}^{(s)}, \mathcal{M}, \mathcal{T}) \right\rangle_{q^{(s)}(\mathcal{T})} - \sum_{s=1}^{S} \mathtt{KL}\left(q^{(s)}(\mathcal{T}) || p(\mathcal{T}|\phi) \right) \end{aligned} \tag{5.13}$$

where an equality is achieved when

$$q^{(s)}(\mathcal{T}) = p(\mathcal{T}|\mathbf{O}^{(s)}, \mathcal{H}^{(s)}, \mathcal{M}) \tag{5.14}$$

It is worth noting that equation (5.13) is derived given a particular canonical model set. Hence, the estimation of transform posterior distributions should also consider the integral over $p(\mathcal{M}|\Phi)$. This makes the optimisation even more complicated. As the transform prior distribution $p(\mathcal{T}|\phi)$ is independent of acoustic conditions while the *multiple* variational transform posterior distributions $q^{(s)}(\mathcal{T})$ are dependent on each homogeneous block respectively, the KL distance in the inequality (5.13) can not be simply minimised to yield a result similar to equation (5.12) as in the canonical model case. In contrast to the Bayesian training of standard HMMs, where no transform prior distribution is used, this is a new issue of Bayesian adaptive training. This issue may be solved given sufficient training data as discussed below.

In practice, when building speech recognition systems, it is possible to control the complexity of the system being trained so that each Gaussian component and transform have "sufficient"

data as discussed in chapter 2 and chapter 3. For example minimum occupancies may be used during the construction of the decision trees for robust canonical model estimates. Transforms may be shared among groups of Gaussian components and homogeneous blocks may be clustered together. With these approaches, it is reasonable to assume that the variances of the parameter distribution is sufficiently small so that the posterior distributions over the parameters can be approximated by a Dirac delta function. Thus

$$p(\mathcal{M}|\mathcal{O},\mathbf{H}) \approx \delta(\mathcal{M}-\hat{\mathcal{M}}) \tag{5.15}$$

$$p(\mathcal{T}|\mathbf{O}^{(s)},\mathcal{H}^{(s)}) \approx \delta(\mathcal{T}-\hat{\mathcal{T}}^{(s)}) \tag{5.16}$$

where $\hat{\mathcal{M}}$ and $\hat{\mathcal{T}}^{(s)}$ are point estimates of the canonical model and transform for homogeneous block s respectively. Using equation (5.15) in equation (5.12) yields a canonical model prior which is also a Dirac delta function

$$p(\mathcal{M}|\Phi) \approx \delta(\mathcal{M}-\hat{\mathcal{M}}) \tag{5.17}$$

Using equation (5.17) in equation (5.8) and considering equation (5.12), it is trivial to show that the point estimate in equation (5.17) is an ML estimate

$$\hat{\mathcal{M}}_{\texttt{ML}} = \arg\max_{\mathcal{M}} p(\mathcal{O}|\mathbf{H},\mathcal{M},\hat{\boldsymbol{\mathcal{T}}}) \tag{5.18}$$

As shown in appendix C, by using equation (5.16) in equation (5.13), the hyper-parameters of the transform prior distribution can be estimated by

$$\hat{\phi} = \arg\max_{\phi} \sum_{s=1}^{S} \log p(\hat{\mathcal{T}}^{(s)}|\phi) \tag{5.19}$$

where the transform estimate for each acoustic condition s, $\hat{\mathcal{T}}^{(s)}$, is also an ML estimate

$$\hat{\mathcal{T}}^{(s)}_{\texttt{ML}} = \arg\max_{\mathcal{T}} p(\mathbf{O}^{(s)}|\mathcal{H}^{(s)},\hat{\mathcal{M}},\mathcal{T}) \tag{5.20}$$

From equation (5.19), the *hyper-parameters* of the transform prior distribution are the ML estimate obtained from the transform "samples" of each homogeneous block. Hence, given sufficient training data, the output of Bayesian adaptive training includes a Dirac delta function of the canonical model with the ML estimate as the hyper-parameters and a non-point transform prior distribution. Therefore, interpreting adaptive training from the Bayesian perspective not only justifies the use of the ML estimate of the canonical model, but also motivates a non-point transform prior distribution. It is worth emphasising that the transform prior distribution is dependent on the canonical model estimate. It can only be used as a counterpart for that particular canonical model set.

5.1.2 Bayesian Adaptive Inference

In the previous section, adaptive training has been formulated within a Bayesian framework. Two prior distributions are estimated and used for inference on test data. The inference on

adaptively trained systems requires the use of the adaptive HMM DBN in figure 5.1(b), referred to as *Bayesian adaptive inference*. Owing to the homogeneity constraint, the inference must be performed for each homogeneous block respectively. The aim of Bayesian adaptive inference is to find the optimal hypothesis sequence $\hat{\mathcal{H}}$ by making use of the marginal likelihood of observation sequence

$$\hat{\mathcal{H}} = \arg\max_{\mathcal{H}} P(\mathcal{H}) p(\mathbf{O}|\mathcal{H}) \tag{5.21}$$

where $\hat{\mathcal{H}}$ is the inferred hypothesis, $\mathbf{O}$ is the test observation sequence of a particular homogeneous block, the index of the homogeneous block is omitted for clarity. $p(\mathbf{O}|\mathcal{H})$ and $P(\mathcal{H})$ are acoustic scores and language scores of each hypothesis respectively. $P(\mathcal{H})$ is the probability of the hypothesis sequence $\mathcal{H}$, which is normally obtained from an N-gram language model. Similar to Bayesian inference with HMM distribution in section 2.6.3, the key problem here is to calculate $p(\mathbf{O}|\mathcal{H})$, the marginal likelihood of the observation sequence given every possible hypothesis. As adaptive HMM DBN is used, this marginalisation is consistent with the marginalisation in equations (5.1) and (5.5) during training except that it is done for one homogeneous block here. From equation (5.15), a point estimate of the canonical model will be used for inference because marginalisation over a Dirac delta function will result in a likelihood given the hyper-parameters of that Dirac delta function. Then, a transform distribution is required to calculate the marginal likelihood given the point estimate of the canonical model[5]

$$p(\mathbf{O}|\mathcal{H}) = \int_{\mathcal{T}} p(\mathbf{O}|\mathcal{H}, \mathcal{T}) p(\mathcal{T})\, d\mathcal{T} \tag{5.22}$$

where $\mathbf{O}$ is the observation sequence for a particular testing homogeneous data block, $\mathcal{H}$ is one possible hypothesis for $\mathbf{O}$. Depending on the nature of the task being addressed,there are also two modes of adaptive inference as discussed in section 3.1.2

- **Supervised mode**

  If both observations and transcriptions for some supervision data are available, a transform posterior distribution, given the supervision, can be estimated using

  $$p(\mathcal{T}|\mathbf{O}_{\text{supv}}, \mathcal{H}_{\text{supv}}) = \frac{p(\mathbf{O}_{\text{supv}}|\mathcal{T}, \mathcal{H}_{\text{supv}}) p(\mathcal{T}|\phi)}{p(\mathbf{O}_{\text{supv}}|\mathcal{H}_{\text{supv}})} \tag{5.23}$$

  where $\mathbf{O}_{\text{supv}}$ and $\mathcal{H}_{\text{supv}}$ are the observations and hypothesis of the supervision data, which comes from the same source as the data to be recognised and $p(\mathcal{T}|\phi)$ is the prior transform distribution. This transform posterior distribution is then used as $p(\mathcal{T})$ to calculate the marginal likelihood of the test data in equation (5.22). Due to the use of the transform posterior distribution, this form of Bayesian adaptive inference is sometimes referred to as *posterior adaptation* [38]. Direct calculation of the transform posterior distribution is hard. Approximations, such as variational Bayes, may be used [9]. As the amount of adaptation

[5]Though the likelihood calculation is performed given the point estimate of the canonical model $\hat{\mathcal{M}}$, the notation $\hat{\mathcal{M}}$ is omitted for clarity.

data increases, the posterior distribution may be approximated by Dirac delta function. In this case, the marginal likelihood is the likelihood given the hyper-parameters of the Dirac delta function, which is a point estimate. In this work, supervised mode will not be further discussed as there is no supervision data available for the tasks considered.

- **Unsupervised mode**

  In this mode, there is no supervision data available for the target domain. It is then necessary to rely on the prior information gathered about the transform from the training data. Hence, the marginal likelihood of the test data will be calculated using transform prior distribution $p(\mathcal{T}|\phi)$ in equation (5.22). This is the form of Bayesian adaptive inference considered in this work.

Bayesian adaptive inference described above is a strict implementation of adaptive HMM DBN in inference. It is interesting to compare this to the standard two-step adaptation/recognition approach. In standard supervised adaptation approach, a point estimate of the transforms is first obtained given some supervision data to adapt the canonical model. The adapted model is then used to recognise the test data. This is equivalent to the supervised mode of Bayesian adaptive inference given sufficient supervision data. When there is no supervision data available, it is not possible to directly decode using the canonical model with the standard adaptation/recognition process. A commonly used hack approach is to generate initial hypothesis using a multi-style trained model. Then, a process similar to supervised adaptation/recognition is applied. In contrast, Bayesian adaptive inference gives a strict framework for using canonical model in unsupervised mode. Due to the nature of adaptive HMMs, it is not possible to construct a standard set of "adapted" HMMs[6]. Therefore, there is no separation between "adaptation" and "recognition" steps. The whole process of Bayesian adaptive inference is to calculate the inference evidence for each possible hypothesis and select the one with the best evidence. Calculation of marginal likelihood for each possible hypothesis is the key problem, which is done by integrating out the transform prior distribution.

In recognition with standard HMMs, *Viterbi* algorithm [116] is usually used to calculate the likelihood of observation sequence as discussed in section 2.6.2.1. This relies on the conditional independence assumption of HMMs. As the assumption is not valid for adaptive HMM due to the additional dependency on transforms, Viterbi algorithm is not suitable for Bayesian adaptive inference. Instead, *N-Best rescoring* [103] is used in this work to reflect the nature of adaptive HMM. Though the N-Best rescoring may limit the performance gain, and loss, due to the limited number of candidate hypothesis sequences, given sufficient hypothesis candidates, this rescoring process is likely to produce the "best" hypothesis. In N-Best rescoring, marginal likelihood given every possible hypothesis, $p(\mathbf{O}|\mathcal{H})$ needs to be calculated. Due to the coupling of transform parameters and hidden state/component sequence, the Bayesian integral in equation (5.22) is again intractable. The calculation of the marginal likelihood $p(\mathbf{O}|\mathcal{H})$ then requires

[6]Unless the prior transform distribution is a Dirac delta function. However, as indicated in section 5.1, prior transform distribution is not a Dirac delta function for non-homogeneous data.

approximations. Various approximation approaches will be discussed in detail in section 5.3.

5.2 Discriminative Bayesian Adaptive Training and Inference

Section 5.1 described Bayesian adaptive training where the likelihood is used as the training criterion. This Bayesian framework can be extended to other criteria. In particular, discriminative criteria, such as MMI, are of interest. The general form of equations (5.1) and (5.5) using a general training criterion $\mathcal{F}(\cdot)$ may be expressed as

$$\mathcal{F}(\mathcal{O},\mathbf{H}) = \int_{\mathcal{M}} \mathcal{F}(\mathcal{O},\mathbf{H};\mathcal{M})p(\mathcal{M}|\Phi)\,d\mathcal{M} \tag{5.24}$$

$$\mathcal{F}(\mathcal{O},\mathbf{H};\mathcal{M}) = \prod_{s=1}^{S}\int_{\mathcal{T}} \mathcal{F}(\mathbf{O}^{(s)},\mathcal{H}^{(s)};\mathcal{M},\mathcal{T})p(\mathcal{T}|\phi)\,d\mathcal{T} \tag{5.25}$$

where s is the index of homogeneous block, $p(\mathcal{M}|\Phi)$ and $p(\mathcal{T}|\phi)$ are the prior distributions of the canonical model and transforms respectively. These prior distributions are associated with a particular training criterion. $\mathcal{F}(\mathbf{O}^{(s)},\mathcal{H}^{(s)};\mathcal{M},\mathcal{T})$ is the general training criterion given the parameter sets $\mathcal{M}$ and $\mathcal{T}$, $\mathcal{F}(\mathcal{O},\mathbf{H})$ is the marginalised criterion.

Equations (5.24) and (5.25) give a general Bayesian adaptive training framework for any training criterion. If the likelihood criterion is used, i.e.

$$\mathcal{F}(\mathbf{O}^{(s)},\mathcal{H}^{(s)};\mathcal{M},\mathcal{T}) = p(\mathbf{O}^{(s)}|\mathcal{H}^{(s)},\mathcal{M},\mathcal{T}) \tag{5.26}$$

the general form becomes the Bayesian adaptive training framework discussed in section 5.1.

State-of-the-art speech recognition systems use discriminative training criteria to obtain the best performance. Discriminative training has previously been studied within the adaptive training framework with various forms of transforms [60, 119, 140] as discussed in chapter 4. In these works, both sets of parameters are assumed to be deterministic. In this section, discriminative adaptive training and inference are discussed within a Bayesian framework by using a discriminative criterion as the general criterion in equations (5.24) and (5.25). The maximum mutual information (MMI) criterion [60] is used as an example for the discussion in this section. It can be expressed as the posterior distribution of the correct transcription and may be written as

$$\mathcal{F}(\mathcal{O},\mathbf{H}) = P(\mathbf{H}|\mathcal{O}) = \int_{\mathcal{M}} P(\mathbf{H}|\mathcal{O},\mathcal{M})p(\mathcal{M}|\Phi)\,d\mathcal{M} \tag{5.27}$$

and from the conditional independence of different homogeneous blocks,

$$P(\mathbf{H}|\mathcal{O},\mathcal{M}) = \prod_{s=1}^{S}\int_{\mathcal{T}} P(\mathcal{H}^{(s)}|\mathbf{O}^{(s)},\mathcal{M},\mathcal{T})p(\mathcal{T}|\phi)\,d\mathcal{T} \tag{5.28}$$

Using Bayes rule, the posterior of the transcription given both sets of parameters can be written as[7]

$$P(\mathcal{H}^{(s)}|\mathbf{O}^{(s)},\mathcal{M},\mathcal{T}) = \frac{p(\mathbf{O}^{(s)}|\mathcal{H}^{(s)},\mathcal{M},\mathcal{T})P(\mathcal{H}^{(s)})}{\sum_{\breve{\mathcal{H}}^{(s)}} p(\mathbf{O}^{(s)}|\breve{\mathcal{H}}^{(s)},\mathcal{M},\mathcal{T})P(\breve{\mathcal{H}}^{(s)})} \tag{5.29}$$

[7]The likelihood scaling factor in [102] is ignored here as it does not affect the discussion in this section.

where $\breve{\mathcal{H}}^{(s)}$ is drawn from the set of all possible hypotheses for homogeneous block s.

In many situations, there is no information about the hyper-parameters of prior distributions in advance. Hence, they also need to be empirically estimated from the training data. When using a discriminative criterion, the prior distributions $p(\mathcal{M}|\Phi)$ and $p(\mathcal{T}|\phi)$ are estimated so that the discriminative criterion, rather than the likelihood criterion, is optimised. As in likelihood based Bayesian adaptive training, the forms the prior distributions need to be determined before training. To the author's knowledge, there is little discussion about the appropriate conjugate prior distributions for generative models trained with discriminative criteria. This is an important issue remained for further investigation as mentioned in chapter 8. However, independent of the form of the prior, the same general estimation process of the hyper-parameters may be used. Applying Jensen's inequality to equations (5.27) and (5.28) yields inequalities similar to inequalities (5.8) and (5.13)

$$\log P(\mathbf{H}|\mathcal{O}) \geq \langle \log P(\mathbf{H}|\mathcal{O},\mathcal{M})\rangle_{q(\mathcal{M})} - \mathtt{KL}\left(q(\mathcal{M})||p(\mathcal{M}|\Phi)\right) \tag{5.30}$$

$$\log P(\mathbf{H}|\mathcal{O},\mathcal{M}) \geq \sum_{s=1}^{S}\left\langle \log P(\mathcal{H}^{(s)}|\mathbf{O}^{(s)},\mathcal{M},\mathcal{T})\right\rangle_{q^{(s)}(\mathcal{T})} - \sum_{s=1}^{S}\mathtt{KL}\left(q^{(s)}(\mathcal{T})||p(\mathcal{T}|\phi)\right) \tag{5.31}$$

The above become equalities when

$$q(\mathcal{M}) = p(\mathcal{M}|\mathcal{O},\mathbf{H}) \tag{5.32}$$

$$q^{(s)}(\mathcal{T}) = p(\mathcal{T}|\mathbf{O}^{(s)},\mathcal{H}^{(s)},\mathcal{M}) \tag{5.33}$$

Using similar proof for likelihood based Bayesian adaptive training in appendix C, it can be shown that maximising equation (5.30) yields

$$p(\mathcal{M}|\Phi) = p(\mathcal{M}|\mathcal{O},\mathbf{H}) \tag{5.34}$$

and $p(\mathcal{T}|\phi)$ is estimated using the transform posterior distributions $p(\mathcal{T}|\mathbf{O}^{(s)},\mathcal{H}^{(s)},\mathcal{M})$[8].

With appropriate model complexity control and sufficient training data, the posterior distributions of both sets of parameters can also be approximated using Dirac delta functions in the same fashion as ML adaptive training. The hyper-parameters of these Dirac delta functions are point estimates of the canonical model and the transforms for each homogeneous block respectively. However, these point estimates are *discriminative* estimates rather than the ML estimates as in section 5.1.1. They can be estimated using the standard discriminative training approach as discussed in chapter 4. For example, using MMI criterion [60]

$$\hat{\mathcal{M}}_{\mathtt{MMI}} = \arg\max_{\mathcal{M}} P(\mathbf{H}|\mathcal{O},\mathcal{M},\hat{\boldsymbol{\mathcal{T}}}) \tag{5.35}$$

Though the transform posterior distribution associated with each homogeneous block can be approximated as a Dirac delta function given sufficient training data, the transform prior distribution is a non-point distribution due to the multiple posterior distributions associated

[8]The estimation of the discriminative posterior distribution estimation was also rarely investigated due to the lack of an appropriate conjugate prior to the discriminative criterion.

with different homogeneous blocks. The estimation formula of the hyper-parameters is similar to likelihood based training case in equation (5.19)[9]. However, there is a fundamental difference during training. The point estimates of the transform posterior distributions are discriminatively estimated, rather than ML estimated. These can be estimated using the standard discriminative training approaches, for example, using MMI criterion [60]

$$\hat{\mathcal{T}}^{(s)}_{\texttt{MMI}} = \arg\max_{\mathcal{T}} P(\mathcal{H}^{(s)}|\mathbf{O}^{(s)}, \hat{\mathcal{M}}, \mathcal{T}) \tag{5.36}$$

Using the above approximate discriminative Bayesian adaptive training will yield a Dirac delta prior distribution of the canonical model parameters and a non-point prior distribution of the transform parameters. As both prior distributions are obtained by maximising a discriminative criterion, they should also be used to calculate "discriminative" evidence in inference. Thus, the inference criterion for discriminative adaptive systems on a homogeneous block can be written as

$$\hat{\mathcal{H}} = \arg\max_{\mathcal{H}} P(\mathcal{H}|\mathbf{O}) \tag{5.37}$$

where $\mathbf{O}$ is the observation sequence of the test data, $\mathcal{H}$ is one possible hypothesis sequence, $P(\mathcal{H}|\mathbf{O})$ is the inference evidence which can be written as

$$P(\mathcal{H}|\mathbf{O}) = \int_{\mathcal{T}} p(\mathcal{H}|\mathbf{O}, \hat{\mathcal{M}}_{\texttt{MMI}}, \mathcal{T}) p(\mathcal{T}|\phi)\, d\mathcal{T} = \int_{\mathcal{T}} \frac{p(\mathbf{O}|\mathcal{H}, \hat{\mathcal{M}}_{\texttt{MMI}}, \mathcal{T}) P(\mathcal{H})}{\sum_{\breve{\mathcal{H}}} p(\mathbf{O}|\breve{\mathcal{H}}, \hat{\mathcal{M}}_{\texttt{MMI}}, \mathcal{T}) P(\breve{\mathcal{H}})} p(\mathcal{T}|\phi)\, d\mathcal{T} \tag{5.38}$$

where $\hat{\mathcal{M}}_{\texttt{MMI}}$ is the MMI estimate of the canonical model[10], $\breve{\mathcal{H}}$ are all possible hypotheses including the current hypothesis $\mathcal{H}$, $p(\mathcal{T}|\phi)$ is the prior distribution of discriminative transform parameters. Thus, the distinct *discriminative evidence* for each possible hypothesis, $P(\mathcal{H}|\mathbf{O})$, is calculated by integrating out over the discriminative transform prior distribution. The hypothesis with the best discriminative evidence is selected as the recognition output. This is a similar N-Best rescoring process to likelihood based Bayesian adaptive inference except for the calculation of inference evidence.

It is interesting to compare the discriminative evidence in equation (5.37) to the likelihood based evidence in equation (5.21). In likelihood based evidence in equation (5.21), only the acoustic part, i.e., the likelihood $p(\mathbf{O}|\mathcal{H}, \mathcal{T}, \mathcal{M})$, is marginalised because the canonical model parameters and transform prior are estimated by maximising the *marginal likelihood* criterion during training. The language model and confusion hypotheses are not included in the marginalisation integral. In contrast, the discriminative inference criterion in equation (5.37) is a marginalisation of the conditional likelihood $P(\mathcal{H}|\mathbf{O})$. It is consistent with the MMI training criterion. As both training and inference use the consistent discriminative criterion, it is believed to have more discriminative power than the marginal likelihood criterion. Hence, it may give better recognition performance.

[9] The proof is similar to appendix C except that the posterior $P(\mathbf{H}|\mathcal{O})$ is used as the criterion instead of the likelihood criterion $p(\mathcal{O}|\mathbf{H})$.

[10] This is the result of integrating out over a Dirac delta canonical model prior distribution.

One advantage of using a discriminative Bayesian adaptive training and inference framework is that adaptive inference can be effectively performed in unsupervised mode given the two prior distributions[11]. As there is no separate "adaptation" stage, there is no requirement of a correct transcription in unsupervised mode. The whole process is to calculate the discriminative evidence for each hypothesis given all possible hypotheses and then infer on those evidence values. However, as mentioned before, there is little discussion about the form and hyper-parameters estimation of discriminative prior distributions. Though the problem is interesting for further investigation as indicated chapter 8, it is not the focus of this work. The complete framework of discriminative Bayesian adaptive training and adaptive inference will not be further discussed.

An alternative strategy to complete discriminative adaptive training has been discussed in section 4.1. This is referred to as *simplified discriminative adaptive training* [119]. Here only the canonical model is updated using a discriminative criterion with the ML estimated transforms fixed during discriminative training. In adaptation/recognition, the ML criterion is used to estimate test set transforms for the discriminative canonical model. This strategy can also be extended to discriminative Bayesian adaptive training and adaptive inference. The canonical model prior distribution (a Dirac delta distribution) is obtained using discriminative criterion, whereas, the transform prior distribution (a non-point distribution) is obtained using likelihood criterion. As the likelihood criterion is used to estimate transforms, a conjugate prior distribution to the likelihood of the complete data set may be effectively estimated on the ML estimates of transforms. This transform prior distribution can then be consistently used for likelihood based Bayesian adaptive inference. This is the training and inference process adopted in this work. The minimum phone error (MPE) criterion [93], rather than the maximum mutual information (MMI) criterion [60] will be used as the discriminative criterion for training, which has been detailed in chapter 4. A slight difference in procedure from section 4.1 is that an additional step is required to estimate the transform prior distribution. Given the discriminatively estimated model $\hat{\mathcal{M}}_{\texttt{MPE}}$, ML criterion is again used to generate a new set of transform estimates $\{\hat{\mathcal{T}}_{\texttt{ML}}^{(1)}, \cdots, \hat{\mathcal{T}}_{\texttt{ML}}^{(S)}\}$. The hyper-parameters of the transform prior distribution $p(\mathcal{T}|\phi)$ are then estimated by using equation (5.19) with these ML transform estimates.

It is worth emphasising that during inference the marginal likelihood is calculated given the discriminative canonical model rather than an ML model. The use of a discriminative canonical model may significantly improve the recognition performance [118]. However, as the estimation of the transform prior distribution is based on ML transform estimates and the transform prior is applied in a non-discriminative way during inference, the discriminative ability of the discriminative adaptive system may be limited. This effect will be discussed in chapter 7.

[11] As discussed in section 4.1, within the standard adaptation/recognition framework, even if the initial hypothesis may be generated, unsupervised discriminative adaptation is still a problem because the recognised initial hypothesis can not be effectively used as the correct transcription for discriminative transform estimation. Hence, the feasibility of doing unsupervised mode discriminative inference within this Bayesian framework is even more interesting than likelihood based Bayesian adaptive inference and needs to be further investigated.

5.3 Approximate Inference Schemes

The marginal likelihood required for Bayesian adaptive inference, is an integral over a transform distribution, which is shown in equation (5.22). It is re-written here[12]

$$p(\mathbf{O}|\mathcal{H}) = \int_{\mathcal{T}} p(\mathbf{O}|\mathcal{H},\mathcal{T})p(\mathcal{T})\, d\mathcal{T} \tag{5.39}$$

where $\mathbf{O}$ is the observation sequence for a particular testing homogeneous data block, $\mathcal{H}$ is one possible hypothesis for $\mathbf{O}$ and $p(\mathcal{T})$ is a transform distribution. As unsupervised mode is the focus of this work, in the rest of this work, $p(\mathcal{T})$ refers to the transform prior distribution $p(\mathcal{T}|\phi)$[13]. This integral is generally intractable, hence approximations are required. In this section, two main categories of approximation approaches are described. One approach is to iteratively tighten a lower bound on the integral, referred to as *lower bound* approximation. The second is to directly approximate the integral, referred to as *direct* approximation. The application of those general approximation approaches to specific types of transforms will be discussed in section 5.5.

5.3.1 Lower Bound Approximations

As described in Bayesian adaptive training, a lower bound may be constructed to approximate the marginal likelihood in equation (5.39). Introducing a joint distribution, $q(\boldsymbol{\theta},\mathcal{T})$, over the component sequence, $\boldsymbol{\theta}$, and transform parameters, $\mathcal{T}$, and applying Jensen's inequality yields a lower bound as below

$$\log p(\mathbf{O}|\mathcal{H}) \geq \mathcal{L}(\mathbf{O}|\mathcal{H}) = \left\langle \log \frac{p(\mathbf{O},\boldsymbol{\theta}|\mathcal{T},\mathcal{H})p(\mathcal{T})}{q(\boldsymbol{\theta},\mathcal{T})} \right\rangle_{q(\boldsymbol{\theta},\mathcal{T})} \tag{5.40}$$

where $\mathcal{L}(\mathbf{O}|\mathcal{H})$ denotes the lower bound to $\log p(\mathbf{O}|\mathcal{H})$. The above becomes an equality when

$$q(\boldsymbol{\theta},\mathcal{T}) = p(\boldsymbol{\theta},\mathcal{T}|\mathbf{O},\mathcal{H}) = P(\boldsymbol{\theta}|\mathbf{O},\mathcal{H},\mathcal{T})p(\mathcal{T}|\mathbf{O},\mathcal{H}) \tag{5.41}$$

Using equation (5.41) is impractical because the calculation of the transform posterior $p(\mathcal{T}|\mathbf{O},\mathcal{H})$ requires the marginal likelihood $p(\mathbf{O}|\mathcal{H})$, so approximations are used. Two tractable forms of variational distributions are used instead of $P(\boldsymbol{\theta}|\mathbf{O},\mathcal{H},\mathcal{T})$ and $p(\mathcal{T}|\mathbf{O},\mathcal{H})$. To obtain good approximation, the lower bound needs to be made as tight as possible. Hence, an iterative *learning* process is then used to update the two variational distributions so that the lower bound is guaranteed not to decrease at each iteration. The tightness of the bound is then dependent on the form of the variational distributions and the number of iterations. Therefore, the tightness of the lower bound can be efficiently controlled, which is an advantage of the lower bound approximation.

[12] The canonical model $\hat{\mathcal{M}}$ and the index of homogeneous block are also omitted for clarity

[13] Given a transform posterior distribution, all approximation approaches in this section can also apply.

5.3.1.1 Lower Bound Based Inference

When using the lower bound approximation in Bayesian inference, it is assumed that the rank ordering of the inference evidence $P(\mathcal{H})p(\mathbf{O}|\mathcal{H})$ in equation (5.21) is similar to the ordering of the lower bound based evidence, i.e.

$$\begin{aligned} & \mathcal{L}(\mathbf{O}|\mathcal{H}_1) + \log P(\mathcal{H}_1) > \mathcal{L}(\mathbf{O}|\mathcal{H}_2) + \log P(\mathcal{H}_2) \\ \Rightarrow \quad & \log p(\mathbf{O}|\mathcal{H}_1) + \log P(\mathcal{H}_1) > \log p(\mathbf{O}|\mathcal{H}_2) + \log P(\mathcal{H}_2) \end{aligned} \tag{5.42}$$

There is no guarantee that this is the case. However, as the lower bound approximation becomes tighter to the real marginal likelihood, the rank ordering of the evidence should also become more similar. Thus, it is important to make the lower bound approximation as tight as possible. In addition to the form of the variational distributions and the number of iterations, the tightness is also dependent on the hypothesis to which the lower bound is optimised. In order to get a tight lower bound, it is necessary to optimise the lower bound with respect to *every possible* hypothesis respectively. During this process, multiple iterations are used to optimise a distinct variational transform distribution for $\mathcal{H}$. The specifically optimised transform distribution is then used to calculate $\mathcal{L}(\mathbf{O}|\mathcal{H})$ for inference.

This general approach of estimating a separate transform distribution for each candidate hypothesis is similar to *N-Best supervision* based adaptation [82]. In [82] an ML estimate of a bias vector was obtained, and used to calculate the likelihood for each of the N-Best candidates. No theoretical justification for the approach was proposed. In contrast, the work here motivates it from a viewpoint of tightening the lower bound during adaptive inference. It is interesting to compare the N-Best supervision to the standard *1-Best* supervision adaptation approaches such as iterative MLLR [132]. In iterative MLLR, a transform is estimated using the 1-Best hypothesis of the test data as supervision. This transform is then used to calculate inference evidence for *all* possible hypothesis and the process repeated if necessary. Maximising the lower bound with respect to a single hypothesis (the 1-Best hypothesis) may lead to a tight lower bound for that particular hypothesis. However, for the other competing hypotheses, the lower bounds are not as tight as they could be. They could have been made greater by optimising with respect to each individual hypothesis. Using 1-Best supervision may significantly affect the performance especially for complex transforms or short sentences as shown in the experimental results in section 7.2.2.2. This hypothesis-bias problem was also discussed in [38] but from a standard unsupervised adaptation perspective.

5.3.1.2 Point Estimates

In the same fashion as ML adaptive training, a Dirac delta function may be used as the transform posterior resulting in a point version of equation (5.41)

$$q(\boldsymbol{\theta}, \mathcal{T}) = P(\boldsymbol{\theta}|\mathbf{O}, \mathcal{H}, \mathcal{T})\delta(\mathcal{T} - \hat{\mathcal{T}}) \tag{5.43}$$

where $\hat{\mathcal{T}}$ is a point estimate of the transform for the target domain. The lower bound expression in equation (5.40) may then be re-expressed as

$$\log p(\mathbf{O}|\mathcal{H}) \geq \mathcal{L}_{\texttt{MAP}}(\hat{\mathcal{T}}) = \log p(\mathbf{O}|\mathcal{H}, \hat{\mathcal{T}}) + \log p(\hat{\mathcal{T}}) + \texttt{H}\left(\delta(\mathcal{T} - \hat{\mathcal{T}})\right) \tag{5.44}$$

where $\mathcal{L}_{\texttt{MAP}}(\hat{\mathcal{T}})$ is a brief notation to the lower bound of the log-likelihood given the MAP estimate of transform $\mathcal{L}_{\texttt{MAP}}(\mathbf{O}|\mathcal{H}, \hat{\mathcal{T}})$ and $\texttt{H}(\cdot)$ is the entropy function defined in equation (2.16). For all point estimates of $\hat{\mathcal{T}}$, the entropy of the Dirac delta function remains $-\infty$ [24]. As $\texttt{H}\left(\delta(\mathcal{T} - \hat{\mathcal{T}})\right)$ is a negative constant with infinite value, it can be ignored without affecting the rank ordering of the lower bound. The rank ordering of $\mathcal{L}_{\texttt{MAP}}(\hat{\mathcal{T}})$ can be derived from

$$\mathcal{K}_{\texttt{MAP}}(\hat{\mathcal{T}}) = \log p(\mathbf{O}|\mathcal{H}, \hat{\mathcal{T}}) + \log p(\hat{\mathcal{T}}) \tag{5.45}$$

Equation (5.45) is a maximum a posteriori (MAP) based point estimate. However, $\mathcal{K}_{\texttt{MAP}}(\hat{\mathcal{T}})$ is no longer a lower bound for $\log p(\mathbf{O}|\mathcal{H})$. Using equation (5.45) for inference, an assumption is made that the inference evidence calculated with $\mathcal{K}_{\texttt{MAP}}(\hat{\mathcal{T}})$ will yield similar rank ordering as the inference evidence calculated with the real marginal likelihood. This assumption tends to be true for sufficient test data, where the point estimate is a reasonable approximation.

As discussed before, it is important to make the lower bound as close to the marginal likelihood as possible to get a good rank ordering approximation. Due to the negative infinity entropy term, the lower bound in the point estimate case, $\mathcal{L}_{\texttt{MAP}}(\hat{\mathcal{T}})$, is a very loose bound. However, it should still be tightened to make the rank ordering approximation better. Tightening the lower bound $\mathcal{L}_{\texttt{MAP}}(\hat{\mathcal{T}})$ is equivalent to tightening $\mathcal{K}_{\texttt{MAP}}(\hat{\mathcal{T}})$ in terms of rank ordering. The EM algorithm may be used to optimise $\mathcal{K}_{\texttt{MAP}}(\hat{\mathcal{T}})$. The auxiliary function for equation (5.45) is the standard MAP auxiliary function [21, 15]

$$\mathcal{Q}_{\texttt{MAP}}(\mathcal{T}_{k+1}; \hat{\mathcal{T}}_k) = \langle \log p(\mathbf{O}, \boldsymbol{\theta}|\mathcal{T}_{k+1}, \mathcal{H}) \rangle_{P(\boldsymbol{\theta}|\mathbf{O}, \mathcal{H}, \hat{\mathcal{T}}_k)} + \log p(\mathcal{T}_{k+1}) \tag{5.46}$$

where $P(\boldsymbol{\theta}|\mathbf{O}, \mathcal{H}, \hat{\mathcal{T}}_k)$ is the component sequence posterior calculated based on $\hat{\mathcal{T}}_k$. The transform estimate is iteratively updated until the estimate reaches a local maximum. This estimate will be a MAP estimate.

The transform prior $p(\mathcal{T})$ in equation (5.46) has been assumed to be a conjugate prior distribution to the likelihood of the complete data set. For example, a single Gaussian distribution may be used as the conjugate prior distribution for mean transforms. If a multiple-component prior, for example a GMM prior as defined in equation (5.7), is used, it is no longer a conjugate prior to the likelihood of the complete data set. However, Jensen's inequality may be applied to the mixture prior to yield a lower bound approximation. At iteration $k+1$

$$\begin{aligned} \log p(\mathcal{T}_{k+1}) &= \log \sum_n c_n p(\mathcal{T}_{k+1}|n) \\ &\geq \left\langle \frac{\log c_n p(\mathcal{T}_{k+1}|n)}{q_k(n)} \right\rangle_{q_k(n)} \end{aligned} \tag{5.47}$$

where $p(\mathcal{T}_{k+1}|n)$ is the n^{th} conjugate distribution component of the prior, c_n is the component weight, $q_k(n)$ is the variational component weight, which is introduced to allow the EM algorithm to be used. It is calculated using the transform parameters of the k^{th} iteration.

$$q_k(n) = P(n|\hat{\mathcal{T}}_k) = \frac{c_n p(\hat{\mathcal{T}}_k|n)}{\sum_n c_n p(\hat{\mathcal{T}}_k|n)} \tag{5.48}$$

Using this iterative form with a multiple-component prior introduces a further lower bound in additional to the lower bound to marginal likelihood. Hence, it will increase the difference between the lower bound and the actual likelihood. However, it simplifies the use of multiple-component prior. Substituting this into the transform estimation auxiliary function yields (ignoring constants)

$$\mathcal{Q}_{\texttt{MAP}}(\mathcal{T}_{k+1}; \hat{\mathcal{T}}_k) \quad = \quad \langle \log p(\mathbf{O}, \boldsymbol{\theta}|\mathcal{T}_{k+1}, \mathcal{H}) \rangle_{P(\boldsymbol{\theta}|\mathbf{O},\mathcal{H},\hat{\mathcal{T}}_k)} + \langle \log p(\mathcal{T}_{k+1}|n) \rangle_{P(n|\hat{\mathcal{T}}_k)} \tag{5.49}$$

Optimising the above auxiliary function, a locally optimal MAP estimate of transforms can be obtained. This estimate can then be used in equation (5.45) to calculate $\mathcal{K}_{\texttt{MAP}}(\hat{\mathcal{T}})$ for inference. It is worth noting that, within the N-Best supervision framework, the MAP estimate is distinct for *each* possible hypothesis. Therefore, when calculating $\mathcal{K}_{\texttt{MAP}}(\hat{\mathcal{T}})$, the prior term $p(\hat{\mathcal{T}})$ is different for each hypothesis. This gives an optimal lower bound for each individual hypothesis. It is interesting to compare this to the standard MAP inference [15]. In the standard approach, the transform parameters are estimated using the 1-Best hypothesis. The same set of transform parameters is then used to calculate the lower bound value, hence, the prior term $p(\hat{\mathcal{T}})$ does not affect the inference at all. Though the particular set of transform parameters may give an optimal lower bound for the 1-Best hypothesis, it gives a looser bound for the other hypotheses compared to N-Best supervision. Hence, the inference performance may be affected. The effect on error rate of this is discussed in section 7.2.2.2. The MAP estimate will tend to the standard ML estimate, if a non-informative prior is used or the data is sufficient. In this case, the prior term in equation (5.45) disappears and the observation sequence likelihood given the ML estimate can be directly used for inference.

5.3.1.3 Variational Bayes

Though the point estimate schemes allow the "true" marginal likelihood to be calculated given sufficient data[14], they are not robust for limited data. It is preferable to marginalise over a distribution to achieve more robust inference. In order to make the marginal likelihood calculation tractable, a variational Bayesian (VB) approximation may be used [9]. In the VB approximation, the component sequence posterior distribution $q(\boldsymbol{\theta}|\mathbf{O}, \mathcal{H})$ and the transform posterior distribution $q(\mathcal{T}|\mathbf{O}, \mathcal{H})$ are assumed to be conditionally independent. Thus

$$q(\boldsymbol{\theta}, \mathcal{T}) = q(\boldsymbol{\theta}|\mathbf{O}, \mathcal{H}) q(\mathcal{T}|\mathbf{O}, \mathcal{H}) \tag{5.50}$$

[14] Given sufficient data, equation (5.41) is equivalent to the point estimate assumption in equation (5.43). In that case, the rank ordering of the optimal lower bound value is the same as that of the marginal likelihood.

For simplicity of notation the two posteriors will be denoted as $q(\boldsymbol{\theta})$ and $q(\mathcal{T})$ in later derivations. This assumption is necessary to obtain a tractable mathematical form. The lower bound in equation (5.40) may be re-written as an auxiliary function. At the k^{th} iteration, it may be expressed as[15]

$$\begin{aligned}\log p(\mathbf{O}|\mathcal{H}) &\geq \mathcal{L}_{\mathtt{VB}}\left(q_k(\mathcal{T})\right) = \mathcal{Q}_{\mathtt{VB}}\left(q_{k+1}(\boldsymbol{\theta}), q_k(\mathcal{T})\right) \\ &= \left\langle \log p(\mathbf{O}, \boldsymbol{\theta}|\mathcal{T}, \mathcal{H})\right\rangle_{q_{k+1}(\boldsymbol{\theta})q_k(\mathcal{T})} + \mathtt{H}\left(q_{k+1}(\boldsymbol{\theta})\right) - \mathtt{KL}(q_k(\mathcal{T})||p(\mathcal{T})) \end{aligned} \quad (5.51)$$

where $\mathtt{H}(\cdot)$ is the entropy and $\mathtt{KL}(\cdot||\cdot)$ is the KL distance. $\mathcal{L}_{\mathtt{VB}}\left(q(\mathcal{T})\right)$ is the brief notation for the lower bound of the log-likelihood given the VB transform posterior distribution $\mathcal{L}_{\mathtt{VB}}\left(\mathbf{O}|\mathcal{H}, q(\mathcal{T})\right)$, $q_k(\boldsymbol{\theta})$ and $q_k(\mathcal{T})$ are the variational component sequence and transform posterior distributions at the k^{th} iteration respectively. The aim is to obtain forms of $q(\boldsymbol{\theta})$ and $q(\mathcal{T})$ that maximise this auxiliary function, thus making the lower bound as tight as possible. The "optimal" variational transform distribution will then be used to calculate the lower bound for inference.

Taking the functional derivatives of the auxiliary function in equation (5.51) with respect to the two distributions, $q(\boldsymbol{\theta})$ and $q(\mathcal{T})$, respectively, an EM-like algorithm can be obtained, referred to as *variational Bayesian EM* (VBEM) [9]. VBEM is guaranteed not to loosen the bound at each iteration. The process is described below:

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

2. **VB Expectation (VBE):**

   The optimal variational posterior component sequence distribution is shown to be [9]

   $$q_k(\boldsymbol{\theta}) = \frac{1}{\mathcal{Z}_{\boldsymbol{\Theta}}(\mathbf{O}, \mathcal{H})} \exp\left(\left\langle \log p(\mathbf{O}, \boldsymbol{\theta}|\mathcal{T}, \mathcal{H})\right\rangle_{q_{k-1}(\mathcal{T})}\right) \quad (5.52)$$

   where $\mathcal{Z}_{\boldsymbol{\Theta}}(\mathbf{O}, \mathcal{H})$ is the normalisation term to make $q_k(\boldsymbol{\theta})$ a valid distribution, $\boldsymbol{\Theta}$ denotes the set of all possible component sequences, and $q_k(\mathcal{T})$ is the variational transform distribution of the k^{th} iteration. As $\log p(\mathbf{O}, \boldsymbol{\theta}|\mathcal{T}, \mathcal{H})$ can be factorised at frame-level, the expectation with respect to $q_k(\mathcal{T})$ can be computed at frame-level in the logarithm domain.

   $$\left\langle \log p(\mathbf{O}, \boldsymbol{\theta}|\mathcal{T}, \mathcal{H})\right\rangle_{q_{k-1}(\mathcal{T})} = \left\langle \log P(\boldsymbol{\theta})\right\rangle_{q_{k-1}(\mathcal{T})} + \sum_t \left\langle \log p(\mathbf{o}_t|\mathcal{T}, \theta_t)\right\rangle_{q_{k-1}(\mathcal{T})} \quad (5.53)$$

   This allows $q_k(\boldsymbol{\theta})$ to be viewed as a component sequence posterior distribution of a model set with a modified Gaussian component[16]

   $$\tilde{p}(\mathbf{o}_t|\theta_t) = \exp\left(< \log p(\mathbf{o}_t|\mathcal{T}, \theta_t) >_{q_{k-1}(\mathcal{T})}\right) \quad (5.54)$$

[15] In VB, the notation of auxiliary function $\mathcal{Q}_{\mathtt{VB}}(\cdot, \cdot)$ is slightly different from the auxiliary function for point estimate $\mathcal{Q}(\cdot;\cdot)$. The VB auxiliary function is used for deriving calculation formulae for both component sequence $q(\boldsymbol{\theta})$ and variational transform distribution $q(\mathcal{T})$. Hence, both variational distributions are regarded as independent variables. In contrast, the point estimate auxiliary function is only used for deriving update formula of model or transform parameters *given* the posterior component sequence calculated based on the current parameter estimates.

[16] As the transform $\mathcal{T}$ is assumed to only work on Gaussian parameters, the other parameters such as transition probability will not change at all.

$\tilde{p}(\mathbf{o}|\theta)$ is referred to as a *pseudo-distribution* because it is not necessarily correctly normalised to be a valid distribution. Similar to the standard likelihood calculation in equation (5.6), $\mathcal{Z}_{\boldsymbol{\Theta}}(\mathbf{O},\mathcal{H})$ can then be simply calculated using the forward algorithm with $\tilde{p}(\mathbf{o}|\theta)$,

$$\mathcal{Z}_{\boldsymbol{\Theta}}(\mathbf{O},\mathcal{H}) = \sum_{\boldsymbol{\theta}} P(\boldsymbol{\theta}|\mathcal{H},\mathcal{M}) \prod_{t} \tilde{p}(\mathbf{o}_t|\theta_t) \tag{5.55}$$

This normalisation term ensures that $q_k(\boldsymbol{\theta})$ is always a valid distribution.

3. **VB Maximisation (VBM):**

   Given the variational component sequence posterior $q_k(\boldsymbol{\theta})$, the optimal $q_k(\mathcal{T})$ can be found as

$$q_k(\mathcal{T}) = \frac{1}{\mathcal{Z}_{\mathcal{T}}(\mathbf{O},\mathcal{H})} p(\mathcal{T}) \exp\left(\langle \log p(\mathbf{O},\boldsymbol{\theta}|\mathcal{T},\mathcal{H})\rangle_{q_k(\boldsymbol{\theta})}\right) \tag{5.56}$$

   where $\mathcal{Z}_{\mathcal{T}}(\mathbf{O},\mathcal{H})$ is also a normalisation term to make $q_k(\mathcal{T})$ a valid distribution. With an appropriate form of prior, normally a conjugate prior, $q(\mathcal{T})$ has the same form as the prior. Hence, the optimisation of $q(\mathcal{T})$ requires updating the hyper-parameters of the prior $p(\mathcal{T})$. The exact form is dependent on the transform form and will be discussed in section 5.5.

4. $k = k+1$. **Goto (2)** unless converged.

Having obtained the transform variational distribution $q(\mathcal{T})$, the value of the lower bound in equation (5.40) is required for inference. By calculating $q(\boldsymbol{\theta})$ based on $q(\mathcal{T})$ with equation (5.52) and using the resulting $q(\boldsymbol{\theta})$ in equation (5.51), the lower bound can be re-expressed as

$$\log p(\mathbf{O}|\mathcal{H}) \geq \mathcal{L}_{\mathtt{VB}}\left(q(\mathcal{T})\right) = \log \mathcal{Z}_{\boldsymbol{\Theta}}(\mathbf{O},\mathcal{H}) - \mathtt{KL}\left(q(\mathcal{T})||p(\mathcal{T})\right) \tag{5.57}$$

where the normalisation term $\mathcal{Z}_{\boldsymbol{\Theta}}(\mathbf{O},\mathcal{H})$ is calculated using equation (5.55) and $\mathtt{KL}(\cdot||\cdot)$ is the KL distance defined in equation (5.9). Due to the use of non-point distributions, the negative infinity entropy term is avoided. Hence, this bound is tighter than the bound obtained with point estimates. This should lead to more robust inference for limited test data case.

Similar to the MAP approach in section 5.3.1.2, if the transform prior distribution $p(\mathcal{T})$ is a mixture model, the VB lower bound can not be directly optimised. A variational component weight $q_k(n)$ has to be introduced and the Jensen's inequality re-applied. The logarithm of the marginal likelihood is then approximated by

$$\begin{aligned}
\log p(\mathbf{O}|\mathcal{H}) &= \log \sum_{n} c_n \int_{\mathcal{T}} p(\mathbf{O}|\mathcal{H},\mathcal{T}) p(\mathcal{T}|n)\, d\mathcal{T} \\
&\geq \left\langle \log \frac{c_n}{q_k(n)} \int_{\mathcal{T}} p(\mathbf{O}|\mathcal{H},\mathcal{T}) p(\mathcal{T}|n)\, d\mathcal{T} \right\rangle_{q_k(n)} \\
&\geq \left\langle \mathcal{Q}_{\mathtt{VB}}\left(q_{k+1}(\boldsymbol{\theta}), q_k(\mathcal{T}|n)\right)\right\rangle_{q_k(n)} - \mathtt{KL}\left(q_k(n)||c_n\right)
\end{aligned} \tag{5.58}$$

where $\mathtt{KL}(\cdot||\cdot)$ is the discrete KL distance defined in equation (5.10). Note that, the total variational distributions include *one* variational component sequence distribution, N transform distributions and N transform component weights.[17] VBEM algorithm can then be applied to optimise equation (5.58) given $q_k(n)$. As shown in appendix D.1, in the VBE step, $q_k(\boldsymbol{\theta})$ can still be obtained using equation (5.52) except that the pseudo-distribution $\tilde{p}(\mathbf{o}_t|\theta_t)$ is now based on a mixture variational transform distribution rather than a single component prior,

$$q(\mathcal{T}) = \sum_n q(n)q(\mathcal{T}|n) \tag{5.59}$$

In the VBM step, the update formulae for each transform distribution component $q(\mathcal{T}|n)$ is similar to equation (5.56) except that the n^{th} prior component $p(\mathcal{T}|n)$ is used instead of $p(\mathcal{T})$.

An additional issue introduced by using mixture priors is the estimation of variational component weights $q(n)$. To simplify estimating $q(n)$, the components of the transform prior are assumed to be independent of each other. With this assumption, the hidden component sequence $\boldsymbol{\theta}$ may change from one prior component to another. Then a distinct $q(\boldsymbol{\theta}|n)$ is introduced for each prior component. The number of variational component sequence distributions changes from 1 to N, where N is the number of prior components. It is shown in appendix D.1 that the component weight can be updated by

$$q_k(n) = \frac{c_n \exp\left(\mathcal{L}_{\mathtt{VB}}\left(q_k(\mathcal{T}|n)\right)\right)}{\sum_n c_n \exp\left(\mathcal{L}_{\mathtt{VB}}\left(q_k(\mathcal{T}|n)\right)\right)} \tag{5.60}$$

where $\mathcal{L}_{\mathtt{VB}}\left(q_k(\mathcal{T}|n)\right)$ is calculated using equation (5.57). With a mixture prior, the *overall* lower bound, equation (5.58), can be re-expressed as an extended form of equation (5.57)

$$\mathcal{L}_{\mathtt{VB}}\left(q(\mathcal{T})\right) = \log \mathcal{Z}_{\boldsymbol{\Theta}}(\mathbf{O}, \mathcal{H}) - \sum_n q(n)\mathtt{KL}\left(q(\mathcal{T}|n)||p(\mathcal{T}|n)\right) - \mathtt{KL}\left(q(n)||c_n\right) \tag{5.61}$$

where $\mathcal{Z}_{\boldsymbol{\Theta}}(\mathbf{O}, \mathcal{H})$ is the normalisation term calculated based on the *complete* variational transform distribution in equation (5.59).

The above derivations assume that a global transform is used for all Gaussian components. It can be extended to a multiple base class case, where a separate (independent) transform is associated with a group of Gaussians. In that case, let $\mathcal{T} = \{\mathcal{T}_1, \cdots, \mathcal{T}_{N_\mathcal{R}}\}$, where $\mathcal{T}_r$ is the transform associated with base class r, $\mathcal{R}$ is the set of regression base classes, $N_\mathcal{R}$ is the total number of base classes. Then the prior can be regarded as a product of the individual transforms distributions

$$p(\mathcal{T}) = \prod_{r \in \mathcal{R}} p(\mathcal{T}_r) \tag{5.62}$$

Variational transform distributions $q(\mathcal{T}_r)$ are used for each regression base class r. The resulting VBEM algorithm is similar to the global case except that the sufficient statistics for each variational transform distribution are accumulated on the corresponding group of Gaussians. In the

[17] N is the total number of components of $p(\mathcal{T})$.

mixture prior case, as shown in appendix D.2, the variational component weight can be updated by

$$q_k(n_r) = \frac{c_{n_r} \exp\left(\mathcal{L}_{\text{VB}}\left(q_k(\mathcal{T}|n_r)\right)\right)}{\sum_{n_r} c_{n_r} \exp\left(\mathcal{L}_{\text{VB}}\left(q_k(\mathcal{T}|n_r)\right)\right)} \tag{5.63}$$

where k is the iteration index, n_r is the n^{th} prior component associated with base class r, c_{n_r} denotes the prior component weight, $q_k(n_r)$ is the corresponding variational weight. Equation (5.63) is a multiple regression base class version of equation (5.60). The lower bound $\mathcal{L}_{\text{VB}}\left(q_k(\mathcal{T}|n_r)\right)$ is calculated based on $q_k(\mathcal{T}|n_r)$, which is defined as

$$q_k(\mathcal{T}|n_r) = q_k(\mathcal{T}_r|n_r) \prod_{i \in \mathcal{R}_{-r}} q_k(\mathcal{T}_i) \tag{5.64}$$

where $\mathcal{R}_{-r}$ denotes the regression base class set without r, $q_k(\mathcal{T}_i)$ is the *complete* mixture variational transform distribution for the i^{th} regression base class at iteration k.

5.3.2 Direct Approximations

There are a number of other approaches, which do not require an iterative process to tighten the lower bound, that can be used to approximate the intractable likelihood integral. This kind of approximation is referred to as *direct marginalisation approximations*. One advantage of these approaches is that no iterative process is required during Bayesian adaptive inference. However, it is hard to know how close the direct approximation is to the real likelihood. It may be greater or less than the real likelihood.

5.3.2.1 Sampling Approach

Sampling approaches are a standard method for directly approximating intractable probabilistic integrals. The basic idea is to draw samples from the distribution and use the average integral function value to approximate the real probabilistic expectation [99]. Thus

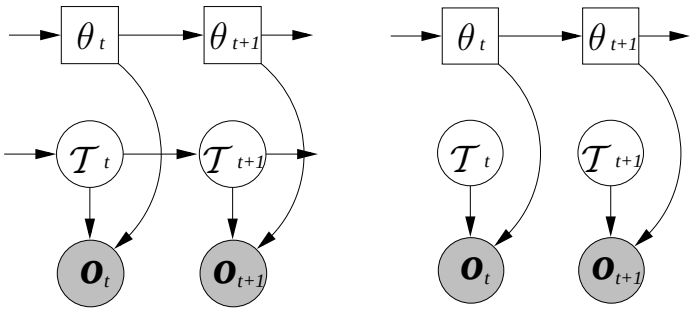
$$p(\mathbf{O}|\mathcal{H}) \approx \frac{1}{N} \sum_{n=1}^{N} p(\mathbf{O}|\mathcal{H}, \hat{\mathcal{T}}_n) \tag{5.65}$$

where N is the total number of samples and $\hat{\mathcal{T}}_n$ is the n^{th} sample drawn from $p(\mathcal{T})$. In the limit as $N \rightarrow \infty$ this will tend to the true integral [99].

There is a fundamental issue associated with this form of approximation. As the number of transform parameters increases the number of samples required to obtain good estimates dramatically increases. As a separate inference evidence calculation is required for each sample to find the final best hypothesis, it is hard to efficiently control the computational cost for large systems. This approach is only applicable to systems with small number of transform parameters such as CAT.

5.3.2.2 Frame-Independent Assumption

Rather than approximating the integral, an alternative approach is used to modify the dynamic Bayesian network (DBN) associated with the Bayesian adaptive inference process. The approach is to allow the transform change at each time instance. Figure 5.2 shows the comparison of the DBN for strict adaptive inference and the DBN for the approximated adaptive inference. Here, θ_t is the Gaussian component at time t.



(a) Strict Inference **(b) FI Assumption**

Figure 5.2 *Dynamic Bayesian network comparison between strict inference and frame-independent assumption*

As shown in figure 5.2(a), in strict Bayesian inference, both component and transform parameters transition process are first-order Markov processes. With the homogeneity constraint on data blocks, the transform transition probability can be expressed as a Dirac delta function as in equation (5.2). This means that the transform parameters are constrained to be constant over all frames within one homogeneous block. Mathematically this yields the integral in equation (5.39).

If the constraint that the transform is the same for all observations is relaxed, the DBN in figure 5.2(b) will be obtained. This allows the transform to vary from time instance to time instance and will be referred to as the *frame-independent* (FI) assumption. As discussed in section 2.6.3, this assumption has been implicitly used in the Bayesian prediction approaches for HMM parameters [62, 120], where the resultant distribution is called *Bayesian predictive distribution*. It has been also investigated for inference with standard two-step adaptation/recognition process [37, 14, 107]. Using this approximation in equation (5.6) yields

$$p(\mathbf{O}|\mathcal{H}) \approx \sum_{\boldsymbol{\theta}} P(\boldsymbol{\theta}|\mathcal{H},\mathcal{M}) \prod_{t} \bar{p}(\mathbf{o}_t|\mathcal{M},\theta_t) \tag{5.66}$$

where

$$\bar{p}(\mathbf{o}_t|\mathcal{M},\theta_t) = \int_{\mathcal{T}} p(\mathbf{o}_t|\mathcal{M},\mathcal{T},\theta_t) p(\mathcal{T})\, d\mathcal{T} \tag{5.67}$$

is the predictive distribution at component θ_t. With the form of $p(\mathcal{T})$ being a conjugate prior to the likelihood of observation, this frame-level integral is tractable. For example, in MLLR where mean based linear transforms are used, a single Gaussian distribution or GMM may be used as

$p(\mathcal{T})$ to obtain a tractable predictive distribution [37, 14]. More details of application of FI will be discussed in section 5.5.

This FI assumption breaks the DBN of adaptive HMM and is more similar to the multi-style training approach, such as the speaker independent (SI) model, where the acoustic condition can usually change from frame to frame (the standard HMM assumption) [38]. Therefore, unless the posterior distributions of each homogeneous block are used, the results with FI approximation will be similar to the SI performance. One advantage of this FI assumption is that the observation is conditionally independent given the hidden state and transform distribution. This makes it possible to use the standard Viterbi decoding with the predictive distribution in equation (5.67). As Viterbi decoding is not generally applicable within the Bayesian adaptive inference framework, the use of FI in Viterbi decoding will not be further discussed.

5.4 Incremental Bayesian Adaptive Inference

The Bayesian adaptive inference discussed in section 5.3 is described in a *batch* adaptation mode, where all the test data are assumed to be available for decoding in a single block. However, in some applications, test data only becomes available gradually. For these applications, *incremental* adaptive inference is often used. Here information from the previous utterances is propagated in adaptation to decode the current utterance. This section will discuss incremental unsupervised adaptation within a Bayesian framework [141]. Here, only lower bound approximations are concerned[18]. Variational Bayes is discussed first. Then the point estimate version is then described.

For incremental adaptive inference, a homogeneous data block comprises multiple utterances. $\mathbf{O} = \mathbf{O}_{1:U} = \{\mathbf{O}_1, \cdots, \mathbf{O}_U\}$ denotes the observation sequence from 1^{st} to the U^{th} utterance. Similarly, the hypothesis for the U utterances, $\mathcal{H}$, consists of a set of hypotheses for utterances within it, $\mathcal{H}_{1:U} = \{\mathcal{H}_1, \ldots, \mathcal{H}_U\}$. Adaptation information is propagated to the U^{th} utterance from the previous $U-1$ utterances. The key question here is what information to propagate between utterances and how to use the propagated information. Various levels of information propagation are discussed below using the VB approximation as the example for lower bound based inference:

1. **No information**:

   No information is propagated between utterances. When the U^{th} utterance comes, the lower bound for all utterances needs to be optimised. This is similar to repeated application of batch mode inference with increased number of utterances. This involves rescoring all U utterances, yielding a new hypothesis sequence for all utterances, $\hat{\mathcal{H}}_{1:U}$. The U^{th} utterance may change the "best" hypothesis for the preceding utterances. This approach

[18] If direct approximations, for example, the FI assumption, are used, the incremental adaptive inference means incrementally update of the transform prior distribution. As the prior update or posterior estimation is not concerned in this work, direct approximations will not be discussed in this section.

breaks the causal aspects of incremental adaptive inference and is highly computationally expensive.

2. **Inferred hypothesis sequence**:

   If the causal constraint is enforced, then the best hypothesis sequence for the previous $U-1$ utterances is fixed as $\hat{\mathcal{H}}_{1:U-1}$. The optimisation of the lower bound is then only based on each hypothesis for the U^{th} utterance respectively. The variational posteriors in equation (5.50) can then be written as

$$q(\boldsymbol{\theta}|\mathbf{O},\mathcal{H}) = q(\boldsymbol{\theta}|\mathbf{O},\hat{\mathcal{H}}_{1:U-1},\mathcal{H}_U) \tag{5.68}$$

$$q(\mathcal{T}|\mathbf{O},\mathcal{H}) = q(\mathcal{T}|\mathbf{O},\hat{\mathcal{H}}_{1:U-1},\mathcal{H}_U) \tag{5.69}$$

   In this configuration, there is a choice of the initial transform distribution to use. The transform prior, $p(\mathcal{T})$, can be used to initialise the VBEM process. Alternatively, the distribution from the previous $U-1$ utterances may be used. Thus

$$q_0(\mathcal{T}|\mathbf{O},\hat{\mathcal{H}}_{1:U-1},\mathcal{H}_U) = q_K(\mathcal{T}|\mathbf{O}_{1:U-1},\hat{\mathcal{H}}_{1:U-1}) \tag{5.70}$$

   where K is the number of VBEM iterations used[19]. The VBEM algorithm remains unchanged except that $\mathbf{O}_{1:U-1}$ only needs to be re-aligned against $\hat{\mathcal{H}}_{1:U-1}$, rather than all possibilities. Hence, the inference only involves possible hypotheses for the U^{th} utterance.

3. **Posterior component sequence distribution and inferred hypotheses**:

   Propagating the hypotheses still requires the posterior component sequence distribution for all U utterances to be computed. This posterior may also be fixed and propagated to the next utterance. Thus equation (5.68) becomes

$$q(\boldsymbol{\theta}|\mathbf{O},\mathcal{H}) = q(\boldsymbol{\theta}_U|\mathbf{O}_U,\mathcal{H}_U)\prod_{u=1}^{U-1} q_K(\boldsymbol{\theta}_u|\mathbf{O}_u,\hat{\mathcal{H}}_u) \tag{5.71}$$

   The previous $U-1$ utterances do not need to be re-aligned. Only $q(\boldsymbol{\theta}_U|\mathbf{O}_U,\mathcal{H}_U)$ needs to be computed, i.e., only the sufficient statistics of the U^{th} utterance need to be accumulated. This is the most efficient form and will be used in this work. It is interesting to compare the information strategy described above to the standard incremental adaptation [73]. In standard incremental adaptation, an ML transform is re-estimated for each utterance using the accumulated statistics from the previous and the current utterances. In this estimation, the state/component alignment of the previous utterances are often assumed to be unchanged. The updated transform is then used to decode the next utterance. As ML estimate is a form of lower bound approximation, standard incremental adaptation can be viewed as a special case of the incremental Bayesian adaptive inference.

Using the information propagation strategy 3, an efficient, modified version of the VBEM algorithm described can be derived as shown in appendix E

[19] Note that the prior distribution, $p(\mathcal{T})$, is still used in the update of variational transform distribution.

1. **Initialisation**:

   Set $k = 1$. The initial transform distribution is given by equation (5.70). For the first utterance, set $q_0(\mathcal{T}) = p(\mathcal{T})$.

2. **VBE step**:

   Only the U^{th} component sequence posterior needs to be computed

$$q_k(\boldsymbol{\theta}_U|\mathbf{O}_U, \mathcal{H}_U) = \frac{1}{\mathcal{Z}_{\boldsymbol{\Theta}}(\mathbf{O}_U, \mathcal{H}_U)} \exp\left(\langle \log p(\mathbf{O}_U, \boldsymbol{\theta}_U|\mathcal{T}, \mathcal{H}_U)\rangle_{q_{k-1}(\mathcal{T}|\mathbf{O}, \hat{\mathcal{H}}_{1:U-1}, \mathcal{H}_U)}\right) \quad (5.72)$$

   where k denotes the iteration index, $\mathcal{Z}_{\boldsymbol{\Theta}}(\mathbf{O}_U, \mathcal{H}_U)$ can be calculated using equation (5.55) based on the pseudo-distribution with $q_{k-1}(\mathcal{T}|\mathbf{O}, \hat{\mathcal{H}}_{1:U-1}, \mathcal{H}_U)$.

3. **VBM step**:

   The single component transform distribution can be updated as below

$$\begin{aligned}\log q_k(\mathcal{T}|\mathbf{O}, \hat{\mathcal{H}}_{1:U-1}, \mathcal{H}_U) = &\sum_{u=1}^{U-1} \left\langle \log p(\mathbf{O}_u, \boldsymbol{\theta}_u|\mathcal{T}, \hat{\mathcal{H}}_u)\right\rangle_{q_K(\boldsymbol{\theta}_u|\mathbf{O}_u, \hat{\mathcal{H}}_u)} \\ &+ \langle \log p(\mathbf{O}_U, \boldsymbol{\theta}_U|\mathcal{T}, \mathcal{H}_U)\rangle_{q_k(\boldsymbol{\theta}_U|\mathbf{O}_U, \mathcal{H}_U)} + \log p(\mathcal{T}) - \mathcal{Z}_{\mathcal{T}}(\mathbf{O}, \hat{\mathcal{H}}_{1:U-1}, \mathcal{H}_U) \quad (5.73)\end{aligned}$$

   From equation (5.73), the total sufficient statistics is a summation of those of the current utterance and the previous $U-1$ utterances, which are propagated and do not need to be re-calculated. This recursive formulae significantly reduces the computation cost.

   If a multiple component prior is used, the transform distribution update formulae in equation (5.73) can be used to update the hyper-parameters of individual component of the variational transform distribution. Theoretically, the variational component weights can still be updated using equation (5.60). However, the use of multiple component prior significantly complicates the incremental adaptation, hence is not considered in this work.

4. $k = k + 1$. Goto 2 until $k = K$.

Having obtained the optimal transform distribution, the value of the VB lower bound in equation (5.57) is required for inference. In this calculation, the normalisation term can also be calculated recursively

$$\mathcal{Z}_{\boldsymbol{\Theta}}(\mathbf{O}, \hat{\mathcal{H}}_{1:U-1}, \mathcal{H}_U) = \mathcal{Z}_{\boldsymbol{\Theta}}(\mathbf{O}_U, \mathcal{H}_U) \prod_{u=1}^{U-1} \mathcal{Z}_{\boldsymbol{\Theta}}(\mathbf{O}_u, \hat{\mathcal{H}}_u) \quad (5.74)$$

Note, a normalisation term is calculated for *each* possible hypothesis $\mathcal{H}_U$. The inferred hypothesis for the U^{th} utterance, $\hat{\mathcal{H}}_U$, is obtained by using the inference criterion in equation (5.21). Then the $(U+1)^{th}$ utterance is processed and this is repeated until all the utterances are processed.

With the point estimate approximations, a similar incremental EM algorithm and inference process can be derived. The main difference is that the transform estimate, rather than the transform distribution, is propagated. The EM algorithm for incremental MAP adaptation is described as below:

1. **Initialisation**: set $k = 1$ and

$$\hat{\mathcal{T}}_0(\mathbf{O}, \hat{\mathcal{H}}_{1:U-1}, \mathcal{H}_U) = \hat{\mathcal{T}}_K(\mathbf{O}_{1:U-1}, \hat{\mathcal{H}}_{1:U-1}) \tag{5.75}$$

where $\hat{\mathcal{T}}_K(\mathbf{O}_{1:U}, \mathcal{H}_{1:U})$ denotes the transform estimated based the utterances $\mathbf{O}_{1:U}$ and the hypothesis sequence $\mathcal{H}_{1:U}$ at iteration K. For the first utterance, the expectation of the prior may be used, $\hat{\mathcal{T}}_0(\mathbf{O}_1, \mathcal{H}_1) = E[p(\mathcal{T})]$. If an uninformative prior is used, i.e. the ML estimate is to be obtained, an identity matrix may be used for initialisation.

2. **Expectation step**:

As the previous $U-1$ component sequence posteriors, $P_K(\boldsymbol{\theta}_u|\mathbf{O}_u, \hat{\mathcal{H}}_u), u \in \{1, \cdots, U-1\}$, are propagated, only the U^{th} component posterior distribution, $P_k(\boldsymbol{\theta}_U|\mathbf{O}_U, \mathcal{H}_U)$ needs to be figured out. This calculation is based on the transform estimate of the previous iteration, $\hat{\mathcal{T}}_{k-1}(\mathbf{O}, \hat{\mathcal{H}}_{1:U-1}, \mathcal{H}_U)$. The sufficient statistics to update the transform can then be accumulated.

3. **Maximisation step**:

With similar derivation to VB, the transform distribution at the U^{th} utterance can be updated using the following auxiliary function

$$\begin{aligned}\mathcal{Q}_{\texttt{MAP}}(\mathcal{T}_{k+1}; \hat{\mathcal{T}}_k) \quad = \quad & \sum_{u=1}^{U-1} \sum_{t,\theta_u} \hat{\gamma}_{\theta_u}(t) \log p(\mathbf{o}_u(t)|\theta_u, \mathcal{T}_{k+1}, \hat{\mathcal{H}}_u) \\ & + \sum_{t,\theta_U} \gamma_{\theta_U}(t) \log p(\mathbf{o}_U(t)|\theta_U, \mathcal{T}_{k+1}, \mathcal{H}_U) + \log p(\mathcal{T}_{k+1})\end{aligned} \tag{5.76}$$

where $\mathcal{T}_k$ is the brief notation for $\mathcal{T}_k(\mathbf{O}, \hat{\mathcal{H}}_{1:U-1}, \mathcal{H}_U)$, $\hat{\gamma}_{\theta_u}, u \in \{1, \cdots, U-1\}$ is the posterior occupancy calculated from the propagated component sequence distribution $P_K(\boldsymbol{\theta}_u|\mathbf{O}_u, \hat{\mathcal{H}}_u)$, γ_{θ_U} is the posterior occupancy calculated on the U^{th} utterance. The optimal transform of the current iteration is found by

$$\hat{\mathcal{T}}_{k+1}(\mathbf{O}, \hat{\mathcal{H}}_{1:U-1}, \mathcal{H}_U) = \arg\max_{\mathcal{T}_{k+1}} \mathcal{Q}_{\texttt{MAP}}(\mathcal{T}_{k+1}; \hat{\mathcal{T}}_k) \tag{5.77}$$

4. $k = k + 1$. Goto 2 step until $k = K$.

Having obtained the optimal transform estimate $\hat{\mathcal{T}}_K(\mathbf{O}, \hat{\mathcal{H}}_{1:U-1}, \mathcal{H}_U)$, the MAP lower bound in equation (5.45) can be calculated. Then, the best hypothesis $\hat{\mathcal{H}}_U$ can be inferred and propagated to the next utterance.

5.5 Applications to Model Based Transformations

The previous sections introduces the general form of various Bayesian inference approximation schemes. In this section, these schemes are applied to specific types of model based transforms [138]. As described in chapter 3, there are three main types of widely used model

based transforms: mean based Maximum likelihood linear regression (MLLR) [74], interpolation weight vectors in cluster adaptive training (CAT) [36] or eigenvoices [69] and constrained MLLR (CMLLR) [34]. These are discussed in the following sections.

5.5.1 Interpolation Weights in Cluster Adaptive Training

Cluster adaptive training (CAT) has been discussed in detail in section 3.4.2.2 and section 4.3.2. This section gives the exact forms of Bayesian approximations applied to CAT [138] (see appendix F for derivations). In CAT, the adapted mean vector is an interpolation of mean clusters, which is re-rewritten as below[20]

$$\hat{\boldsymbol{\mu}}^{(m)} = \sum_{p=1}^{P} \lambda_p \boldsymbol{\mu}_p^{(m)} = \mathbf{M}^{(m)} \boldsymbol{\lambda} \tag{5.78}$$

where $\hat{\boldsymbol{\mu}}^{(m)}$ is the adapted mean vector of Gaussian component m, $\mathbf{M}^{(m)} = [\boldsymbol{\mu}_1^{(m)}, \cdots, \boldsymbol{\mu}_P^{(m)}]$ is the cluster mean vectors for component m, P is the number of clusters, and $\boldsymbol{\lambda}$ is a $P \times 1$ interpolation weight vector. For the interpolation weight vector, a Gaussian distribution may be used as the conjugate prior. Hence, GMM may be used as an enhanced form of prior. As the number of parameters is small, typically only 2 or 3, it is possible to use the sampling approach discussed in section 5.3.2.1

For the frame independent assumption it is necessary to obtain the predictive distribution in equation (5.67). For both the interpolation weights here and the linear transform in the next section, if the original distribution is Gaussian and the transform prior distribution is GMMs, then the resultant predictive distribution will also be a GMM. For a particular Gaussian component m, the predictive distribution is

$$\bar{p}(\mathbf{o}|m) = \int_{\mathcal{T}} p(\mathbf{o}|\mathcal{T}, m) p(\mathcal{T})\, d\mathcal{T} = \sum_{n=1}^{N} c_n \mathcal{N}(\mathbf{o}; \bar{\boldsymbol{\mu}}^{(mn)}, \bar{\boldsymbol{\Sigma}}^{(mn)}) \tag{5.79}$$

where the prior distribution $p(\mathcal{T})$ is a GMM given in equation (5.7). In the CAT case, the GMM is expressed as

$$p(\mathcal{T}) = p(\boldsymbol{\lambda}) = \sum_{n=1}^{N} c_n \mathcal{N}(\boldsymbol{\lambda}; \boldsymbol{\mu}_\lambda^{(n)}, \boldsymbol{\Sigma}_\lambda^{(n)}) \tag{5.80}$$

where $\boldsymbol{\mu}_\lambda^{(n)}$ and $\boldsymbol{\Sigma}_\lambda^{(n)}$ are hyper-parameters of the n^{th} prior component and c_n is the weight of the prior component n. Hence, given the predictive distribution for each Gaussian component, the resultant state output distribution has $M \times N$ components, where M and N are component numbers for the original distribution and the prior respectively. For CAT, the parameters of this

[20] Note that only the global interpolation weight vector is discussed in this section. The formulae for multiple base classes case are similar except for accumulating statistics over the particular base class when estimating the interpolation weight vectors. The index of acoustic condition s is dropped for clarity.

distribution can be shown as (see appendix F)

$$\bar{\boldsymbol{\mu}}^{(mn)} = \mathbf{M}^{(m)}\boldsymbol{\mu}_{\lambda}^{(n)} \tag{5.81}$$

$$\bar{\boldsymbol{\Sigma}}^{(mn)} = \mathbf{M}^{(m)}\boldsymbol{\Sigma}_{\lambda}^{(n)}\mathbf{M}^{(m)T} + \boldsymbol{\Sigma}^{(m)} \tag{5.82}$$

It is interesting to note that even if the prior and original Gaussian distribution both have diagonal covariance matrices, the resultant predictive distribution has a full covariance matrix.

The ML estimate for CAT interpolation weights have been well investigated in [36]. Hence, for lower bound based approximations, only MAP estimate and VB approximation are discussed here. Though the MAP estimate for the single component prior case has already been derived for CAT [33], the multiple component case has not been considered. Optimising the auxiliary function in equation (5.49), the MAP estimate of $\boldsymbol{\lambda}$ can be shown as

$$\hat{\boldsymbol{\lambda}} = \left(\sum_{n=1}^{N} q(n)\boldsymbol{\Sigma}_{\lambda}^{(n)-1} + \mathbf{G}_{\mathtt{ML}}\right)^{-1}\left(\sum_{n=1}^{N} q(n)\boldsymbol{\Sigma}_{\lambda}^{(n)-1}\boldsymbol{\mu}_{\lambda}^{(n)} + \mathbf{k}_{\mathtt{ML}}\right) \tag{5.83}$$

where $q(n)$ is obtained using equation (5.48) given the current weights estimate, $\mathbf{G}_{\mathtt{ML}}$ and $\mathbf{k}_{\mathtt{ML}}$ are the standard ML CAT sufficient statistics in equation (3.54) and equation (3.55)[21], which is obtained given the canonical model and the current weights estimate. Having obtained the MAP estimate of the interpolation weight vector for each hypothesis, the value of $\mathcal{K}_{\mathtt{MAP}}(\hat{\mathcal{T}})$ in equation (5.45) can be calculated for inference.

When using VB as the approximation, it is necessary to find the pseudo-distribution, $\tilde{p}(\mathbf{o}|m)$, given in equation (5.54). As shown in appendix F, for CAT, the pseudo-distribution for component m is

$$\log\tilde{p}(\mathbf{o}|m) = \sum_{n=1}^{N} q(n)\Big(\log\mathcal{N}(\mathbf{o};\mathbf{M}^{(m)}\tilde{\boldsymbol{\mu}}_{\lambda}^{(n)},\boldsymbol{\Sigma}^{(m)}) - \frac{1}{2}\mathrm{tr}(\tilde{\boldsymbol{\Sigma}}_{\lambda}^{(n)}\mathbf{M}^{(m)T}\boldsymbol{\Sigma}^{(m)-1}\mathbf{M}^{(m)})\Big) \tag{5.84}$$

where $q(n)$ is the variational posterior weight for component n calculated using equation (5.60). For single Gaussian prior, $q(n)$ is always 1. $\tilde{\boldsymbol{\mu}}_{\lambda}^{(n)}$ and $\tilde{\boldsymbol{\Sigma}}_{\lambda}^{(n)}$ are the hyper-parameters of the below variational transform distribution, which is also a GMM

$$q(\boldsymbol{\lambda}) = \sum_{n=1}^{N} q(n)\mathcal{N}(\boldsymbol{\lambda};\tilde{\boldsymbol{\mu}}_{\lambda}^{(n)},\tilde{\boldsymbol{\Sigma}}_{\lambda}^{(n)}) \tag{5.85}$$

The hyper-parameters of the variational distribution $q(\boldsymbol{\lambda})$ can be estimated using VBEM as

$$\tilde{\boldsymbol{\Sigma}}_{\lambda}^{(n)} = \left(\boldsymbol{\Sigma}_{\lambda}^{(n)-1} + \mathbf{G}_{\mathtt{ML}}\right)^{-1} \tag{5.86}$$

$$\tilde{\boldsymbol{\mu}}_{\lambda}^{(n)} = \tilde{\boldsymbol{\Sigma}}_{\lambda}^{(n)}\left(\boldsymbol{\Sigma}_{\lambda}^{(n)-1}\boldsymbol{\mu}_{\lambda}^{(n)} + \mathbf{k}_{\mathtt{ML}}\right) \tag{5.87}$$

where $\boldsymbol{\mu}_{\lambda}^{(n)}$ and $\boldsymbol{\Sigma}_{\lambda}^{(n)}$ are the hyper-parameters of the transform prior distribution, $\mathbf{G}_{\mathtt{ML}}$ and $\mathbf{k}_{\mathtt{ML}}$ are again the ML sufficient statistics given in equation (3.54) and equation (3.55), except that $\gamma_m^{\mathtt{ML}}(t)$ is now calculated based on the pseudo-distribution given in equation (5.84). After K iterations, the final variational distribution $q_K(\boldsymbol{\lambda})$ is used to calculate the lower bound in equation (5.61) for inference.

[21] Note that $\boldsymbol{\lambda}$ here is a global weight.

5.5.2 Mean MLLR in Speaker Adaptive Training

Mean based linear transforms were originally introduced within a ML framework, referred to as maximum likelihood linear regression (MLLR) [74]. It has been reviewed in detail in section 3.2.2.1. The adapted mean vector can be written as[22]

$$\hat{\boldsymbol{\mu}}^{(m)} = \mathbf{A}\boldsymbol{\mu}^{(m)} + \mathbf{b} = \mathbf{W}\boldsymbol{\xi}^{(m)} \tag{5.88}$$

where $\boldsymbol{\xi}^{(m)} = [\boldsymbol{\mu}^{(m)T} \quad 1]^T$ is the extended mean vector, $\mathbf{W} = [\mathbf{A} \quad \mathbf{b}]$ is the extended linear transform. Similar to the interpolation weight vector, which is also a linear "transform" on mean vectors, a Gaussian distribution may be used as the conjugate prior [37, 14]. As discussed in section 5.1.1, a GMM is also used as an extension. In this case, to be consistent with the diagonal covariance matrix used for HMM systems [37], each row of the transform is assumed to be independent given the prior component. Thus

$$p(\mathcal{T}) = p(\mathbf{W}) = \sum_{n=1}^{N} c_n \prod_{d=1}^{D} \mathcal{N}(\mathbf{w}_d; \boldsymbol{\mu}_{\mathbf{w}_d}^{(n)}, \boldsymbol{\Sigma}_{\mathbf{w}_d}^{(n)}) \tag{5.89}$$

where D is the size of the feature vector, $\mathbf{w}_d^T$ is the d^{th} row of $\mathbf{W}$.

As indicated in section 5.3.2.1, MLLR transforms have too many parameters to robustly use the sampling approximation. Hence, only frame-independent assumption is used as a direct approximation. The resultant predictive distribution for Gaussian component m has a similar form as equation (5.79), where the parameters have been derived in [37] and [14] for the case of single Gaussian prior. These parameters are reproduced here at element level in a consistent form as equation (5.81) and equation (5.82)

$$\begin{aligned} \bar{\mu}_d^{(mn)} &= \mathbf{w}_d^{(n)T}\boldsymbol{\xi}^{(m)} \\ \bar{\sigma}_{dd}^{(mn)} &= \sigma_{dd}^{(m)} + \boldsymbol{\xi}^{(m)T}\boldsymbol{\Sigma}_{\mathbf{w}_d}^{(n)}\boldsymbol{\xi}^{(m)} \end{aligned}$$

where $\bar{\mu}_d^{(mn)}$ and $\bar{\sigma}_{dd}^{(mn)}$ are parameters of the predictive distribution, $d \in \{1, \cdots, D\}$ is the element index, $\boldsymbol{\Sigma}^{(m)}$ is assumed to be a diagonal covariance matrix, of which $\sigma_{dd}^{(m)}$ is the d^{th} diagonal element. In contrast to CAT, due to the row-independent assumption in prior, the resultant covariance matrix predictive distribution is also diagonal.

MAP Linear Regression (MAPLR) with single Gaussian prior was presented in [13]. The multiple component prior MAP estimate is a straightforward extension, yielding forms similar to that for CAT in equation (5.83). Given the ML sufficient statistics $\mathbf{G}_{\mathtt{ML},d}$ and $\mathbf{k}_{\mathtt{ML},d}$ in equation (3.8) and equation (3.9), the d^{th} row of MAPLR transform $\mathbf{W}$ is updated by

$$\hat{\mathbf{w}}_d = \left(\sum_n q(n)\boldsymbol{\Sigma}_{\mathbf{w}_d}^{(n)-1} + \mathbf{G}_{\mathtt{ML},d}\right)^{-1} \left(\sum_n q(n)\boldsymbol{\Sigma}_{\mathbf{w}_d}^{(n)-1}\boldsymbol{\mu}_{\mathbf{w}_d}^{(n)} + \mathbf{k}_{\mathtt{ML},d}\right) \tag{5.90}$$

where $q(n)$ is again calculated using equation (5.48) given the current transform estimate.

[22] As in the discussion of interpolation weight vector, only global MLLR transform is discussed in this section.

For the VB approximation, the pseudo-distribution is first required. Again it can be shown that this is an unnormalised distribution, where component m has the form

$$\log \tilde{p}(\mathbf{o}|m) = \sum_{n=1}^{N} q(n) \Big(\log \mathcal{N}(\mathbf{o}; \tilde{\mathbf{W}}_{\boldsymbol{\mu}}^{(n)} \boldsymbol{\xi}^{(m)}, \boldsymbol{\Sigma}^{(m)}) - \frac{1}{2} \sum_{d=1}^{D} \frac{\boldsymbol{\xi}^{(m)T} \tilde{\boldsymbol{\Sigma}}_{\mathbf{w}_d}^{(n)} \boldsymbol{\xi}^{(m)}}{\sigma_{dd}^{(m)}} \Big) \tag{5.91}$$

where $\tilde{\mathbf{W}}_{\boldsymbol{\mu}}^{(n)} = [\tilde{\boldsymbol{\mu}}_{\mathbf{w}_1}^{(n)}, \cdots, \tilde{\boldsymbol{\mu}}_{\mathbf{w}_D}^{(n)}]^T$ is the mean transform of the n^{th} component of the variational posterior transform distribution. Given the pseudo-distribution, the hyper-parameters of the complete variational transform posterior, $q(\mathbf{W})$, can be updated. This is similar to equation (5.85), but modified to reflect the independence assumption between rows of the transform shown in equation (5.89). The n^{th} component's mean and covariance for row d can be estimated by

$$\begin{aligned} \tilde{\boldsymbol{\Sigma}}_{\mathbf{w}_d}^{(n)} &= \left(\boldsymbol{\Sigma}_{\mathbf{w}_d}^{(n)-1} + \mathbf{G}_{\mathtt{ML},d} \right)^{-1} \\ \tilde{\boldsymbol{\mu}}_{\mathbf{w}_d}^{(n)} &= \tilde{\boldsymbol{\Sigma}}_{\mathbf{w}_d}^{(n)} \left(\boldsymbol{\Sigma}_{\mathbf{w}_d}^{(n)-1} \boldsymbol{\mu}_{\mathbf{w}_d}^{(n)} + \mathbf{k}_{\mathtt{ML},d} \right) \end{aligned} \tag{5.92}$$

where $\boldsymbol{\mu}_{\mathbf{w}_d}^{(n)}$ and $\boldsymbol{\Sigma}_{\mathbf{w}_d}^{(n)}$ are the parameters of the n^{th} prior component, $\mathbf{G}_{\mathtt{ML},d}$ and $\mathbf{k}_{\mathtt{ML},d}$ are the ML sufficient statistics in equation (3.8) and equation (3.9) except that the component posterior, $\gamma_m^{\mathtt{ML}}(t)$, is based on the pseudo-distribution given in equation (5.91). The variational component weights $q(n)$ can also be updated using equation (5.60). After K iterations, the final distribution $q_K(\mathbf{W})$ is used to calculate the VB lower bound in equation (5.61) for inference.

5.5.3 Constrained MLLR in Speaker Adaptive Training

The mean based linear transform only adapts the mean vectors. An alternative is to use constrained linear transforms [34], which has been discussed in detail in section 3.2.2.3. Here, the same transform is applied to both mean vector and covariance matrix.

$$\hat{\boldsymbol{\mu}}^{(m)} = \mathbf{A}' \boldsymbol{\mu}^{(m)} - \mathbf{b}' \tag{5.93}$$

$$\hat{\boldsymbol{\Sigma}}^{(m)} = \mathbf{A}' \boldsymbol{\Sigma}^{(m)} \mathbf{A}'^{T} \tag{5.94}$$

where $\mathbf{A}'$ is the constrained linear transform, $\mathbf{b}'$ is the bias on the mean vector, and $\boldsymbol{\mu}^{(m)}$ and $\boldsymbol{\Sigma}^{(m)}$ are the original Gaussian parameters. This constrained transform can be effectively implemented as a feature space transform. The transformed Gaussian distribution is expressed as

$$\mathcal{N}(\mathbf{o}; \hat{\boldsymbol{\mu}}^{(m)}, \hat{\boldsymbol{\Sigma}}^{(m)}) = |\mathbf{A}| \mathcal{N}(\mathbf{A}\mathbf{o} + \mathbf{b}; \boldsymbol{\mu}^{(m)}, \boldsymbol{\Sigma}^{(m)}) \tag{5.95}$$

where $\boldsymbol{\mu}^{(m)}$ and $\boldsymbol{\Sigma}^{(m)}$ are original Gaussian parameters, $\mathbf{A} = \mathbf{A}'^{-1}$ and $\mathbf{b} = \mathbf{A}'^{-1}\mathbf{b}'$.

The above equation shows that the computational cost of likelihood calculation almost remains unchanged after adaptation. However, due to the determinant term $|\mathbf{A}|$ in equation (5.95), it is hard to find a conjugate prior for constrained linear transform. To the author's knowledge, this is still an open problem. Therefore, in this work, Bayesian adaptive inference with constrained linear transforms is not considered further.

5.6 Summary

This chapter has presented a consistent Bayesian framework for adaptive training and adaptive inference by treating the parameters of the canonical model and the transforms as random variables. As the complexity can be controlled during training to reflect the amount of data, the standard point estimate adaptive training schemes, either with likelihood criterion or discriminative criteria, can be justified. In addition, a transform prior distribution is motivated from this Bayesian framework as well. It is then possible to do unsupervised adaptive inference using the adaptively trained systems with the transform prior distribution. This adaptive inference is a Bayesian generalisation of the standard two-step adaptation/recognition process. The key issue here is to calculate the marginal likelihood (or posterior of hypothesis in the case of discriminative adaptive inference) over the transform distribution.

As the marginalisation over the transform distribution is intractable for HMM based systems, approximations are required. Two forms of approximations are examined in this chapter. Lower bound approximations, both point estimates (MAP or ML) and variational Bayesian approach are investigated. In addition, direct approximations, sampling and the frame-independent predictive distribution, are also discussed. These approximations are investigated in both batch and incremental modes. The Bayesian approximation approaches are then applied to two specific transforms: interpolation weights in CAT and mean based linear transforms in SAT.

6

Experiments on Discriminative Adaptive Training

In this chapter, experiments concerning discriminative adaptive training, in particular, discriminative cluster adaptive training, are presented. All experiments were conducted on a LVCSR conversational telephone speech (CTS) task. The experimental setups, including the training and the test datasets, building of the acoustic models and recognition setup are introduced in section 6.1. Section 6.2 discusses various issues in discriminative cluster adaptive training including the effect of the I-smoothing prior and initialisation approach. Section 6.3 gives the comparisons between different discriminative adaptive training approaches.

6.1 Experimental Setup

The performance of various discriminative adaptive training schemes was evaluated on a large vocabulary speech recognition system, conversational telephone speech (CTS) task. Each conversation of CTS speech corpora is recorded with two distinct sides, one from each end of the telephone circuit. Each side is recorded and stored as a standard telephone coded signal with 8 KHz sampling rate, 8-bit μ-law encoding. The training dataset consists of 3 corpora recorded with slightly different acoustic conditions and collection framework. These are the LDC Callhome English (che), Switchboard (Swbd) and Switchboard-Cellular (SwCell) datasets. In total, they consist of 5446 speakers (2747 female, 2699 male) about 296 hours of data. Two held out test sets are used to evaluate recognition performance. A smaller test set of half of the 2001 development data for CTS (dev01 test data) is about 3 hours, called the dev01sub test data. The data is from the Swbd corpus and consists of 59 speakers (30 female, 29 male), 2663 utterances, 30K words. The second, larger test dataset is the eval03 dataset, about 6 hours. It comes from two different corpora (Swbd and Fisher), consisting of 144 speakers (77 female, 67 male), 7074 utterances, 76K words. All systems used a 12-dimensional MF-PLP front-end [127] with the C0 energy and its first, second and third derivatives with Cepstral mean and variance normalisation. An HLDA transform was applied to reduce the feature dimension to 39. VTLN was also used. The use of the linear projection scheme, i.e. HLDA, and simple feature normalisation schemes, mean and variance normalisation and VTLN, decreased the possible gains that could

be obtained using adaptive training, especially for constrained MLLR. However it gave a more realistic baseline. Single pass Viterbi decoding was used to recognise the test data sets using the adapted acoustic models and a tri-gram language model. The tri-gram language model was trained on 1044M words. It was built by interpolating various components, each was trained on different text data sources. The dictionary used was a multiple pronunciation dictionary with a vocabulary size of 58K words.

Two kinds of system were built. A simple system was used for initial development. The second is a more complex system similar to the systems used for NIST RT03 evaluation at CUED [25]. Both systems were built using the same state-clustered decision trees with 6189 distinct states. 16 component-per-state systems with 4 ML training iterations and 4 MPE training iterations were used for rapid development. 28 component-per-state systems with 4 ML training iterations and 8 MPE training iterations were built for generating results of more complex systems. I-smoothing using a dynamic ML prior was used to build all baseline MPE systems. As unsupervised adaptation was conducted, a simplified discriminative adaptive training procedure was used for all MPE adaptive training, where only the canonical model was discriminatively updated given the ML estimated transforms.

Speaker-independent (SI) [1], and gender-dependent (GD) MPE systems were built using standard MPE training [90]. Due to insufficient training data, the GD MPE system often gave poor performance. To obtain good performance from a GD system trained with the MPE criterion, the MPE-MAP technique was used, which takes into account the static prior information in I-smoothing [91]. Here, rather than using a MAP prior as in [91], the static MPE-SI models were directly used as the I-smoothing prior as this has been shown to give slightly better performance [26]. These systems are referred to as GD MPE-MAP systems. The GD MPE-MAP systems originated from the MPE-SI models and used 2 further MPE-MAP training iterations with only mean and Gaussian component weights updated. Therefore, it actually used more MPE training iterations, which should be a little more powerful than the other MPE systems. In this work, gender labels were assumed to be known for both standard GD MPE and GD MPE-MAP systems in decoding [2].

Several adaptive MPE systems were constructed using the simplified MPE adaptive training procedure. They include MPE-SAT (CMLLR), MPE-SAT (MLLR), MPE-CAT and MPE-ST which used CAT weights and CMLLR transforms as the structured transforms (ST). The systems using the ST of CAT weights combined with MLLR transforms were not built due to memory limitation given the complexity of the systems and our current experimental condition. During adaptive training and testing adaptation, global interpolation weights were estimated for CAT and separate speech and silence transforms were used for MLLR and CMLLR. Initial supervision hy-

[1] The speaker-independent system in this thesis is also gender-independent.

[2] Automatic gender detection may be done by aligning the hypothesised supervision with the GD models and then determining gender labels for each speaker using the average log-likelihood values. The gender detection error rates were shown by additional experiments to be 5.1% for `dev01sub` and 4.2% for `eval03`. Experiments showed that automatic gender detection had insignificant effects on the performance of GD systems.

potheses for estimating test transforms were generated using the ML-SI and the MPE-SI models respectively.

In the following experiments, wherever the term "*significant*" is used, a pair-wise significance test was done using NIST provided scoring toolkit `sctk-1.2`. The significance difference was reported using the Matched-Pair Sentence-Segment Word Error (MAPSSWE) test at a significance level of 5%, or 95% confidence [43].

6.2 Discriminative Cluster Adaptive Training

This section presents the development experiments of discriminative cluster adaptive training (CAT). All systems in this section are 16 Gaussian component per state systems with 4 ML and 4 MPE training iterations. The effects of the I-smoothing prior and initialisation approach were investigated in detail. With these investigations, configuration of MPE CAT is determined for further comparison with the other discriminative adaptive training techniques.

6.2.1 Baseline Performance

Table 6.1 shows the performance of various ML and MPE baseline 16 component systems for MPE-CAT development. They include ML and MPE systems of SI and GD training. A GD MPE-MAP system was also constructed, which gave the state-of-the-art GD MPE performance. Finally, the ML performance of a 2-cluster gender-initialised CAT system is given as the ML baseline for the MPE-CAT system to be developed.

System	`dev01sub`		`eval03`	
	ML	MPE	ML	MPE
SI	33.4	30.4	32.6	29.2
GD	32.7	30.3	32.2	29.3
GD (MPE-MAP)	—	29.6	—	28.7
CAT	32.6	—	31.9	—

Table 6.1 *16-component ML and MPE SI and GD performance and ML performance for the 2-cluster CAT system initialised with gender information. SI and GD MPE training used a standard dynamic ML prior and GD (MPE-MAP) used the MPE-SI model as the static MPE prior.*

From table 6.1, as expected the performance gain from MPE training is large, around 3% absolute for both test sets on SI and GD systems. This shows why the study of using discriminative criterion is important. Direct MPE-GD training did not get much gain compared to MPE-SI and was even worse on `eval03`. However, with the static MPE prior, the GD MPE-MAP system got the expected gain. This is consistent with the gains that were obtained on the broadcast news task [91]. For this reason, in the following experiments, only the GD (MPE-MAP) number is quoted as the MPE performance of a GD system.

The gender-initialised ML-CAT system was significantly better than the ML-SI baseline for both test sets according to the significance test. It was slightly, not significantly, better than the ML-GD system in `dev01sub`, while significantly better in the larger test set `eval03`. This shows that with the same number of clusters, the soft choice of clusters (CAT) outperformed the hard choice (GD) for ML systems.

6.2.2 Effect of I-smoothing Priors

I-smoothing is essential for obtaining good test set performance using MPE training [90]. The standard form of I-smoothing for training HMMs is to use a dynamic, single-cluster ML prior. As discussed in section 4.3.2, the selection of the prior parameters is of additional interest for MPE-CAT systems as the number of model parameters to be estimated is greater than that of the equivalent standard HMM systems. Thus the form of prior used may have a greater influence than for the standard HMM systems. A range of priors may be used, as described in section 4.3.2.2. A single-cluster static MPE prior may be obtained from the standard MPE-SI model. Alternatively a dynamic MPE prior can be obtained from the single-cluster MPE statistics generated during training. Furthermore, a multiple-cluster dynamic ML prior can be obtained from the multiple-cluster ML statistics during training. These three forms of prior, along with using a standard dynamic single-cluster ML prior were investigated first. It is worth noting that all CAT systems in this section were initialised using the gender information yielding 2-cluster systems.

Prior			Test sets	
Form	Type	Criterion	`dev01sub`	`eval03`
multiple	dynamic	ML	29.7	28.9
single	dynamic	ML	29.7	29.1
		MPE	29.3	28.4
	static	MPE	29.3	28.5

Table 6.2 *16-component 2-cluster MPE-CAT systems with different forms of I-smoothing prior. The MPE-CAT systems were initialised using gender information.*

Table 6.2 shows the 16 component development system performance of the MPE-CAT systems using different I-smoothing priors. It can be observed that the form of the prior greatly affects the error rate. For example the performance on `dev01sub` varied from 29.3% upto 29.7%, which was statistically significant. It should be noted that all these values are better than the performance of the ML-CAT baseline and the MPE-SI systems shown in table 6.1. The worst performance was obtained with the standard dynamic ML prior. The best performance was obtained using either the static or the dynamic single-cluster MPE priors. The performance of the MPE-CAT system using either of the MPE priors shown in table 6.2 was significantly better than both the ML-CAT system and the MPE-SI system. They were also better than the GD MPE-MAP system

in table 6.1. This shows that using a robust discriminative prior can benefit discriminative CAT training [3].

There was little difference in performance between the dynamic and the static MPE single-cluster prior systems. However, since a dynamic MPE prior requires additional accumulates [137] (unless a bias cluster is used), thus all the following MPE experiments of CAT and ST used the single-cluster static MPE prior.

As indicated in section 4.3.2.2 and appendix B, a multiple-cluster MAP estimates may also be used as the I-smoothing prior. The MAP prior is a trade-off between the multiple-cluster dynamic ML prior and the single-cluster static MPE-SI prior. The MAP parameter $\tau^{\texttt{MAP}}$ is the coefficient to balance the two. A MAP prior with $\tau^{\texttt{MAP}} = 0$ is equivalent to a multiple-cluster dynamic ML prior. On the other end, a MAP prior with $\tau^{\texttt{MAP}} = \infty$ is actually a single-cluster static MPE-SI prior.

$\tau^{\texttt{MAP}}$	dev01sub	eval03
0	29.7	28.9
25	29.7	28.8
50	29.6	28.8
100000	29.4	28.6
∞	29.3	28.5

Table 6.3 *16-component 2-cluster MPE-CAT systems with MAP priors. A MAP prior is a trade-off between the multiple-cluster dynamic ML prior and the single-cluster static MPE-SI prior, $\tau^{\texttt{MAP}}$ is the coefficient to balance the two. The MPE-CAT systems were initialised using gender information.*

Table 6.3 shows the MPE-CAT systems with the MAP prior with different $\tau^{\texttt{MAP}}$ values. It can be observed that there is a trend of improving performance from ML-based prior to MPE-based prior[4]. The static single-cluster MPE prior ($\tau^{\texttt{MAP}} = \infty$) was significantly better than the dynamic multiple-cluster ML prior ($\tau^{\texttt{MAP}} = 0$). This again motivates the use of static MPE prior for discriminative CAT in the following experiments.

6.2.3 Effect of Initialisation Approaches

A second interesting aspect for CAT systems is the number of clusters and how they are initialised. Various forms of intialisation have been discussed in section 3.4.2.2. Three forms of cluster initialisation were investigated here. The first two were cluster based schemes, where the interpolation weights were initialised using either gender information, or the corpus information. The gender initialised CAT system has 2 clusters while the corpus initialised system has 3 clusters, corresponding to `che`, `Swbd` and `SwCell` in training data respectively. The third

[3] Though dynamic MMI prior has also been used as a discriminative prior for MPE training of standard HMMs [101], it was not investigated here for MPE-CAT systems. The choice of investigating MPE priors is to keep consistent with the GD MPE-MAP baseline where the MPE-SI model was used as the I-smoothing prior.

[4] This observation is also consistent with the observation for GD MPE-MAP, which is the motivation of using static MPE prior for GD MPE-MAP [26].

form of initialisation used an eigen-decomposition approach, as described in section 3.4.2.2. For the eigen-decomposition initialised systems, the bias cluster interpolation weight was either constrained to stay at one, called a bias cluster system or allowed to vary after initialisation, called a no bias cluster system. Again the 16 component development system was used in this experiment.

Initialisation	Bias	#Clst	dev01sub		eval03	
			ML	MPE	ML	MPE
gender	no	2	32.6	29.3	31.9	28.5
corpus		3	32.3	29.2	31.7	28.3
eigen		3	32.3	29.0	31.5	28.2
eigen	yes	2	32.8	29.3	32.0	28.5
		3	32.3	29.0	31.6	28.3
		4	32.3	29.0	31.5	28.3

Table 6.4 *16-component MPE-CAT systems with different initialisation approaches and number of clusters. The MPE-SI model, a single-cluster static MPE prior, was used in I-smoothing of MPE-CAT.*

Table 6.4 shows the performance with different numbers of clusters and initialisation. Initially examining the form of the initialisation with no bias, the use of a 3-cluster eigen-decomposition system was significantly better than the 2-cluster gender initialised systems for both ML and MPE training. However, there was no significant difference between the eigen-decomposition initialised system and the corpus initialised system. For the eigen-decomposition scheme, various systems using a bias were also constructed. For these systems the use of 3 or 4 clusters was better than the 2-cluster system. However there was no significant difference in performance between any of the 3 or 4 cluster systems.

It is interesting to contrast the forms of systems used here, where there are relatively few clusters, with the large number of clusters used in many eigenvoices systems. Many eigenvoices systems [68] use large number of clusters, but with relatively simple acoustic models, for example single Gaussian component per state models. These simple models are not appropriate for LVCSR. More complex systems have been built using maximum likelihood eigenspace [87]. However, on the same task, and starting from a better baseline, greater gains were obtained using a simple 2-cluster CAT system [39][5]. One reason for this is that CAT updates all the model parameters in an adaptive training framework, whereas only the eigenvoices are updated in maximum likelihood eigenspace training [87]. The results from table 6.4 indicate that on this task, training all the canonical model parameters, the performance has approximately saturated at about 4 clusters. This effect has also been observed for eigenvoices in [11] where a large

[5]Few strict comparisons exist on large vocabulary systems. One close comparison is on the WSJ task where a 20 cluster eigenvoices system was built [87], and a 2 cluster CAT system [39] on the SI-284 training data. Despite starting from a better baseline, the CAT system showed a greater relative reduction in WER over the GI system than the equivalent eigenvoices system.

number of eigenvoices did not help performance and in some cases degraded the performance. The use of a relatively small number of clusters is advantageous when using MPE training. MPE, in common with other discriminative training criteria, is more likely to lead to overtraining than the ML criterion. Thus the generalisation with large numbers of clusters would be expected to be poor.

Though the 3-cluster no-bias eigen-decomposition approach gave similar results as the corpus initialisation in table 6.4, it is interesting to further investigate the breakdown of the `eval03` results. As described in the experimental setup, the `eval03` includes some `Fisher` data, which comes from a source not included in the training data. Performance gain on different parts of the test data may indicate the difference of the two initialisation schemes.

System	Initialization	`Swbd`	`Fisher`	Overall
SI	—	33.6	24.5	29.2
CAT	corpus info.	32.6	23.6	28.3
	eigenvoices	32.4	23.7	28.2

Table 6.5 *Breakdown WER on* `eval03` *of 16-component MPE-SI and 3-cluster non-bias MPE-CAT systems. The MPE-SI model, a single-cluster static MPE prior, was used in I-smoothing of MPE-CAT.*

Table 6.5 shows the breakdown between `Swbd` and `Fisher` on `eval03` of MPE-SI and the two MPE-CAT systems [6]. Both MPE-CAT systems obtained significant gains on both corpus compared to the MPE-SI system. For eigen-decomposition initialisation, the gain on `Swbd` corpus (1.2%) is much larger than the gain on `Fisher` corpus (0.8%). However, for corpus initialisation, the gain on `Swbd` corpus (1.0%) is similar to the gain on `Fisher` (0.9%). This is expected as the eigen-decomposition approach is a data-driven approach, which makes better description of the training data. Hence, it is likely to obtain gains on the test data which is similar to the training data. In contrast, the corpus initialisation used some prior knowledge and deliberately discriminated different corpus at the beginning. The final CAT system was not specifically tuned to the training corpus as much as the eigen-decomposition initialisation. Hence, the gains on different corpus may be more evenly distributed. This also implies that even with the same number of clusters, the initialisation approach does have impact on the recognition performance.

6.3 Comparison of Discriminative Adaptive Training Techniques

In the previous section, various configurations of discriminative cluster adaptive training were investigated. This section will compare the performance of discriminative adaptive training techniques on both the development (16 component per state)and the more complex (28 component per state) systems. The simplified discriminative adaptive training procedure was adopted for all adaptively trained MPE systems. To be consistent with the form of MPE training used to train

[6] `dev01sub` has the same corpus with training data, thus no need to show breakdown numbers.

the MPE-SI systems, the dynamic ML prior was also used as the I-smoothing prior for training MPE-SAT with MLLR and MPE-SAT with CMLLR, where the canonical models are also standard HMMs. As GD MPE-MAP is the baseline for cluster adaptation, the MPE-CAT systems in this section used an I-smoothing prior based on the static MPE-SI model. As a multiple-cluster system, the MPE-ST systems, where the structured transforms are CAT and CMLLR, adopted static MPE-SI model as the I-smoothing prior, too. Due to memory limitation, it is only possible to build a 2-cluster MPE-CAT system for the 28-component configuration. Hence, the 2 cluster gender initialisation scheme was used here, though systems with more clusters may be expected to yield greater gains. The comparison on 16-component systems is given first. The next section presents the results of 28-component systems.

6.3.1 16-Component Development Systems Performance

This section gives the performance of 16-component development systems, which are regarded as baselines for the investigation on various discriminative adaptive training techniques. Initially, rapid adaptation schemes, CAT and GD models, were investigated. The CAT systems were initialised with gender information. The ML results for GD were generated using the standard ML GD models, while the MPE results for GD were generated using the GD MPE-MAP models to give state-of-the-art GD MPE performance. Due to the small amount of parameters to be estimated during adaptation [7], both CAT and GD adaptation can be rapidly done. However, for the same reason, the adaptation gain may be limited. The comparison of rapid adaptation techniques is shown in table 6.6. From table 6.6, all MPE systems significantly outperformed

System	#Clst	dev01sub		eval03	
		ML	MPE	ML	MPE
SI	1	33.4	30.4	32.6	29.2
GD (MPE-MAP)	2	32.7	29.6	32.2	28.7
CAT	2	32.6	29.3	31.9	28.5

Table 6.6 *ML and MPE performance for 16-component SI, GD, and CAT systems. ML-GD models were trained on gender-specific training data. GD (MPE-MAP) was built on top of MPE-SI model and used a single-cluster static MPE prior in I-smoothing. The CAT systems were initialised using gender information and used the same single-cluster static MPE prior in I-smoothing.*

the corresponding ML systems. The gains of the GD MPE-MAP and MPE-CAT systems were both over 3% absolute over the ML GD and ML-CAT systems. This is similar to the gain obtained for the MPE-SI model. This shows that discriminative training can be effectively applied to multiple-cluster models. It can also be observed that both GD and CAT systems significantly outperformed the ML and the MPE SI baselines on both test sets. Due to the use of soft weights, CAT systems,

[7] For GD adaptation, gender label is the parameters to estimate. In this work, all test gender labels were assumed to be known. This gives an upper bound of the GD adaptation performance.

either ML or MPE, were better than the GD systems. However, the gains of the MPE-CAT system compared to the GD MPE-MAP system were not significant. This is due to the limited number of clusters of the MPE-CAT system.

System	Adaptation		dev01sub		eval03	
	Training	Test	ML	MPE	ML	MPE
SI	—	MLLR	31.1	28.5	30.2	27.0
SAT	MLLR		30.4	27.4	29.3	26.4
SI	—	CMLLR	31.5	28.3	30.4	27.1
SAT	CMLLR		31.0	27.8	30.0	26.8
CAT	CAT	ST(M)	30.5	27.4	29.5	26.4
CAT	CAT	ST	30.8	28.0	29.7	26.5
ST	ST		30.6	27.5	29.6	26.1

Table 6.7 *ML and MPE performance of adaptation using complex schemes on 16-component SI, SAT, CAT and ST systems. MPE-SI, MLLR and CMLLR based MPE-SAT systems both used the standard dynamic ML prior in I-smoothing. ST(M) transform was CAT weights combined with MLLR transforms. ST transforms were CAT weights combined with CMLLR transforms. MPE-CAT and MPE-ST systems used the single-cluster static MPE prior in I-smoothing. They were both initialised using gender information, hence have 2 clusters.*

To obtain more gains of adaptation, more complex adaptation schemes may be used, where the number of adaptation parameters is considerably larger than the rapid adaptation schemes. Table 6.7 shows the results of adaptation using more complex schemes. Both CAT and ST systems were 2 clusters and initialised using gender information. The form of ST used here for adaptive training was CAT weights combined with CMLLR transforms. To implement ST adaptation for the CAT models, first the interpolation weights for the CAT models were estimated and then CMLLR transforms were further estimated on top of the adapted CAT models. Another possible form of ST is CAT weights combined with MLLR transforms. As indicated in section 3.4.3, using this form of ST in adaptive training require a large amount of memory. Especially for discriminative training, the situation is even worse due to the additional denominator statistics required. Hence, the MLLR based ST system was not implemented given our current experimental condition. However, in testing adaptation, as the estimation of MLLR transform does not require large memory, it is possible to estimate MLLR transform on top of the adapted CAT models. This process is similar to the CMLLR based ST adaptation on top of the CAT models and is referred to as ST(M) in table 6.7.

Comparing the performance of the SI system with MLLR or CMLLR adaptation in table 6.7 to the performance of unadapted SI systems in table 6.6, significant gains were obtained by using complex adaptation schemes. The performance of adaptively trained systems were significantly better than that of multi-style, SI, trained systems using the same form of adaptation. For example, the MPE-SAT system with MLLR obtained 1.1% absolute gain on `dev01sub` over the MPE-SI system after MLLR adaptation; the MPE-SAT system with CMLLR got 0.5% gain on

dev01sub over the MPE-SI system after CMLLR adaptation. These gains show the advantage of adaptive training compared to multi-style training. However, there is a large difference between the gains of the CMLLR based SAT system and the MLLR based SAT system. It can be observed, the gains of the CMLLR based SAT system over the SI system are always much smaller (only about half) than that of the MLLR based SAT systems. This is due to the use of various feature normalisation techniques, such as CMN, CVN and VTLN, and linear projection schemes such as HLDA. As CMLLR is a feature transform, its adaptation power will be limited if the features have been previously normalised or projected to a feature space with reduced acoustic mismatch. In contrast, as a mean transform, MLLR is relatively independent to those feature normalisations and the adaptation power is more additive.

For a similar reason, when comparing the absolute performance of the two adaptively trained systems, the MLLR based SAT system is significantly better than those of the CMLLR based SAT system on both test sets. This difference is believed to mainly come from the nature of the transforms used rather than the adaptively trained canonical models. To investigate this, appropriate MLLR adaptation [8] was performed on the CMLLR based MPE-SAT system, which yielded a WER of 26.2%. In contrast, performing appropriate CMLLR adaptation on the MLLR based MPE-SAT model yielded 26.1%. The difference is not significant showing that both canonical models have similar adaptabilities.

The last three lines give the results of using complex adaptation schemes on multiple-cluster models. Compared to the rapid CAT adaptation in table 6.6, the CAT system with ST and ST(M) adaptation obtained significant gains on both test sets. The performance of ST(M) on top of the CAT system outperformed the corresponding performance of ST. This is again due to the feature normalisations and projections, which limit the power of CMLLR. As the most complex transform representing non-speech variabilities, both ML and MPE ST systems yielded the lowest WER on both test sets among comparable CMLLR or CAT systems.

6.3.2 28-Component Systems Performance

The previous section presents comparisons of different discriminative adaptive training techniques using 16-component development systems. In this section, more complex 28-component systems were built to compare these techniques. MLLR based SAT systems were not built for this configuration due to memory limitation as discussed in section 4.2.1. Hence, in this section, CMLLR based systems are the focus.

Table 6.8 shows the performance using rapid adaptation techniques. Comparing the baseline MPE-SI numbers to the development system numbers in table 6.6, significant gains, 1.5% absolute gain on dev01sub and 0.9% on eval03 were obtained by using the more complex 28-component systems. Similar gains due to increased complexity can be observed for the other

[8] The "appropriate MLLR adaptation" here means CMLLR transforms need to be first estimated to adapt the CMLLR canonical model. MLLR transforms are then estimated on top of the adapted model. Similar process applies to "appropriate CMLLR adaptation" except for applying MLLR first and then CMLLR.

System	#Clst	dev01sub		eval03	
		ML	MPE	ML	MPE
SI	1	32.2	28.9	31.5	28.3
GD (MPE-MAP)	2	32.2	28.7	31.2	28.0
CAT	2	31.9	28.7	31.2	27.8

Table 6.8 *ML and MPE performance for 28-component SI, GD, and CAT systems. ML-GD models were trained on gender-specific training data. GD (MPE-MAP) was built on top of MPE-SI model and used a single-cluster static MPE prior in I-smoothing. The CAT systems were initialised using gender information and used the same single-cluster static MPE prior in I-smoothing.*

28-component systems. The GD MPE-MAP and the MPE-CAT systems still outperformed MPE-SI in this setup. For example, the MPE-CAT system got 0.5% gain on eval03 over the MPE-SI system. However, this gain is smaller compared to the gains of the 16-component development system (0.7%) in table 6.6. This is due to the increased model complexity. Though no gain on dev01sub, the MPE-CAT system still showed gains over the GD MPE-MAP system on the larger test set eval03.

System	Adaptation		dev01sub	eval03
	Training	Test		
SI	—	CMLLR	27.1	26.1
SAT	CMLLR		26.8	25.9
CAT	CAT	ST	27.1	25.7
ST	ST		26.6	25.5

Table 6.9 *MPE performance of using complex adaptation schemes on 28-component SI, CAT and ST systems. MPE-SI used the standard dynamic ML prior in I-smoothing. ST transforms were CAT weights combined with CMLLR transforms. MPE-CAT and MPE-ST systems used the single-cluster static MPE prior in I-smoothing. They were both initialised using gender information, hence have 2 clusters.*

Table 6.9 shows the performance of using complex adaptation schemes on the 28-component MPE systems. Compared to the performance of the MPE-SI and the MPE-CAT systems in table 6.8, complex adaptation schemes show significant gains over unadapted models or simple rapid adaptation schemes. This is consistent with the observation in the development experiments shown in table 6.7. There was no consistent gain found between CMLLR based SAT and CAT after ST adaptation. When the gains of ST over other systems were compared to the gains of the development system, they were also found reduced due to increased model complexity. For example, on eval03 dataset, ST obtained 0.2% over CAT and 0.4% over SAT, while from table 6.7, ST got 0.4% over CAT and 0.7% over SAT. However the use of the ST for both training and testing still showed significant gains over all the other systems.

6.4 Summary

This chapter has described experiments concerning discriminative adaptive training techniques. A new technique, discriminative cluster adaptive training based on MPE criterion, was investigated first. MPE training requires a I-smoothing distribution for updating multiple-cluster model parameters. The form of I-smoothing prior was shown to be important in discriminative CAT. A static MPE prior was finally used as the configuration for all the other MPE-CAT experiments. With the static MPE prior, a 2-cluster MPE-CAT system initialised by gender information was significantly better than the MPE-SI system and also outperformed the GD MPE-MAP system. The effect of the number of clusters and initialisation approach was also discussed. With more clusters, the CAT systems gave improved performance. Given the complexity of the systems used, 3 clusters led to saturated performance. A data-driven approach, eigen-decomposition initialisation, was shown to give better description on the training data than the corpus initialisation with the same number of clusters.

The second part of this chapter compared the performance of different discriminative adaptive training techniques. In all experiments, adaptively trained systems outperformed multi-style trained systems, which showed the advantage of adaptive training. In the experiments of rapid adaptation schemes with 16-component development systems, the GD MPE-MAP system and the 2-cluster MPE-CAT system both significantly outperformed the MPE-SI baseline. The MPE-CAT system was always better than the GD MPE-MAP system. Using more complex adaptation schemes gave further significant gains over the simple rapid adaptation schemes. The CMLLR based MPE-SAT system got smaller gain than the MLLR based MPE-SAT system due to various feature normalisation and projection techniques used in this experimental configuration. As the most powerful adaptively trained systems, the ST systems always yielded the lowest WER among comparable adaptively trained systems. In the final experiments with 28-component systems, due to the increased model complexity, most gains were reduced. However, similar trends were found as the development systems. the MPE-CAT systems outperformed the GD MPE-MAP systems and the ST systems still yielded the best performance among all comparable adaptively trained systems.

7

Experiments on Bayesian Adaptive Inference

This chapter presents the experiments concerning Bayesian adaptive training and adaptive inference using both ML and MPE adaptively trained systems. All experiments were conducted on the conversational telephone speech (CTS) task with a similar setup to the previous chapter. However, in contrast to the previous section, for all experiments, N-Best list rescoring was used as the inference scheme instead of Viterbi decoding due to the nature of Bayesian adaptive inference. Both cluster adaptive training (CAT) and speaker adaptive training (SAT) with MLLR were investigated in this chapter. Section 7.1 discusses the form of the transform prior distribution and the use of N-Best list rescoring. Experiments on utterance level Bayesian adaptive inference are described in section 7.2. The effect of the tightness of lower bounds and the effect of using multiple component prior are also discussed in the section. Then, section 7.3 investigates incremental Bayesian adaptive inference. Conclusions are given in the final section.

7.1 Experimental Setup

The performance of various Bayesian adaptive inference approximations were evaluated on the same conversational telephone speech task as chapter 6. The training data and front-end used are the same as those introduced in section 6.1. The test set used in this chapter is the `eval03` data set, consisting of 144 speakers, about 6 hours. A standard decision-tree state-clustered tri-phones with an average of 16 Gaussian components per state was constructed as the starting point for the adaptive training. This is the baseline speaker-independent (SI) model. 4 iterations of standard MPE training [93] were then used to produce the baseline MPE-SI model.

Two forms of ML and MPE adaptively trained systems were built. The first were 2-cluster CAT systems, initialised using gender information. A global transform (interpolation weight vector) was used for the CAT systems. For the ML-CAT system, a 2-component GMM transform prior distribution was estimated from the training transforms. Given the ML weights, an MPE-CAT system was built using the simplified discriminative adaptive training procedure as discussed in section 5.2. The I-smoothing prior used here was again the static MPE-SI prior in accordance with the setup in the previous chapter. The second form was SAT systems constructed using

MLLR. For the ML-SAT system, a single Gaussian and a 2-component GMM transform prior distributions were both estimated from the training data. The single Gaussian prior was used for most experiments in this chapter. Separate speech and silence transforms were used, the priors for which were independently estimated. Given the ML linear transforms, a simplified discriminative adaptively trained systems, MPE-SAT[1], was also built. As the canonical model for MPE-SAT is a standard HMM model set, the ML sufficient statistics was used as the I-smoothing prior to keep consistent with the standard MPE training technique [93]. Having trained the MPE-CAT and the MPE-SAT systems, transforms for each training speaker can again be estimated using the ML criterion based on the MPE models. The resultant ML transforms were then used to estimate prior distributions for the simplified discriminative adaptive models as described in section 5.2. With similar approaches, transform prior distributions for non-adaptively trained systems, ML-SI or MPE-SI, can also be obtained. Bayesian adaptive inference techniques can then be applied to the non-adaptively trained systems.

As the canonical model is trained given the transforms representing non-speech variabilities, it is not suitable to be directly used in inference. To illustrate this, standard Viterbi decoding results were generated for ML-SI and ML-SAT systems without any adaptation[2]. The ML-SI unadapted performance was 32.6%, while the ML-SAT unadapted performance was 33.8%. Thus, without adaptation, the ML-SAT system was significantly[3] worse than the ML-SI system by over 1.0%. This shows that the adaptively trained system can not work well without taking into account the non-speech variabilities of the test domain. Therefore, adaptive inference is required for adaptively trained systems.

As discussed in section 5.1.2, the Viterbi algorithm is not appropriate for Bayesian adaptive inference. Hence, N-Best rescoring was used for inference in the experiments. In this rescoring process, the likelihood calculation was implemented using the forward-backward algorithm because this is consistent with the formulae derived in chapter 5. Two 150-Best lists were generated for ML and MPE systems respectively with the corresponding SI models. A 300-Best list was also generated for ML systems to check the performance of 150-Best list rescoring. The Viterbi decoding, 150-Best list and 300-Best list rescoring results for the ML-SI system are shown in table 7.1.

As the Viterbi algorithm calculates the likelihood of an observation sequence in a different way from the forward-backward algorithm, there is significant difference between the results of Viterbi decoding and N-Best list rescoring in table 7.1. The difference between the two N-Best lists is small. This shows that the hypothesis sequences included in the 150-Best list may be sufficient for the ML-SI system. Performing a spot-check on the best VB configuration for the ML-SAT system in section 7.3 with a 300-Best list further illustrated that this was not a major

[1] Without introducing confusions, the term "MPE-SAT" and "ML-SAT" in this chapter refer to the systems built with unconstrained mean MLLR transforms.

[2] Note that for the ML-SAT system, no adaptation is equivalent to applying an identity transform.

[3] As in the previous chapter, wherever the term "*significant*" is used, a pair-wise significance test was done using NIST provided scoring tool `sctk-1.2`, which uses a MAPSSWE approach to conduct significance tests with the significance level of 5% [43].

Likelihood Cal.	N-Best List	
	150	300
Viterbi	32.6	
Forward-Backward	32.8	32.8

Table 7.1 *Viterbi decoding and N-Best list rescoring performance of the 16-component ML-SI system on* `eval03`

problem. All results shown in the rest of this chapter are based on the two 150-Best lists unless explicitly noted.

7.2 Utterance Level Bayesian Adaptive Inference

To illustrate the effects of the Bayesian approaches on adaptive inference, the homogeneous blocks considered in this section were based on a *single* utterance, not as in the standard case on a side (speaker) basis[4]. For the `eval03` test set the average utterance length was 3.13 seconds, compared to the average side length of 153.75 seconds. This dramatically limits the available data for estimating transforms. Results of Bayesian adaptive inference on CAT and SAT systems are shown in the below sections respectively.

7.2.1 Experiments Using CAT

Table 7.2 shows the 150-Best list rescoring results of different Bayesian approximation techniques on both ML-CAT and MPE-CAT systems.

System	Bayesian Approx.	Train	
		ML	MPE
SI	—	32.8	29.2
CAT	Sampling	32.2	28.5
	FI	32.5	28.8
	ML	32.2	28.6
	MAP	32.2	28.6
	VB	32.1	28.6

Table 7.2 *ML and MPE 150-Best list rescoring performance on* `eval03` *of utterance level Bayesian adaptive inference on 2-cluster CAT systems initialised with gender information . The weight prior distribution was a 2-component GMM. 1 transform distribution update iteration was used for the lower bound approximations.*

[4]The experiments in chapter 6 were all based on speaker basis. Hence, they are not comparable to the results in this section.

The performance of the baseline SI systems are shown in the first row. For CAT, the simple sampling approximation to the Bayesian integral could be used. Here, 200 samples were drawn from the two GMM CAT prior distributions and used to rescore the ML and MPE 150-Best lists respectively. The sampling approach gave about 0.7% absolute better than the SI systems, for both ML and MPE. This may be viewed as a bound on the performance from the Bayesian approximation perspective. Though the FI assumption results in full-covariance matrices for CAT as discussed in section 5.5.1, the FI performance improved WER by only about 0.4% absolute over the SI systems in both ML and MPE training. It was worse than the other approximation schemes. This illustrates the effect of not constraining the transform to be constant for each homogeneous block.

The last three sets of results were obtained using different forms of lower bound approximations. These lower bound approximations were based on an iterative learning process to tighten the lower bound before inference. Thus initialisation and number of iterations must be considered. Depending on the approximation used, different initialisations of the transform variational distribution were used. The ML approach used the component posterior occupancies generated by the SI models for initialisation, consistent with the standard CAT adaptation [36]. The MAP approach used the mean of the prior transform distribution. The prior distribution was used in the zeroth iteration of VB approximation. Due to the limited amount of the available data, only a single iteration was used in the iterative learning process to estimate the transform distribution used for final inference.

Examining the ML-CAT results, the performance of the ML approximation on the first iteration was about 0.6% absolute better than the ML-SI model. Though the MAP and the VB approximations were slightly better than the ML approximation, they were all about the same as that of the sampling approach. This is expected as the number of parameters for CAT is very small, the transform variational posterior distribution can be effectively approximated by a Dirac delta function. Therefore, the use of a point estimate is reasonable. For the MPE trained systems, the relative gains of different approximation approaches were similar to those of the ML-CAT systems. Pair-wise comparison on each row shows that the MPE-CAT systems always significantly outperformed the ML-CAT systems by over 3.5%, illustrating the power of discriminative training. The gain of MPE-CAT system compared to MPE-SI system was only slightly worse than the ML gain. This is because the number of the transform parameters for CAT is limited, hence, the effect of using ML-based transform prior distribution on a discriminatively trained model is small.

An important comparison in table 7.2 is to compare the SI performance (32.8% for ML and 29.2% for MPE) to the performance of Bayesian adaptive inference using VB approximation on top of the CAT model (32.1% for ML and 28.6% for MPE). As compared in figure 5.1, the SI performance is the result of using HMM assumption, the DBN in figure 5.1(a), in both training and inference, whereas Bayesian adaptive inference on top of adaptively trained model uses adaptive HMM assumption, the DBN in figure 5.1(b), in both training and inference. Therefore, the above comparison gives the best possible gains that can be obtained when using a consistent

adaptive HMM framework in both training and inference. As expected, the gains are significant (0.7% for both ML and MPE), showing the advantage of using consistent adaptive HMM assumption compared to the standard HMM assumption.

7.2.2 Experiments Using SAT with MLLR

The CAT systems have few transform parameters, hence, there is no significant difference between different Bayesian approximations. To reveal the difference between the approximation schemes, the SAT systems which have far more transform parameters are investigated in detail in the following sections.

7.2.2.1 Performance of SAT Systems

Table 7.3 shows the performance of Bayesian adaptive inference with a single Gaussian transform prior distribution on SI and SAT systems. The initialisation schemes for lower bound approximations were similar to the CAT systems except that an identity matrix was used for ML initialisation. Again, transform distributions used for inference were updated only for a single iteration.

Bayesian Approx	ML Train		MPE Train	
	SI	SAT	SI	SAT
—	32.8	—	29.2	—
FI	—	32.9	—	29.7
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 7.3 *ML and MPE 150-Best list rescoring performance on* `eval03` *of utterance level Bayesian adaptive inference on SI and MLLR based SAT systems with a single Gaussian transform prior distribution. 1 transform distribution update iteration was used for lower bound approximations.*

Initially, the performance of the direct, FI, approximation in table 7.3 is investigated. The FI performance on the ML-SAT system was slightly worse than the ML-SI system, while for MPE-SAT was about 0.5% worse than the MPE-SI. The degradation of the performance on the ML-SAT system is mainly because of the FI assumption and the transform prior distribution used. First, the transform prior distribution for SAT is not a full covariance Gaussian. As the number of parameters for linear transforms (over 1500) is much more than that of the CAT interpolation weights (only 2), a single Gaussian prior may not be complex enough to model the variability of linear transforms. Hence, unless a GMM prior distribution is used, the FI approximation on the ML-SAT system is slightly worse than the ML-SI system[5]. Second, the estimation of the

[5]Experiments have shown that complex GMM prior distributions have more gains on non-HLDA systems than HLDA systems. While for HLDA system, due to the normalisation effect of HLDA, GMM prior distribution only got

hyper-parameters of the SAT prior is not as robust as CAT due to the dramatically increased parameters. However, the performance degradation for the MPE-SAT system may have another cause. As discussed in section 5.2, the transform prior distribution for the MPE-SAT system was obtained based on the ML transform estimates and applied to calculate the likelihood rather than the discriminative evidence. Thus though the canonical model parameters are discriminatively trained, the discriminative power is reduced by applying an ML based transform prior in a non-discriminative way. This problem may be solved if a real discriminative prior distribution is estimated and appropriately used in discriminative Bayesian inference, but it is not addressed in this work.

The last three rows of table 7.3 show the performance of the lower bound approximations. In contrast to the ML-CAT system, the ML approximation performance of ML-SAT system was about 2.4% absolute worse than that of the ML-SI baseline (32.8%). This is expected as the MLLR transform parameters were estimated using an average of only 300 frames. This problem was reduced by using MAP estimation. It gave a 1% absolute gain over the ML-SI baseline, showing the importance of using prior information when estimating MLLR transforms with little data. The final approximation considered was the VB approximation. This should yield more robust estimates as a distribution over the transform parameters is used rather than a point estimate. It was 0.3% absolute better than the MAP approach, which was statistically significant using the pair-wise significance test. It shows that the robustness issue is more important for SAT than CAT due to the large number of parameters. Hence, different approximation approaches showed different effects here.

As a comparison to the adaptively trained system, Bayesian adaptive inference on non-adaptively trained systems was also investigated, where the prior distribution was estimated based on the non-adaptively trained models. This was a mixture use of the two DBNs in figure 5.1. HMM assumption was employed for training whereas adaptive HMM assumption was used for Bayesian adaptive inference. As training on non-homogeneous data with the HMM DBN is actually multi-style training, the resultant system is not compact and contains both speech and non-speech variabilities. Hence, the adaptability of the multi-style models may be smaller than adaptively trained systems. From table 7.3, the ML-SAT system always significantly outperformed the ML-SI baseline system by about 0.3% for all approximation schemes. This shows the importance of using adaptive HMM DBN in training. Again, comparing using adaptive HMM assumption to using HMM assumption in both training and inference, significant gain of 1.3% (SAT+VB compared to SI baseline) can be observed.

For MPE trained systems, the MAP approximation again significantly outperformed the ML approximation and the VB approximation got the best performance for the MPE-SAT system. Though the MPE-SAT systems got about 3% absolute gain over the ML-SAT systems for all approximation approaches, the gains over the baseline MPE-SI system (29.2%) was greatly reduced compared to the gains of ML-SAT over ML-SI. For example, the VB gain for the MPE-SAT system over the MPE-SI baseline is only about 0.6%, which is significantly smaller than the gain

slight gains over single Gaussian prior as shown in section 7.2.2.3.

of 1.3% for the ML-SAT systems. The gains of non-adaptively trained MPE systems were also reduced compared to the ML gains. This again shows the effect of using ML based transform prior distributions in a non-discriminative way.

7.2.2.2 Effect of Tightness of Lower Bound

As discussed in section 5.3.1, the approximation quality is dependent on the tightness of the lower bound. This section will investigate this effect.

As all the lower bound approximations use an iterative tightening process, the number of iteration will affect the tightness of the lower bounds. With more iterations, the lower bounds are expected to be tighter. The lower bound approximations in table 7.2 and 7.3 all used a single iteration to update transform distributions. Table 7.4 gives the performance with a different number of update iterations for the ML-SAT system using the VB approximation.

# Iteration for $q(\mathcal{T})$	ML Train (SAT)
0	34.1
1	31.5
2	31.6

Table 7.4 *150-Best list rescoring performance on* `eval03` *of utterance level VB adaptive inference on the MLLR based ML-SAT system with different number of iterations. The transform prior distribution was a single Gaussian.*

In the zeroth iteration, the prior distribution is used to calculate the VB lower bound. As the distribution was not updated using the test data, it yielded a very loose lower bound. Consequently, the VB performance on ML-SAT was significantly worse than the SI baseline performance (32.8%) by 1.3%. This illustrates the sort of degradation that can result when the bound is too loose. After one iteration, the performance was significantly improved by 2.6%. The importance of tightening the lower bound was proved. One further iteration gave slight degradation, showing that the tightening process had converged. Therefore, in the rest of experiments, only 1 iteration was used for lower bound based Bayesian adaptive inference.

The above experiments on lower bound approximations were all based on the N-Best supervision, where one transform distribution was generated for each possible hypothesis. As discussed in section 5.3.1.1, using the 1-Best hypothesis as supervision to learn the transform distributions may bias to the particular hypothesis. This transform distribution may give a looser lower bound approximation for the other hypotheses and consequently degrade the performance. Here, an experiment was done to contrast the standard 1-Best supervision to N-Best supervision. MAP and VB approaches were examined using 1-Best hypothesis as supervision. One transform distribution was estimated per utterance for all possible hypotheses. The results are shown in table 7.5. Only 1 iteration was used for the transform distribution update.

Comparing the 1-Best supervision results to the N-Best supervision performance, the VB and the MAP approximations both degraded significantly. Though the estimated transform distribu-

Bayesian Approx.	Supervision	
	N-Best	1-Best
MAP	31.8	32.0
VB	31.5	32.0

Table 7.5 *150-Best list rescoring performance on* `eval03` *of utterance level Bayesian adaptive inference on the MLLR based ML-SAT system with 1-Best or N-Best supervision (N=150). The transform prior distribution was a single Gaussian. 1 transform distribution update iteration was used for lower bound approximations.*

tion may lead to a tight lower bound for the 1-Best hypothesis, for all the other hypotheses, the particular transform distribution is unlikely to give as tight lower bound as using the the N-Best supervision. The above results illustrates the impact of this on WER. It is also interesting to note that the degradation for the VB approximation (0.5%) was larger than MAP (0.2%). This is because the VB approximation is more likely to be tuned to the supervision than the MAP approximation due to the use of non-point distributions. Then, the estimated transform distribution was biased to the 1-Best hypothesis more heavily and led to more mismatch to the other hypotheses, consequently more degradation. This shows that VB may be more sensitive to the supervision used.

7.2.2.3 Effect of Multiple Component Transform Prior Distribution

The above SAT experiments all used a single Gaussian as the form of the transform prior distribution. It is also interesting to investigate whether a multiple component prior can benefit Bayesian adaptive inference. A 2-component GMM prior distribution for SAT transform was estimated and then used with the various approximations. Table 7.6 gives the performance of the ML-SAT system with different Bayesian approximation approaches with a GMM transform prior distribution.

Bayesian Approx.	# Prior Component	
	1	2
FI	32.9	32.7
MAP	31.8	31.7
VB	31.5	31.5

Table 7.6 *150-Best list rescoring performance on* `eval03` *of utterance level Bayesian adaptive inference on the MLLR based ML-SAT system with different Gaussian transform prior distributions. 1 transform distribution update iteration was used for lower bound approximations.*

For the FI assumption, the use of the GMM prior distribution obtained a slight gain of 0.2%. The performance is still similar to the SI performance (32.8%). This is expected because the FI assumption is similar to the assumption of standard HMMs as discussed in section 5.3.2.2. For MAP approximation, the gain reduced to 0.1%. While for VB approximation, there was no gain

by using the GMM prior distribution. These very slight improvements imply that for this task, a single Gaussian distribution is good enough.

7.3 Incremental Bayesian Adaptive Inference

In the previous section, the adaptive inference was performed in a batch mode with each homogeneous block assumed to be based on one utterance. This section will investigate Bayesian adaptive inference in incremental mode with the *side-based* homogeneous block. The incremental inference procedure has been described in section 5.4. For `eval03`, on average, each side has about 49 utterances, which has considerably larger amount of data than the previous section. In this section, only the lower bound approximations were examined. During adaptive inference, a single iteration [6] was used for estimating the variational transform distribution. It is worth emphasising that the adaptive inference in this section was still run in an unsupervised mode, hence, the adaptation data was also the data to be recognised.

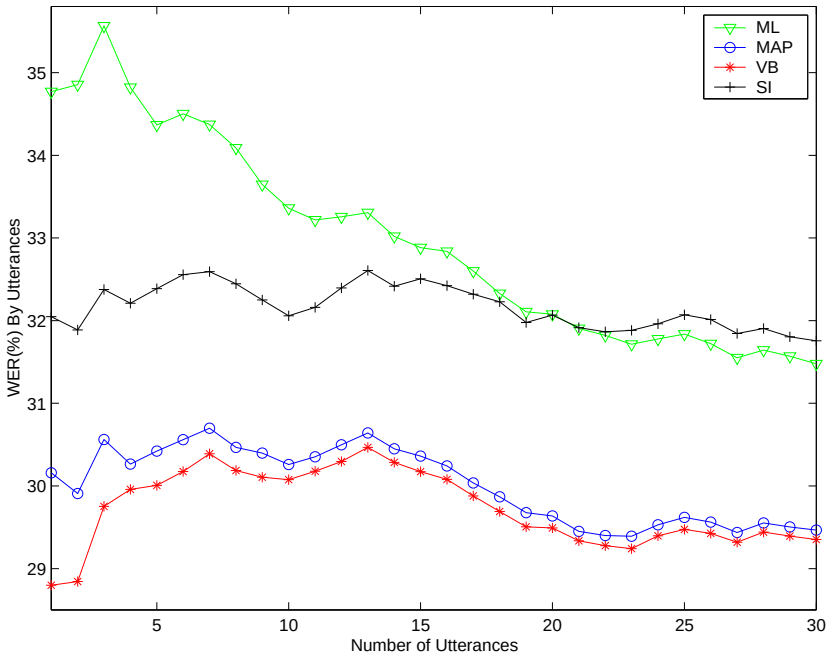


Figure 7.1 *150-Best list rescoring cumulative WER (%) on* `eval03` *of incremental Bayesian adaptive inference of different number of utterances on the MLLR based ML-SAT system. 30 utterances are shown here. The transform prior distribution was a single Gaussian.*

In incremental adaptive inference, it is interesting to investigate the change of the performance as the number of utterances varies. Thus, cumulative WERs of the first 30 utterances of the ML-SAT system are shown in figure 7.1 to compare the performance of different Bayesian approximations. The SI line in figure 7.1 is for the unadapted ML-SI baseline. The overall average

[6] Using the notations in section 5.4, this means the final iteration index is $K = 2$.

performance of this system is shown in table 7.3. As expected, for a limited number of utterances, the ordering of performance is similar to that shown for the ML-SAT system in table 7.3. The VB approximation had the best performance. As the number of utterances increased the difference between the VB and the MAP approximations became far smaller[7]. Given sufficient adaptation data, the point transform estimates are reasonable approximations to the transform posterior distribution. Hence the two are expected to be close to each other. The ML approximation was significantly worse than all the others at the beginning because of the insufficient adaptation data. The ML performance was gradually improved as more data came and outperformed the unadapted SI system after 20 utterances. However, due to the poor performance at the beginning, the cumulative WER was still significantly worse than MAP and VB after 30 utterances.

The cumulative WERs of the first 30 utterances of the MPE-SAT system are also plotted in figure 7.2. At the first utterance, the WER of VB and MAP was the same. This is due to the ordering of the utterances because in section 7.2.2 the VB approximation has shown to outperform the MAP approximation when averaging errors over all utterances. The general trend similar to figure 7.1 appeared after the first utterance. The VB approximation again outperformed the MAP approximation when adaptation data was limited. With more adaptation data, the two approaches became closer and the performance of the ML approximation was greatly improved. It is also interesting to compare the gap between SI and VB of the two figures. The gap in figure 7.1 is obviously larger than the gap in figure 7.2. This means for ML systems, the gain of ML-SAT compared to ML-SI was always larger than the gains of corresponding MPE systems during the whole incremental adaptive inference process. This is also due to the use of ML based transform prior on the MPE-SAT system.

Table 7.7 shows the overall average performance on the complete test set. As a baseline for incremental Bayesian adaptive inference, the ML-SI model was adapted during inference using the standard robust ML adaption technique. Here, a threshold was used during the ML adaptive inference to determine the minimum posterior occupancy to estimate a robust ML transform, referred to as the `ML+thresh` in table 7.7 [8].

As expected, the incremental adaptive inference of `ML+thresh` showed significant gains over both the ML-SI and the MPE-SI systems in table 7.3, around 1.4%. The performance of ML approximation with the ML-SAT system was about 1% absolute better than the unadapted ML-SI

[7]The WER curves in figure 7.1 are not monotonically decreasing due to the order of the utterances. As shown in table 7.3, the average performance of *all* utterances for VB approximation was 31.5%. However, the average WER for the *first* utterances of all speakers was below 29% as shown in figure 7.1. This means that, the first utterances of all speakers, *on average*, happened to be "easier" to recognise than some later utterances. When the more "difficult" utterances came, the absolute WER of those utterances may increase and lead to the fluctuations in figure 7.1. From the cumulative WER curve of the unadapted SI system, the intrinsic difficulty of the utterances can be observed. Similar phenomenon can be observed in figure 7.2.

[8]In contrast to the standard robust ML adaptation technique, Bayesian adaptive inference approaches did not use any threshold because prior information is considered in the Bayesian adaptive inference. ML approach in second row of table 7.7 was viewed as a Bayesian approximation approach, hence, no threshold was set, either.

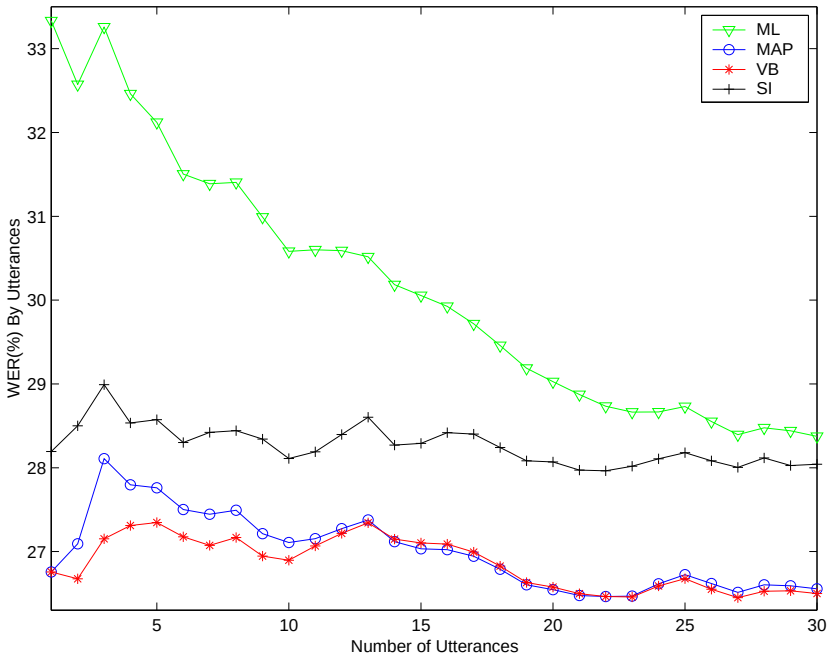


Figure 7.2 *ML and MPE 150-Best list rescoring cumulative WER (%) on* `eval03` *of incremental Bayesian adaptive inference of different number of utterances on the MLLR based MPE-SAT system. 30 utterances are shown here. The transform prior distribution was a single Gaussian.*

Bayesian Approx	ML Train		MPE Train	
	SI	SAT	SI	SAT
ML+thresh	31.2	—	27.8	—
ML	32.2	31.8	28.9	28.7
MAP	30.9	30.4	27.7	27.5
VB	30.9	30.3	27.7	27.4

Table 7.7 *150-Best list rescoring performance on* `eval03` *of incremental Bayesian adaptive inference on SI and MLLR based SAT systems using lower bound approximations. The transform prior distribution was a single Gaussian. 1 transform distribution update iteration was used for lower bound approximations.*

performance (32.8%). This is the effect of data accumulation in incremental adaptive inference, which eventually resulted in a robust transform estimate. However, this result was still 0.6% worse than the `ML+thresh` approach, which shows the ML approximation without threshold is not robust enough. Using the prior information, MAP and VB both significantly outperformed the ML approximation and the standard `ML+thresh` approach and gave about the same gain. Comparing the performance of the ML-SAT system to the ML-SI system shows that the adaptively trained system consistently and significantly outperformed the non-adaptively trained system by over 0.4% for all approximations. For MPE training, there were similar trends as the ML case. However, the gains of adaptively trained system were all reduced. For example, comparing the VB adaptive inference on the MPE-SAT system to the baseline `ML+thresh` performance on the MPE-SI system, the gain was 0.4%. In contrast, the comparable gain of the ML-SAT system was 0.9%. This gain reduction is also because a ML based transform prior distribution was used in adaptive inference in a non-discriminative way.

The above experiments were based on N-Best rescoring on a 150-Best hypothesis list. To check whether the 150-Best list included sufficient candidates or not, the best ML configuration of adaptive inference, the incremental VB adaptive inference on ML-SAT (30.3%), was tested on a 300-Best list. With twice the number of hypotheses, the performance was also 30.3%. The improvement was very slight, only 23 word errors from the 76K words in total. This shows that the majority of correct hypotheses have been included in the 150-Best list.

The adaptive inference experiments in this section were run in an *incremental* mode at speaker basis within the *N-Best rescoring (N-Best supervision)* inference framework. It is interesting to contrast these results to the discriminative adaptive training results in chapter 6. Though adaptation also performed at speaker basis, the results in chapter 6 were generated using *Viterbi* algorithm with model adaptation run in a *batch* mode with *1-Best supervision*. Hence, they are not strictly comparable to the results in table 7.7. However, due to the use of sufficient adaptation data (speaker basis), the results in chapter 6 were expected to be robust though ML approximation was used. Comparing the `eval03` performance of adaptively trained system (MLLR-SAT) in table 6.7 to the performance of the unadapted SI system in table 6.6, similar conclusions can be drawn as the above discussions on robust adaptive inference schemes (such as VB) in table 7.7. These conclusions include: adaptively trained systems significantly outperformed multi-style systems; discriminative systems significantly outperformed ML systems; the gains of adaptive training for MPE systems were smaller than the gains for ML systems. As no prior transform distribution was used in chapter 6, the reduction of the gains for MPE systems was mainly due to the way of the test transform estimation.

7.4 Summary

This chapter gives experimental results using Bayesian adaptive inference on adaptively trained systems. Initial experiments examined the performance of adaptively trained systems with very limited adaptation data. The homogeneous block was a single utterance in these experiments.

For a 2-cluster CAT system, sampling approach was used to generate a bound of the adaptive inference performance. Compared to the performance of sampling approach, other approximations gave similar performance. This is because the number of transform parameters required for CAT is very small. MLLR based SAT systems were then examined. With considerably larger number of transform parameters and limited adaptation data, the FI assumption and the ML approximation showed degraded performance. In contrast, VB significantly outperformed all the other approximation approaches in the utterance level adaptive inference. The robustness of Bayesian adaptive inference was effectively illustrated. The tightness of lower bound was also investigated. The number of update iterations was shown to be a good control on the tightness of the lower bound, and hence affect the adaptive inference performance. Optimising the lower bound for every possible hypothesis was also demonstrated to give a tighter lower bound and better performance (N-Best supervision) than using only 1-Best supervision to update the transform distribution. In addition to utterance level adaptive inference in batch mode, incremental Bayesian adaptive inference at side level was investigated. At the first few utterances, as the most robust approximation approach, VB outperformed all the other schemes as in utterance level adaptive inference. As more adaptation data became available, the performance of VB and MAP became closer. This is expected as after a few utterances, the variational transform posterior distribution can be effectively approximated using a Dirac delta function with the MAP estimate as the hyper-parameters.

The Bayesian adaptive inference was performed on both adaptively trained and non-adaptively trained systems. The results showed that by using consistent adaptive HMM assumption in both training and inference, significant gains were obtained compared to using the standard HMM assumption in both stages. Even when the adaptive HMM assumption is used in inference for both adaptively trained systems and non-adaptive systems, adaptively trained systems got consistent gains over systems without adaptive training. In addition to the ML trained systems, MPE trained systems were also examined. Those MPE systems significantly outperformed the ML trained systems. Comparing adaptively trained MPE systems to non-adaptively trained MPE systems, similar gains were obtained as in the ML case. However, the MPE gains were smaller (about half) than the ML gains of adaptive training. This is due to the application of an ML based transform prior in a non-discriminative way distribution in Bayesian adaptive inference.

8

Conclusion and Future Work

Adaptive training is a powerful approach to build systems on non-homogeneous training data. The concept and the standard ML framework of adaptation and adaptive training have been reviewed in chapter 3. The main contributions of this work have been described in chapter 4 and chapter 5. The first contribution, described in chapter 4, is to use a discriminative criterion in adaptive training to increase the discrimination ability of the parameter estimates. A novel discriminative cluster adaptive training (CAT) technique which allows rapid adaptation to be performed on discriminatively trained models is proposed. This contribution will be summarised in section 8.1. The second contribution, described in chapter 5, provides a consistent Bayesian framework for adaptive training and adaptive inference. This framework allows the adaptively trained systems to be directly used in inference and addresses the robustness issue when there is only limited adaptation data. This will be summarised in section 8.2. Possible future research directions are discussed section 8.3.

8.1 Discriminative Cluster Adaptive Training

Adaptive training is normally described using the ML criterion. It yields two sets of parameter estimates: a canonical model and a set of transforms. However, discriminative training has been used to achieve good performance for most state-of-the-art systems. Linear transform based discriminative adaptive training has been previously investigated. An alternative form of adaptive training suitable for rapid adaptation on discriminative systems, *discriminative cluster adaptive training* (CAT), is investigated in detail in chapter 4 with the minimum phone error (MPE) criterion. This technique allows rapid adaptation to be performed on discriminatively trained systems.

As the canonical model in CAT is a multiple-cluster model, there are a number of changes to be made in optimising the weak-sense auxiliary function of the MPE criterion. Modified versions of the I-smoothing distribution and the smoothing function are derived in chapter 4. As the multiple-cluster model has a larger amount of parameters than the standard model, the choice of the prior in I-smoothing distribution is more important. Various types of priors, multiple or

single cluster, dynamic or static, are discussed. The selection of the smoothing constant also becomes complicated for the multiple-cluster model. Except for two or three cluster models, there is no closed-form solution to work out the smoothing constant. However, the performance is not sensitive to the exact value of the smoothing constant, provided that a sensible approximation is used. An MPE update of interpolation weight vectors is also derived in this work to form a complete discriminative cluster adaptive training framework. The form of discriminative cluster adaptive training is applicable to other multiple-cluster systems, such as eigenvoices. It is also used to build a more complex multiple-cluster adaptively trained system with structured transforms, which combine both interpolation weights and constrained linear transforms to model highly non-homogeneous training data.

Due to the lack of a correct transcription, it is not possible to discriminatively update transforms during unsupervised adaptation. Therefore, to keep a consistent transform update criterion for both training and adaptation, a simplified discriminative adaptive training strategy is used in the experiments of this work. In this strategy, only the canonical model is discriminatively updated given the ML estimated transforms. In recognition, the ML criterion is used to estimate the test domain specific transforms given the discriminative canonical model. Experiments performed on a state-of-the-art conversational telephone speech task are reported in chapter 6. Discriminative cluster adaptive training not only yielded significant gains over the ML-CAT systems, but also consistently outperformed the non-adaptive MPE systems (MPE-SI) and state-of-the-art cluster-dependent MPE systems (GD MPE-MAP). Using the more complex structured transforms showed significant gains over both non-adaptively trained and the other comparable adaptively trained MPE systems.

8.2 Bayesian Adaptive Training and Adaptive Inference

Standard ML adaptive training assumes that both canonical model and transform parameters are deterministic; a two-step adaptation/recognition process is performed after training. In this framework, the canonical model can not be directly used in unsupervised adaptation due to the unavailability of the test domain specific transforms. To address the problem, this work presents a consistent Bayesian framework for both adaptive training and adaptive inference in chapter 5. In this framework, an adaptive HMM assumption is used for both training and inference, where the two sets of parameters, the canonical model and the transform, are assumed to be random variables and marginalised out to calculate the likelihood of training data. Prior distributions of both sets of parameters can be estimated by using empirical Bayes in training. By controlling the model complexity to reflect the amount of training data, the use of the ML estimate of the canonical model can be justified within this framework. In contrast, the transform prior distribution is a non-point distribution. The transform prior distribution is then used in unsupervised mode Bayesian adaptive inference. In contrast to the standard two-step adaptation/recognition process, the Bayesian framework motivates an integrated adaptive inference process. The key problem in this process is to calculate the marginal likelihood of the observation sequence given

every possible hypothesis by integrating out the transform prior distribution. This motivates the use of N-Best list rescoring as the inference approach in this work. The Bayesian framework can also be extended to discriminative criteria. In this work, the *complete* discriminative Bayesian adaptive training and adaptive inference is briefly discussed but not implemented. Instead, the simplified discriminative adaptive training strategy described in chapter 4 is used. Hence, during inference, the likelihood based Bayesian approximations are applicable to the discriminative canonical model.

As Bayesian marginalisation over the transform distribution is intractable, approximations are required. Lower bound approximations, point estimates and a new variational Bayesian (VB) approximation, are investigated. In addition to this, direct approximations, sampling and the frame-independent assumption, are also discussed. In the situation where there is only limited adaptation data, the standard adaptation framework has a limitation that the ML estimate of transform is unreliable. This limitation is effectively addressed by using an appropriate approximation approach within the Bayesian framework. The Bayesian adaptive inference is first discussed in batch mode and then extended to incremental mode in this work. With lower bound approximations, an efficient recursive adaptive inference algorithm is developed. The general approximation approaches are applied to two specific mean based transforms, interpolation weights and mean based linear transforms.

Experiments on adaptively trained system with both ML and MPE criteria were performed on the same conversational telephone speech task as in chapter 6. The results in chapter 7 show that using the adaptive HMM assumption in both training and inference can obtain significant gains compared to using the standard HMM assumption in both stages. Regarding the different approximation approaches, they yielded similar performance for CAT systems due to the small amount of transform parameters. In contrast, for speaker adaptive training (SAT) systems with mean transforms, various approximation schemes showed great differences in performance. With very limited adaptation data, the VB approach significantly outperformed all the other approximation approaches on average. Hence, the robustness of Bayesian adaptive inference was effectively illustrated. As more data became available in incremental adaptive inference, the performance of the VB and the maximum a posteriori (MAP) approaches became closer to each other. This is because after a few utterances, a Dirac delta function, with the MAP point estimate being the hyper-parameters, becomes a reasonable approximation to the variational transform posterior distribution. However, the various gains for the MPE trained systems were smaller than those of the ML trained systems. The reason is believed to be that the transform prior distribution was obtained from ML estimated transforms and was applied in a non-discriminative way during adaptive inference.

8.3 Future work

The research on adaptive training and adaptation for large vocabulary continuous speech recognition (LVCSR) systems may be further carried out in a number of directions:

- **More clusters for discriminative CAT:** 2-cluster gender initialised CAT systems were used in the 28 component experiments in this work. More clusters are expected to yield improved results as in the development experiments in chapter 6. The large memory requirement and the risk of overtraining due to the increased number of parameters need to be addressed. One possible direction is to try transform based CAT [36] rather than the model based CAT in this work.

- **Structured transforms:** CAT interpolation weights combined with constrained MLLR (CMLLR) or MLLR have been used as the structured transform (ST) in this work. There are other possible combinations to be investigated within the adaptive training framework. For example, model-based joint uncertainty compensation, which may be viewed as a linear transform of the features with a bias on the model covariances, may be combined with constrained MLLR in compensating for noise [75]. The power of ST based systems may also be illustrated by applying them in applications where data is more non-homogeneous than the conversational telephone speech task considered in this work. To train a meaningful ST based system, more accurate information about the acoustic conditions of the training data is required.

- **Conjugate prior for CMLLR transforms:** As indicated in section 5.5.3, the constrained linear transform has not been implemented within the Bayesian adaptive training and adaptive inference framework due to the lack of a conjugate prior. Future research may be carried out to investigate this problem.

- **Efficient Bayesian approximations for Viterbi decoding:** Due to the nature of Bayesian adaptive inference, N-Best list rescoring is used as the inference approach in this work. Though it strictly ties in with the proposed Bayesian framework, it is not practical for state-of-the-art LVCSR systems due to the high computational load. As indicated in section 5.1.2, traditional Viterbi decoding can not be directly used within the Bayesian framework. Thus, it is interesting to investigate further approximations that may allow Viterbi-like recursive formulae to be derived for the various Bayesian approximation approaches proposed in this work. Such formulae may significantly reduce the computational load and allow Bayesian adaptive inference to be used for state-of-the-art LVCSR systems.

- **Complete discriminative Bayesian adaptive training and adaptive inference:** In this work, the gain of discriminative adaptive training is limited due to the ML based transform prior distribution and the way of using it in adaptive inference. To implement complete Bayesian discriminative adaptive training, the form of an appropriate conjugate prior distribution to discriminative criteria needs to be investigated. In addition, Bayesian approximations, such as variational Bayes, for discriminative criteria also need to be studied in the future.

A

Smoothing Functions in Discriminative Training

The generic form of smoothing function for Gaussian parameters was introduced in [86, 102], which can be written as

$$\mathcal{S}(\mathcal{M};\hat{\mathcal{M}}) = \sum_m D_m \int_{\mathbf{o}} p(\mathbf{o}|m,\hat{\mathcal{M}}) \log p(\mathbf{o}|m,\mathcal{M})\, d\mathbf{o} \tag{A.1}$$

where D_m is a component-specific constant to control convergence, $p(\mathbf{o}|m,\mathcal{M})$ is the Gaussian distribution for component m given the model parameters $\mathcal{M}$, $\hat{\mathcal{M}}$ is the current model parameter set. Recall the KL distance between $p(\mathbf{o}|m,\hat{\mathcal{M}})$ and $p(\mathbf{o}|m,\mathcal{M})$

$$\mathtt{KL}\left(p(\mathbf{o}|m,\hat{\mathcal{M}})||p(\mathbf{o}|m,\mathcal{M})\right) = \int_{\mathbf{o}} p(\mathbf{o}|m,\hat{\mathcal{M}}) \log \frac{p(\mathbf{o}|m,\hat{\mathcal{M}})}{p(\mathbf{o}|m,\mathcal{M})}\, d\mathbf{o} \tag{A.2}$$

The KL distance is non-negative, which means, for any $\mathcal{M}$,

$$\int_{\mathbf{o}} p(\mathbf{o}|m,\hat{\mathcal{M}}) \log p(\mathbf{o}|m,\hat{\mathcal{M}})\, d\mathbf{o} \geq \int_{\mathbf{o}} p(\mathbf{o}|m,\hat{\mathcal{M}}) \log p(\mathbf{o}|m,\mathcal{M})\, d\mathbf{o} \tag{A.3}$$

Inequation (A.3) shows that the maximum of equation (A.1) is at the current model parameters $\hat{\mathcal{M}}$. As $\mathcal{S}(\mathcal{M};\hat{\mathcal{M}})$ is usually a smooth or differentiable function with respect to $\mathcal{M}$, at the point of $\hat{\mathcal{M}}$, the gradient of the smoothing function in equation (A.1) should be zero, i.e., it satisfies the constraint equation (2.59) as below

$$\left.\frac{\partial \mathcal{S}(\mathcal{M};\hat{\mathcal{M}})}{\partial \mathcal{M}}\right|_{\hat{\mathcal{M}}} = 0 \tag{A.4}$$

In adaptive training, though there are two sets of parameters, the canonical model and the transforms, the generic form of smoothing function is similar to equation (A.1). Here, it is defined at acoustic condition level as below [60]

$$\mathcal{S}(\mathcal{M},\boldsymbol{\mathcal{T}};\hat{\mathcal{M}},\hat{\boldsymbol{\mathcal{T}}}) = \sum_{s,m} \nu_m^{(s)} D_m \int_{\mathbf{o}} p(\mathbf{o}|m,\hat{\mathcal{M}},\hat{\mathcal{T}}^{(s)}) \log p(\mathbf{o}|m,\mathcal{M},\mathcal{T}^{(s)})\, d\mathbf{o} \tag{A.5}$$

where s denotes index of acoustic condition, $\hat{\mathcal{M}}$ is the current canonical model and $\hat{\boldsymbol{\mathcal{T}}} = \{\hat{\mathcal{T}}^{(1)},\cdots,\hat{\mathcal{T}}^{(s)}\}$ is the set of current transforms, $\nu_m^{(s)}$ is the parameter introduced in this work

to control reflect the proportions of data for the particular component of an acoustic condition, defined as equation (4.6). It should be noted that the update of the canonical model and the set of transforms interleaves in adaptive training. This means when estimating the canonical model parameters $\mathcal{M}$, $\boldsymbol{\mathcal{T}} = \hat{\boldsymbol{\mathcal{T}}}$ and vice versa. With similar derivation using the KL distance, it is trivial to show that each term of an individual Gaussian component satisfies the constraint of gradient being zero at the current sets of parameters. As $\nu_m^{(s)}$ is just a multiplier, for any value, the whole smoothing function, equation (A.5) will satisfy

$$\left.\frac{\partial \mathcal{S}(\mathcal{M}, \boldsymbol{\mathcal{T}}; \hat{\mathcal{M}}, \hat{\boldsymbol{\mathcal{T}}})}{\partial \mathcal{M}}\right|_{\hat{\mathcal{M}}, \hat{\boldsymbol{\mathcal{T}}}} = 0 \qquad \left.\frac{\partial \mathcal{S}(\mathcal{M}, \boldsymbol{\mathcal{T}}; \hat{\mathcal{M}}, \hat{\boldsymbol{\mathcal{T}}})}{\partial \boldsymbol{\mathcal{T}}}\right|_{\hat{\mathcal{M}}, \hat{\boldsymbol{\mathcal{T}}}} = 0 \tag{A.6}$$

Given the generic smoothing function in equation (A.1) and equation (A.5), the exact form for updating a particular set of parameters may be obtained as below.

1. Smoothing function for standard Gaussian parameters

In the standard HMMs, the Gaussian distribution for component m is

$$p(\mathbf{o}|m, \mathcal{M}) = \mathcal{N}(\mathbf{o}; \boldsymbol{\mu}^{(m)}, \boldsymbol{\Sigma}^{(m)}) \tag{A.7}$$

Using the Gaussian distribution in equation (A.1) yields

$$\begin{aligned}
\mathcal{S}(\mathcal{M}; \hat{\mathcal{M}}) &= \sum_m D_m \int_{\mathbf{o}} \mathcal{N}(\mathbf{o}; \hat{\boldsymbol{\mu}}_{\mathsf{c}}^{(m)}, \hat{\boldsymbol{\Sigma}}_{\mathsf{c}}^{(m)}) \log \mathcal{N}(\mathbf{o}; \boldsymbol{\mu}^{(m)}, \boldsymbol{\Sigma}^{(m)}) \, d\mathbf{o} \\
&= K - \sum_m \frac{D_m}{2} \int_{\mathbf{o}} \mathcal{N}(\mathbf{o}; \hat{\boldsymbol{\mu}}_{\mathsf{c}}^{(m)}, \hat{\boldsymbol{\Sigma}}_{\mathsf{c}}^{(m)}) \left\{ \log |\boldsymbol{\Sigma}^{(m)}| + \left(\mathbf{o} - \boldsymbol{\mu}^{(m)}\right)^T \boldsymbol{\Sigma}^{(m)-1} \left(\mathbf{o} - \boldsymbol{\mu}^{(m)}\right) \right\} d\mathbf{o} \\
&= K - \sum_m \frac{D_m}{2} \left\{ \log |\boldsymbol{\Sigma}^{(m)}| + \mathrm{tr}\left((\hat{\boldsymbol{\Sigma}}_{\mathsf{c}}^{(m)} + \hat{\boldsymbol{\mu}}_{\mathsf{c}}^{(m)} \hat{\boldsymbol{\mu}}_{\mathsf{c}}^{(m)T}) \boldsymbol{\Sigma}^{(m)-1} \right) \right. \\
&\quad \left. - 2\boldsymbol{\mu}^{(m)T} \boldsymbol{\Sigma}^{(m)-1} \hat{\boldsymbol{\mu}}_{\mathsf{c}}^{(m)} + \boldsymbol{\mu}^{(m)T} \boldsymbol{\Sigma}^{(m)-1} \boldsymbol{\mu}^{(m)} \right\}
\end{aligned} \tag{A.8}$$

where K is a constant independent of the parameters $\boldsymbol{\mu}^{(m)}$ and $\boldsymbol{\Sigma}^{(m)}$, $\hat{\boldsymbol{\mu}}_{\mathsf{c}}^{(m)}$ and $\hat{\boldsymbol{\Sigma}}_{\mathsf{c}}^{(m)}$ are the current Gaussian parameters. Ignoring the constant K leads to the standard smoothing function in equation (2.61).

2. Smoothing function for Gaussian parameters in adaptive training with mean transforms

In adaptive training with mean transforms, the Gaussian distribution for component m associated with acoustic condition s is expressed as

$$p(\mathbf{o}|m, \mathcal{M}) = \mathcal{N}(\mathbf{o}; \hat{\boldsymbol{\mu}}^{(sm)}, \boldsymbol{\Sigma}^{(m)}) \tag{A.9}$$

where

$$\hat{\boldsymbol{\mu}}^{(sm)} = \mathbf{A}^{(sr_m)} \boldsymbol{\mu}^{(m)} + \mathbf{b}^{(sr_m)} = \mathbf{W}^{(sr_m)} \boldsymbol{\xi}^{(m)} \tag{A.10}$$

where r_m is the regression base class, or Gaussian group, that the Gaussian component m belongs to, $\hat{\boldsymbol{\mu}}^{(sm)}$ is the adapted mean of component m to acoustic condition s, $\boldsymbol{\xi}^{(m)} = [\boldsymbol{\mu}^{(m)T} \;\; 1]^T$ is the extended mean vector, and $\mathbf{W}^{(sr)} = [\mathbf{A}^{(sr)} \;\; \mathbf{b}^{(sr)}]$ is the extended linear transform associated with acoustic condition s and regression base class r. Using the Gaussian distribution in equation (A.5) and considering only the mean update yields

$$\begin{aligned}
\mathcal{S}(\mathcal{M};\hat{\mathcal{M}},\hat{\boldsymbol{\mathcal{T}}}) &= \sum_{s,m} D_m \nu_m^{(s)} \int_{\mathbf{o}} \mathcal{N}(\mathbf{o};\mathbf{W}^{(sr_m)}\hat{\boldsymbol{\xi}}_{\mathsf{c}}^{(m)},\boldsymbol{\Sigma}^{(m)}) \log \mathcal{N}(\mathbf{o};\mathbf{W}^{(sr_m)}\boldsymbol{\xi}^{(m)},\boldsymbol{\Sigma}^{(m)})\, d\mathbf{o} \\
&= K - \frac{1}{2}\sum_{s,m} D_m \nu_m^{(s)} \int_{\mathbf{o}} \mathcal{N}(\mathbf{o};\mathbf{W}^{(sr_m)}\hat{\boldsymbol{\xi}}_{\mathsf{c}}^{(m)},\boldsymbol{\Sigma}^{(m)}) \times \\
&\quad \left(\mathbf{o} - \mathbf{W}^{(sr_m)}\boldsymbol{\xi}^{(m)}\right)^T \boldsymbol{\Sigma}^{(m)-1} \left(\mathbf{o} - \mathbf{W}^{(sr_m)}\boldsymbol{\xi}^{(m)}\right) d\mathbf{o} \\
&= K - \frac{1}{2}\sum_{s,m} D_m \nu_m^{(s)} \left(\boldsymbol{\xi}^{(m)} - \hat{\boldsymbol{\xi}}_{\mathsf{c}}^{(m)}\right)^T \mathbf{W}^{(sr_m)T} \boldsymbol{\Sigma}^{(m)-1} \times \\
&\quad \mathbf{W}^{(sr_m)} \left(\boldsymbol{\xi}^{(m)} - \hat{\boldsymbol{\xi}}_{\mathsf{c}}^{(m)}\right)
\end{aligned} \tag{A.11}$$

Ignoring the constant K, the smoothing function in equation (4.5) in section 4.2.1 is obtained.

As indicated in section 4.2.1, the covariance update is separate from the mean update. Therefore, when updating covariance, $\boldsymbol{\xi}^{(m)} = \hat{\boldsymbol{\xi}}_{\mathsf{c}}^{(m)}$. The smoothing function for covariance update is then expressed as[1]

$$\begin{aligned}
\mathcal{S}(\mathcal{M};\hat{\mathcal{M}},\hat{\boldsymbol{\mathcal{T}}}) &= K - \frac{1}{2}\sum_{s,m} D_m \nu_m^{(s)} \int_{\mathbf{o}} \mathcal{N}(\mathbf{o};\mathbf{W}^{(sr_m)}\boldsymbol{\xi}^{(m)},\boldsymbol{\Sigma}_{\mathsf{c}}^{(m)}) \Big\{ \log|\boldsymbol{\Sigma}^{(m)}| + \\
&\quad \left(\mathbf{o} - \mathbf{W}^{(sr_m)}\boldsymbol{\xi}^{(m)}\right)^T \boldsymbol{\Sigma}^{(m)-1} \left(\mathbf{o} - \mathbf{W}^{(sr_m)}\boldsymbol{\xi}^{(m)}\right) \Big\} d\mathbf{o} \\
&= K - \frac{1}{2}\sum_{s,m} D_m \nu_m^{(s)} \left\{ \log|\boldsymbol{\Sigma}^{(m)}| + \mathrm{tr}\left(\hat{\boldsymbol{\Sigma}}_{\mathsf{c}}^{(m)}\boldsymbol{\Sigma}^{(m)-1}\right)\right\} \\
&= K - \frac{1}{2}\sum_{m} D_m \left\{ \log|\boldsymbol{\Sigma}^{(m)}| + \mathrm{tr}\left(\hat{\boldsymbol{\Sigma}}_{\mathsf{c}}^{(m)}\boldsymbol{\Sigma}^{(m)-1}\right)\right\}
\end{aligned} \tag{A.12}$$

where $\hat{\boldsymbol{\Sigma}}_{\mathsf{c}}^{(m)}$ is the current covariance parameter. Ignoring the constant K, the smoothing function for covariance update, equation (4.26), is obtained.

3. Smoothing function for cluster adaptive training

In cluster adaptive training, the Gaussian distribution is also expressed as equation (A.9). Here the adapted mean $\hat{\boldsymbol{\mu}}^{(sm)}$ is an interpolation between different clusters

$$\hat{\boldsymbol{\mu}}^{(sm)} = \mathbf{M}^{(m)}\boldsymbol{\lambda}^{(sr_m)} \tag{A.13}$$

where $\mathbf{M}$ is the cluster mean matrix, $\boldsymbol{\lambda}$ is the interpolation vector. Using this Gaussian distribution in equation (A.5) and considering the update of the multiple-cluster model parameters

[1] Note that covariance is independent of acoustic conditions.

yields

$$
\begin{aligned}
\mathcal{S}(\mathcal{M};\hat{\mathcal{M}},\hat{\boldsymbol{\mathcal{T}}}) &= \sum_{s,m}\nu_m^{(s)}D_m\int_{\mathbf{o}}\mathcal{N}(\mathbf{o};\mathbf{M}^{(m)}\boldsymbol{\lambda}^{(sr_m)},\boldsymbol{\Sigma}_{\mathsf{c}}^{(m)})\log\mathcal{N}(\mathbf{o};\hat{\mathbf{M}}_{\mathsf{c}}^{(m)}\boldsymbol{\lambda}^{(sr_m)},\boldsymbol{\Sigma}^{(m)})\,d\mathbf{o}\\
&= K-\sum_{m,s}\frac{D_m\nu_m^{(s)}}{2}\int_{\mathbf{o}}\mathcal{N}(\mathbf{o};\hat{\mathbf{M}}_{\mathsf{c}}^{(m)}\boldsymbol{\lambda}^{(sr_m)},\boldsymbol{\Sigma}_{\mathsf{c}}^{(m)})\Big\{\log|\boldsymbol{\Sigma}^{(m)}|+\\
&\quad\left(\mathbf{o}-\mathbf{M}^{(m)}\boldsymbol{\lambda}^{(sr_m)}\right)^T\boldsymbol{\Sigma}^{(m)-1}\left(\mathbf{o}-\mathbf{M}^{(m)}\boldsymbol{\lambda}^{(sr_m)}\right)\Big\}\,d\mathbf{o}\\
&= K-\sum_{m,s}\frac{D_m\nu_m^{(s)}}{2}\Big\{\log|\boldsymbol{\Sigma}^{(m)}|+\mathrm{tr}(\hat{\boldsymbol{\Sigma}}_{\mathsf{c}}^{(m)}\boldsymbol{\Sigma}^{(m)-1})\\
&\quad+\boldsymbol{\lambda}^{(sr_m)T}\left(\mathbf{M}^{(m)}-\hat{\mathbf{M}}_{\mathsf{c}}^{(m)}\right)^T\boldsymbol{\Sigma}^{(m)-1}\left(\mathbf{M}^{(m)}-\hat{\mathbf{M}}_{\mathsf{c}}^{(m)}\right)\boldsymbol{\lambda}^{(sr_m)}\Big\}
\end{aligned}
\tag{A.14}
$$

where $\hat{\mathbf{M}}_{\mathsf{c}}^{(m)}$ and $\boldsymbol{\Sigma}_{\mathsf{c}}^{(m)}$ are current multiple-cluster model parameters. Ignoring the constant K yields the form of the smoothing function in section 4.3.2.1, i.e., equation (4.38).

As the interpolation weight vector is estimated for each acoustic condition, there is no need to use the term $\nu_m^{(s)}$. Then, the generic smoothing function in equation (A.5) can be rewritten for updating weights for a particular acoustic condition s

$$
\begin{aligned}
\mathcal{S}(\mathcal{T}^{(s)};\hat{\mathcal{M}},\hat{\mathcal{T}}^{(s)}) &= \sum_m D_m\int_{\mathbf{o}}\mathcal{N}(\mathbf{o};\mathbf{M}^{(m)}\hat{\boldsymbol{\lambda}}_{\mathsf{c}}^{(sr_m)},\boldsymbol{\Sigma}^{(m)})\log\mathcal{N}(\mathbf{o};\mathbf{M}^{(m)}\boldsymbol{\lambda}^{(sr_m)},\boldsymbol{\Sigma}^{(m)})\,d\mathbf{o}\\
&= K-\sum_m\frac{D_m}{2}\int_{\mathbf{o}}\mathcal{N}(\mathbf{o};\mathbf{M}^{(m)}\hat{\boldsymbol{\lambda}}_{\mathsf{c}}^{(sr_m)},\boldsymbol{\Sigma}^{(m)})\times\\
&\quad\left(\mathbf{o}-\mathbf{M}^{(m)}\boldsymbol{\lambda}^{(sr_m)}\right)^T\boldsymbol{\Sigma}^{(m)-1}\left(\mathbf{o}-\mathbf{M}^{(m)}\boldsymbol{\lambda}^{(sr_m)}\right)\,d\mathbf{o}\\
&= K-\sum_m\frac{D_m}{2}\Big\{\boldsymbol{\lambda}^{(sr_m)T}\mathbf{M}^{(m)T}\boldsymbol{\Sigma}^{(m)-1}\mathbf{M}^{(m)}\boldsymbol{\lambda}^{(sr_m)}\\
&\quad-2\boldsymbol{\lambda}^{(sr_m)T}\mathbf{M}^{(m)T}\boldsymbol{\Sigma}^{(m)-1}\mathbf{M}^{(m)}\hat{\boldsymbol{\lambda}}_{\mathsf{c}}^{(sr_m)}\Big\}
\end{aligned}
\tag{A.15}
$$

where $\hat{\boldsymbol{\lambda}}_{\mathsf{c}}^{(sr_m)}$ is the current weight vector. Ignoring the constant K yields the equation (4.72) in section 4.3.2.4.

B

Maximum a Posteriori (MAP) Estimate of Multiple-Cluster Model Parameters

To obtain a multiple-cluster MAP estimate, a prior parameter distribution is required. One possible form is again the Normal-Wishart distribution in equation (4.43). However, here parameters from a single-cluster model are used as the prior [1]. Ignoring constant terms

$$\begin{aligned}\log p(\mathcal{M}|\Phi_{\texttt{MAP}}) &= -\frac{\tau^{\texttt{MAP}}}{2}\sum_{s,m}\tilde{\nu}_m^{(s)}\Big\{\log|\boldsymbol{\Sigma}^{(m)}| + \text{tr}(\tilde{\boldsymbol{\Sigma}}_{\texttt{MAP}}^{(m)}\boldsymbol{\Sigma}^{(m)-1}) \\ &\quad + \Big(\mathbf{M}^{(m)}\boldsymbol{\lambda}^{(sr_m)} - \tilde{\boldsymbol{\mu}}_{\texttt{MAP}}^{(m)}\Big)^T\boldsymbol{\Sigma}^{(m)-1}\Big(\mathbf{M}^{(m)}\boldsymbol{\lambda}^{(sr_m)} - \tilde{\boldsymbol{\mu}}_{\texttt{MAP}}^{(m)}\Big)\Big\} \end{aligned} \tag{B.1}$$

where $\tilde{\nu}_m^{(s)}$ is similar to equation (4.6), but it is defined by ML occupancy. $\Phi_{\texttt{MAP}} = \{\tau^{\texttt{MAP}}, \tilde{\boldsymbol{\mu}}_{\texttt{MAP}}^{(m)}, \tilde{\boldsymbol{\Sigma}}_{\texttt{MAP}}^{(m)}\}$ is the hyper-parameters of the MAP prior distribution $\log p(\mathcal{M}|\Phi_{\texttt{MAP}})$. $\tau^{\texttt{MAP}}$ is to control the impact of the robust model parameters, $\tilde{\boldsymbol{\mu}}_{\texttt{MAP}}^{(m)}$ and $\tilde{\boldsymbol{\Sigma}}_{\texttt{MAP}}^{(m)}$ are the *single-cluster* prior parameters. Note that the MAP distribution in equation (B.1) is a separate distribution from the I-smoothing distribution, though the form is similar.

With the above MAP prior distribution, a new auxiliary function may be derived by adding the additional prior distribution term to the ML auxiliary function

$$\begin{aligned}\mathcal{Q}_{\texttt{MAP}}(\mathcal{M};\hat{\mathcal{M}},\hat{\boldsymbol{\mathcal{T}}}) &= \log p(\mathcal{M}|\Phi_{\texttt{MAP}}) - \frac{1}{2}\sum_{s,m,t}\gamma_m^{\texttt{ML}}(t)\Big\{\log\Big|\boldsymbol{\Sigma}^{(m)}\Big| \\ &\quad + \Big(\mathbf{o}_t^{(s)} - \mathbf{M}^{(m)}\boldsymbol{\lambda}^{(sr_m)}\Big)^T\boldsymbol{\Sigma}^{(m)-1}\Big(\mathbf{o}_t^{(s)} - \mathbf{M}^{(m)}\boldsymbol{\lambda}^{(sr_m)}\Big)\Big\} \end{aligned} \tag{B.2}$$

where $\hat{\boldsymbol{\mathcal{T}}}$ is now the set of interpolation weights consisting of $\boldsymbol{\lambda}^{(sr)}$, $\gamma_m^{\texttt{ML}}(t)$ is the posterior occupancy of Gaussian component m calculated given the current CAT model and the weights estimate. By differentiating the above auxiliary function with respect to the mean and covariance and setting it to zero, the MAP estimate can be obtained by

$$\hat{\mathbf{M}}^{(m)T} = \mathbf{G}_{\texttt{MAP}}^{(m)-1}\mathbf{K}_{\texttt{MAP}}^{(m)} \tag{B.3}$$

$$\hat{\boldsymbol{\Sigma}}^{(m)} = \text{diag}\Big(\frac{\mathbf{L}_{\texttt{MAP}}^{(m)} - \hat{\mathbf{M}}^{(m)}\mathbf{K}_{\texttt{MAP}}^{(m)}}{\gamma_m^{\texttt{MAP}}}\Big) \tag{B.4}$$

[1] The choice is consistent with [42], where a robust set of single-cluster model parameters is obtained by using some ad-hoc approach such as speaker-independent training.

where the sufficient statistics are

$$\gamma_m^{\texttt{MAP}} = \gamma_m^{\texttt{ML}} + \tau^{\texttt{MAP}} \tag{B.5}$$

$$\mathbf{G}_{\texttt{MAP}}^{(m)} = \mathbf{G}_{\texttt{ML}}^{(m)} + \tau^{\texttt{MAP}} \sum_s \tilde{\nu}_m^{(s)} \boldsymbol{\lambda}^{(sr_m)} \boldsymbol{\lambda}^{(sr_m)T} \tag{B.6}$$

$$\mathbf{K}_{\texttt{MAP}}^{(m)} = \mathbf{K}_{\texttt{ML}}^{(m)} + \tau^{\texttt{MAP}} \Big(\sum_s \tilde{\nu}_m^{(s)} \boldsymbol{\lambda}^{(sr_m)} \Big) \tilde{\boldsymbol{\mu}}_{\texttt{MAP}}^{(m)T} \tag{B.7}$$

$$\mathbf{L}_{\texttt{MAP}}^{(m)} = \mathbf{L}_{\texttt{ML}}^{(m)} + \tau^{\texttt{MAP}} \Big(\tilde{\boldsymbol{\mu}}_{\texttt{MAP}}^{(m)} \tilde{\boldsymbol{\mu}}_{\texttt{MAP}}^{(m)T} + \tilde{\boldsymbol{\Sigma}}_{\texttt{MAP}}^{(m)} \Big) \tag{B.8}$$

where $\gamma_m^{\texttt{ML}}$, $\mathbf{G}_{\texttt{ML}}^{(m)}$, $\mathbf{K}_{\texttt{ML}}^{(m)}$ and $\mathbf{L}_{\texttt{ML}}^{(m)}$ are ML the statistics in equations (3.46) to (3.49).

The above describes how to get the MAP estimate of the multiple-cluster model. The MAP estimate can then be used as the prior in I-smoothing distribution. Substituting the multiple-cluster MAP prior for multiple-cluster ML prior, the statistics for the I-smoothing distribution, equations (4.58) to (4.60), become [2]

$$\mathbf{G}_{\texttt{p}}^{(m)} = \frac{1}{\gamma_m^{\texttt{MAP}}} \mathbf{G}_{\texttt{MAP}}^{(m)} \tag{B.9}$$

$$\mathbf{K}_{\texttt{p}}^{(m)} = \frac{1}{\gamma_m^{\texttt{MAP}}} \mathbf{K}_{\texttt{MAP}}^{(m)} \tag{B.10}$$

$$\mathbf{L}_{\texttt{p}}^{(m)} = \frac{1}{\gamma_m^{\texttt{MAP}}} \mathbf{L}_{\texttt{MAP}}^{(m)} \tag{B.11}$$

In this case, There are two tunable parameters controlling prior occupancy: τ^I for the whole I-smoothing distribution and $\tau^{\texttt{MAP}}$ for the MAP estimate.

[2] Note that in the MAP prior case, $\tilde{\nu}_m^{(s)}$ in the I-smoothing distribution in equation (4.43), need to be re-defined using the MAP occupancy.

C

Estimation of Hyper-Parameters of Prior Distributions in Adaptive Training

As discussed in section 5.1.1, using empirical Bayesian approach, the hyper-parameters of the canonical model prior distribution is obtained by maximising

$$p(\mathcal{O}|\mathbf{H}) = \int_{\mathcal{M}} p(\mathcal{O}|\mathbf{H},\mathcal{M})p(\mathcal{M}|\Phi)\,d\mathcal{M} \tag{C.1}$$

Introducing a variational distribution and applying Jensen's inequality yields a lower bound of equation (C.1)

$$\log p(\mathcal{O}|\mathbf{H}) \geq \left\langle \log \frac{p(\mathcal{O}|\mathbf{H},\mathcal{M})p(\mathcal{M}|\Phi)}{q(\mathcal{M})} \right\rangle_{q(\mathcal{M})} \tag{C.2}$$

where $< f(x) >_{g(x)}$ is the expectation of $f(x)$ with respect to $g(x)$, defined as equation (2.14). The above becomes equality when

$$q(\mathcal{M}) = p(\mathcal{M}|\mathcal{O},\mathbf{H}) \tag{C.3}$$

With sufficient training data and appropriate model complexity control, the canonical model posterior distribution can be effectively approximated by a Dirac delta function

$$q(\mathcal{M}) = \delta(\mathcal{M} - \hat{\mathcal{M}}) \tag{C.4}$$

where $\hat{\mathcal{M}}$ is a point estimate of canonical model. Equation (C.2) can then be re-expressed as

$$\log p(\mathcal{O}|\mathbf{H}) \geq \log p(\mathcal{O}|\mathbf{H},\hat{\mathcal{M}}) - \mathtt{KL}\left(\delta(\mathcal{M} - \hat{\mathcal{M}})||p(\mathcal{M}|\Phi)\right) \tag{C.5}$$

where the KL distance is defined in equation (5.9). As the KL distance between a Dirac delta function and any other distribution is ∞, the optimal $p(\mathcal{M}|\Phi)$ must be identical to the delta function

$$p(\mathcal{M}|\Phi) = \delta(\mathcal{M} - \hat{\mathcal{M}}) \tag{C.6}$$

Given equation (C.6), the transform prior is estimated based on a point estimate of the canonical model. Again, the hyper-parameters of the transform prior distribution is obtained by

maximising the marginal likelihood

$$\begin{aligned} p(\mathcal{O}|\mathbf{H}) &= \int_{\mathcal{M}} p(\mathcal{O}|\mathbf{H},\mathcal{M})\delta(\mathcal{M}-\hat{\mathcal{M}})\,d\mathcal{M} = p(\mathcal{O}|\mathbf{H},\hat{\mathcal{M}}) \\ &= \prod_{s=1}^{S}\int_{\mathcal{T}} p(\mathbf{O}^{(s)}|\mathcal{H}^{(s)},\hat{\mathcal{M}},\mathcal{T})p(\mathcal{T}|\phi)\,d\mathcal{T} \end{aligned} \tag{C.7}$$

where ϕ is the hyper-parameters of the transform prior distribution. Note that transform is constrained to be constant for each homogeneous block, which is the fundamental assumption of adaptive HMM. Introducing one variational transform distribution for each homogeneous block and applying Jensen's inequality yields

$$\log p(\mathcal{O}|\mathbf{H}) \geq \sum_{s=1}^{S}\left\langle \log\frac{p(\mathbf{O}^{(s)}|\mathcal{H}^{(s)},\hat{\mathcal{M}},\mathcal{T})p(\mathcal{T}|\phi)}{q^{(s)}(\mathcal{T})}\right\rangle_{q^{(s)}(\mathcal{T})} \tag{C.8}$$

With sufficient training data and appropriate complexity control, each variational transform distribution can also be approximated as a Dirac delta function with a distinct point estimate

$$q^{(s)}(\mathcal{T}) = \delta(\mathcal{T}-\hat{\mathcal{T}}^{(s)}) \tag{C.9}$$

Hence, inequality (C.8) can be re-expressed as

$$\log p(\mathcal{O}|\mathbf{H}) \geq \sum_{s=1}^{S}\left(\log p(\mathbf{O}^{(s)}|\mathcal{H}^{(s)},\hat{\mathcal{M}},\hat{\mathcal{T}}^{(s)}) + \log p(\hat{\mathcal{T}}^{(s)}|\phi) + \mathtt{H}\left(\delta(\mathcal{T}-\hat{\mathcal{T}}^{(s)})\right)\right) \tag{C.10}$$

where $\mathtt{H}(\cdot)$ is the entropy defined in equation (2.16). The entropy of a Dirac delta function is $-\infty$, hence equation (C.10) is a loose bound of $\log p(\mathcal{O}|\mathbf{H})$. Due to the multiple homogeneous blocks, there is no prior transform distribution that can compensate for the infinity entropy. However, as the entropy term is a constant, the rank ordering of $\log p(\mathcal{O}|\mathbf{H})$ is only dependent on the second term in equation (C.10) in terms of prior hyper-parameter update. Hence, the optimal hyper-parameters of transform prior, $\hat{\phi}$, can be found by

$$\hat{\phi} = \max_{\phi}\sum_{s=1}^{S}\log p(\hat{\mathcal{T}}^{(s)}|\phi) \tag{C.11}$$

The above formulae imply that the hyper-parameters are actually ML estimates of the "parameter samples". Hence. this hyper-parameter estimate is also known as a "ML-II" prior in empirical Bayesian approach [10, 96].

D

Derivations in Variational Bayes

D.1 Derivations of Using Multiple Component Prior in VB

This section gives the derivation of using a multiple-component, or mixture prior distribution in variational Bayes approach. The multiple-component prior is re-produced here

$$p(\mathcal{T}) = \sum_n c_n p(\mathcal{T}|n) \tag{D.1}$$

where $p(\mathcal{T}|n)$ is a prior component, which is valid conjugate distribution, for example, a single Gaussian distribution for MLLR or CAT and c_n is the weight of the n^{th} component. Then the marginal likelihood is given by

$$\begin{aligned} p(\mathbf{O}|\mathcal{H}) &= \sum_n c_n \int_{\mathcal{T}} p(\mathbf{O}|\mathcal{H},\mathcal{T}) p(\mathcal{T}|n)\, d\mathcal{T} \\ &= \sum_n c_n \int_{\mathcal{T}} \sum_{\boldsymbol{\theta}} p(\mathbf{O},\boldsymbol{\theta}|\mathcal{H},\mathcal{T}) p(\mathcal{T}|n)\, d\mathcal{T} \end{aligned} \tag{D.2}$$

where $\boldsymbol{\theta}$ is the hidden component sequence. Similar to the VB approach for single component prior, introducing variational distributions $q(n)$ and applying Jensen's inequality yields a lower bound

$$\begin{aligned} \log p(\mathbf{O}|\mathcal{H}) &\geq \sum_n q(n) \left(\log \frac{c_n}{q(n)} + \log \int_{\mathcal{T}} p(\mathbf{O}|\mathcal{H},\mathcal{T}) p(\mathcal{T}|n)\, d\mathcal{T} \right) \\ &= \langle \log p(\mathbf{O}|\mathcal{H},n) \rangle_{q(n)} - \mathtt{KL}(q(n)||c_n) \end{aligned} \tag{D.3}$$

where $\mathtt{KL}(\cdot||\cdot)$ is the discrete KL distance defined in equation (5.10). One VB lower bound can also be introduced for each $\log p(\mathbf{O}|\mathcal{H},n)$

$$\begin{aligned} \log p(\mathbf{O}|\mathcal{H},n) &= \log \int_{\mathcal{T}} p(\mathbf{O}|\mathcal{H},\mathcal{T}) p(\mathcal{T}|n)\, d\mathcal{T} \\ &\geq \left\langle \log \frac{p(\mathbf{O},\boldsymbol{\theta}|\mathcal{H},n) p(\mathcal{T}|n)}{q(\boldsymbol{\theta},\mathcal{T})} \right\rangle_{q(\boldsymbol{\theta},\mathcal{T})} \end{aligned} \tag{D.4}$$

Again, the variational transform distribution $q(\mathcal{T}|\mathbf{O},\mathcal{H},n)$ is assumed to be conditionally independent of the hidden component distribution $q(\boldsymbol{\theta}|\mathbf{O},\mathcal{H})$. The VB approximation for the n^{th}

prior component is expressed as

$$q(\boldsymbol{\theta}, \mathcal{T}) = q(\boldsymbol{\theta}|\mathbf{O}, \mathcal{H}) q(\mathcal{T}|\mathbf{O}, \mathcal{H}, n) \tag{D.5}$$

It is worth noting that according to equation (D.2), all components of the transform prior distribution are associated with *one* common hidden component sequence $\boldsymbol{\theta}$. It is not allowed to swap between components. Hence, in the VB approximation for each component, the variational hidden component sequence distribution $q(\boldsymbol{\theta}|\mathbf{O}, \mathcal{H})$ is independent to n, which is a consistent result from equation (D.2). With equation (D.5), let $q(\boldsymbol{\theta})$ and $q(\mathcal{T}|n)$ be brief notations for $q(\boldsymbol{\theta}|\mathbf{O}, \mathcal{H})$ and $q(\mathcal{T}|\mathbf{O}, \mathcal{H}, n)$, the lower bound in equation (D.4) can be re-expressed as an auxiliary function similar to equation (5.51)

$$\begin{aligned} \log p(\mathbf{O}|\mathcal{H}, n) &\geq \mathcal{Q}_{\mathtt{VB}}\left(q_{k+1}(\boldsymbol{\theta}), q_k(\mathcal{T}|n)\right) \\ &= \left\langle \log p(\mathbf{O}, \boldsymbol{\theta}|\mathcal{T}, \mathcal{H}, n) \right\rangle_{q_{k+1}(\boldsymbol{\theta}) q_k(\mathcal{T}|n)} + \mathtt{H}\left(q_{k+1}(\boldsymbol{\theta})\right) - \mathtt{KL}(q_k(\mathcal{T}|n)||p(\mathcal{T}|n)) \end{aligned} \tag{D.6}$$

where k is the iteration index. Assuming that the variational weight of the k^{th} iteration $q_k(n)$ has been obtained, the overall VB lower bound with mixture prior can be expressed as

$$\begin{aligned} \log p(\mathbf{O}|\mathcal{H}) &\geq \mathcal{Q}_{\mathtt{VB}}\left(q_{k+1}(\boldsymbol{\theta}), q_k(\mathcal{T})\right) = \mathcal{L}_{\mathtt{VB}}\left(q_k(\mathcal{T})\right) \\ &= \Big\langle \mathcal{Q}_{\mathtt{VB}}\big(q_{k+1}(\boldsymbol{\theta}), q_k(\mathcal{T}|n)\big) \Big\rangle_{q_k(n)} - \mathtt{KL}\left(q_k(n)||c_n\right) \end{aligned} \tag{D.7}$$

where k is the iteration number, $q(\mathcal{T})$ denotes the complete variational distribution for the mixture prior $p(\mathcal{T})$, which is also a mixture model

$$q(\mathcal{T}) = \sum_n q(n) q(\mathcal{T}|n) \tag{D.8}$$

Differentiating the overall auxiliary function equation (D.7) with respect to $q_k(\mathcal{T}|n)$ and $q_{k+1}(\boldsymbol{\theta})$ and setting it to zero leads to a VBEM algorithm similar to the single component prior case.

- **VBE step**:

$$\begin{aligned} \log q_k(\boldsymbol{\theta}) &= \sum_n q_{k-1}(n) \left\langle \log p(\mathbf{O}, \boldsymbol{\theta}|\mathcal{T}) \right\rangle_{q_{k-1}(\mathcal{T}|n)} - \log \mathcal{Z}_{\boldsymbol{\Theta}}(\mathbf{O}, \mathcal{H}) \\ &= \left\langle \log p(\mathbf{O}, \boldsymbol{\theta}|\mathcal{T}) \right\rangle_{q_{k-1}(\mathcal{T})} - \log \mathcal{Z}_{\boldsymbol{\Theta}}(\mathbf{O}, \mathcal{H}) \end{aligned} \tag{D.9}$$

  This is equivalent to equation (5.52) except that the complete mixture variational distribution $q_{k-1}(\mathcal{T})$ is used to calculate the pseudo-distribution $\tilde{p}(\mathbf{o}_t|\theta_t)$.

- **VBM step**:

$$\log q_k(\mathcal{T}|n) = \log p(\mathcal{T}|n) + \left\langle \log p(\mathbf{O}, \boldsymbol{\theta}|\mathcal{T}) \right\rangle_{q_k(\boldsymbol{\theta})} - \log \mathcal{Z}_{\mathcal{T}}^{(n)} \tag{D.10}$$

  This is similar to equation (5.56) except that the corresponding prior component $p(\mathcal{T}|n)$ is used instead of $p(\mathcal{T})$.

An additional issue introduced by using a mixture prior is the estimation of variational component weights $q_k(n)$. Directly optimising equation (D.7) with respect to $q_k(n)$ leads to an update formula requiring the calculation of the component-level lower bound $\mathcal{Q}_{\mathtt{VB}}\left(q_{k+1}(\boldsymbol{\theta}), q_k(\mathcal{T}|n)\right)$. Though in the single component case, the lower bound can be efficiently calculated using equation (5.57), there is no efficient formulae for the component-level lower bound when a mixture prior distribution is used because the hidden component sequence is shared among all components. To simplify the calculation, in the update of $q_k(n)$, different components of the transform prior are assumed to be independent of each other. Hence the hidden component sequence $\boldsymbol{\theta}$ may alter from one prior component to another, a distinct $q(\boldsymbol{\theta}|n)$ is associated with each prior component. Then equation (D.5) changes to

$$q(\boldsymbol{\theta}, \mathcal{T}) = q(\boldsymbol{\theta}|\mathbf{O}, \mathcal{H}, n) q(\mathcal{T}|\mathbf{O}, \mathcal{H}, n) \tag{D.11}$$

The number of variational component sequence distributions increases from 1 to N, where N is the number of prior components. Accordingly, an independent auxiliary function $\mathcal{Q}_{\mathtt{VB}}\left(q_{k+1}(\boldsymbol{\theta}|n), q_k(\mathcal{T}|n)\right)$ can be introduced for each $\log p(\mathbf{O}|\mathcal{H}, n)$. Equation (D.7) becomes

$$\log p(\mathbf{O}|\mathcal{H}) \geq \left\langle \mathcal{Q}_{\mathtt{VB}}\big(q_{k+1}(\boldsymbol{\theta}|n), q_k(\mathcal{T}|n)\big) \right\rangle_{q_k(n)} - \mathtt{KL}\left(q_k(n)||c_n\right) \tag{D.12}$$

The definition of $\mathcal{Q}_{\mathtt{VB}}\left(q_{k+1}(\boldsymbol{\theta}|n), q_k(\mathcal{T}|n)\right)$ is similar to equation (D.6) except that a distinct $q_{k+1}(\boldsymbol{\theta}|n)$ is used instead of the shared $q_{k+1}(\boldsymbol{\theta})$. With the above approximation, the lower bound for each component can be efficiently calculated as in the single component case

$$\mathcal{Q}_{\mathtt{VB}}\left(q_{k+1}(\boldsymbol{\theta}|n), q_k(\mathcal{T}|n)\right) = \mathcal{L}_{\mathtt{VB}}\left(q_k(\mathcal{T}|n)\right) = \log \mathcal{Z}^{(n)}_{\boldsymbol{\Theta}}(\mathbf{O}, \mathcal{H}) - \mathtt{KL}\left(q_k(\mathcal{T}|n)||p(\mathcal{T}|n)\right) \tag{D.13}$$

where the normalisation term for each prior component is calculated using forward-backward algorithm for the particular component n

$$\mathcal{Z}^{(n)}_{\boldsymbol{\Theta}}(\mathbf{O}, \mathcal{H}) = \sum_{\boldsymbol{\theta}} P(\boldsymbol{\theta}|\mathcal{H}, \mathcal{M}, n) \prod_t \tilde{p}(\mathbf{o}_t|\theta_t, n) \tag{D.14}$$

where $\tilde{p}(\mathbf{o}_t|\theta_t, n)$ is the pseudo-distribution calculated based on $q_k(\mathcal{T}|n)$. Differentiating equation (D.12) with respect to $q_k(n)$ and setting it to zero leads to the update formula of variational weights

$$q_k(n) = \frac{c_n \exp\left(\mathcal{L}_{\mathtt{VB}}\left(q_k(\mathcal{T}|n)\right)\right)}{\sum_n c_n \exp\left(\mathcal{L}_{\mathtt{VB}}\left(q_k(\mathcal{T}|n)\right)\right)} \tag{D.15}$$

So far the estimation of the complete variational distribution $q(\mathcal{T})$ is discussed. Having obtained it after the iterative learning, the value of the overall lower bound is required for inference. As indicated before, equation (D.7) is the overall lower bound. Here, an efficient formula to calculate this lower bound can be derived by using equation (D.9) in equation (D.7)

$$\begin{aligned} \mathcal{L}_{\mathtt{VB}}\left(q_k(\mathcal{T})\right) &= \mathcal{Q}_{\mathtt{VB}}\big(q_{k+1}(\boldsymbol{\theta}), q_k(\mathcal{T})\big) \\ &= \log \mathcal{Z}_{\boldsymbol{\Theta}}(\mathbf{O}, \mathcal{H}) - \sum_n q_k(n) \mathtt{KL}\left(q_k(\mathcal{T}|n)||p(\mathcal{T}|n)\right) - \mathtt{KL}\left(q_k(n)||c_n\right) \end{aligned} \tag{D.16}$$

where $\mathcal{Z}_{\boldsymbol{\Theta}}(\mathbf{O}, \mathcal{H})$ is the *overall* normalisation term. The calculation formula is similar to equation (D.14) except that the pseudo-distribution is now based on the *complete* variational transform distribution in equation (D.8). It is worth noting that this normalisation term is different from the normalisation term in equation (D.14). The two normalisation terms have to be calculated separately.

D.2 Derivations for Multiple Regression Base Classes

When multiple regression base class is used, multiple adaptation transforms are used. Each one is shared among a group of Gaussians and assumed to be independent to the others. Let $\mathcal{T} = \{\mathcal{T}_1, \cdots, \mathcal{T}_{N_\mathcal{R}}\}$, where $\mathcal{T}_r$ is the transform associated with base class r, $\mathcal{R}$ is the set of regression base classes, $N_\mathcal{R}$ is the total number of base classes. The overall transform prior distribution can be regarded as a product of the individual transform prior distributions

$$p(\mathcal{T}) = \prod_{r \in \mathcal{R}} p(\mathcal{T}_r) \tag{D.17}$$

Accordingly, an independent variational transform distributions $q(\mathcal{T}_r)$ are introduced for each regression base class r, forming an overall variational transform distribution as

$$q(\mathcal{T}) = \prod_{r \in \mathcal{R}} q(\mathcal{T}_r) \tag{D.18}$$

a. Single component prior

For single component prior $p(\mathcal{T}_r)$, the introduced variational distribution $q(\mathcal{T}_r)$ is also a single component distribution. Then the joint variational distribution in equation (5.50) is expressed as

$$q(\boldsymbol{\theta}, \mathcal{T}) = q(\boldsymbol{\theta}|\mathbf{O}, \mathcal{H}) \prod_{r \in \mathcal{R}} q(\mathcal{T}_r|\mathbf{O}, \mathcal{H}) \tag{D.19}$$

The corresponding auxiliary function in equation (5.51) still applies and is reproduced here

$$\mathcal{Q}_{\mathtt{VB}}\left(q_{k+1}(\boldsymbol{\theta}), q_k(\mathcal{T})\right) = \langle \log p(\mathbf{O}, \boldsymbol{\theta}|\mathcal{T}, \mathcal{H}) \rangle_{q_{k+1}(\boldsymbol{\theta}) q_k(\mathcal{T})} + \mathtt{H}\left(q_{k+1}(\boldsymbol{\theta})\right) - \mathtt{KL}(q_k(\mathcal{T})||p(\mathcal{T})) \tag{D.20}$$

where k is the iteration index. The difference is that $p(\mathcal{T})$ and $q(\mathcal{T})$ are both products of individual distributions associated with the regression base classes. A VBEM algorithm similar to the global case can be derived. There are only slight modifications to be made as below

1. The pseudo-distribution $\tilde{p}(\mathbf{o}_t|\theta_t)$ is now calculated by

$$\tilde{p}(\mathbf{o}_t|\theta_t) = \exp\left(< \log p(\mathbf{o}_t|\mathcal{T}, \theta_t) >_{q_{k-1}(\mathcal{T}_{r(\theta_t)})} \right) \tag{D.21}$$

   where $r(\theta_t)$ is the base class that the component θ_t belongs to.

2. The KL distance $\mathtt{KL}(q(\mathcal{T})||p(\mathcal{T}))$ is calculated by

$$\begin{aligned}\mathtt{KL}(q(\mathcal{T})||p(\mathcal{T})) &= \int_{\mathcal{T}_1,\cdots,\mathcal{T}_{N_\mathcal{R}}} \left(\prod_r q(\mathcal{T}_r)\log\frac{\prod_r q(\mathcal{T}_r)}{\prod_r p(\mathcal{T}_r)}\right) d\mathcal{T}_1\cdots d\mathcal{T}_{N_\mathcal{R}} \\ &= \sum_{r\in\mathcal{R}}\int_{\mathcal{T}_r} q(\mathcal{T}_r)\log\frac{q(\mathcal{T}_r)}{p(\mathcal{T}_r)}\,d\mathcal{T}_r \\ &= \sum_{r\in\mathcal{R}}\mathtt{KL}(q(\mathcal{T}_r)||p(\mathcal{T}_r)) \end{aligned} \tag{D.22}$$

   This derivation is trivial by considering the independence of the base class transforms.

3. When updating the variational distribution $q(\mathcal{T}_r)$, only the statistics associated with the Gaussian components in base class r is accumulated. This is also a natural extension.

b. Multiple component prior

If a multiple component prior is used, $p(\mathcal{T})$ and $q(\mathcal{T})$ are both mixture distributions. The prior is expressed as

$$p(\mathcal{T}) = \prod_{r\in\mathcal{R}}\left(\sum_{n_r} c_{n_r} p(\mathcal{T}_r|n_r)\right) \tag{D.23}$$

where n_r is the n^{th} component of the transform distribution associated with base class r, c_{n_r} is the component weight. For clarity of notations, $\boldsymbol{\omega} = [n_1,\cdots,n_{N_\mathcal{R}}]$ is introduced as an index vector, $N_\mathcal{R}$ is the number of regression base classes. The prior distribution may then be re-expressed as

$$p(\mathcal{T}) = \sum_{\boldsymbol{\omega}} P(\boldsymbol{\omega})p(\mathcal{T}|\boldsymbol{\omega}) \tag{D.24}$$

where $P(\boldsymbol{\omega}) = \prod_r c_{n_r}$ and $p(\mathcal{T}|\boldsymbol{\omega}) = \prod_r p(\mathcal{T}_r|n_r)$. Similarly, the variational distribution is

$$q(\mathcal{T}) = \prod_{r\in\mathcal{R}}\left(\sum_{n_r} q(n_r)q(\mathcal{T}_r|n_r)\right) \tag{D.25}$$

$$= \sum_{\boldsymbol{\omega}} q(\boldsymbol{\omega})q(\mathcal{T}|\boldsymbol{\omega}) \tag{D.26}$$

where $q(\boldsymbol{\omega}) = \prod_r q(n_r)$ and $q(\mathcal{T}|\boldsymbol{\omega}) = \prod_r q(\mathcal{T}_r|n_r)$. Similar to section D.1, a lower bound is obtained by introducing a variational distribution

$$\begin{aligned}\log p(\mathbf{O}|\mathcal{H}) &= \sum_{\boldsymbol{\omega}} P(\boldsymbol{\omega})p(\mathbf{O}|\mathcal{H},\boldsymbol{\omega}) \\ &\geq \langle\log p(\mathbf{O}|\mathcal{H},\boldsymbol{\omega})\rangle_{q(\boldsymbol{\omega})} - \mathtt{KL}(q(\boldsymbol{\omega})||P(\boldsymbol{\omega})) \\ &= \langle\log p(\mathbf{O}|\mathcal{H},\boldsymbol{\omega})\rangle_{q(\boldsymbol{\omega})} - \sum_r \mathtt{KL}(q(n_r)||c_{n_r}) \end{aligned} \tag{D.27}$$

Further lower bounds are introduced for each particular "component-base class" sequence $\boldsymbol{\omega}$

$$\begin{aligned}\log p(\mathbf{O}|\mathcal{H},\boldsymbol{\omega}) &= \log\int_\mathcal{T} p(\mathbf{O}|\mathcal{H},\boldsymbol{\omega},\mathcal{T})p(\mathcal{T}|\boldsymbol{\omega})\,d\mathcal{T} \\ &\geq \left\langle\log\frac{p(\mathbf{O},\boldsymbol{\theta}|\mathcal{H},\mathcal{T},\boldsymbol{\omega})p(\mathcal{T}|\boldsymbol{\omega})}{q(\boldsymbol{\theta},\mathcal{T})}\right\rangle_{q(\boldsymbol{\theta},\mathcal{T})} \end{aligned} \tag{D.28}$$

The VB approximation is then given by

$$q(\boldsymbol{\theta}, \mathcal{T}) = q(\boldsymbol{\theta}|\mathbf{O}, \mathcal{H})q(\mathcal{T}|\mathbf{O}, \mathcal{H}, \boldsymbol{\omega}) \tag{D.29}$$

This is similar to equation (D.5) and also yields the auxiliary function (with similar brief notations as before)

$$\begin{aligned} \log p(\mathbf{O}|\mathcal{H}, \boldsymbol{\omega}) &\geq \mathcal{Q}_{\mathtt{VB}}\left(q_{k+1}(\boldsymbol{\theta}), q_k(\mathcal{T}|\boldsymbol{\omega})\right) \\ &= \left\langle \log p(\mathbf{O}, \boldsymbol{\theta}|\mathcal{H}, \mathcal{T}, \boldsymbol{\omega})\right\rangle_{q_{k+1}(\boldsymbol{\theta})q_k(\mathcal{T}|\boldsymbol{\omega})} + \mathtt{H}\left(q_{k+1}(\boldsymbol{\theta})\right) - \mathtt{KL}(q_k(\mathcal{T}|\boldsymbol{\omega})||p(\mathcal{T}|\boldsymbol{\omega})) \end{aligned} \tag{D.30}$$

Given the variational weights $q(\boldsymbol{\omega})$, similar VBEM algorithm can be derived as in section D.1. Equation (D.9) and equation (D.10) can again apply except that n_r is used instead of the component index n.

However, with multiple regression base classes, the update of component weights n_r is much more complicated than the global case. As discussed before, using *one* common hidden component sequence makes the update of variational weight hard. Here, the base class and prior components are all assumed to be independent to each other. Hence, one distinct hidden component sequence is associated with a particular "component-base class" pair n_r. Similar to equation (D.11), to calculate the variational weights, the variational distribution in equation (D.29) is approximated by

$$q(\boldsymbol{\theta}, \mathcal{T}) = q(\boldsymbol{\theta}|\mathbf{O}, \mathcal{H}, n_r)q(\mathcal{T}|\mathbf{O}, \mathcal{H}, \boldsymbol{\omega}) \tag{D.31}$$

Again, we used $q(\boldsymbol{\theta}|n_r)$ to denote $q(\boldsymbol{\theta}|\mathbf{O}, \mathcal{H}, n_r)$ and $q(\mathcal{T}|\boldsymbol{\omega})$ to denote $q(\mathcal{T}|\mathbf{O}, \mathcal{H}, \boldsymbol{\omega})$. The lower bound in equation (D.27) is then further approximated by

$$\log p(\mathbf{O}|\mathcal{H}) \geq \left\langle \mathcal{Q}_{\mathtt{VB}}\left(q_{k+1}(\boldsymbol{\theta}|n_r), q_k(\mathcal{T}|\boldsymbol{\omega})\right)\right\rangle_{q(\boldsymbol{\omega})} - \sum_r \mathtt{KL}(q(n_r)||c_{n_r}) \tag{D.32}$$

$$= \left\langle \mathcal{Q}_{\mathtt{VB}}\left(q_{k+1}(\boldsymbol{\theta}|n_r), q_k(\mathcal{T}|n_r)\right)\right\rangle_{q_k(n_r)} - \sum_r \mathtt{KL}(q_k(n_r)||c_{n_r}) \tag{D.33}$$

where k is the iteration index, $q_k(\mathcal{T}|n_r)$ is the modified variational distribution associated with the particular component n of base class r

$$q_k(\mathcal{T}|n_r) = q_k(\mathcal{T}_r|n_r) \prod_{i \in \mathcal{R}_{-r}} q_k(\mathcal{T}_i) \tag{D.34}$$

where $\mathcal{R}_{-r}$ denotes the regression base class set without r, $q_k(\mathcal{T}_i)$ is the *complete* mixture variational transform distribution for the i^{th} regression base class at iteration k. Derivation from equation (D.32) to equation (D.33) is actually a re-arrangement based on equation (D.30). Note that transforms associated with $\mathcal{R}_{-r}$ are all associated with the same $q(\boldsymbol{\theta}|n_r)$, which should be considered in the re-arrangement.

Given the form of the modified variational distribution in equation (D.34), the lower bound give each "component-base class" pair n_r can be re-expressed as

$$\begin{aligned} \mathcal{L}_{\mathtt{VB}}\left(q_k(\mathcal{T}|n_r)\right) &= \mathcal{Q}_{\mathtt{VB}}\left(q_{k+1}(\boldsymbol{\theta}|n_r), q_k(\mathcal{T}|n_r)\right) \\ &= \log \mathcal{Z}_{\boldsymbol{\Theta}}^{(n_r)}(\mathbf{O}, \mathcal{H}) - \mathtt{KL}\left(q_k(\mathcal{T}_r|n_r)||p(\mathcal{T}_r|n_r)\right) \\ &\quad - \sum_{i \in \mathcal{R}_{-r}} \sum_{n_i} q_k(n_i)\mathtt{KL}\left(q_k(\mathcal{T}_i|n_i)||p(\mathcal{T}_i|n_i)\right) \end{aligned} \tag{D.35}$$

where the calculation of $\mathcal{Z}_{\boldsymbol{\Theta}}^{(n_r)}(\mathbf{O}, \mathcal{H})$ is similar to equation (D.14) except that the pseudo-distribution is calculated based on the modified variational distribution equation (D.34). Given equation (D.35), differentiating equation (D.33) with respect to $q_k(n_r)$ and setting it to zero leads to the update formula of component weights as below

$$q_k(n_r) = \frac{c_{n_r} \exp\left(\mathcal{L}_{\mathtt{VB}}\left(q_k(\mathcal{T}|n_r)\right)\right)}{\sum_{n_r} c_{n_r} \exp\left(\mathcal{L}_{\mathtt{VB}}\left(q_k(\mathcal{T}|n_r)\right)\right)} \tag{D.36}$$

E

Derivations in Incremental Bayesian Adaptive Inference

The procedure of incremental variational Bayesian adaptive inference has been discussed in section 5.4. Here, the incremental recursions are derived for the general VB bound. In unsupervised mode, the marginal likelihood may be approximated by a lower bound

$$\log p(\mathbf{O}|\mathcal{H}) \geq \left\langle \log \frac{p(\mathbf{O}, \boldsymbol{\theta}|\mathcal{T}, \mathcal{H})p(\mathcal{T})}{q(\boldsymbol{\theta}, \mathcal{T})} \right\rangle_{q(\boldsymbol{\theta}, \mathcal{T})} \tag{E.1}$$

where the homogeneous data block is assumed to be split into U utterances, $\mathbf{O} = \mathbf{O}_{1:U} = \{\mathbf{O}_1, \cdots, \mathbf{O}_U\}$. Information is propagated to the U^{th} utterance from the previous $U-1$ utterances. The hypothesis for the data available consists of a set of hypotheses for utterances within it, $\mathcal{H} = \mathcal{H}_{1:U} = \{\mathcal{H}_1, \ldots, \mathcal{H}_U\}$. Assuming that $U-1$ utterances have been processed, with the information propagation strategy described in section 5.4, the VB approximation is given by

$$\begin{aligned} q(\boldsymbol{\theta}, \mathcal{T}) &= q(\boldsymbol{\theta}|\mathbf{O}, \mathcal{H}) q(\mathcal{T}|\mathbf{O}, \mathcal{H}) \\ &= q(\boldsymbol{\theta}_U|\mathbf{O}_U, \mathcal{H}_U) \prod_{u=1}^{U-1} q_K(\boldsymbol{\theta}_u|\mathbf{O}_u, \hat{\mathcal{H}}_u) q(\mathcal{T}|\mathbf{O}, \hat{\mathcal{H}}_{1:U-1}, \mathcal{H}_U) \end{aligned} \tag{E.2}$$

where K is the total iteration number, $\hat{\mathcal{H}}_{1:U}$ is the inferred hypothesis sequence from the 1^{st} to the U^{th} utterance. Considering that each utterance is independent to another and that the previously recognised hypothesis are propagated, i.e., $\mathcal{H} = \{\hat{\mathcal{H}}_1, \ldots, \hat{\mathcal{H}}_{U-1}, \mathcal{H}_U\}$, the VB lower bound in equation (E.1) can be re-arranged as an auxiliary function

$$\begin{aligned} \log p(\mathbf{O}|\mathcal{H}) &\geq \mathcal{Q}_{\mathtt{VB}}\left(q_{k+1}(\boldsymbol{\theta}), q_k(\mathcal{T})\right) \\ &= \langle \log p(\mathbf{O}, \boldsymbol{\theta}|\mathcal{T}, \mathcal{H}) \rangle_{q_{k+1}(\boldsymbol{\theta}) q_k(\mathcal{T})} + \mathtt{H}\left(q_{k+1}(\boldsymbol{\theta})\right) - \mathtt{KL}(q_k(\mathcal{T})||p(\mathcal{T})) \\ &= \sum_{u=1}^{U-1} \left(\left\langle \log p(\mathbf{O}_u, \boldsymbol{\theta}_u|\mathcal{T}, \hat{\mathcal{H}}_u) \right\rangle_{q_K(\boldsymbol{\theta}_u) q_k(\mathcal{T})} + \mathtt{H}\left(q_K(\boldsymbol{\theta}_u)\right) \right) - \mathtt{KL}(q_k(\mathcal{T})||p(\mathcal{T})) \\ &\quad + \langle \log p(\mathbf{O}_U, \boldsymbol{\theta}_U|\mathcal{T}, \mathcal{H}_U) \rangle_{q_{k+1}(\boldsymbol{\theta}_U) q_k(\mathcal{T})} + \mathtt{H}\left(q_{k+1}(\boldsymbol{\theta}_U)\right) \end{aligned} \tag{E.3}$$

where k is the iteration index and K is the total iteration number, $q(\boldsymbol{\theta}_u), 1 \leq u \leq U-1$ is the brief notation for $q(\boldsymbol{\theta}_u|\mathbf{O}_u, \hat{\mathcal{H}}_u)$, which are propagated from previous utterances. $q(\boldsymbol{\theta}_U)$ is

brief notation for $q(\boldsymbol{\theta}_U|\mathbf{O}_U, \mathcal{H}_U)$ and $q(\mathcal{T})$ is for $q(\mathcal{T}|\mathbf{O}, \hat{\mathcal{H}}_{1:U-1}, \mathcal{H}_U)$, they are the distributions to optimise. Differentiating equation (E.3) with respect to the two free distributions and setting it to zero leads to the exact update formulae in the incremental VBEM algorithm:

$$\log q_k(\boldsymbol{\theta}_U) = \langle \log p(\mathbf{O}_U, \boldsymbol{\theta}_U|\mathcal{T}, \mathcal{H}_U)\rangle_{q_{k-1}(\mathcal{T})} - \log \mathcal{Z}_{\boldsymbol{\Theta}}(\mathbf{O}_U, \mathcal{H}_U) \tag{E.4}$$

$$\begin{aligned}\log q_k(\mathcal{T}) \propto & \sum_{u=1}^{U-1} \left\langle \log p(\mathbf{O}_u, \boldsymbol{\theta}_u|\mathcal{T}, \hat{\mathcal{H}}_u)\right\rangle_{q_K(\boldsymbol{\theta}_u)} \\ & + \langle \log p(\mathbf{O}_U, \boldsymbol{\theta}_U|\mathcal{T}, \mathcal{H}_U)\rangle_{q_k(\boldsymbol{\theta}_U)} + \log p(\mathcal{T})\end{aligned} \tag{E.5}$$

They are equivalent to equation (5.72) and equation (5.73) in section 5.4. As each sentence is assumed to be independent to another, the normalisation term $\mathcal{Z}_{\boldsymbol{\Theta}}(\mathbf{O}_U, \mathcal{H}_U)$ is also independent to others. Considering that the overall component variational distribution is a product of individual variational component distributions as shown in equation (E.2), the overall normalisation term is also a product of each individual normalisation term

$$\mathcal{Z}_{\boldsymbol{\Theta}}(\mathbf{O}, \hat{\mathcal{H}}_{1:U-1}, \mathcal{H}_U) = \mathcal{Z}_{\boldsymbol{\Theta}}(\mathbf{O}_U, \mathcal{H}_U) \prod_{u=1}^{U-1} \mathcal{Z}_{\boldsymbol{\Theta}}(\mathbf{O}_u, \hat{\mathcal{H}}_u) \tag{E.6}$$

The derivation for point estimate approximation is similar except that the point version of variational distributions is used in equation (E.2).

F

Application of Bayesian Approximations to Mean Based Transforms

Interpolation weights in CAT and linear transforms in MLLR are both mean based linear transforms. The derivations of the two are quite similar. Hence, in this section, we only give the derivations for CAT. The MLLR formulae can be easily analogised.

F.1 Derivations in Frame-Independent (FI) Assumption

For a mixture prior $p(\mathcal{T}) = \sum_{n=1}^{N} c_n p(\mathcal{T}|n)$, the predictive distribution in FI is generally expressed as

$$\bar{p}(\mathbf{o}|m) = \int_{\mathcal{T}} p(\mathbf{o}|\mathcal{T}, m) p(\mathcal{T})\, d\mathcal{T} = \sum_{n} c_n \bar{p}(\mathbf{o}|m, n) \tag{F.1}$$

where $p(\mathbf{o}|\mathcal{T}, m)$ is a Gaussian distribution in HMM, $\bar{p}(\mathbf{o}|m, n)$ is an individual component of the predictive distribution

$$\bar{p}(\mathbf{o}|m, n) = \int_{\mathcal{T}} p(\mathbf{o}|\mathcal{T}, m) p(\mathcal{T}|n) \tag{F.2}$$

In the case of CAT, equation (F.2) can be written as

$$\bar{p}(\mathbf{o}|m, n) = \int_{\boldsymbol{\lambda}} \mathcal{N}(\mathbf{o}; \mathbf{M}^{(m)}\boldsymbol{\lambda}, \boldsymbol{\Sigma}^{(m)}) \mathcal{N}(\boldsymbol{\lambda}; \boldsymbol{\mu}_{\lambda}^{(n)}, \boldsymbol{\Sigma}_{\lambda}^{(n)})\, d\boldsymbol{\lambda} \tag{F.3}$$

In the rest derivations, the index m and n are omitted for clarity.

As $\boldsymbol{\lambda}$ is applied only to cluster means, by some re-arrangement, $\mathcal{N}(\mathbf{o}; \mathbf{M}^{(m)}\boldsymbol{\lambda}, \boldsymbol{\Sigma}^{(m)}) \mathcal{N}(\boldsymbol{\lambda}; \boldsymbol{\mu}_{\lambda}^{(n)}, \boldsymbol{\Sigma}_{\lambda}^{(n)})$ is a Gaussian distribution for the joint variable $(\mathbf{o}, \boldsymbol{\lambda})$ [1]. Then, the marginal distribution $\bar{p}(\mathbf{o}|m, n)$ is also a Gaussian distribution. The problem here is to derive the parameters of the resultant Gaussian, i.e.

$$\bar{p}(\mathbf{o}) = \mathcal{N}(\mathbf{o}; \bar{\boldsymbol{\mu}}, \bar{\boldsymbol{\Sigma}}) = \int_{\boldsymbol{\lambda}} \mathcal{N}(\mathbf{o}; \mathbf{M}\boldsymbol{\lambda}, \boldsymbol{\Sigma}) \mathcal{N}(\boldsymbol{\lambda}; \boldsymbol{\mu}_{\lambda}, \boldsymbol{\Sigma}_{\lambda})\, d\boldsymbol{\lambda} \tag{F.4}$$

[1] They all have quadratic forms in the exponential part and can be merged together.

According to the definition of mean and covariance, we have

$$\bar{\boldsymbol{\mu}} = \int_{\mathbf{o}} \mathbf{o} \int_{\boldsymbol{\lambda}} \mathcal{N}(\mathbf{o}; \mathbf{M}\boldsymbol{\lambda}, \boldsymbol{\Sigma}) \mathcal{N}(\boldsymbol{\lambda}; \boldsymbol{\mu}_\lambda, \boldsymbol{\Sigma}_\lambda)\, d\boldsymbol{\lambda}\, d\mathbf{o} \tag{F.5}$$

$$\bar{\boldsymbol{\Sigma}} = \int_{\mathbf{o}} (\mathbf{o} - \bar{\boldsymbol{\mu}})(\mathbf{o} - \bar{\boldsymbol{\mu}})^T \int_{\boldsymbol{\lambda}} \mathcal{N}(\mathbf{o}; \mathbf{M}\boldsymbol{\lambda}, \boldsymbol{\Sigma}) \mathcal{N}(\boldsymbol{\lambda}; \boldsymbol{\mu}_\lambda, \boldsymbol{\Sigma}_\lambda)\, d\boldsymbol{\lambda}\, d\mathbf{o} \tag{F.6}$$

By swapping the order of the integral and doing the integration over $\mathbf{o}$ first, it is easy to obtain the parameters as

$$\bar{\boldsymbol{\mu}} = \mathbf{M}\boldsymbol{\mu}_\lambda \tag{F.7}$$

$$\bar{\boldsymbol{\Sigma}} = \mathbf{M}\boldsymbol{\Sigma}_\lambda \mathbf{M}^T + \boldsymbol{\Sigma} \tag{F.8}$$

The above are equivalent to equation (5.81) and equation (5.82) when considering the index for prior component and HMM Gaussian component. Similar derivation applies to MLLR and has been mentioned in [37]. In [14], similar formulae were also obtained but using a much more complicated derivation.

F.2 Derivations in Variational Bayes (VB)

In VB, the form of the pseudo-distribution $\tilde{p}(\mathbf{o}|m)$ and the update formulae for variational transform distribution $q(\boldsymbol{\lambda})$ need to be worked out. Given the complete variational distribution

$$q(\boldsymbol{\lambda}) = \sum_{n=1}^{N} q(n) \mathcal{N}(\boldsymbol{\lambda}; \tilde{\boldsymbol{\mu}}_\lambda^{(n)}, \tilde{\boldsymbol{\Sigma}}_\lambda^{(n)}) \tag{F.9}$$

the log-likelihood of the pseudo-distribution has shown to be

$$\log \tilde{p}(\mathbf{o}|m) = \sum_n q(n) \int_{\boldsymbol{\lambda}} \log p(\mathbf{o}|m) \mathcal{N}(\boldsymbol{\lambda}; \tilde{\boldsymbol{\mu}}_\lambda^{(n)}, \tilde{\boldsymbol{\Sigma}}_\lambda^{(n)})\, d\boldsymbol{\lambda} \tag{F.10}$$

where $p(\mathbf{o}|m)$ is a Gaussian component of HMM, hence

$$\log p(\mathbf{o}|m) = -\frac{1}{2}\left(D \log 2\pi + \log |\boldsymbol{\Sigma}^{(m)}| + (\mathbf{o} - \mathbf{M}^{(m)}\boldsymbol{\lambda})^T \boldsymbol{\Sigma}^{(m)-1} (\mathbf{o} - \mathbf{M}^{(m)}\boldsymbol{\lambda}) \right) \tag{F.11}$$

where D is the observation dimension size. As equation (F.11) is a quadratic form of $\boldsymbol{\lambda}$, it is easy to do the integral in equation (F.10). The result is

$$\log \tilde{p}(\mathbf{o}|m) = \sum_n q(n) \left(\log \mathcal{N}(\mathbf{o}; \mathbf{M}^{(m)} \tilde{\boldsymbol{\mu}}_\lambda^{(n)}, \boldsymbol{\Sigma}^{(m)}) - \frac{1}{2} \mathrm{tr}(\tilde{\boldsymbol{\Sigma}}_\lambda^{(n)} \mathbf{M}^{(m)T} \boldsymbol{\Sigma}^{(m)-1} \mathbf{M}^{(m)}) \right) \tag{F.12}$$

Given the pseudo-distribution $\tilde{p}(\mathbf{o}|m)$ calculated using the current $q(\boldsymbol{\lambda})$, sufficient statistics can be accumulated to update $q(\boldsymbol{\lambda})$. Here, we only demonstrate the single component case. It can be generalised to multiple component case as described in appendix D.1. From the VBEM algorithm, the general form of $q_k(\boldsymbol{\lambda})$ is

$$\begin{aligned} \log q_k(\boldsymbol{\lambda}) &= \log p(\boldsymbol{\lambda}) + \langle \log p(\mathbf{O}, \boldsymbol{\theta}|\boldsymbol{\lambda}, \mathcal{H}) \rangle_{q_k(\boldsymbol{\theta})} - \log \mathcal{Z}'_\lambda(\mathbf{O}, \mathcal{H}) \\ &= \log p(\boldsymbol{\lambda}) + \sum_m \sum_t \gamma_m(t) \log \mathcal{N}(\mathbf{o}_t; \mathbf{M}^{(m)}\boldsymbol{\lambda}, \boldsymbol{\Sigma}^{(m)}) - \log \mathcal{Z}''_\lambda(\mathbf{O}, \mathcal{H}) \end{aligned} \tag{F.13}$$

where m is distinct component index, $\gamma_m(t)$ is the posterior occupancy calculated using forward-backward algorithm based on the pseudo-distribution. $\log \mathcal{Z}'_\lambda(\mathbf{O}, \mathcal{H})$ and $\log \mathcal{Z}''_\lambda(\mathbf{O}, \mathcal{H})$ are normalisation terms to subsume constants. Considering that $p(\boldsymbol{\lambda})$ is a Gaussian distribution and using it in equation (F.13) yields

$$\log q_k(\boldsymbol{\lambda}) \propto -\frac{1}{2}\left(\boldsymbol{\lambda}^T \tilde{\boldsymbol{\Sigma}}_\lambda \boldsymbol{\lambda} - 2\tilde{\boldsymbol{\Sigma}}_\lambda^{-1}\tilde{\boldsymbol{\mu}}_\lambda\right) \tag{F.14}$$

From the above formula, with an appropriate normalisation term, $q_k(\boldsymbol{\lambda})$ is a Gaussian distribution. The parameters are updated by[2]

$$\begin{aligned}
\tilde{\boldsymbol{\Sigma}}_\lambda &= \left(\boldsymbol{\Sigma}_\lambda^{-1} + \mathbf{G}_{\texttt{ML}}\right)^{-1} \\
\tilde{\boldsymbol{\mu}}_\lambda &= \tilde{\boldsymbol{\Sigma}}_\lambda\left(\boldsymbol{\Sigma}_\lambda^{-1}\boldsymbol{\mu}_\lambda + \mathbf{k}_{\texttt{ML}}\right)
\end{aligned}$$

where the ML sufficient statistics are

$$\begin{aligned}
\mathbf{G}_{\texttt{ML}} &= \sum_m \sum_t \gamma_m(t)\mathbf{M}^{(m)T}\boldsymbol{\Sigma}^{(m)-1}\mathbf{M}^{(m)} \\
\mathbf{k}_{\texttt{ML}} &= \sum_m \mathbf{M}^{(m)T}\boldsymbol{\Sigma}^{(m)-1}\left(\sum_t \gamma_m(t)\mathbf{o}_t\right)
\end{aligned}$$

[2]This can be proved by re-arranging equation (F.13) in terms of sufficient statistics.

Bibliography

[1] 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.

[2] 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] T. Anastasakos, J. Mcdonough, R. Schwartz, and J. Makhoul. A compact model for speaker adaptive training. In *Proc. ICSLP*, pages 1137–1140, 1996.

[4] 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.

[5] S. Axelrod, R. Gopinath, and P. Olsen. Modeling with a subspace constraint on inverse covariance matrices. In *Proc. ICSLP*, 2002.

[6] 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.

[7] J. K. Baker. The dragon system - an overview. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 23(1):24–29, 1975.

[8] 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.

[9] M. J. Beal. *Variational Algorithms for Approximate Bayesian Inference*. PhD thesis, University College London, 2003.

[10] J. O. Berger. *Statistical Decision Theory and Bayesian Analysis*. Springer-Verlag, 1993.

[11] H. Botterweck. Very fast adaptation for large vocabulary continuous speech recognition using eigenvoices. In *ICSLP*, pages 354–357, 2000.

[12] N. Campbell. Canonical variate analysis - a general formulation. *Australian Journal of Statistics*, 26:86–96, 1984.

[13] C. Chesta, O. Siohan, and C. Lee. Maximum a posteriori linear regression for hidden Markov model adaptation. *Proc. EuroSpeech*, 1:211–214, 1999.

[14] J. T. Chien. Linear regression based Bayesian predictive classification for speech recognition. *IEEE transactions on speech and audio processing*, 11:70–79, 2003.

[15] W. Chou. Maximum a-posterior linear regression with elliptical symmetric matrix variate priors. In *Proc. ICASSP*, pages 1–4, 1999.

[16] 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.

[17] H. Davis, R. Biddulph, and S. Balashek. Automatic recognition of spoken digits. *Journal of the Acoustical Society of America*, 24(6):637–642, 1952.

[18] 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.

[19] T. Dean and K. Kanazawa. A model for reasoning about persistence and causation. *Computational Intelligence*, 5:142–150, 1989.

[20] M. DeGroot. *Optimal statistical decisions*. New Yori: McGraw-Hill, 1970.

[21] 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.

[22] 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.

[23] V. Doumpiotis, S. Tsakalidis, and W. Byrne. Lattice segmentation and minimum Bayes risk discriminative training. In *Proc. EuroSpeech*, 2003.

[24] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. John Wiley & Sons, Inc, 2001.

[25] 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.

[26] G. Evermann, H. Y. Chan, M. J. F. Gales, B. Jia, D. Mrva, P. C. Woodland, and K. Yu. Training LVCSR systems on thousands of hours of data. In *Proc. ICASSP*, 2005.

[27] K. Fukunaga. *Introduction to statistical pattern recognition*. Academic Press Professional, 1972.

[28] 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.

[29] S. Furui. Unsupervised speaker adaptation method based on hierarchical spectral clustering. In *Proc. ICASSP*, volume 1, pages 286–289, 1989.

[30] M. J. F. Gales. *Model-based techniques for noise-robust speech recognition*. PhD thesis, Cambridge University, 1995.

[31] 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.

[32] M. J. F. Gales. Transformation smoothing for speaker and environmental adaptation. In *Proc. EuroSpeech*, 1997.

[33] M. J. F. Gales. Cluster adaptive training for speech recognition. In *Proc. ICSLP*, pages 1783–1786, 1998.

[34] M. J. F. Gales. Maximum likelihood linear transformations for HMM-based speech recognition. *Computer Speech and Language*, 12:75–98, 1998.

[35] M. J. F. Gales. Semi-tied covariance matrices for hidden Markov models. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 7:272–281, 1999.

[36] M. J. F. Gales. Cluster adaptive training of hidden Markov models. *IEEE Transactions on Speech and Audio Processing*, 8:417–428, 2000.

[37] M. J. F. Gales. Acoustic factorization. In *Proc. ASRU*, 2001.

[38] M. J. F. Gales. Adaptive training for robust ASR. In *Proc. ASRU*, 2001.

[39] M. J. F. Gales. Multiple-cluster adaptive training schemes. In *Proc. ICASSP*, 2001.

[40] 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.

[41] M. J. F. Gales and P. C. Woodland. Mean and variance adaptation within the MLLR framework. *Computer Speech and Language*, 10:249–264, 1996.

[42] 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.

[43] L. Gillick and S. J. Cox. Some statistical issues in the comparison of speech recognition. In *Proc. ICASSP*, pages 532–535, 1989.

[44] V. Goel and W. Byrne. Minimum Bayes-risk automatic speech recognition. *Computer Speech and Language*, 14(2), 2000.

[45] I. J. Good. The population frequency of species and the estimation of population parameters. *Biometrika*, 40:237–264, 1953.

[46] P. S. Gopalakrishnan, D. Kanevsky, A. Nadas, and D. Nahamoo. A generalization of the Baum algorithm to rational objective functions. In *Proc. ICASSP*, 1989.

[47] A. Gunawardana and W. Byrne. Discriminative speaker adaptation with conditional maximum likelihood linear regression. In *Proc. EuroSpeech*, 2001. Scandinavia.

[48] A. Gunawardana and W. J. Byrne. Discriminative adaptation with conditional maximum likelihood linear regression. *Presented at NIST Hub5 Workshop*, 2001.

[49] T. Hain. *Hidden Model Sequence Models for Automatic Speech Recognition*. PhD thesis, Cambridge University, 2001.

[50] 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*, 2004.

[51] T. Hazen. *The use of speaker correlation information for automatic speech recognition*. PhD thesis, Mass. Inst. Technol., 1998.

[52] T. Hazen and J. Glass. A comparison of novel techniques for instantaneous speaker adaptation. In *Proc. EuroSpeech*, pages 2047–2050, 1997.

[53] H. Hermansky. Perceptual linear prediction of speech. *Journal of the acoustic society of America*, 87(4):1738–1752, 1990.

[54] H. Hermansky, N. Morgan, A. Bayya, and P. Kohn. RASTA-PLP speech analysis technique. In *Proc. ICASSP*, 1992.

[55] S. Homma, K. Aikawa, and S. Sagayama. Improved estimation of supervision in unsupervised speaker adaptation. In *Proc. ICASSP*, volume 2, pages 1023–1026, 1997.

[56] 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.

[57] 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.

[58] 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.

[59] 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.

[60] T. Schaaf J. McDonough and A. Waibel. On maximum mutual information speaker-adapted training. In *Proc. ICASSP*, Florida, USA, May, 2002.

[61] F. Jelinek. Continuous speech recognition by statistical methods. *Procs. of the IEEE*, 64:532–556, 1976.

[62] 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.

[63] N. L. Johnson and S. Kotz. *Distribution in statistics*. New York: Wiley, 1972.

[64] 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.

[65] B. H. Juang, S. E. Levinson, and M. M. Sondhi. Maximum likelihood estimation for multivariate mixture observations of Markov chains. *IEEE Transactions on Information Theory*, 32:307–309, 1986.

[66] 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.

[67] T. Kosaka and S. Sagayama. Tree structured speaker clustering for fast speaker adaptation. In *Proc. ICASSP*, volume 1, pages 245–248, 1994.

[68] R. Kuhn, J. C. Junqua, P. Nguyen, and N. Niedzielski. Rapid speaker adaptation in eigenvoice space. *IEEE Transactions on Speech and Audio Processing*, 8(6):695–707, 2000.

[69] 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.

[70] N. Kumar. *Investigation of Silicon-Auditory Models and Generalization of Linear Discriminant Analysis for Improved Speech Recognition*. PhD thesis, John Hopkins University, 1997.

[71] L. Lee and R. C. Rose. Speaker normalization using efficient frequency warping procedures. *Proc. ICASSP*, 1:353–356, 1996.

[72] C. J. Leggetter. *Improved acoustic modelling for HMMs using linear transformations*. PhD thesis, Cambridge University, 1995.

[73] 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.

[74] 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.

[75] H. Liao and M. J. F. Gales. Joint uncertainty decoding for robust speech recognition with found data. Technical Report CUED/F-INFENG/TR552, Cambridge University Engineering Department, 2006.

[76] 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.

[77] A. Ljolje. The importance of cepstral parameter correlations in speech recognition. *Computer speech and language*, 8:223–232, 1994.

[78] A. Ljolje. The AT&T LVCSR-2001 system. In *Proc. the NIST LVCSR Workshop*, NIST, 2001.

[79] J. Luo, Z. Ou, and Z. Wang. Discriminative speaker adaptation with eigenvoices. In *Proc. INTERSPEECH*, 2005.

[80] W. Macherey, L. Haferkamp, R. Schlüter, and H. Ney. Investigations on error minimizing training criteria for discriminative training in automatic speech recognition. In *Proc. InterSpeech*, pages 2133–2136, 2005.

[81] S. Matsoukas, R. Schwartz, H. Jin, and L. Nguyen. Practical implementations of speaker adaptive training. *Proc. DARPA Speech Recognition Workshop*, 1997.

[82] T. Matsui and S. Furui. N-Best-based unsupervised speaker adaptation for speech recognition. *Computer Speech and Language*, 12:41–50, 1998.

[83] L. Neumeyer, A. Sankar, and V. Digilakis. A comparative study of speaker adaptation techniques. *Proc. EuroSpeech*, pages 1127–1130, 1995.

[84] 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.

[85] L. Nguyen, B. Xiang, M. Afify, S. Abdou, S. Matsoukas, R. Schwartz, and J. Makhoul. The BBN RT04 english broadcast news transcription system. In *Proc. INTERSPEECH*, 2005.

[86] Y. Normandin. *Hidden Markov models, maximum mutual information estimation, and the speech recognition problem*. PhD thesis, McGill University, 1991.

[87] P. Nuguyen, R. Kuhn, L. Rigazio, J. C. Junqua, and C. Wellekens. Self-adaptation using eigenvoices for large vocabulary continuous speech recognition. In *Proc. ITRW on Adaptation*, 2001.

[88] P. Nuguyen, C. Wellekens, and J. C. Junqua. Maximum likelihood eigenspace and MLLR for speech recognition in noisy environments. In *Proc. EuroSpeech*, pages 2519–2522, 1999.

[89] M. Padmanabhan, G. Saon, and G. Zweig. Lattice-based unsupervised MLLR for speaker adaptation. *Proc. ISCA ITRW ASR2000*, pages 128–131, 2000.

[90] D. Povey. *Discriminative Training for Large Vocabulary Speech Recognition*. PhD thesis, Cambridge University, 2003.

[91] D. Povey, M. J. F. Gales, D. Y. Kim, and P. C. Woodland. MMI-MAP and MPE-MAP for acoustic model adaptation. In *Proc. EuroSpeech*, 2003.

[92] D. Povey and P. C. Woodland. Improved discriminative training techniques for large vocabulary continuous speech recognition. In *Proc. ICASSP*, 2001.

[93] D. Povey and P. C. Woodland. Minimum phone error and I-smoothing for improved discriminative training. In *Proc. ICASSP*, 2002. Orlando.

[94] D. Pye and P. C. Woodland. Experiments in speaker normalization and adaptation for large vocabulary speech recognition. *Proc. ICASSP*, pages 1047–1050, 1997.

[95] L. Rabiner and B. H. Juang. *Fundamentals of Speech Recognition*. Prentice Hall PTR, 1993.

[96] B. D. Ripley. *Pattern recognition and neural networks*. Cambridge University Press, 1996.

[97] H. Robbins. An empirical Bayes approach to statistics. In *Proc. Third Berkeley Symposium on Math. Statist. and Prob.*, pages 157–164, 1955.

[98] H. Robbins. The empirical Bayes approach to statistical decision problems. *Ann. Math. Stastist.*, 35:1–20, 1964.

[99] C. P. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer-Verlag, 1999.

[100] G. Saon, A. Dharanipragada, and D. Povey. Feature space Gaussianization. In *Proc. ICASSP*, 2004.

[101] G. Saon, D. Povey, and G. Zweig. CTS decoding improvements at IBM. In *EARS STT Workshop*, 2003.

[102] R. Schlüter and W. Macherey. Comparison of discriminative training criteria. In *Proc. ICASSP*, 1998.

[103] 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.

[104] K. Shinoda and C. H. Lee. Structural MAP speaker adaptation using hierarchical priors. *Proc. ASRU*, pages 381–388, 1997.

[105] 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.

[106] 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.

[107] A. C. Surendran and Chin-Hui Lee. Transformation based Bayesian prediction for adaptation of HMMs. *Speech Communication*, 34:159–174, 2001.

[108] L. Tierney and J. B. Kadane. Accurate approximations for posterior moments and marginal densities. *J. Ameri. Statist. Assoc.*, 81(393):82–86, 1986.

[109] S. E. Tranter, K. Yu, G. Evermann, and P. C. Woodland. Generating and evaluation segmentations for automatic speech recognition of conversational telephone speech. In *Proc. ICASSP*, 2004.

[110] 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.

[111] L. F. Uebel. *Speaker normalisation and adaptation in large vocabulary speech recognition*. PhD thesis, Cambridge University, 2002.

[112] L. F. Uebel and P. C. Woodland. An investigation into vocal tract length normalization. *Proc. EuroSpeech*, pages 911–914, 1999.

[113] L. F. Uebel and P. C. Woodland. Discriminative linear transforms for speaker adaptation. *Proc. ISCA ITR-Workshop on Adaptation Methods for Speech Recognition*, 2001.

[114] L. F. Uebel and P. C. Woodland. Improvements in linear transforms based speaker adaptation. *Proc. ICASSP*, 2001.

[115] L. F. Uebel and P. C. Woodland. Speaker adaptation using lattice-based MLLR. *Proc. ISCA ITR-Workshop on Adaptation Methods for Speech Recognition*, 2001.

[116] A. J. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13:260–269, 1967.

[117] F. Wallhoff, D. Willett, and G. Rigoll. Frame-discriminative and confidence-driven adaptation for LVCSR. In *Proc. ICASSP*, volume 3, pages 1835–1838, 2000.

[118] L. Wang. *Discriminative linear transforms for adaptation and adaptive training*. PhD thesis, Cambridge University, 2006.

[119] L. Wang and P. C. Woodland. Discriminative adaptive training using the MPE criterion. In *Proc. ASRU*, 2003.

[120] S. Watanabe, Y. Minami, A. Nakamura, and N. Ueda. Application of variational Bayesian approach to speech recognition. In *NIPS 15*, 2003.

[121] S. Wegmann, D. McAllaster, J. Orloff, and B. Peskin. Speaker normalisation on conversational telephone speech. In *Proc. ICASSP*, pages 339–341, 1996.

[122] F. Wessel, K. Macherey, and H. Ney. A comparison of word graph and N-Best list based confidence measures. In *Proc. EuroSpeech*, volume 1, pages 315–318, 1999.

[123] F. Wessel, K. Macherey, and R. Schlüter. Using word probabilities as confidence measures. In *Proc. ICASSP*, volume 6, pages 225–228, 1998.

[124] F. Wessel, R. Schlutr, K. Macherey, and H. Ney. Confidence measures for large vocabulary continuous speech recognition. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 9(3):288–298, 2001.

[125] 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.

[126] P. C. Woodland. Speaker adaptation for continuous density HMMs: A review. *Proc. ISCA ITR-Workshop on Adaptation Methods for Speech Recognition*, 2001.

[127] P. C. Woodland, M. J. F. Gales, D. Pye, and S. J. Young. The development of 1996 broadcast news transcription system. *Proc. DARPA Speech Recognition Workshop*, pages 73–78, 1997.

[128] P. C. Woodland, J. J. Odell, V. Valtchev, and S. J. Young. Large vocabulary continuous speech recognition using htk. In *Proc. ICASSP*, 1994.

[129] 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.

[130] 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.

[131] 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.

[132] 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.

[133] S. J. Young, D. Kershaw, J. J. Odell, D. Ollason, V. Valtchev, and P. C. Woodland. *The HTK Book (for HTK version 3.0)*. Cambridge University Engineering Department, 2000.

[134] 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.

[135] S. J. Young and P. C. Woodland. The use of state tying in continuous speech recognition. In *Proc. EuroSpeech*, pages 2207–2210, 1993.

[136] K. Yu and M. J. F. Gales. Adaptive training using structured transforms. In *Proc. ICASSP*, 2004.

[137] K. Yu and M. J. F. Gales. Discriminative cluster adaptive training. Technical Report CUED/F-INFENG/TR486, Cambridge University Engineering Department, 2004.

[138] K. Yu and M. J. F. Gales. Bayesian adaptation and adaptively trained systems. In *Proc. ASRU*, 2005.

[139] K. Yu and M. J. F. Gales. Bayesian adaptation and adaptive training. Technical Report CUED/F-INFENG/TR542, Cambridge University Engineering Department, 2006.

[140] K. Yu and M. J. F. Gales. Discriminative cluster adaptive training. *IEEE Transactions on Speech and Audio Processing*, 14(5):1694–1703, 2006.

[141] K. Yu and M. J. F. Gales. Incremental adaptation using Bayesian inference. In *Proc. ICASSP*, 2006.

[142] 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.