Bayesian and Predictive Techniques for Speaker Adaptation

Seyed Mohammad Ahadi-Sarkani

Hughes Hall

January 1996

Dissertation submitted to the University of Cambridge

for the degree of Doctor of Philosophy

Declaration

This dissertation is the result of my own work and includes nothing which is the outcome of work done in collaboration, except where stated. It has not been submitted in whole or part for a degree at any other university.

The length of this thesis including appendices, bibliography, footnotes and tables is approximately 48000 words.

Summary

HMM-based speech recognition systems have recently demonstrated impressive recognition performance. Many of these systems attempt to provide low error rates for a large range of speakers. However, the performance of these speaker independent systems is generally inferior to speaker dependent systems trained for a specific speaker.

In this thesis, the problem of speaker adaptation using small amounts of speaker-specific data in order to improve speaker independent performance is addressed. Two different approaches to solving this problem are considered: a Bayesian model parameter adaptation technique and a model parameter prediction technique.

The Bayesian model adaptation technique, also called Maximum *a posteriori* (MAP) estimation, tries to update the HMM parameters using the available utterances from a new speaker and prior information to overcome the sparse training data problem. An implementation of this approach using the Forward-Backward algorithm is reported and several issues regarding the implementation and prior parameter estimation are evaluated. Furthermore, the use of MAP estimation for supervised and unsupervised adaptation using both batch and incremental adaptation modes is discussed. A speaker clustering approach to prior parameter improvement is also introduced.

The second adaptation technique called Regression-based Model Prediction (RMP) is a predictive approach which uses linear regression to find the phone model relationships in an HMM system. These model relationships are used in a predictive fashion to help a model parameter adaptation scheme improve further when only sparse training data is available. In this way the parameters of unadapted or poorly adapted models are predicted from the better trained model parameters. In this work, RMP is applied to the models already adapted by MAP estimation for further improvement, and is found to be useful for very fast speaker adaptation purposes. Several issues which can help in improving the performance of an RMP adapted system have been reported such as the use of multiple regression, dynamic setting of regression order and iterative RMP adaptation.

Experiments on the above techniques using data from ARPA RM and WSJ databases are described.

Keywords: speech recognition, hidden Markov models, speaker adaptation, maximum *a posteriori* estimation, regression-based model prediction.

Acknowledgements

Firstly I would like to express my deepest gratitude to my supervisor, Phil Woodland, for his patience, dedication, valuable guidance and assistance throughout the period of my study in Cambridge. It has been a marvellous opportunity to work under his supervision.

I should also like to thank the Iranian Ministry of Culture and Higher Education for sponsoring my studies and the Committee of Vice-Chancellors and Principals of the Universities of the United Kingdom for providing me with ORS award, as well as the Engineering Department, the IEEE UK & RI Signal Processing Chapter and Hughes Hall for providing grants towards my attendance at ICASSP-95.

It has been a great privilege to be a member of the Speech, Vision and Robotics group at the University of Cambridge, and I would like to thank all those people who have assisted me during my stay here and especially for being good friends. Many thanks to Jason Humphries, Kamran Kordi, Chris Leggetter, Parham Zolfaghari and Mark Gales for the time and effort they have spent reading and proof reading my thesis, and for their discussions and comments. I am also thankful to Steve Young and Phil Woodland for providing the group with the HTK toolkit, which has spared me much effort and time. Many thanks also to everyone responsible for maintaining the high standard of the research computer facilities of the Speech, Vision and Robotics group.

Finally, I would like to thank my wife for her continued support, encouragement and company during these years, and my mother and late father for giving me all I have in life.

Contents

Notation and List of Symbols

a	a scalar variable
$\boldsymbol{a}$, $\boldsymbol{A}$	a column vector, a matrix or a vector of parameters (bold)
$\lvert\boldsymbol{A}\rvert$	the determinant of matrix $\boldsymbol{A}$
$\boldsymbol{A}'$	the transpose of matrix $\boldsymbol{A}$
$tr(\boldsymbol{A})$	the trace of matrix $\boldsymbol{A}$
$\tilde{a}$	an estimate of parameter a
$\boldsymbol{m}$, $\boldsymbol{\mu}$	Gaussian mean vector
$\boldsymbol{\Sigma}$	Gaussian covariance matrix
$\boldsymbol{r}$	Gaussian precision matrix (inverse of covariance matrix)
ω	Gaussian mixture weight
$\boldsymbol{\theta}$	p.d.f. parameter vector
$\boldsymbol{\lambda}$	HMM parameter vector
$\boldsymbol{o}$	a set of observations
τ	normal prior distribution parameter
$\boldsymbol{u}$	precision matrix of Wishart prior distribution
ψ	parameter of prior Dirichlet distribution for mixture weights
η	parameter of prior Dirichlet distribution for state transition probabilities
ρ	simple correlation coefficient
R	multiple correlation coefficient
x	a source regression component
y	a target regression component

Abbreviations

HMM	Hidden Markov Models
CDHMM	Continuous Density Hidden Markov Models
SI	Speaker Independent
SD	Speaker Dependent
MAP	Maximum *a posteriori* estimation
ML	Maximum Likelihood estimation
EM	Expectation Maximisation
RMP	Regression-based Model Prediction
VFS	Vector Field Smoothing
MLLR	Maximum Likelihood Linear Regression
SC	Speaker Clustering
EMAP	Extended Maximum *a posteriori* estimation
ICM	Intra-Cluster Model set
ICPM	Intra-Cluster Phone Model set
ARPA	Advanced Research Projects Agency
RM	Resource Management database
WSJ	Wall Street Journal database

Chapter 1

Introduction

1.1 The Speech Recognition Problem

Speech recognition has proved to be one of the most difficult problems in the field of pattern recognition. It is also one of the most important ones since the most natural means of human communication is speech. Leaving the hands and eyes free, speech provides an easy way for communication and control while other important tasks are in progress. As an example, a driver could communicate with others or control various things within his car while driving, without any need to use his hands or eyes, which are needed for driving. Speech recognition can also help in the automation of many tasks, such as answering telephones, which is currently being largely performed by human beings, or man-machine interfacing.

Although speech recognition appears to be a very simple task for human beings, it has been found to be a very difficult and complicated job for machines. In spite of the tremendous improvements in the fields of computer and information technology, the overall performance of the speech recognition systems are substantially lower than that of human beings. To achieve the very low error rates, a number of restrictions have to be placed on the speech recognition task. Many realistic problems, such as out of vocabulary words (a large vocabulary problem), background and channel noise, channel limitations (e.g. limited spectral bandwidth usually imposed to speech signals on telephone lines), change in speech signal level and non-native accents can however lead to much lower recognition accuracies. Much work has been carried out to overcome the different problems faced in speech recognition and have led to more robust speech recognisers and this will continue until satisfactory recognition performance is achieved.

1.2 Historical Background

After the introduction of digital computers and their application to signal analysis and processing problems, the notion of automatically recognising speech using computers emerged. The attraction of *Automatic Speech Recognition* (ASR) initiated a challenge that after about three decades still continues.

While the more basic problems were explored during the 1960s, some important techniques in this field were developed and more realistic speech recognisers based on these techniques were implemented during the 1970s. There is no doubt that one of the most important improvements made during this period was the application of the theory of *Hidden Markov Models* (HMMs) to speech recognition. This was done for the first time by Baker at CMU [3] and Jelinek and his colleagues at IBM [52]. The systems based on HMMs became more popular during the 1980s and many different problems encountered by speech recognisers, which caused them to be far from ideal, were tackled during this period. Many researchers reported successful implementations of algorithms to overcome, at least partly, some important difficulties in the field, such as large vocabulary, speaker independence, noise, continuous speech, etc.

Many of these efforts are still continuing to enable researchers to approach the ideal speech recogniser system, capable of understanding speech of a large vocabulary size in normal noise conditions independent of speaker and his/her position and physical conditions.

1.3 Speaker Independence and Speaker Dependence

The concept of *speaker independence* has always challenged the researchers in this field. A Speaker Independent (SI) system, as the name implies, is a system which is capable of recognising speech from any speaker. In contrast, a *Speaker Dependent* (SD) system is a system which is trained to recognise speech from a particular speaker. Such systems normally need a large amount of training data from the proposed speaker before being able to recognise his/her speech with high accuracy. Most of the speech recognition systems built during the 1980s or before were of this type and hence, not able to recognise speech from a new speaker without a considerable amount of degradation in the system performance.

Speaker independent systems have a number of advantages over speaker dependent systems. It is preferable that any new speaker need not to be obliged to speak for hours[1] in order to train the recogniser, before it becomes able to recognise his/her speech. Even after that, he/she must wait for a long time while the system is trained.

Another problem with speaker dependent systems is that they are usable only by the speaker for which they are trained, so if several speakers are to be supported a large amount of memory is required to save the models for different speakers. Even with several model sets, automatic switching between speakers can be a difficult task. There are even further problems with SD systems that urge researchers to try to move toward speaker independence. Aging, health condition, fatigue and stress are some of the problems which can have more adverse effects on SD systems compared to an SI system. Moreover, multi-speaker environments and conditions need an absolutely speaker independent system.

[1] The amount of speech required depends on the complexity of the system, so an isolated word small vocabulary system may require just a few repetitions of each word.

SI systems, with many advances in different aspects of speech recognition, have recently become more popular and can achieve low error rates. The most popular technique to achieve speaker independent performance in speech recognition is found to be speaker pooling. In this technique, typically a large speech database involving a large number of speakers, each uttering a few sentences is used to train the recogniser. The recogniser trained by this approach, if properly trained, is capable of recognising any new speaker's speech with an acceptable level of performance.

1.4 Speaker Adaptation

Generally, a speaker dependent system, trained with a large amount of speech data from a single speaker and tested on that speaker, outperforms a similar speaker independent system, regardless of how good it is, when tested on the same speaker. Speaker adaptation aims to approach SD performance with as little data from the speaker as possible.

This means that any new speaker can use the recognition system by simply supplying a few sentences. These sentences are used to improve the system performance in recognising the speech of this particular speaker. Recent speaker adaptation techniques usually use the speaker independent system as a baseline system, so that the final performance will be superior to that of the SI system. The ultimate goal in speaker adaptation is to reach the performance of the SD system, while some speaker adaptation techniques, under certain conditions and with large amounts of speaker data, can even achieve better performance compared to that of the equivalent SD system [42]. The amount of improvement obtained is very much dependent on the technique used for adaptation and the amount of data required. Due to some properties of the conventional training techniques used in statistical speech recognition, they may even lead to system performance degradation if only a small amount of speaker-specific data is available [36]. This means that special adaptation techniques need to be developed to not only overcome this problem, but also to make the most of the limited amounts of adaptation data available.

Some adaptation techniques, such as the one described in Chapter 5, claim *fast* speaker adaptation. This means that the system is able to adapt to a new speaker with very little adaptation data from that speaker and can be of great importance for certain applications. This goal is especially difficult in large vocabulary continuous speech recognition tasks where, due to the very large number of system parameters, usually a considerable amount of adaptation data should be available before any improvement in the system performance for a new speaker can be achieved.

1.5 Outline of the Thesis

The rest of this thesis is organised as follows: Chapter 2 gives a brief introduction to the main topics in automatic speech recognition including front-end processing, the use of statistical speech recognition methods and their problems, other problems associated with speech recognition and the solutions found for them so far.

Chapter 3 focuses on the introduction of possible sources of differences between speakers and the use of speaker adaptation to overcome them. Also a classification of different speaker adaptation techniques developed is given in this chapter. Chapter 4 discusses the important issue of the application of Bayesian training to Hidden Markov Models as a means of adaptation, which has been used extensively during this research. In this chapter, the prior parameter estimation problem in Bayesian adaptation has also been discussed and a few solutions to this problem are suggested.

Chapter 5 includes a detailed discussion on the Regression-based Model Prediction approach to speaker adaptation. This approach uses a linear regression method to predict undertrained model parameters using the parameters of the better-trained models. In Chapter 6 the experimental evaluation of the above adaptation techniques is discussed. Furthermore, the use of available continuous speech databases in this work and several other issues concerned with the experimental implementations and evaluations are explained.

Finally, a review of the adaptation techniques introduced and the experimental results discussed in this thesis are given in Chapter 7 which concludes and gives some suggestions for the extension of the work and future research.

Chapter 2

Automatic Speech Recognition Using HMMs

The use of hidden Markov models in speech recognition has dramatically improved the performance of speech recognisers during recent years. In this chapter, a brief introduction to modern automatic speech recognition using hidden Markov models and the developments of these systems to improve their capabilities and overcome several speech recognition problems is given. Also, an overview of the speech parametrisation techniques used in continuous speech recognition is included in the first section.

2.1 Speech Signal Representations

In any modern speech recogniser, the speech signal is first converted into a parametric representation. This results in a sequence of parameter vectors representing the speech waveform, which is assumed to be stationary during the frame covered by each single vector. The parametrisation technique used has an indisputable influence on the output of the recogniser (for example see [86]). In order to keep the parametrised data compact, to eliminate the storage problems and also so as not to dispose of phonetically important information in the speech signal, special feature analysis techniques are used.

Many of the current HMM-based continuous speech recognition systems implement the following basic steps in their acoustic feature analysis section:

1. Sampling. In order not to lose the important information available in the speech, an appropriate sampling frequency should be chosen. Due to the sampling theory, this frequency should be at least twice the highest frequency in the speech that should be preserved. Depending on the recording conditions of the speech signal, a sampling frequency between 8 to 16 KHz is usual. In some systems, an extra pre-emphasis step is carried out by digital filtering to spectrally shape the signal [61].

2. Setting Frames. A predefined number of consecutive samples are grouped as a frame,

with a certain amount of frame overlap by setting the spacing of the frames. This allows for the speech signal within that frame to be assumed to be stationary.

3. Windowing. A window function is applied to the signal at this stage to overcome the possible effects of the assumptions made so far in cutting the frame from the running signal. A Hamming window is used for this purpose (for specifications refer to Appendix B).

4. MFCC computations. Mel-Frequency Cepstral Coefficients (MFCCs) have shown superior performance in speech recognition compared to other parametrisation techniques [22]. For MFCC computation, a number of usually highly overlapped, triangular bandpass filters are used. These filters are normally equally-spaced along the mel-scale. The output of each filter can be considered as representing the energy of the signal within the passband of that filter. A discrete cosine transform is applied to the log-energy outputs of these filters in order to calculate the MFCC parameters [22] as shown in Equation B.4 in appendix B.

5. Inclusion of energy and transitional parameters. Rabiner and Wilpon [88] have reported improvements in overall recognition performance of isolated and connected word recognition using log-energy parameters, while it is believed that the performance improvement due to the inclusion of these parameters for continuous speech is less significant [61]. Furui [33] and Soong and Rosenberg [95] have shown that the joint use of instantaneous and first order transitional features can lead to improved speech signal representation. It has also been shown by Ney [78], Wilpon et al. [102] and Lee et al. [61] that the use of second order cepstral time derivatives can significantly improve the performance of a speech recognition system. Lee et al. [61] have also reported on improvements due to the use of differential log-energy parameters. In order to benefit from all the above, usually the log-energy is added as an extra parameter to the set of cepstral parameters and the first and second order time derivatives are also calculated and included.

2.2 HMMs in Speech Recognition

Hidden Markov models are statistical models which can represent parametric random processes. Speech signals have been found to be quite well represented by these statistical models and the parameters of such models can be automatically estimated.

The theory behind HMMs was developed by Baum and his colleagues, in a series of papers, such as [6, 4]. The first applications of the HMMs to the problem of speech recognition were reported by Baker [3] and Jelinek [52].

2.2.1 Specifications of HMMs

A hidden Markov model is a probabilistic finite state machine, i.e. a set of states, connected to each other by transition links, with probabilities associated with each link. At any specified time, the system can be considered to be in one of the available states and at regular intervals, a transition to another state occurs (or the same state if a self transition is available), according to the probabilities associated with the transitional links. Associated with each state, there also exists a probability density function (p.d.f.) which defines the probability of emitting an observation vector, once an HMM state is entered. If a given speech signal is parametrised into T observation vectors, represented by $\boldsymbol{O} = \{\boldsymbol{o}_1, \boldsymbol{o}_2, \ldots, \boldsymbol{o}_T\}$, then it would be possible to calculate the likelihood of generating this speech signal using any sequence of states, given the above HMM.

According to above discussion, in an HMM system, the sequence of states for generating the given speech signal is not known. This is in fact one of the main problems associated with the HMMs, i.e. finding the optimal state sequence for a given sequence of observations, and is the reason for calling these models *hidden* Markov models.

In the current work, the HMM specifications used are those supported by the HTK toolkit for continuous speech recognition [112]. These specifications limit the general HMM architecture in the following ways:

1. The observation densities are continuous multivariate Gaussians.

2. The model topology is one with non-emitting entry and exit states.

An extra limitation is generally present in the models used in this research which is the use of a left-to-right model topology. An example of such an HMM is given in Figure 2.1. Such HMMs are defined by a set of parameters known as $\boldsymbol{\lambda} = (\boldsymbol{A}, \boldsymbol{\theta})$, where

- $\boldsymbol{A} = \{a_{ij}\}$ is the state transition probability matrix, where

$$a_{ij} = P(s_{t+1} = j | s_t = i), \qquad 2 \leq i, j \leq N - 1 \tag{2.1}$$

  which covers transition probabilities between emitting states.

- $\boldsymbol{\theta} = \{b_i(\boldsymbol{o}_t)\}$ is a vector of parameters defining the state output p.d.f. where

$$b_i(\boldsymbol{o}_t) = P(\boldsymbol{o}_t | s_t = i), \qquad 2 \leq i \leq N - 1. \tag{2.2}$$

The non-emitting states in the above-mentioned HMM topology are provided to facilitate the connection of HMMs in continuous speech recognition, which will be discussed later, and are called the entry and exit states of the HMM. Due to the use of this special HMM topology, the general definition of the HMM parameters has changed somewhat, i.e. the initial state occupation probability usually used in HMM parameter definitions (e.g. see [85]) is not used, but two transition probabilities have been defined as follows

$$a_{1i} = P(s_{t=1} = i), \qquad i = 2, \ldots, N-1 \tag{2.3}$$
$$a_{iN} = P(s_{t=T} = i), \qquad i = 2, \ldots, N-1 \tag{2.4}$$

which are the probabilities of making a transition from the entry state to any other state and from any other state to the exit state. Hence, in order to cover the case of entry and exit states, either the above definition of the parameter set $\boldsymbol{\lambda}$ should be changed to include a_{1i} and a_{iN}, or the parameters $\boldsymbol{A}$ should be modified to also include these extra transitions. Here, the first case will be used by adding these two transition parameters to the above set of HMM parameters $\boldsymbol{\lambda}$.

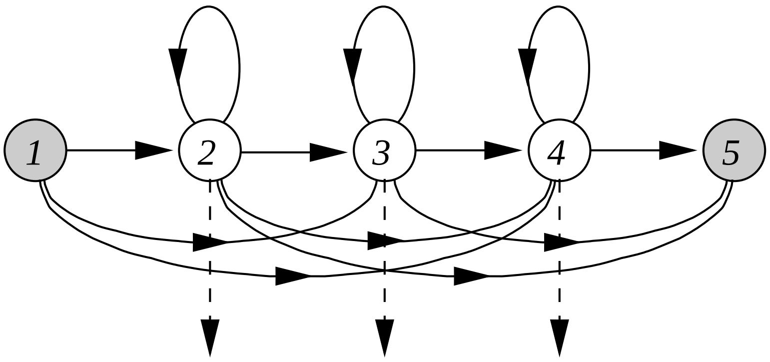


Figure 2.1: A simple 5-state left-right version of the proposed HMM structure with non-emitting entry and exit states.

The use of continuous mixture densities allows direct precise modelling of the speech parameters, which can lead to better recognition results compared to discrete densities [87, 63]. In an N-state HMM topology, these observation densities are represented by

$$b_i(\boldsymbol{o}_t) = \sum_{k=1}^{K} \omega_{ik} \mathcal{N}(\boldsymbol{o}_t; \boldsymbol{\mu}_{ik}, \boldsymbol{\Sigma}_{ik}), \qquad 2 \le i \le N-1 \tag{2.5}$$

where ω_{ik} are the mixture weights and $\boldsymbol{\mu}_{ik}$ and $\boldsymbol{\Sigma}_{ik}$ are the mean vectors and covariance matrices of the multivariate Gaussian mixture components $\mathcal{N}$, defined by

$$\mathcal{N} = \frac{1}{\sqrt{(2\pi)^V |\boldsymbol{\Sigma}_{ik}|}} \exp[-\frac{1}{2}(\boldsymbol{o}_t - \boldsymbol{\mu}_{ik})' \boldsymbol{\Sigma}_{ik}^{-1} (\boldsymbol{o}_t - \boldsymbol{\mu}_{ik})] \tag{2.6}$$

where V represents the dimension of the feature vector.

Also, although densities with full covariance matrices require fewer Gaussians to perform as well as those with diagonal covariance matrices, in most practical implementations, due to the large reduction in computations and memory needed, a diagonal covariance matrix is usually used [87]. Note that an HMM state with a mixture density is equivalent to a multi-state single Gaussian density model [54]. Hence, the introduction of mixture densities does not affect the mathematical derivations.

2.2.2 The Basics of HMM Speech Recognition

In a recognition system, a set of observations are assumed to be available and representing a certain word, or in general an utterance, to be recognised. These can be denoted by

$\boldsymbol{O} = \{\boldsymbol{o}_1, \boldsymbol{o}_2, \ldots, \boldsymbol{o}_T\}$, where $\boldsymbol{o}_t$ is the observation vector at time t. Then, given a limited vocabulary of words, the recognition task can be carried out by computing (e.g. see [85, 112])

$$\arg\max_i \{P(w_i|\boldsymbol{O})\} \tag{2.7}$$

where w_i is the ith word of the vocabulary. Using the Bayes rule, one can write

$$P(w_i|\boldsymbol{O}) = \frac{P(\boldsymbol{O}|w_i)P(w_i)}{P(\boldsymbol{O})}. \tag{2.8}$$

Hence, since $P(\boldsymbol{O})$ may be considered to be constant for a given input, the recognition task involves finding the model which maximises $P(\boldsymbol{O}|w_i)P(w_i)$ in place of $P(w_i|\boldsymbol{O})$. In such problems, the prior probabilities $P(w_i)$ are usually determined by a language model, hence the initial probability calculation now reduces to finding the class conditional observations $P(\boldsymbol{O}|w_i)$.

Assuming that the sequence of observations corresponding to each word w_i is generated by a hidden Markov model h_i, to solve the problem of speech recognition, one needs to be able to compute $P(\boldsymbol{O}|h_i)$, where it is assumed that

$$P(\boldsymbol{O}|h_i) = P(\boldsymbol{O}|w_i). \tag{2.9}$$

Due to the hidden nature of the state sequence in a hidden Markov model, assuming a particular state sequence $\boldsymbol{s}$ so that

$$\boldsymbol{s} = \{s_1, s_2, \ldots, s_T\}, \tag{2.10}$$

this likelihood can be calculated as follows

$$\begin{aligned} P(\boldsymbol{O}|h_i) &= \sum_{\boldsymbol{s}\in\boldsymbol{S}} P(\boldsymbol{O}|\boldsymbol{s}, h_i)\ P(\boldsymbol{s}|h_i) && (2.11)\\ &= \sum_{\boldsymbol{s}\in\boldsymbol{S}} a_{s_0 s_1} \prod_{t=1}^{T} a_{s_t s_{t+1}} b_{s_t}(\boldsymbol{o}_t), && (2.12) \end{aligned}$$

where $\boldsymbol{S}$ is the set of all possible state sequences of length T in model h_i and s_0 and s_{T+1} are the above mentioned entry and exit states of the model.

This computation, however, is not trivial, even for small numbers of states (N) and time frames (T), since at every time instant, $N-2$ states can be reached (not counting a skip transition to the exit state), leading to $(N-2)^T$ possible state sequences, with about $2T$ calculations for each sequence.

The efficient calculation of the above likelihood can be carried out by the so-called *Forward-Backward* procedure, which is a recursive algorithm and will be discussed in Section 2.2.3. However, only the forward pass of this algorithm is enough in this case to calculate $P(\boldsymbol{O}|h_i)$.

2.2.3 The Training Problem

A major problem in HMM speech recognition is the issue of model training. The training problem involves the estimation of model parameters $\boldsymbol{\lambda} = (\boldsymbol{A}, \boldsymbol{\theta})$, given the observation sequence $\boldsymbol{O} = \{\boldsymbol{o}_1, \boldsymbol{o}_2, \ldots, \boldsymbol{o}_T\}$ as the training data, so that $P(\boldsymbol{O}|\boldsymbol{\lambda})$ is maximised.

Typically, the method used to achieve this is *Maximum Likelihood* (ML) estimation. For the case of incomplete data (such as a hidden state sequence) ML estimation can be computed using the *Expectation-Maximisation* (EM) algorithm [24, 89]. The particular instance of the EM algorithm for HMMs is known as *Baum-Welch algorithm*, developed by Baum and his colleagues [5, 6, 4]. The EM algorithm is an iterative approach to maximum likelihood calculation which is used to find an estimate of the parameter set $\boldsymbol{\lambda}$ in each pass and then tries to maximise the likelihood of generating the training data using the model, so that the new likelihood is greater or equal to the previous one. Having defined the state sequence $\boldsymbol{s}$ as belonging to a state sequence space $\boldsymbol{S}$, which includes all possible state sequences, the maximisation of the above likelihood can be achieved by maximising an auxiliary function given by

$$Q(\boldsymbol{\lambda}, \hat{\boldsymbol{\lambda}}) \triangleq \sum_{\boldsymbol{s} \in \boldsymbol{S}} P(\boldsymbol{O}, \boldsymbol{s}|\boldsymbol{\lambda}) \log P(\boldsymbol{O}, \boldsymbol{s}|\hat{\boldsymbol{\lambda}}). \tag{2.13}$$

The convergence of the above algorithm was first proved by Baum [6]. Later, Liporace [70] extended it to the case of the elliptically symmetric multivariate distributions and vector distributions and Juang further widened the scope of the algorithm, and also included the concept of mixture distributions [53, 54].

The above auxiliary function can be expanded and decomposed into separate auxiliary functions which can then be maximised independently to give the state transitions, mixture weights and Gaussian parameters for the proposed HMM. This leads to the following equations for the estimation of mixture weights and Gaussian parameters

$$\hat{\boldsymbol{\mu}}_{ik} = \frac{\sum_{t=1}^{T} L_{ik}(t)\boldsymbol{o}_t}{\sum_{t=1}^{T} L_i(t)} \tag{2.14}$$

$$\hat{\boldsymbol{\Sigma}}_{ik} = \frac{\sum_{t=1}^{T} L_{ik}(t)(\boldsymbol{o}_t - \boldsymbol{\mu}_{ik})(\boldsymbol{o}_t - \boldsymbol{\mu}_{ik})'}{\sum_{t=1}^{T} L_i(t)} \tag{2.15}$$

$$\hat{\omega}_{ik} = \frac{\sum_{t=1}^{T} L_{ik}(t)}{\sum_{t=1}^{T} L_i(t)} \tag{2.16}$$

where $L_{ik}(t)$ is the *a posteriori* probability of being in mixture k of state i at time t and $L_i(t)$ is the *a posteriori* probability of being in state i at time t. Equations for the estimation of state transition probabilities can also be found.

However, for the calculation of the above parameters, the *a posteriori* probabilities mentioned above need to be found. This can be computed efficiently using the Forward-Backward algorithm. A forward variable $\alpha_i(t)$ is defined as

$$\alpha_i(t) = P(\boldsymbol{o}_1, \boldsymbol{o}_2, \cdots, \boldsymbol{o}_t,\ s_t = i|\boldsymbol{\lambda}). \tag{2.17}$$

This is the likelihood of being in state i at time t, having seen the first t observation vectors. Then the following procedure could be followed:

1. Initialisation: Given that the entry and exit states are non-emitting, the initial conditions would be

$$\alpha_1(1) = 1 \tag{2.18}$$
$$\alpha_i(1) = a_{1i}b_i(\boldsymbol{o}_1), \qquad 2 \le i \le N-1 \tag{2.19}$$

and also for the final condition

$$\alpha_N(T) = \sum_{j=2}^{N-1} \alpha_j(T)a_{jN} \tag{2.20}$$

2. Induction:

$$\alpha_i(t) = \left[\sum_{j=2}^{N-1} \alpha_j(t-1)a_{ji}\right] b_i(\boldsymbol{o}_t), \qquad \begin{array}{l} 2 \le i \le N-1 \\ 2 \le t \le T \end{array} \tag{2.21}$$

3. Termination:

$$P(\boldsymbol{O}|\boldsymbol{\lambda}) = \alpha_N(T). \tag{2.22}$$

The forward probability can be calculated recursively for all states of the HMM using the induction step, at any time t, starting from $t = 2$ and continuing towards $t = T$, where the termination step is reached.

The backward probability, can also be calculated similarly. Considering the backward variable to be defined as

$$\beta_i(t) = P(\boldsymbol{o}_{t+1}, \boldsymbol{o}_{t+2}, \cdots, \boldsymbol{o}_T | s_t = i, \boldsymbol{\lambda}), \tag{2.23}$$

which is the likelihood of observing the sequence from $t+1$ to the end, given that the process is in state i at time t. A similar recursive procedure can also be used to calculate $\beta_i(t)$:

1. Initialisation:

$$\beta_i(T) = a_{iN}, \qquad 2 \le i \le N-1 \tag{2.24}$$

2. Induction:

$$\beta_i(t) = \sum_{j=2}^{N-1} a_{ij}b_j(\boldsymbol{o}_{t+1})\beta_j(t+1), \qquad \begin{array}{l} 2 \le i \le N-1 \\ 1 \le t \le T-1 \end{array} \tag{2.25}$$

3. Termination:

$$\beta_1(1) = \sum_{j=2}^{N-1} a_{1j} b_j(\boldsymbol{o}_1) \beta_j(1). \tag{2.26}$$

From the above definitions, one can write

$$\alpha_i(t)\beta_i(t) = P(\boldsymbol{O}, s_t = i|\boldsymbol{\lambda}). \tag{2.27}$$

Since

$$L_i(t) = P(s_t = i|\boldsymbol{O}, \boldsymbol{\lambda}), \tag{2.28}$$

then, it can be written as

$$L_i(t) = \frac{1}{P(\boldsymbol{O}|\boldsymbol{\lambda})} \alpha_i(t)\beta_i(t). \tag{2.29}$$

Similarly $L_{ik}(t)$ for the case of mixture Gaussians can be derived with slightly more effort.

The probabilities found from the application of the Forward-Backward algorithm explained above can then be used in Equations (2.14) to (2.16) in order to estimate new model parameters. Several iterations of the above algorithm may be carried out to achieve the desired parameter training.

2.2.4 The Recognition Problem

Viterbi decoding [100] is usually used to find the highest probability state sequence for generating a given set of observations. This can be done using the same basic algorithm used for forward probability calculation, except that in this case, a maximisation has to be carried out on probabilities in place of summation, since the single best path is to be found. One can define a parameter $\delta_i(t)$ as

$$\delta_i(t) = \max_{s_1,\ldots,s_{t-1}} P[s_1 s_2 \ldots s_{t-1},\ s_t = i, \boldsymbol{o}_1 \boldsymbol{o}_2 \ldots \boldsymbol{o}_t|\boldsymbol{\lambda}], \tag{2.30}$$

i.e. the highest likelihood, along a path, of observing the first t speech vectors and reaching state i at time t. In a similar way to Equation (2.21), one can write

$$\delta_i(t) = [\max_j \delta_j(t-1) a_{ji}]\ b_i(\boldsymbol{o}_t). \tag{2.31}$$

The initial conditions in this case are

$$\delta_1(1) = 1 \tag{2.32}$$

$$\delta_i(1) = a_{1i} b_i(\boldsymbol{o}_1), \qquad 2 \le i \le N-1. \tag{2.33}$$

The termination step then would be

$$\delta_N(T) = [\max_j \delta_j(T) a_{jN}]. \tag{2.34}$$

The single best path can then be retrieved by computing the above highest likelihood at every time instance and keeping track of the arguments that maximised it. It should be noted that both here and in Forward-Backward calculations, direct calculation of the resultant probabilities, which are the product of a large number of probabilities, might lead to computation underflow. Hence, all the probabilities are usually handled in logarithmic form.

2.2.5 Recognition Units

A basic problem in speech recognition is to select the appropriate units for modelling speech. Since the final output of the recogniser is usually a string of words, one natural candidate for this purpose could be the whole words. However, any smaller unit can similarly be used for this purpose.

2.2.5.1 Whole Word Modelling

As stated, one possible candidate for a recognition unit is the whole word. This has been the unit of choice for many early speech recogniser systems e.g. [87]. In this framework, each HMM is used to model a whole word and the speech recogniser can be used to recognise unknown utterances, using these isolated word models. The implementation of recognition algorithm using these models will be discussed together with the case of sub-word models in Section 2.2.6.

However, extending the task of speech recognition to more difficult problems, such as large vocabularies, makes the system more complicated due to the huge number of models required (i.e. proportional to vocabulary size). Hence a more appropriate recognition unit is required.

2.2.5.2 Sub-Word Modelling

Sub-word modelling of the speech units can be performed at different linguistic levels, such as syllable, diphone and phone. However, a common speech unit, the phone, seems to be the most appropriate one for the larger tasks, since the number of basic phones in a language is quite small and this small number of phones can be used to construct any word in that language. This is usually done by combining any number of these phone models to make up the desired word model.

Sub-word models show many advantages over whole word models such as

- Reduce the memory and computation requirements of the system.
- Reduced system size and number of parameters leads to better training of those parameters since more effective use is made from limited amounts of training data.
- Allow the system to recognise words unseen in the training data, which is a typical problem in large vocabulary speaker independent speech recognition.
- Allow modelling of inter-word co-articulation variabilities.

Limiting the number of acoustic models to those corresponding to basic phones has some disadvantages too. For example, the acoustic specifications of a phone can greatly change in the vicinity of other phones. Thus, the use of a limited number of models for the basic phones will have the undesirable effect of poor modelling in different contexts. This problem will be addressed in Section 2.2.7. Also, in the case of whole word modelling, the recognition structure could be simpler due to the direct use of a word lexicon.

2.2.6 Continuous Speech Recognition

The above sub-word speech units are appropriate tools for continuous speech recognition. The HMM training procedure discussed previously can be extended for use in this case, taking into account the availability of data for several models in the training utterance. A composite HMM can be built by joining the HMMs corresponding to the training utterance, according to the training transcriptions. This eliminates the need to exactly specify the word boundaries in continuous speech, which is not trivial, and the usefulness of the non-emitting entry and exit states should be clear in this stage.

While the change to continuous speech recognition is not very difficult to implement at the training stage, many more changes need to be carried out before recognition is possible. However, the principle of finding the best path through a composite network of HMMs is exactly the same. An example of such an approach which can be followed in this case is *token passing* [109]. This method is designed to make the implementation of Viterbi algorithm easier. It can be used with both whole word and sub-word models and hence can easily be extended to the case of continuous speech recognition. Briefly, in this method, a token is assigned to each state i of an HMM at time t, containing the log likelihood of that state having generated all the observation vectors up to that time according to (2.31), together with the token's route through the network. The extension of the path in the Viterbi algorithm is then replaced by passing the token to the next HMM state at each time frame, updating its log probability and then discarding all but the highest probability token in every state.

Transition of tokens out of one model into another is performed in a similar fashion, with the exception that the identity of the model from which the token emerges is stored in a *link record*, together with the specification of the previous link record. These link records are then used in the final stage to trace back the path of the best matching token so as to find the best sequence of models.

2.2.7 Model Refinement

The last few sections gave a very brief overview of using continuous density HMMs in continuous speech recognition. However, there are several important issues that also affect the quality of the models and hence the accuracy of a speech recognition system. The issues of *context-dependent modelling* and *parameter tying* will be explained briefly here. There are also several other popular techniques for achieving more refined models, such as *model parameter interpolation*, *function-word modelling* [63, 64] and incorporation of *cross-word effects* [105] which are beyond the scope of this discussion.

2.2.7.1 Context Dependent Modelling

A problem which is more obvious in sub-word modelling, compared to whole word modelling case, is the variability in the acoustic specification of such units caused by context. This variability can have a large degrading effect on the performance of a speech recognition system.

A context-dependent system can be used to improve the acoustic modelling by modelling phones in different contexts separately [92, 64]. An example could be a triphone system which takes into account the left and right phonetic contexts in modelling a phone. This results in a large increase in the number of models which in turn can result in poor training of the models. However, it is generally agreed that context-dependent models can outperform context-independent ones if enough training data for robust estimation of the model parameters is provided.

2.2.7.2 Parameter Tying

A major problem with the training of the HMM systems is the problem of insufficient training data. Due to the finite size of the observation sequences, there could always exist model parameters with insufficiently trained parameters.

One way to overcome this problem is to increase the size of the training data. This is usually impractical. Another way would be to reduce the number of free model parameters to be trained. Although this may be possible in most cases, for example by reducing the number of states, number of mixture components, or using context-independent models in place of context-dependent ones, there are always practical reasons for using a certain model architecture.

One of the practical solutions which is used in many different situations to reduce the number of free parameters in an HMM system is to tie some of the parameters, i.e. let two or more models share some of their parameters. This can be implemented at different levels such as models [64], states [107, 111], or mixture components [47, 46]. The tied parameters are trained by pooling the available data for these parameters, wherever it occurs, leading to more robust parameter estimates. The tying process can be implemented based on prior linguistic knowledge of the models, or by assessing, mathematically, the similarities after some level of training has been achieved.

Parameter tying has been found very useful, especially for context-dependent models, to reduce the number of free system parameters and improve model training.

2.2.8 Other Issues Concerning HMM Speech Recognition

There are a number of other important issues concerning the speech recognition problem, two of which are listed below.

Large Vocabulary introduces several major problems to speech recognition. As it was pointed out earlier, in large vocabularies, words cannot be modelled individually and there will be a need for using sub-word units. Furthermore, with large vocabularies there is an increase in the number of confusable words. This can lead to performance degradation of the recogniser.

The complexity of search also increases which necessitates the use of pruning techniques, which can in turn increase the error level in the system output.

For the case of *unlimited vocabularies*, another problem is that of *Out Of Vocabulary* (OOV) words, which increases the error rate of the recogniser.

Grammar perplexity[1] is another factor concerning a speech recognition system. High grammar perplexities can result in an increase in the system error rate.

The use of an appropriate grammar can reduce the test set perplexity and plays a crucial role during recognition by preventing the search space from becoming extremely large by giving proper weights to the word sequences. Different types of grammar have been used in speech recognisers such as *bigrams* and *trigrams* and *word pair grammars*. However, it should be pointed out that the implementation of more complicated grammars is not trivial due to the extremely large search space and the complexity of search.

2.3 Summary

This chapter aimed to provide an introduction to the problem of speech recognition using HMMs and several important issues in this field.

The representation of speech signals in modern speech recognisers was discussed in the first section, while the rest of the chapter was dedicated to HMM speech recognition with the main focus being continuous density HMMs. The problems of model training and recognition were addressed and the importance of specifying appropriate recognition units explained. Also, the problems faced in continuous speech recognition and their usual solutions were counted. Finally, the use of context-dependent models, parameter tying and large vocabulary recognition were discussed.

[1]Perplexity is an information theoretic measurement of the average uncertainty at each decision point. It can roughly be considered as the average number of possible words following any string of $(N-1)$ words based on an N-gram language model and specifies the degree of sophistication in a recognition task.

Chapter 3

Speaker Differences and Adaptation

Speech recognition systems which are tuned to the speech of a specific speaker tend to perform less well on other speakers due to variations in speech characteristics. Speaker independent speech recognition, as the name implies, deals with any given utterance from any speaker, while speaker dependent recognition only deals with the speech from a single speaker. The usual approach to speaker independent system construction is by pooling data from several speakers. However, systems trained with such amounts of training data from each speaker fail to approach speaker dependent system performance. In this chapter, the sources of differences in the speech of different speakers are introduced and some of the adaptation techniques that try to overcome the above problem are discussed.

3.1 Speaker-Related Speech Differences

Factors that affect the recognition performance of a speech recognition system are numerous. However, they are generally divided into two groups: inter-speaker and intra-speaker.

3.1.1 Inter-Speaker Differences

Inter-speaker differences are those which usually originate from two main sources: anatomical differences and speaking habits of the speakers [2]. Anatomical differences are related to the natural differences between the shape, size and dynamic features of the vocal apparatus of different speakers, while the speaking habits are mostly concerned with the way a speaker has learned to speak. Such habits have a direct influence on the vocal tract dynamics resulting in co-articulation effects and variations in formant transition rates. However, according to the rather complex nature of speech production, this simple classification of the inter-speaker differences has been criticised by Nolan and an attempt has been made to formulate an overall model for speaker-specific information [79]. Nevertheless, this simple classification will be used for this discussion due to its usefulness in analysing the basic inter-speaker variations.

3.1.1.1 Anatomical Differences

The physiological differences have important influence on the fundamental frequencies of a speaker's speech, producing different acoustic features in different speakers.

The most extreme case in this category is the influence of the speaker's gender on the spectral parameters of speech. The analysis of the pronunciation of some vowels by male and female speakers has unveiled that their main difference is in having different formant frequencies. Some further studies have shown that vowel pronunciations also differ between genders; male vowels have a lower fundamental frequency, narrower formant bandwidths and perhaps a less steeply sloping spectrum [30].

Due to above differences a speaker dependent system trained on the speech from a male speaker generally will perform relatively poorly for a female speaker and vice versa.

3.1.1.2 Speaking Habits

The second source of differences is related to the speaking habits of a speaker such as rate of speech or accent.

While the average rate of speech might be different in different speakers, which is usually considered to be a personal characteristic, accent can be related to speaker's belonging to certain socio-economic communities or groupings, i.e. it is a linguistic phenomenon. This can be the main cause of differences between native and non-native speakers of a language, as well as between the native speakers from different geographical regions or communities.

For example, speakers from the north of England pronounce the long vowel /ɑː/ of southern England in words such as *bath* and *master* like the sound /ɑ/ in the word *trap* [101]. Also, in Manchester and Leeds, the pronunciation of the last vowel in the word *happy* changes from /ɪː/ to /ɪ/ as in the word *kit*. These differences are so important that sometimes they even give human listeners recognition problems.

3.1.2 Intra-Speaker Differences

There are also intra-speaker differences which cause changes in speech recognition performance. These differences include speaking rate, emotional state, health conditions of the speaker, etc. [2]. Speech recognisers have been found sensitive to these effects and a change in any of the above characteristics can result in performance degradation of a speech recognition system trained for that speaker.

In some fields such as speaker recognition, it is very important to distinguish between the inter- and intra-speaker differences, in order to be able to properly recognise a certain speaker among the others using the inter-speaker differences by factoring out the intra-speaker differences. Nevertheless, in such cases as speaker independent speech recognition both issues have to be taken into account. Any changes in the speech should be compensated for, whether it is due to change of speaker, or only due to changes in the speaker's speech generation conditions.

3.2 Speaker Adaptation

Speaker Adaptation is the general name for techniques which try to improve the modelling accuracy for a certain speaker with less speaker-specific data than needed to train a speaker dependent system. Although historically, work on speaker adaptation algorithms started only slightly after building the first generation of speech recognisers, with further improvements in speech recognition techniques, more attention has recently been paid to the speaker adaptation issue.

In this section, we will present a brief overview of different approaches to the problem of speaker adaptation and have a general discussion on the available techniques. The following terms are used in speaker adaptation and each defines the mode in which adaptation is carried out:

Incremental adaptation is where the adaptation data is available to the system sequentially, i.e. the adaptation algorithm continues to adapt the system to the new speaker as further adaptation data becomes available. This adaptation style is also known as *Dynamic Adaptation*.

Batch adaptation is when the adaptation is performed using all the data at once. This is also known as *Static Adaptation*.

Supervised adaptation is when the adaptation data is labelled correctly.

Unsupervised adaptation is the condition in which the adaptation data is not labelled and so any required labelling is automatic (e.g. from a recogniser) and not necessarily correct.

The usual choices are batch supervised, where adaptation data has been acquired in an enrolment session, or incremental unsupervised, where adaptation data is acquired as recognition proceeds. Under conditions where the amount of adaptation data cannot be predicted, or it is advantageous to use any further adaptation data that becomes available to the system, an incremental adaptation approach should be adopted. In such conditions both supervised and unsupervised adaptation might be possible, but unsupervised adaptation is more likely under normal circumstances.

3.2.1 Speaker Normalisation

Speaker normalisation algorithms, as illustrated in Figure 3.1, attempt to convert the characteristics of speech from a new speaker to those of a reference speaker, so that the

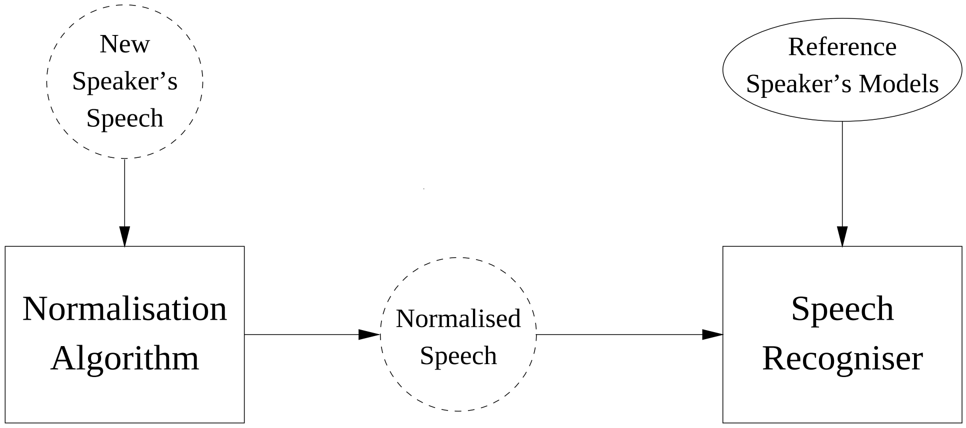


Figure 3.1: The block diagram for the basic speaker normalisation technique.

speaker dependent system already available for the reference speaker can be used for recognition of the utterances of the new one.

The normalisation idea seems to have originated from the natural fact that humans can recognise speech from a wide range of speakers of different age and gender without much difficulty. Researchers have concluded that there exists relationships between different characteristics of these speakers' speech signals that enable the human listener to understand the speech from any new speaker. Hence, it is believed that a transformation, or a series of transformations, might be able to provide the proper means for this conversion.

Several implementations of such techniques have been reported in the literature. These implementations include techniques which are also known as *Spectral Mapping* or *Feature Vector Transformation*. As an example Zhao [114, 115] has linearly transformed the speech spectra from a new speaker to those of a reference speaker, using a bias vector calculated in the logarithmic spectral domain and has reported some improvements due to acoustic normalisation section. Several other cases reporting improvements in the speech recognition performance by speaker normalisation via a spectral shift in the speaker's speech signal have also been reported e.g. [14, 75, 82, 99].

In some approaches, in place of applying the normalisation to the speech from the new speaker which is to be recognised, i.e. test data, it is applied to training utterances of several speakers to map them to the feature vector of the reference speaker and then train the HMM system using all the data from both the reference and other speakers. The resultant HMM system is then transformed for any new speaker using a transformation method with the transformation matrix being calculated from the adaptation data [58, 15].

In most of these cases simple linear functions have been used for the purpose of feature vector transformation, e.g. [20, 21, 74]. However, some researchers have used a piecewise linear function for this purpose [75, 8, 7], while Huang has implemented a non-linear function using a neural network to perform the normalisation [43, 44].

The main drawback in the case of speaker normalisation is that simple mapping functions are not able to provide large improvements in the system performance for the new speakers but more complex functions require more data in order to be well estimated. There also exists a problem of assessing the effectiveness of most of the speaker normalisation techniques reported. In most cases, there clearly exists an improvement in the performance of the SD system of the reference speaker on the new speaker, after the application of the normalisation technique. However, often no results from an SI system are given and hence it is very difficult to judge on the performance of the normalisation technique in comparison with SI system performance.

Recent research has tended to concentrate more on speaker adaptation of an SI system than speaker normalisation of an SD system.

3.2.2 Bayesian Adaptation

In Bayesian adaptation, *a priori* information is utilised to improve the adaptation procedure. This approach usually maximises *a posteriori* probabilities and hence is also called MAP (*Maximum a posteriori*) estimation, in contrast to Maximum Likelihood (ML) estimation.

One of the first attempts to use MAP estimation was to adapt the parameters of discrete density HMMs to new speakers [10]. Later work used versions of MAP estimation for continuous density HMM parameters [62, 76]. This was further developed by Gauvain and Lee to cover the parameters of HMMs with mixture Gaussian output distributions [34, 60, 36].

A simple MAP estimation of the HMM output distribution mean parameters can be considered as a simple interpolation between the ML estimated mean parameter and the prior mean parameter, each with an appropriate weight [62]. The prior parameters in this simple case are usually derived from the parameters of SI system. However, the estimation of good prior parameters is one of the most difficult problems in MAP estimation.

Recently, MAP estimation has attracted much attention, so that several applications of this technique have been implemented. Huo has reported the application of this technique to discrete and semi-continuous HMMs [48, 49]. One extension to MAP estimation is the approach of Stern, Rozzi and Lasry [91, 59, 96], where a class correlation factor, namely the contribution of other classes of parameters to the calculation of the parameters of the current class, is also included in the parameter estimation formulae. This approach is called *Extended MAP* (EMAP) and has recently been applied successfully to a large continuous speech database [113]. EMAP, however, has the disadvantage of needing very large correlation matrices and a large amount of computation for every new observation which can be prohibitive for larger systems [91], unless techniques to reduce the system parameter size are employed. Also, because of using the class correlations together with prior parameters in the main estimation formula, with small amounts of adaptation data, due to the larger weight, which is usually given to prior parameters in such cases, the correlations do not have much effect on the adaptation process.

MAP estimation is widely accepted among the researchers as a good adaptation technique due to a number of important characteristics. Implementation of MAP estimation can be regarded as an extension of ML estimation, which is widely used as a standard training procedure. It also has these desirable properties that its performance, usually starting from SI performance, tends to improve almost monotonically by introduction of adaptation data and approaches to SD performance with large amounts of adaptation data.

The MAP estimation technique will be discussed in detail in Chapter 4, thus further discussion on this subject will not be presented here.

3.2.3 Model Parameter Transformation Approaches

Speaker model parameter transformation is another widely used approach to speaker adaptation. Sometimes these methods are grouped with speaker feature vector transformation methods, which were described as being normalisation techniques in Section 3.2.1

One approach to transforming the model parameters is the method used by Shinoda [93]. A refined version of this approach was introduced by Ohkura [81] and Hattori and Sagayama [40] and was named *Vector Field Smoothing* (VFS). In this technique, a set of models for a reference speaker are trained and the available adaptation utterances are used to retrain these model parameters for a new speaker. Dividing the resultant model parameters into two trained and untrained subsets represented by K_1 and K_2 respectively, a transfer vector is calculated, relating the retrained and initial model parameters, i.e. for mean vectors, this is given by

$$\boldsymbol{v}_k = \boldsymbol{\mu}_k^R - \boldsymbol{\mu}_k^I, \qquad k \in K_1 \tag{3.1}$$

where $\boldsymbol{\mu}_k^R$ and $\boldsymbol{\mu}_k^I$ are the kth retrained and initial mean vectors respectively and $\boldsymbol{v}_k$ is the corresponding transfer vector, while k is a member of the trained subset of mean parameters K_1.

In the next step, the interpolation is carried out on the untrained subset of mean parameters, using new transfer vectors $\boldsymbol{v}_n$, so that

$$\boldsymbol{\mu}_n^R = \boldsymbol{\mu}_n^I + \boldsymbol{v}_n, \qquad n \in K_2 \tag{3.2}$$

where $\boldsymbol{\mu}_n^R$ and $\boldsymbol{\mu}_n^I$ are the nth mean vectors of the retrained and initial model sets, while n represents membership of the untrained subset of mean parameters K_2. Here, $\boldsymbol{v}_n$ is found using $\boldsymbol{v}_k$ for all $k \in K_1$ and a fuzzy membership function. In the final stage, all the transfer vectors are smoothed according to a new fuzzy membership function to obtain a new set of model parameters. The reported results show an improvement in the recognition performance when insufficient training data is provided. This approach has also been developed for the use of multiple reference speakers, where the best reference speaker is found according to the distances between the transfer vectors of all the reference speakers and the new speaker [80].

In an alternative approach, a set of transformation parameters $\boldsymbol{\theta} = [\boldsymbol{A}, \boldsymbol{b}]$ is estimated from the adaptation data by maximising the likelihood of the adaptation data given the corresponding word string using the Expectation-Maximisation (EM) algorithm [25, 26]. The estimated parameters are then used to find the adapted model parameters $\boldsymbol{x}$ from the SI model parameters $\boldsymbol{y}$ through the transformation

$$\boldsymbol{x} = \boldsymbol{A}\boldsymbol{y} + \boldsymbol{b}. \tag{3.3}$$

Other similar model parameter transformation approaches have also been reported [38, 39].

Jaschul [51] has used a linear matrix transformation to project the speech of a reference speaker to the new speaker using a *least squares* error approach to compute transformation matrix parameters. This approach is further investigated by Hewett [41], who used a transformation matrix to perform transformation in a DTW template matching system. The transformation matrix was found by a least squares regression method. Both approaches reported improvements in the system performance according to adaptation.

Leggetter and Woodland have also used a model parameter transformation approach to speaker adaptation called *Maximum Likelihood Linear Regression* (MLLR) [66]. MLLR is similar to Hewett's work in using the concept of transformation. However, here the transformation matrix is estimated using a maximum likelihood estimate in place of the least squares estimate in an HMM framework. In this approach linear regression transformation is used to map SI means, $\boldsymbol{\mu}_j$, of the mixture component j to the estimates of SD means, $\hat{\boldsymbol{\mu}}_j$, by

$$\hat{\boldsymbol{\mu}}_j = \boldsymbol{W}_j \boldsymbol{\nu}_j, \tag{3.4}$$

where $\boldsymbol{W}_j$ is the $n \times (n+1)$ transformation matrix and $\boldsymbol{\nu}_j$ is the extended mean vector,

$$\boldsymbol{\nu}_j = [1, \mu_{j_1}, \ldots, \mu_{j_n}]'. \tag{3.5}$$

The Equations 3.3 and 3.4 represent identical transformation actions, however, the transformation parameters are estimated differently. In this method, the regression matrix is estimated using the observation vector from the new speaker in a maximum likelihood estimation framework. Tying of transformation matrices is also carried out in this approach to reduce the number of parameters to be estimated from the adaptation data. This results in more robust estimates and can provide adaptation for the parameters not represented in the training data. A considerable performance improvement is reported by the application of this approach with moderate number of adaptation sentences in a continuous speech recognition task [69, 68]. A dynamic tying approach is also carried out to control the number of classes and the tied components within each class during the adaptation process, using a tree-based approach. This is reported to have further improved the adaptation process [68, 67].

In some approaches, the model parameter transformation is also accompanied by feature vector transformation to further improve the adaptation performance [13, 77].

The model parameter transformation techniques usually perform well in adapting models to a new speaker. The application of parameter reduction techniques, such as tying, can result in further performance improvement, especially in the case of systems with larger number of parameters. Not all transformation techniques have reported good results with very small or very large amounts of adaptation data, but some of them claim improvements with as small as 3 adaptation sentences in a continuous speech recognition task [65]. In the case of large number of adaptation sentences, some of these approaches, such as MLLR, can be considered as equivalent to retraining and hence approach SD system performance. Good performance with such data sizes has also been reported by Digalakis [26].

3.2.4 Speaker Clustering

The concept of speaker clustering has also been investigated for the purpose of speaker adaptation. The basic idea is to have several different model sets for different speakers. Then, each new speaker's acoustic features are compared to the available model set characteristics, which will determine the best model set to be used for that speaker in place of a general SI system. This is believed to improve the recognition performance for the new speaker.

The main problem in this case is the large variety of speakers available which makes the generation and storage of HMM sets impractical. A solution to this is to cluster speakers in a few groups according to their acoustic similarities and train a single model set for the speakers within that cluster. Allocation of a new speaker to the cluster with the closest acoustic characteristics can then be carried out.

Speaker clustering, for its obvious advantage over a standard SI system, has been used for speaker adaptation by some researchers. Lee [63] has implemented the basic idea using an agglomerative clustering technique. However, the results obtained were not satisfactory, which was believed to be due to the small number of available training data for each cluster compared to the data available for training the SI system. Imamura [50] has implemented the same idea by using a *cross-entropy* based distance measure for clustering the already trained speaker (hidden) Markov model sets. This technique has led to some improvements in the speech recognition performance.

A hierarchical tree clustering approach was utilised by Mathan [72] to assign the speakers to different clusters, where in each stage the HMM networks corresponding to the leaves of a binary tree are trained using the available speakers to that leaf. For a new speaker, the words uttered by him/her are used to descend the tree network until the depth of terminal network is reached.

Another tree-structured clustering approach was implemented by Kosaka [56], where at every step, the cluster with a maximum sum of Bhattacharyya distance between models is divided. The clustering process is controlled by a threshold and the allocation of speakers to clusters is achieved by finding the most likely cluster model sets for a new speaker.

Most of the above approaches to speaker clustering are similar in principle. The problems of undertrained model parameters within a cluster, uneven distribution of speakers in different clusters and unavailability of an appropriate cluster for any new speaker, especially when the number of clusters is kept small to overcome the problem of undertrained parameters, are the most important problems faced by this approach. However, the extent to which each case deals with any of these problems is also dependent on the database used for that purpose.

Gender dependent modelling can also be counted as a speaker clustering approach. It consists of dividing speakers into two gender groups (clusters) and preparing separate recognition systems for these groups by pooling only from the speakers of the same gender during training session. This technique, however, is slightly different to regular clustering techniques, since the allocation of any new speaker to the appropriate cluster can readily be done. Many researchers have reported improvements using this technique, e.g. [45, 35, 104], and its use is now widespread for improving speaker independent recognition.

3.2.5 Predictive Approaches

An approach to improve modelling can be the use of prediction for finding the model parameters which are not normally updated by standard adaptation or training techniques for reasons such as lack of training data. This approach has been implemented by a few researchers and has been found very useful in providing more accurate models.

Furui [32] introduced a technique based on the use of linear regression in a predictive fashion in order to use a fraction of a proposed vocabulary for training an isolated word recognition system. In this technique, multiple regression was used to estimate a regression matrix and a residual vector for relating the parameters of the log-area-ratios (LARs) of any phone from the whole vocabulary to any other phone from the fraction of vocabulary and reported improvements in the system performance.

Cox [17, 19, 18] has also used linear regression for the purpose of speaker adaptation and tried to predict model parameters for a test class consisting of half of the English alphabet using the parameters of models of the other half. Even with small number of enrolment classes, i.e. limited adaptation data, marked improvements in the results have been reported. This unveils the important fact that the relationships between different phones can be used in training unseen model parameters in order to improve adaptation performance. A predictive approach based on a similar idea, using multiple regression for extracting the phone model relationships and its use in improving Bayesian adaptation technique in continuous speech recognition has been developed which will be discussed in Chapter 5.

3.2.6 Other Approaches

Another recent development in the area of speaker adaptation has been to control the number of free parameters in the system according to the amount of available adaptation data to achieve optimum performance. As mentioned in Section 3.2.3, Leggetter and Woodland have used a tree-based tying approach to improve their adaptation technique [67]. A similar method has been applied to a model parameter transformation adaptation technique, while this linear transformation is confined to a simple shift in the value of the mean parameters [94]. Using a tree structure for the level of tying of parameters, values of shifts for all leaf and non-leaf nodes are calculated and the same tree is used to control the level of tying by setting a threshold on the amount of available node data. Another similar idea has been used for controlling the smoothing rate, in place of the number of system parameters, in a VFS approach, according to the amount of adaptation data available [98]. Both methods have been reported effective in balancing the system parameter or adaptation conditions according to available adaptation data.

3.3 Summary and Discussion

In this chapter, several issues concerned with speaker adaptation were discussed. The differences among speakers and their sources and also changes in the speaking conditions of a speaker, which could result in performance degradation of a speech recognition system were outlined.

These issues are the main targets of a speaker adaptation system. Several important speaker adaptation techniques were also discussed and examples of each were given. Speaker normalisation techniques can help in adaptation to some extent, but they suffer from two disadvantages: global transformation and need for a reference SD system. Some feature vector transformation techniques have partly overcome these disadvantages by performing local transformations in place of a global one.

Bayesian adaptation has attracted many researchers because of its continuing improvement with the increase in the amount of adaptation data and its desirable convergence towards speaker dependent system performance. However, like some other model adaptation techniques, it only updates seen parameters, i.e. the parameters for which there exists adaptation data. Model parameter transformation approaches have also shown good adaptation performance, but again in the case of limited training data they might suffer the same performance degradation problem, because they only rely on the information extracted from the adaptation utterances. However, they may be able to perform better by taking into account the relationships learnt from larger training data sets. Although these approaches do not claim the same asymptotic characteristics of MAP estimation, they can be equivalent to model retraining with enough transformations and experimentally have been found to be able to equal SD performance on ARPA RM1 data [69].

Many researchers have achieved better adaptation performance by applying more than one of the techniques discussed above. As an example, a feature vector transformation can be used in conjunction with a model parameter transformation to achieve improved results. Some researchers have also tried to optimise the system size, or the adaptation procedure according to the size of the data available.

In the following chapters, several approaches to improved speaker adaptation will be introduced and the adaptation results presented and discussed.

Chapter 4

Bayesian Adaptation of CDHMMs

The Bayesian approach to the estimation problem in statistics takes into account the existence of prior information and utilises it in the process of estimation. This technique has recently received much attention for different estimation problems. A potential problem, where such an approach can be useful, is the estimation of the *Continuous Density HMM* (CDHMM) parameters. The application of this technique to the problem of speaker adaptation for the case of CDHMMs is the subject of this chapter and several important issues such as prior parameter estimation are discussed in detail.

4.1 Introduction

Since in speaker adaptation, the system is expected to adapt to a new speaker using a limited amount of adaptation data, the problem of *sparse training data* is often faced. However, HMM training procedures such as *Maximum Likelihood* (ML) estimation are optimal only under certain conditions, one of which is the availability of large amounts of training data. Hence, standard ML training of HMM parameters, in the presence of sparse training data, is found unsuccessful (for examples, refer to Section 6.3.1.2).

A successful technique, recently applied to the problem of speaker adaptation, is the Bayesian training of the HMM parameters for the new speaker in a framework known as *Maximum a posteriori* (MAP) estimation [10, 62, 36, 48]. In this approach, the application of prior parameters leads to better estimates of the HMM parameters, compared to conventional estimation techniques. However, some issues such as prior parameter estimation remain as difficult tasks due to the complexity of the prior distributions used.

In this chapter, after the introduction to the problem of MAP estimation of the parameters of multivariate single Gaussian and mixture Gaussian distributions and CDHMMs, the issue of the estimation of the prior parameters is discussed and a technique for improving the prior parameter estimation for the purpose of speaker adaptation in a CDHMM-based speech recogniser suggested.

4.2 MAP and ML estimation

The use of a prior distribution of the parameters to be estimated is the fundamental difference between MAP and ML estimations. If $\boldsymbol{O} = \{\boldsymbol{o}_1, \ldots, \boldsymbol{o}_T\}$ is a sequence of observations with a p.d.f. $P(\boldsymbol{O})$ and $\boldsymbol{\lambda}$ is the parameter set defining the distribution, given a sequence of training data $\boldsymbol{O}$, $\boldsymbol{\lambda}$ is to be estimated. If $\boldsymbol{\lambda}$ is assumed to be fixed but unknown, the maximum likelihood estimate for $\boldsymbol{\lambda}$ is found by solving

$$\frac{\partial}{\partial \boldsymbol{\lambda}} P(\boldsymbol{o}_1, \ldots, \boldsymbol{o}_T | \boldsymbol{\lambda}) = 0. \tag{4.1}$$

However, if $\boldsymbol{\lambda}$ is assumed random with *a priori* distribution function $P_0(\boldsymbol{\lambda})$, then the MAP estimate for $\boldsymbol{\lambda}$ is found by solving

$$\frac{\partial}{\partial \boldsymbol{\lambda}} P(\boldsymbol{\lambda} | \boldsymbol{o}_1, \ldots, \boldsymbol{o}_T) = 0. \tag{4.2}$$

Using Bayes theorem,

$$P(\boldsymbol{\lambda} | \boldsymbol{o}_1, \ldots, \boldsymbol{o}_T) = \frac{P(\boldsymbol{o}_1, \ldots, \boldsymbol{o}_T | \boldsymbol{\lambda}) P_0(\boldsymbol{\lambda})}{P(\boldsymbol{o}_1, \ldots, \boldsymbol{o}_T)}. \tag{4.3}$$

Thus, the MAP estimation procedure, compared to the maximum likelihood approach, involves a prior distribution function $P_0(\boldsymbol{\lambda})$ for the random parameter $\boldsymbol{\lambda}$.

It should also be noted that the use of informative priors, i.e. distributions which contain some information about the parameters to be estimated[1], is the main reason for the success of this algorithm. An important simplification in the process of MAP estimation can result if the concept of *conjugate families of distributions* can be used [23]. A conjugate prior for a random vector, drawn from a family of distributions for which there is a sufficient statistic of fixed dimension is defined as the prior distribution for the parameters of the p.d.f. of the random vector, such that the posterior distribution of these parameters and their prior distribution $P_0(\boldsymbol{\lambda})$ belong to the same distribution family for any sample size n and any values of the observation samples. For example the conjugate prior for the mean of the Gaussian density is known to also be a Gaussian density.

4.3 MAP Estimation for Single Gaussians

Before proceeding with MAP estimates for HMM parameters, MAP estimates for single and mixture Gaussians are discussed. Then MAP estimates for the HMM parameters are presented as their extension.

[1] This is in contrast to the case of *non-informative priors*, where no (or small) prior information is available, but where the use of Bayesian analysis is still desirable [9]. Further discussion on this subject is beyond the scope of this thesis.

4.3.1 MAP Estimation of the Gaussian Mean

Let $\boldsymbol{m}$ be the V-dimensional mean vector and $\boldsymbol{r}$ be the $V \times V$ precision matrix (inverse of the covariance) of the sample observation vector. If the mean $\boldsymbol{m}$ is assumed to be random with a prior distribution $P_0(\boldsymbol{m})$ and the precision $\boldsymbol{r}$ is assumed to be known and fixed, then the conjugate prior for $\boldsymbol{m}$ is also Gaussian with mean $\boldsymbol{\mu}$ and precision $\boldsymbol{r}'$. Using the concept of conjugate priors, the MAP estimate for $\boldsymbol{m}$ is

$$\begin{aligned} \tilde{\boldsymbol{m}} &= (\boldsymbol{r}' + T\boldsymbol{r})^{-1}(\boldsymbol{r}'\boldsymbol{\mu} + T\boldsymbol{r}\bar{\boldsymbol{o}}) \\ &= (\boldsymbol{r}' + T\boldsymbol{r})^{-1}\boldsymbol{r}'\boldsymbol{\mu} + (\boldsymbol{r}' + T\boldsymbol{r})^{-1}T\boldsymbol{r}\bar{\boldsymbol{o}} \end{aligned} \tag{4.4}$$

where T is the total number of samples observed and $\bar{\boldsymbol{o}}$ is the sample mean [23, 27]. Note that the MAP estimate of $\boldsymbol{m}$, is a weighted average of the prior mean $\boldsymbol{\mu}$ and the sample mean $\bar{\boldsymbol{o}}$, where the weights are functions of the parameters and the number of observations. When $T = 0$, the MAP estimate is simply the prior mean (no samples case). However, as T grows larger ($T \to \infty$), the MAP estimate converges to $\bar{\boldsymbol{o}}$ asymptotically (ML estimate). Also, if the elements of $\frac{1}{T}\,\boldsymbol{r}'$ become negligible in comparison to the elements of $\boldsymbol{r}$, then the MAP estimate is approximately equal to the ML estimate ($\bar{\boldsymbol{o}}$), which corresponds to the case of using non-informative priors.

Due to the use of conjugate priors, the posterior distribution is also Gaussian with the mean vector $\hat{\boldsymbol{\mu}} = \tilde{\boldsymbol{m}}$ and a precision matrix which can be written as [23, 27]

$$\hat{\boldsymbol{r}}' = \boldsymbol{r}' + T\boldsymbol{r}. \tag{4.5}$$

The above discussion on the effect of increasing the number of observed samples is valid here too. As would be expected, as T increases the posterior distribution is more concentrated around the sample mean.

4.3.2 MAP Estimation of both the Gaussian Mean and Precision

When the (full) precision matrix of the observation vector, $\boldsymbol{r}$, is unknown, the prior joint distribution can be considered as a normal-Wishart distribution [23]. If the precision matrix is diagonal, a normal-gamma distribution will be used for each component of the precision vector. This case will be dealt with in Section 4.4 in the context of mixture Gaussians.

4.4 MAP Estimation for Mixture Gaussians

In this section, the results of the previous section for the single Gaussian density case will be extended to the case of mixture Gaussians.

The *Expectation-Maximisation* (EM) algorithm which locally maximises the likelihood function of the observed data is used for the purpose of MAP estimation. Dempster et al. have shown that EM is also applicable to the case of MAP estimation [24].

Here, once again, the problem is to maximise the posterior distribution of a parameter vector, say $\boldsymbol{\theta}$, defined as

$$\boldsymbol{\theta} = (\omega_1, \ldots, \omega_K, \boldsymbol{m}_1, \ldots, \boldsymbol{m}_K, \boldsymbol{r}_1, \ldots, \boldsymbol{r}_K) \tag{4.6}$$

where ω_k is the mixture weight for the kth mixture component, $\boldsymbol{m}_k$ is the V-dimensional mean vector and $\boldsymbol{r}_k$ is the $V \times V$ precision matrix of the kth mixture of the multivariate normal density, while for the mixture weights, there exists a constraint of

$$\sum_{k=1}^{K} \omega_k = 1.$$

For the observation vector $\boldsymbol{O} = (\boldsymbol{o}_1, \ldots, \boldsymbol{o}_T)$ of T independent and identically distributed observation samples, the joint p.d.f. is

$$f(\boldsymbol{O}|\boldsymbol{\theta}) = \prod_{t=1}^{T} \sum_{k=1}^{K} \omega_k \mathcal{N}(\boldsymbol{o}_t|\boldsymbol{m}_k, \boldsymbol{r}_k) \tag{4.7}$$

where

$$\mathcal{N}(\boldsymbol{o}_t|\boldsymbol{m}_k, \boldsymbol{r}_k) \propto |\boldsymbol{r}_k|^{1/2} \exp[-\frac{1}{2}(\boldsymbol{o}_t - \boldsymbol{m}_k)' \boldsymbol{r}_k (\boldsymbol{o}_t - \boldsymbol{m}_k)]. \tag{4.8}$$

The specification of a joint conjugate prior density is restricted by the fact that there does not exist a sufficient statistic of fixed dimension for vector $\boldsymbol{\theta}$ [23]. However, $f(\boldsymbol{O}|\boldsymbol{\theta})$ can be expressed as the product of a multinomial density and multivariate normal densities. Hence, for this case, the appropriate candidate for the conjugate prior density of the mixture weight parameter vector is a Dirichlet density of the form

$$g(\omega_1, ..., \omega_K|\psi_1, ..., \psi_K) \propto \prod_{k=1}^{K} \omega_k^{\psi_k - 1} \tag{4.9}$$

where $\psi_k > 0$ are the components of the parametric vector of the Dirichlet distribution [23].

For $(\boldsymbol{m}_k, \boldsymbol{r}_k)$, due to the multivariate normal distribution of $\boldsymbol{O}$, the joint conjugate prior density is a normal-Wishart density of the form

$$g(\boldsymbol{m}_k, \boldsymbol{r}_k|\tau_k, \boldsymbol{\mu}_k, \alpha_k, \boldsymbol{u}_k) \propto |\boldsymbol{r}_k|^{(\alpha_k - V)/2} \exp[-\frac{\tau_k}{2}(\boldsymbol{m}_k - \boldsymbol{\mu}_k)' \boldsymbol{r}_k (\boldsymbol{m}_k - \boldsymbol{\mu}_k)] \exp[-\frac{1}{2} tr(\boldsymbol{u}_k \boldsymbol{r}_k)]. \tag{4.10}$$

Here $\tau_k > 0$ and $\boldsymbol{\mu}_k$ (vector of dimension V) are the multivariate normal prior density parameters and $\alpha_k > V - 1$ is the degree of freedom and $\boldsymbol{u}_k (V \times V$ matrix) is the precision matrix of the prior Wishart distribution [23].

The joint prior density $g(\boldsymbol{\theta})$, assuming independence between the parameters of individual mixture components and the mixture weights, could then be written as the product of these prior p.d.f.'s, i.e.

$$g(\boldsymbol{\theta}) = g(\omega_1, ..., \omega_K) \prod_{k=1}^{K} g(\boldsymbol{m}_k, \boldsymbol{r}_k). \tag{4.11}$$

Having defined the appropriate prior densities, the MAP estimates of the Gaussian mixture parameters can be found by the application of the EM algorithm. This consists of the maximisation of the auxiliary function $Q(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}})$ at each iteration [4, 24, 89], where $\hat{\boldsymbol{\theta}}$ is the current fit for $\boldsymbol{\theta}$. This leads to the following estimations of the Gaussian mixture parameters [36]

$$\tilde{\omega}_k = \frac{\psi_k - 1 + \sum_{t=1}^{T} c_{kt}}{\sum_{k=1}^{K} \psi_k - K + \sum_{k=1}^{K} \sum_{t=1}^{T} c_{kt}}, \tag{4.12}$$

$$\tilde{\boldsymbol{m}}_k = \frac{\tau_k \boldsymbol{\mu}_k + \sum_{t=1}^{T} c_{kt} \boldsymbol{o}_t}{\tau_k + \sum_{t=1}^{T} c_{kt}}, \tag{4.13}$$

and

$$\tilde{\boldsymbol{r}}_k^{-1} = \frac{\boldsymbol{u}_k + \sum_{t=1}^{T} c_{kt} (\boldsymbol{o}_t - \tilde{\boldsymbol{m}}_k)(\boldsymbol{o}_t - \tilde{\boldsymbol{m}}_k)' + \tau_k (\tilde{\boldsymbol{m}}_k - \boldsymbol{\mu}_k)(\tilde{\boldsymbol{m}}_k - \boldsymbol{\mu}_k)'}{\alpha_k - V + \sum_{t=1}^{T} c_{kt}} \tag{4.14}$$

where

$$\begin{aligned} c_{kt} &= \hat{\omega}_k f(\boldsymbol{o}_t | \hat{\boldsymbol{\theta}}_k) f^{-1}(\boldsymbol{o}_t | \hat{\boldsymbol{\theta}}) \\ &= \hat{\omega}_k \mathcal{N}(\boldsymbol{o}_t | \hat{\boldsymbol{m}}_k, \hat{\boldsymbol{r}}_k) f^{-1}(\boldsymbol{o}_t | \hat{\boldsymbol{\theta}}). \end{aligned} \tag{4.15}$$

The above equations are the MAP estimations of the parameters of a mixture Gaussian distribution. A full derivation of the above formulae is given in Appendix A.

4.4.1 Mixture Observations with Diagonal Covariance Matrices

If the covariance matrices of the multivariate Gaussian mixture distributions are assumed to be diagonal[2], then the joint conjugate prior density for (m_{kv}, r_{kv}), due to the Gaussian distribution of the elements of $\boldsymbol{o}_t$, could be considered to be a normal-gamma distribution [23] of the form

$$g(m_{kv}, r_{kv} | \tau_{kv}, \mu_{kv}, \alpha_{kv}, \beta_{kv}) \propto r_{kv}^{(2\alpha_{kv}-1)/2} \ \exp[-\frac{\tau_{kv} r_{kv}}{2} (m_{kv} - \mu_{kv})^2] \ \exp(-\beta_{kv} r_{kv}) \tag{4.16}$$

where $\tau_{kv} > 0$ and μ_{kv} are the prior normal distribution parameters and α_{kv} and β_{kv} are the parameters of the prior gamma distribution.

Substituting (4.10) with (4.16) for every element of the $\boldsymbol{o}_t$ vector and proceeding with the rest of the derivation would lead to the following re-estimation formulae for the mean and variance vector elements of the mixture Gaussian distribution

$$\tilde{m}_{kv} = \frac{\tau_{kv} \mu_{kv} + \sum_{t=1}^{T} c_{kt} o_{tv}}{\tau_{kv} + \sum_{t=1}^{T} c_{kt}}, \tag{4.17}$$

and

$$\tilde{r}_{kv}^{-1} = \frac{2\beta_{kv} + \sum_{t=1}^{T} c_{kt} (o_{tv} - \tilde{m}_{kv})^2 + \tau_{kv} (\tilde{m}_{kv} - \mu_{kv})^2}{2\alpha_{kv} - 1 + \sum_{t=1}^{T} c_{kt}}. \tag{4.18}$$

[2]In this part of discussion, for the sake of simplicity, in place of a $V \times V$ diagonal covariance matrix, a vector of dimension V of the diagonal elements is used.

4.5 MAP Estimation for HMM Parameters

The discussion carried out in the last section is the general case of the application of MAP estimation to mixture Gaussian densities. In order to apply this approach to CDHMMs, the method requires some further extensions.

In HMMs with finite mixture densities, due to the hidden nature of the process, there is a lack of sufficient statistic [23, 27] of a fixed dimension, which makes MAP estimation much more difficult. This is also the case with ML estimates under such conditions. ML estimates are usually obtained by locally maximising the likelihood function for the observed data using the EM algorithm. The main idea here is to use the same algorithm for MAP estimation, which, as stated earlier, has already been proved to be possible [24]. This was first carried out for a general HMM structure by Gauvain and Lee [36]. However, they have not implemented this approach using a Forward-Backward algorithm, but their implementation is based on the so-called *Segmental MAP estimation* which is named in analogy with *Segmental K-means* algorithm.

This work deals with HMMs with non-emitting states and hence is somewhat different to that presented in [36]. In order to extend the results obtained in the last discussion to the HMM parameters, a hidden Markov model with parameter vector $\boldsymbol{\lambda} = (\boldsymbol{A}, \boldsymbol{\theta})$ is considered[3] where similar to the definition in Section 2.2.1, $\boldsymbol{A}$ is the transition probability matrix with elements a_{ij} and $\boldsymbol{\theta}$ is the p.d.f. parameter vector, i.e. for each state i, $\boldsymbol{\theta}_i = \{\omega_{ik}, \boldsymbol{m}_{ik}, \boldsymbol{r}_{ik}\}_{k=1,\ldots,K}$.

At this point, it is clear that the transitions from or to the non-emitting states can be included in $\boldsymbol{A}$. However, for the sake of simplicity in notation, in current discussions, $\boldsymbol{A}$ will include only the transitions between any two emitting states, i.e. $\boldsymbol{A} = \{a_{ij}\}$, where $i, j = 2, \cdots, N-1$. Hence, there is a $(N-2) \times (N-2)$ matrix of state transition probabilities and the parameter vector of the HMM is extended to $\boldsymbol{\lambda} = (\boldsymbol{a}_1, \boldsymbol{a}_N, \boldsymbol{A}, \boldsymbol{\theta})$, where $\boldsymbol{a}_1$ and $\boldsymbol{a}_N$ refer to the vectors of elements in (2.3) and (2.4)[4].

In this case, for a vector of observation samples $\boldsymbol{O} = (\boldsymbol{o}_1, ..., \boldsymbol{o}_T)$, the complete data is $\boldsymbol{U} = (\boldsymbol{O}, \boldsymbol{s}, \boldsymbol{l})$, where $\boldsymbol{s} = (s_1, ..., s_T)$ is the unobserved state sequence, $\boldsymbol{l} = (l_1, ..., l_T)$ is the vector of unobserved mixture component labels where $s_t \in [1, N]$ and $l_t \in [1, K]$. Then the joint p.d.f. for the complete data is defined as

$$h(\boldsymbol{O}, \boldsymbol{s}, \boldsymbol{l}|\boldsymbol{\lambda}) = a_{1s_1}\left[\prod_{t=1}^{T} a_{s_{t-1}s_t}\omega_{s_t l_t} f(\boldsymbol{o}_t|\boldsymbol{\theta}_{s_t l_t})\right] a_{s_T N}. \tag{4.19}$$

If the parameters $\boldsymbol{A}$, $\boldsymbol{a}_1$ and $\boldsymbol{a}_N$ are assumed to be fixed and known, the prior density would be

$$G(\boldsymbol{\lambda}) = \prod_{i=2}^{N-1} g(\boldsymbol{\theta}_i) \tag{4.20}$$

[3]The parameter vector for ordinary HMM structures is usually defined as $\lambda = (\pi, A, \theta)$ where π is the initial probability vector. In line with the discussion in Chapter 2 this parameter will not be present in current work.

[4]Note that in the above discussion, to alleviate further complexity, the transition from the non-emitting entry state to the non-emitting exit state is not considered.

where $g(\boldsymbol{\theta}_i)$ is defined by (4.11). In a more general case, $\boldsymbol{A}$, $\boldsymbol{a}_1$ and $\boldsymbol{a}_N$ are assumed to be random and the MAP estimates for the observation density parameters, as well as the state transition probabilities, are to be derived. Once again, the proper candidates for the prior density of both the entry and exit states transition probability vectors $\boldsymbol{a}_1$ and $\boldsymbol{a}_N$ and also each row of the transition probability matrix $\boldsymbol{A}$ are Dirichlet densities. This choice is easily derived from the definition of the Dirichlet density [23] and the discussion of the last section, since $h(\boldsymbol{O}, \boldsymbol{s}, \boldsymbol{l}|\boldsymbol{\lambda}) = h(\boldsymbol{s}|\boldsymbol{\lambda})h(\boldsymbol{O}, \boldsymbol{l}|\boldsymbol{s}, \boldsymbol{\lambda})$, where $h(\boldsymbol{s}|\boldsymbol{\lambda})$ is the product of N multinomial densities with parameter sets $\{a_{12}, ..., a_{1(N-1)}\}$, $\{a_{2N}, ..., a_{(N-1)N}\}$ and $\{a_{i2}, ..., a_{i(N-1)}\}_{i=2,...,N-1}$. Thus, a relationship for all the HMM parameters can be derived as follows

$$G(\boldsymbol{\lambda}) \propto \prod_{i=2}^{N-1} \left[a_{1i}^{\eta_i - 1} a_{iN}^{\eta_i' - 1} g(\boldsymbol{\theta}_i) \prod_{j=2}^{N-1} a_{ij}^{\eta_{ij} - 1} \right] \tag{4.21}$$

where η_i, η_i' and η_{ij} are the elements of the parametric vectors of Dirichlet densities of the entry, exit and other state transition probabilities respectively. Here, the inner multiplication makes up the Dirichlet density for each row of $\boldsymbol{A}$, while the outer, extends it to all the $\boldsymbol{A}$ matrix.

In order to find the MAP estimates of HMM parameters using the EM algorithm, the auxiliary function Q, to be maximised in each iteration, is defined. As shown in Appendix A, it can be decomposed into the sum of four functions of the form

$$Q_{a_1}(\boldsymbol{a}_1, \hat{\boldsymbol{\lambda}}) = \sum_{i=2}^{N-1} \gamma_{i1} \log a_{1i} \tag{4.22}$$

$$Q_{a_N}(\boldsymbol{a}_N, \hat{\boldsymbol{\lambda}}) = \sum_{i=2}^{N-1} \gamma_{iT} \log a_{iN} \tag{4.23}$$

$$Q_A(\boldsymbol{A}, \hat{\boldsymbol{\lambda}}) = \sum_{t=1}^{T} \sum_{i=2}^{N-1} \sum_{j=2}^{N-1} P(s_{t-1} = i, s_t = j|\boldsymbol{O}, \hat{\boldsymbol{\lambda}}) \log a_{ij} \tag{4.24}$$

$$Q_\theta(\boldsymbol{\theta}, \hat{\boldsymbol{\lambda}}) = \sum_{t=1}^{T} \sum_{i=2}^{N-1} \sum_{k=1}^{K} P(s_t = i, l_t = k|\boldsymbol{O}, \hat{\boldsymbol{\lambda}}) \log \omega_{ik} f(\boldsymbol{o}_t|\boldsymbol{\theta}_{ik}), \tag{4.25}$$

where $\gamma_{it} = P(s_t = i|\boldsymbol{O}, \hat{\boldsymbol{\lambda}})$. The derivation of the equations for MAP estimation of HMM parameters can be carried out by maximising the auxiliary function constructed by adding the above Q functions and the logarithm of the prior p.d.f. [24]. These equations are as follows

$$\tilde{a}_{1i} = \frac{\eta_i - 1 + \gamma_{i1}}{\sum_{j=2}^{N-1}(\eta_j - 1) + \sum_{j=2}^{N-1} \gamma_{j1}} \tag{4.26}$$

$$\tilde{a}_{iN} = \frac{\eta_i' - 1 + \gamma_{iT}}{\sum_{j=2}^{N-1}(\eta_j' - 1) + \sum_{j=2}^{N-1} \gamma_{jT}} \tag{4.27}$$

$$\tilde{a}_{ij} = \frac{\eta_{ij} - 1 + \sum_{t=1}^{T} \xi_{ijt}}{\sum_{j=2}^{N-1}(\eta_{ij} - 1) + \sum_{j=2}^{N-1} \sum_{t=1}^{T} \xi_{ijt}}. \tag{4.28}$$

$$\tilde{\omega}_{ik} = \frac{\psi_{ik} - 1 + \sum_{t=1}^{T} c_{ikt}}{\sum_{k=1}^{K} \psi_{ik} - K + \sum_{k=1}^{K} \sum_{t=1}^{T} c_{ikt}}, \tag{4.29}$$

$$\tilde{\boldsymbol{m}}_{ik} = \frac{\tau_{ik}\boldsymbol{\mu}_{ik} + \sum_{t=1}^{T} c_{ikt}\boldsymbol{o}_t}{\tau_{ik} + \sum_{t=1}^{T} c_{ikt}}, \tag{4.30}$$

and

$$\tilde{\boldsymbol{r}}_{ik}^{-1} = \frac{\boldsymbol{u}_{ik} + \sum_{t=1}^{T} c_{ikt}(\boldsymbol{o}_t - \tilde{\boldsymbol{m}}_{ik})(\boldsymbol{o}_t - \tilde{\boldsymbol{m}}_{ik})' + \tau_{ik}(\tilde{\boldsymbol{m}}_{ik} - \boldsymbol{\mu}_{ik})(\tilde{\boldsymbol{m}}_{ik} - \boldsymbol{\mu}_{ik})'}{\alpha_{ik} - V + \sum_{t=1}^{T} c_{ikt}}, \tag{4.31}$$

where $\xi_{ijt} = P(s_{t-1} = i, s_t = j | \boldsymbol{O}, \hat{\boldsymbol{\lambda}})$ and

$$c_{ikt} = \gamma_{it}\hat{\omega}_{ik}\mathcal{N}(\boldsymbol{o}_t | \hat{\boldsymbol{m}}_{ik}, \hat{\boldsymbol{r}}_{ik})\ f^{-1}(\boldsymbol{o}_t | \hat{\boldsymbol{\theta}}_i). \tag{4.32}$$

A detailed derivation of the above equations is also given in Appendix A. The above results can be seen to have great similarity with the Baum-Welch re-estimation formulae, except that here the effect of the prior parameters is also included. The formulae for the case of HMMs with diagonal covariance matrices can be derived in the same manner.

The generalisation of the MAP estimation formulae presented in this section, for the cases of multiple observations and multiple model training in an embedded model training procedure can also be carried out. However, this is not performed here so as to avoid further complication of the formulae due to the large number of subscripts and superscripts needed.

4.6 Prior Parameter Estimation

A key issue in the implementation of the MAP estimation technique is the estimation of the prior parameters since the application of these parameters in the process of estimation is the basic difference between MAP estimation and other estimation techniques such as ML. However, this has proved to be a rather difficult problem in many cases. In this section a discussion on the estimation of the prior parameters within a Bayesian estimation framework and the problems associated with it are presented. Some possible solutions to these problems are suggested.

During the previous section, the prior density $G(\boldsymbol{\lambda})$, defined by (4.21), was assumed to be a member of a preassigned family of prior distributions. If a pure Bayesian approach to parameter estimation is considered, subjective knowledge about the process is also required in order to enable one to assume the parameter vector $\boldsymbol{\phi}$ of this prior family of p.d.f.'s, $G(\cdot|\boldsymbol{\phi})$, known. However, in a real situation, it is usually difficult to have such subjective knowledge, especially in such cases where the parameters are continuous and multi-dimensional [9]. Hence the application of a complete pure Bayesian approach in this case is extremely difficult.

4.6.1 Empirical Bayes Approach to Prior Parameter Estimation

An alternative approach to a pure Bayesian one is the *Empirical Bayes* approach [71, 90], where a frequency interpretation is given to the prior distribution. In this approach, it is assumed that a current observation $\boldsymbol{o}$ is to be used for estimating $\boldsymbol{\lambda}$ and also, when the current observation is made, $\boldsymbol{o}_1, \boldsymbol{o}_2, \ldots, \boldsymbol{o}_n$ are the past observations which are obtained with independent past realisations $\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2, \ldots, \boldsymbol{\lambda}_n$ of $\boldsymbol{\Lambda}$. Hence, $(\boldsymbol{O}, \boldsymbol{\Lambda})$ denotes a sequence of independent sets of past observations and their associated unknown HMM parameters. Usually, the actual values of $\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2, \ldots, \boldsymbol{\lambda}_n$ are assumed not to be ever known, but that there exists a common prior p.d.f. $G(\cdot|\boldsymbol{\phi})$ for them all. Thus the empirical p.d.f. of $\boldsymbol{O}$, $f_n(\boldsymbol{O})$, is an estimate of the marginal p.d.f. of $\boldsymbol{O}$, $f(\boldsymbol{O}|\boldsymbol{\Lambda})$ so that as $n \to \infty$ for every $\boldsymbol{O}$, $f_n(\boldsymbol{O}) \to f(\boldsymbol{O}|\boldsymbol{\Lambda})$. Thus, it might be possible to find a p.d.f. $G(\boldsymbol{\Lambda})$ such that in

$$f(\boldsymbol{O}|\boldsymbol{\phi}) = \int f(\boldsymbol{O}|\boldsymbol{\Lambda})\hat{G}(\boldsymbol{\Lambda}|\boldsymbol{\phi})d\boldsymbol{\Lambda}, \tag{4.33}$$

$\hat{G}(\boldsymbol{\Lambda}|\boldsymbol{\phi}) \to G(\boldsymbol{\lambda}|\boldsymbol{\phi})$ as $n \to \infty$. Note that here $\boldsymbol{O} = (\boldsymbol{o}_1, \boldsymbol{o}_2, \cdots, \boldsymbol{o}_n)$, $\boldsymbol{\Lambda} = (\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2, \cdots, \boldsymbol{\lambda}_n)$, $f(\boldsymbol{O}|\boldsymbol{\Lambda}) = \prod_{i=1}^{n} f(\boldsymbol{o}_i|\boldsymbol{\lambda}_i)$ and $G(\boldsymbol{\Lambda}|\boldsymbol{\phi}) = \prod_{i=1}^{n} G(\boldsymbol{\lambda}_i|\boldsymbol{\phi})$. However, due to the difficulty of finding maximum likelihood estimates based on $f(\boldsymbol{O}|\boldsymbol{\phi})$, a simpler approach should be followed.

One alternative to the maximum likelihood approach for solving this problem is the modified likelihood approach [71]. In this approach, the pairs $(\boldsymbol{o}_i, \boldsymbol{\lambda}_i)$ of the Empirical Bayes scheme are regarded as independent realisations of $(\boldsymbol{O}_i, \boldsymbol{\Lambda}_i)$ with joint p.d.f. $f(\boldsymbol{o}_i|\boldsymbol{\lambda}_i)g(\boldsymbol{\lambda}_i|\boldsymbol{\phi})$. Then for all pairs, the joint p.d.f. would be

$$L(\boldsymbol{o}, \boldsymbol{\lambda}) = \prod_{i=1}^{n} f(\boldsymbol{o}_i|\boldsymbol{\lambda}_i)g(\boldsymbol{\lambda}_i|\boldsymbol{\phi}). \tag{4.34}$$

Also the marginal p.d.f. for the $\boldsymbol{o}_i$'s, is given by

$$h_n(\boldsymbol{o}|\boldsymbol{\phi}) = \int L(\boldsymbol{o}, \boldsymbol{\lambda})d\boldsymbol{\lambda}. \tag{4.35}$$

These two functions are also known as likelihood functions for $\boldsymbol{\lambda}$ and $\boldsymbol{\phi}$.

The modified likelihood approach then involves finding a likelihood estimate of $\boldsymbol{\lambda}$, i.e. $\hat{\boldsymbol{\lambda}}(\boldsymbol{o}, \boldsymbol{\phi})$ using $L(\boldsymbol{o}, \boldsymbol{\lambda})$. An estimate of $\boldsymbol{\phi}$, say $\hat{\boldsymbol{\phi}}$, is obtained using $h_n(\boldsymbol{o}|\boldsymbol{\phi})$, and a final estimate for $\boldsymbol{\lambda}$ is given by $\hat{\boldsymbol{\lambda}}(\boldsymbol{o}, \hat{\boldsymbol{\phi}})$.

The justification for using the above function is under question. This is due to the unobservability of the random variables $\boldsymbol{\lambda}$, which make the likelihood function $L(\boldsymbol{o}, \boldsymbol{\lambda})$ an unusual one [71]. Thus, the properties of this likelihood function are not completely known. Furthermore, obtaining the estimates of proposed parameters $\boldsymbol{\Lambda}$ and $\boldsymbol{\phi}$ by maximising the above modified likelihood function in this case does not seem to be trivial.

The method of moments is another alternative for estimating the prior parameters [9, 71]. In this approach, in order to estimate the parameters of the marginal p.d.f., the standard method of moments is used. This method consists of equating the first few sample moments with the corresponding population moments, to obtain as many equations as unknown parameters. As an example, in the current case, the observation sets $\boldsymbol{o}_1, \boldsymbol{o}_2, \ldots, \boldsymbol{o}_n$ can be used for the estimation of the corresponding parameters $\hat{\boldsymbol{\lambda}}_1, \hat{\boldsymbol{\lambda}}_2, \ldots, \hat{\boldsymbol{\lambda}}_n$, using ordinary parameter re-estimation procedures such as the Baum-Welch algorithm. These estimated parameters can then be assumed to be observations with the density $g(\boldsymbol{\lambda})$. In the case of Gaussian mixtures with diagonal covariance matrices, $g(\boldsymbol{\lambda})$ is defined according to (4.11), (4.16) and (4.21). Hence, the Dirichlet distribution properties can be used to find the population moments for the entry and exit state probabilities, i.e. [23, 48]

$$E(a_{1i}) = \frac{\eta_i}{\sum_{i=2}^{N-1} \eta_i} \tag{4.36}$$

$$\begin{aligned} \mathrm{Var}(a_{1i}) &= \frac{\eta_i(\sum_{i=2}^{N-1} \eta_i - \eta_i)}{(\sum_{i=2}^{N-1} \eta_i)^2(\sum_{i=2}^{N-1} \eta_i + 1)} \\ &= \frac{E(a_{1i})[1-E(a_{1i})]}{\sum_{i=2}^{N-1} \eta_i + 1}. \end{aligned} \tag{4.37}$$

Thus,

$$\eta_i = \left\{ \frac{E(a_{1i})[1-E(a_{1i})]}{\mathrm{Var}(a_{1i})} - 1 \right\} E(a_{1i}) \tag{4.38}$$

and similarly

$$\eta_i' = \left\{ \frac{E(a_{iN})[1-E(a_{iN})]}{\mathrm{Var}(a_{iN})} - 1 \right\} E(a_{iN}). \tag{4.39}$$

According to the Dirichlet distributions of state transition probabilities and mixture weights, the same procedure can be followed for the parameters of these distributions and would lead to

$$\eta_{ij} = \left\{ \frac{E(a_{ij})[1-E(a_{ij})]}{\mathrm{Var}(a_{ij})} - 1 \right\} E(a_{ij}) \tag{4.40}$$

$$\psi_{ik} = \left\{ \frac{E(\omega_{ik})[1-E(\omega_{ik})]}{\mathrm{Var}(\omega_{ik})} - 1 \right\} E(\omega_{ik}). \tag{4.41}$$

For the remaining prior parameters, i.e. α_{ikv}, β_{ikv}, μ_{ikv} and τ_{ikv}, the properties of the normal-gamma distribution may be used to find the population moments, i.e. [23]

$$E(r_{ikv}) = \frac{\alpha_{ikv}}{\beta_{ikv}} \tag{4.42}$$

$$\mathrm{Var}(r_{ikv}) = \frac{\alpha_{ikv}}{\beta_{ikv}^2} \tag{4.43}$$

$$E(m_{ikv}) = \mu_{ikv} \tag{4.44}$$

$$\begin{aligned} \mathrm{Var}(m_{ikv}) &= \frac{1}{\tau_{ikv} E(r_{ikv})} \\ &= \frac{\beta_{ikv}}{\tau_{ikv}\alpha_{ikv}}. \end{aligned} \tag{4.45}$$

Thus,

$$\alpha_{ikv} = \frac{[E(r_{ikv})]^2}{\mathrm{Var}(r_{ikv})} \tag{4.46}$$

$$\beta_{ikv} = \frac{E(r_{ikv})}{\mathrm{Var}(r_{ikv})} \tag{4.47}$$

$$\mu_{ikv} = E(m_{ikv}) \tag{4.48}$$

$$\tau_{ikv} = \frac{\beta_{ikv}}{\mathrm{Var}(m_{ikv})\alpha_{ikv}}. \tag{4.49}$$

Note that the above equations are written for every element of vectors individually. In order to find the estimates of the above parameters, one can replace $E(r_{ikv})$, $\mathrm{Var}(r_{ikv})$, $E(m_{ikv})$ and $\mathrm{Var}(m_{ikv})$ with their corresponding sample moments.

For the full covariance matrix case, the same procedure can once again be followed. In this case, the prior densities for the mixture weights, entry and exit states and state transition probabilities are the same and hence the same equations also apply. However, for the output mixture distributions, the normal-Wishart prior density properties should be used. According to DeGroot [23], these can be written as

$$E(\boldsymbol{m}_{ik}) = \boldsymbol{\mu}_{ik} \tag{4.50}$$

$$\mathrm{Cov}(\boldsymbol{m}_{ik}) = [\tau_{ik} E(\boldsymbol{r}_{ik})]^{-1} \tag{4.51}$$

$$E(\boldsymbol{r}_{ik}) = \alpha_{ik} \boldsymbol{u}_{ik}^{-1} \tag{4.52}$$

$$\mathrm{Prec}(\boldsymbol{r}_{ik}) = \boldsymbol{u}_{ik}, \tag{4.53}$$

where the function Prec implies the precision matrix of the Wishart distribution. However, in this case, it is more difficult to derive an adequate number of equations for the parameters of the distribution to be estimated. Hence, in most cases, some constraints are applied to make the parameter estimation feasible [34, 36, 48]. As an example, a more restrictive family of distributions has been introduced by Gauvain and Lee [36] by setting the following slightly simplifying constraints

$$\tau_{ik} = \psi_{ik} - 1 \tag{4.54}$$

$$\alpha_{ik} = \tau_{ik} + V. \tag{4.55}$$

Then, appropriate equations for the rest of parameters can be found by using the above population moment equations, i.e.

$$\boldsymbol{\mu}_{ik} = E(\boldsymbol{m}_{ik}) \tag{4.56}$$

$$\boldsymbol{u}_{ik}^{-1} = \frac{1}{\alpha} E(\boldsymbol{r}_{ik}). \tag{4.57}$$

These equations can be used to obtain moment estimates of the prior parameters by replacing $E(\boldsymbol{m}_{ik})$ and $E(\boldsymbol{r}_{ik})$ by their sample moment estimates.

As stated before, the use of a classical model training procedure such as the Baum-Welch algorithm can lead to an initial estimate of the parameters $\hat{\boldsymbol{\lambda}}_i$ from the observation sets $\boldsymbol{o}_1, \boldsymbol{o}_2, \ldots, \boldsymbol{o}_n$, which can be used in the method of moments to provide estimates of the prior parameters for the MAP estimation process. A further improved MAP estimation result can be obtained by using an iterative procedure as follows: the above method of moments can be used to obtain an initial estimate of the hyperparameter $\boldsymbol{\phi}$. Using this initial estimate and the observation sets $\boldsymbol{o}_1, \boldsymbol{o}_2, \ldots, \boldsymbol{o}_n$, MAP estimations of the HMM parameter sets $\hat{\boldsymbol{\lambda}}_1, \hat{\boldsymbol{\lambda}}_2, \ldots, \hat{\boldsymbol{\lambda}}_n$ can be obtained, which can in turn lead to an improved estimate of the hyperparameter ϕ. This process can be iterated by again applying the above method of moments to the results of last iteration to obtain a new estimate of the hyperparameter and using that result in the MAP estimation process.

In the case of speaker adaptation, the initial set of model parameters for prior parameter estimation are derived from either the parameters of a currently available speaker independent system, or by polling from several speaker dependent trained systems. In either case the above method of moments is applicable.

4.7 Speaker Clustering for Prior Parameter Improvement

The prior parameter estimation techniques discussed above are able to provide reasonably acceptable estimates of prior parameters for MAP estimation of HMMs. However, under certain conditions, some better estimated priors can lead to better model parameter estimations. In this section, a technique for improving the MAP estimation results by improving the initial model parameters used for prior parameter estimation will be discussed.

Usually, speaker independent parameters act as $\hat{\boldsymbol{\lambda}}_i$, viewed as prior parameters with the prior density $G(\boldsymbol{\lambda})$ and used for the derivation of prior estimates. The SI models are trained on a large number of different speakers and are the most suitable models available for any new speaker. However, the use of only one set of model parameters could lead to some limitations which necessitate the application of further artificial constraints to make prior parameter estimation feasible.

To obtain a better set of models for any speaker, the concept of *Speaker Clustering* (SC) has been investigated. This concept, as discussed in Chapter 3, has already been used in different approaches to speech recognition, and especially speaker adaptation problems [56, 72, 37]. In the current approach, however, speaker clustering has essentially been used for prior parameter estimation improvements. Therefore, the usual SI system can be replaced by several sets of models, each one specially trained for a certain cluster.

A basic block diagram of the proposed technique is displayed in Figure 4.1. As is shown in this block diagram, the whole database of speaker independent speech is divided into separate databases for each of the K individual speakers and used for training K different speaker-specific HMM sets. The speakers of the SI speech database are then divided into a few clusters, say L, according to the similarity of their speaker-specific model parameters. For each of these speaker clusters, an intra-cluster HMM system (or ICM for *Intra-Cluster Model set*) is trained using all the speech data available from the speakers of that cluster,

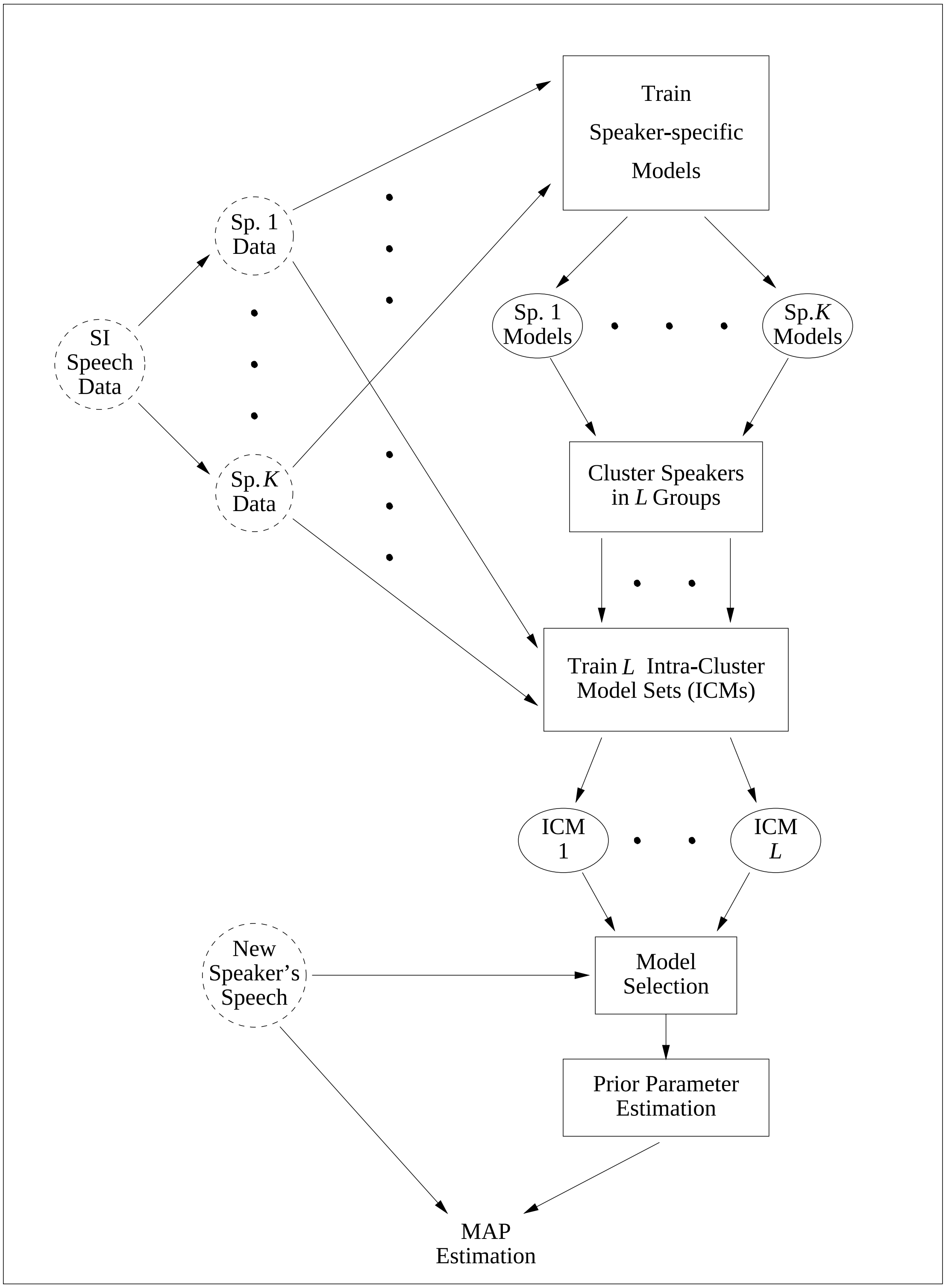


Figure 4.1: Block diagram of the speaker clustering scheme for improved prior parameter estimation.

leading to L such ICMs. Once the speech data from a new speaker to whom HMMs are to be adapted is available, these utterances (adaptation data) are used to select the closest ICM to the new speaker. Therefore, a system different from the usual SI system, which is believed to be closer to the new speaker's actual acoustic models, is selected and used in the prior parameter estimation process already discussed.

Different stages of the above approach and problems associated with each of them will be discussed during the next few sections.

4.7.1 Speaker-specific Model training

In order to provide a few speaker groups to be used for the speaker clustering approach discussed above, a number of speaker-specific HMM systems should be initially available. The speakers available for this purpose are the speakers usually used for training purposes in a speaker independent system. However, according to the needs and specifications of an SI system, training is usually based on a larger number of speakers with limited training data from each. A large number of speakers can be beneficial to our approach, since the larger the number, the better are they distributed according to their acoustic features and hence better clustering might be achieved. Meanwhile, a problem arises from the limited amount of training data available from each speaker, which makes the training of such speaker-specific models unsatisfactory.

The problem of sparseness of the training data in this case can cause major subsequent problems by reducing the robustness and reliability of the clustering process. This problem can damage the whole process if the amount of training data for each speaker is less than a certain amount. A possible way to reduce the effect of the sparseness of data on the process of speaker-specific model training, as discussed earlier in this chapter, is the use of MAP estimation.

Although the whole process is supposed to provide better prior parameter estimations for a certain task, this does not necessarily prevent us from using MAP estimation during this process. However, the usual prior parameter estimation methods can be used in this case for MAP estimation of speaker-specific models.

It should also be noted that the use of MAP estimation for the above purpose does not necessarily guarantee improved models for the proposed speaker compared to the speaker independent models, especially when the amount of training data is too small. In this case, the best model set to be used for deriving the prior parameters is the same SI HMM system trained using all of the above mentioned available speakers. This will provide us with an initial system which tends to improve using the data received from the proposed speaker. However, in order to have better results in this case, it is necessary to use as much training data as possible from every speaker.

4.7.2 Clustering Speakers

Several clustering techniques are introduced in the literature for different clustering problems [29, 63, 111]. Hierarchical clustering in which the data is partitioned into a number of clusters gradually, usually in several steps, is the basis for many clustering and classification approaches. Hierarchical techniques can be subdivided into *agglomerative* and *divisive* clustering methods. Both these methods have widely been used in classification problems and implemented in different ways. The issue of choosing the proper clustering technique for any problem can be viewed as an open one since it depends very much on the nature of problem, the type of data faced and the different parameters concerned [29].

In this section the problem of clustering, i.e. dividing the available speakers into groups according to their similarities, is discussed and the approach adopted here to clustering explained. This approach is based on an agglomerative clustering method.

4.7.2.1 Finding Speaker Similarities

A number of different measures of similarity could be used to group the speakers [29]. Two metrics often used to find the distance between Gaussians are divergence and the Bhattacharyya distance [97, 31]. The divergence between two Gaussians is found by

$$d = \frac{1}{2}tr(\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\Sigma}_2 + \boldsymbol{\Sigma}_2^{-1}\boldsymbol{\Sigma}_1 - 2\boldsymbol{I}) + \frac{1}{2}(\boldsymbol{m}_1 - \boldsymbol{m}_2)'(\boldsymbol{\Sigma}_1^{-1} + \boldsymbol{\Sigma}_2^{-1})(\boldsymbol{m}_1 - \boldsymbol{m}_2) \tag{4.58}$$

where $\boldsymbol{m}_1$ and $\boldsymbol{m}_2$ are the mean vectors, and $\boldsymbol{\Sigma}_1$ and $\boldsymbol{\Sigma}_2$ are the covariance matrices of the two Gaussians and $\boldsymbol{I}$ is the identity matrix.

The distance between the HMM states can be found using the square root of the divergence between Gaussians. Hence, assuming the covariance matrices to be diagonal, this distance, between any two HMM states, is given by

$$D_{12} = \left\{ \frac{1}{V} \sum_{v=1}^{V} \left[\frac{\sigma_{2v}^2}{\sigma_{1v}^2} + \frac{\sigma_{1v}^2}{\sigma_{2v}^2} - 2 + (\frac{1}{\sigma_{1v}^2} + \frac{1}{\sigma_{2v}^2})(m_{1v} - m_{2v})^2 \right] \right\}^{\frac{1}{2}}. \tag{4.59}$$

Here σ_{1v}^2 and σ_{2v}^2 are the diagonal elements of the covariance matrices of the first and second proposed states respectively, while m_{1v} and m_{2v} are the corresponding mean vector elements. The following simpler distance metric has been found to be less sensitive to undertrained variances while performing as well as the previous one [111]

$$D_{12} = \left[\frac{1}{V} \sum_{v=1}^{V} \frac{(m_{1v} - m_{2v})^2}{\sqrt{\sigma_{1v}^2 \sigma_{2v}^2}} \right]^{\frac{1}{2}}. \tag{4.60}$$

Where the class covariance matrices are the same for two Gaussians the divergence measure becomes the Mahalanobis distance [97]

$$d = (\boldsymbol{m}_1 - \boldsymbol{m}_2)'\boldsymbol{\Sigma}^{-1}(\boldsymbol{m}_1 - \boldsymbol{m}_2). \tag{4.61}$$

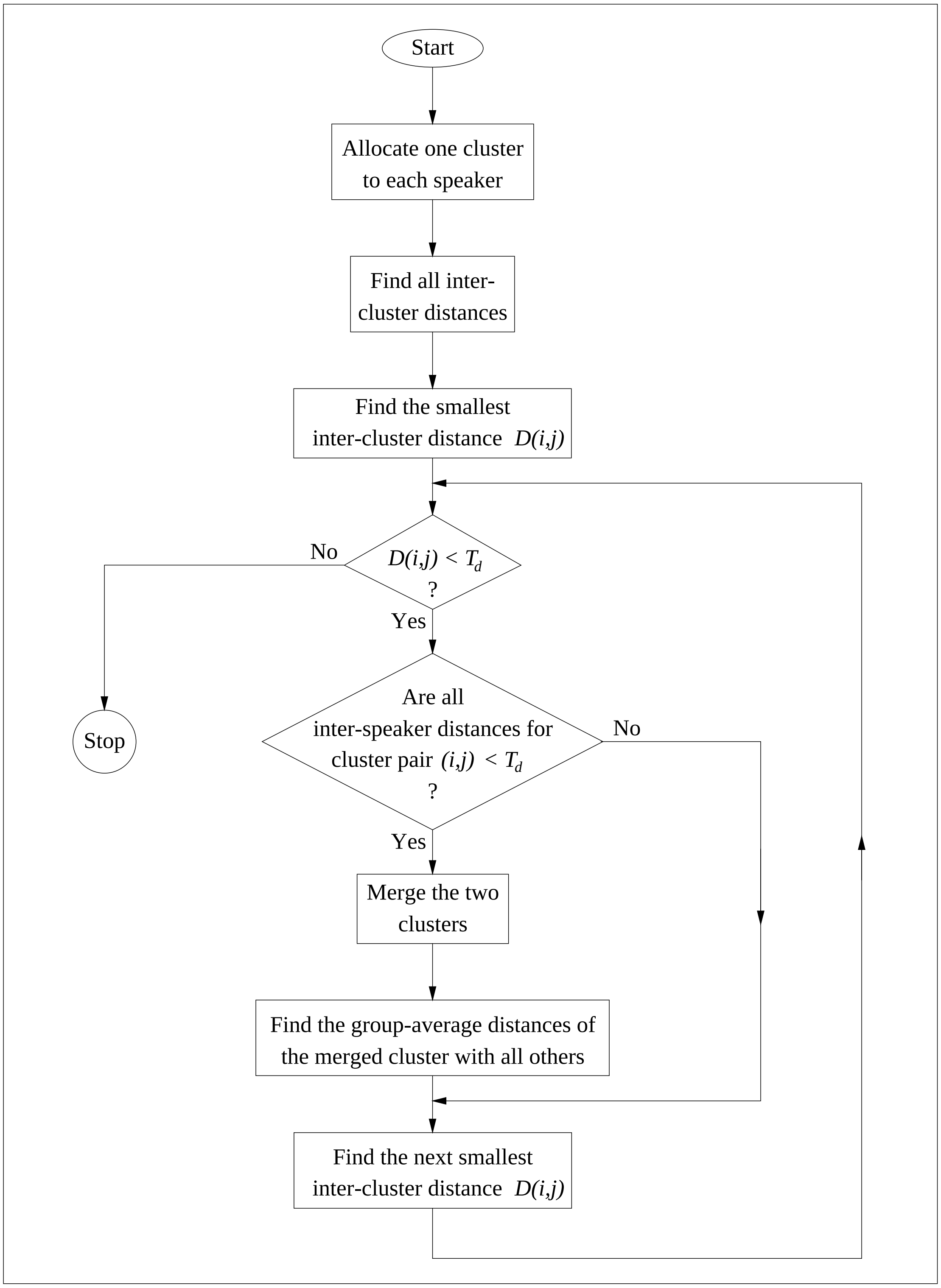


Figure 4.2: Flow diagram the clustering algorithm.

4.7.2.2 The Clustering Algorithm

The algorithm for clustering the speakers is displayed in the flow diagram of Figure 4.2 and can be summarised as follows:

1. One training speaker is allocated to every cluster initially.
2. The average inter-cluster distances are found by using (4.60) for all similar states and averaging over all the HMM states.
3. The smallest inter-cluster distance $D(i, j)$ is found.
4. If the smallest inter-cluster distance found, $D(i, j)$, is not smaller than a predefined inter-cluster distance threshold T_d, the clustering process is terminated, otherwise continue. If the distance between any two speakers in the two closest clusters (when clusters have more than one speaker) is not less than T_d the next two steps are skipped, i.e. the two clusters are not merged.
5. These two closest clusters are merged.
6. All average inter-cluster distances are found for the newly formed cluster with all the rest, based on the group-average distance principle. This means finding and averaging the distances between all speakers of the new cluster with all speakers of any other cluster. The inter-speaker distances are found as stated in step 2.
7. The next smallest inter-cluster distance $D(i, j)$ is found.
8. The algorithm is continued from step 4.

The purpose of using threshold T_d in the above algorithm is twofold. First it acts as an intra-cluster distance threshold. Thus the speakers within two clusters to be merged are not allowed to be further apart than T_d. It is also used to terminate the clustering process when the two merging candidates are not close enough. Although another threshold may be more appropriate for this purpose, setting two thresholds at the same time could be very difficult.

4.7.3 Training Intra-Cluster Models

After the speakers have been allocated to clusters, L Intra-Cluster Model sets (ICMs) are trained. These model sets will be used later, together with the speech of the new speaker to specify the most suitable cluster for that speaker.

In this stage, as long as the number of speakers allocated to any cluster is high enough, standard training techniques can be utilised for the purpose of model training. This means that the amount of training data available for the training of every cluster's models be such that the models receive adequate training. In other words, the problem of sparse training data, as discussed before, is not faced. However, in most clustering techniques this cannot be guaranteed.

Thus, once again the problem of sparse training data may be encountered here which can lead to poor training of the intra-cluster model sets. Accordingly it is still preferable to use MAP estimation for the purpose of training at this stage, whether the training data could be considered as sparse or not. As in the discussion on the use of MAP estimation for training speaker-specific models, the use of prior parameters extracted from SI models can also be considered to be suitable for this application.

4.7.4 Model Selection for the New Speaker

Once L ICMs are trained and available, the most appropriate of them can be used as the source for calculating the prior parameters for adaptation of HMMs to a new speaker.

The main question in this case would be "Which ICM is the most appropriate one for the new speaker?" The answer should be found using the adaptation data from the new speaker and the available ICMs and try to find the best match.

A supervised adaptation procedure is being followed in this case, hence a Viterbi alignment procedure has been used to produce a forced alignment of the new speaker's utterances with the HMMs. The algorithm for selecting the best ICM for the new speaker is as follows:

1. Given any ICM i (starting from $i = 1$), a forced alignment of the new speaker's adaptation utterances with the existing models is performed. This gives a log likelihood per frame for all the utterances.
2. The log likelihoods obtained in step 1 are averaged over all the sentences.
3. If $i < L$ (L is the total number of clusters), i is incremented and the algorithm will continue from step 1, otherwise it will continue.
4. The ICM with highest average log likelihood per frame for the utterances of this speaker is found.

An alternative to the above Viterbi alignment, in a supervised adaptation framework, is to perform full recognition which is also possible at this stage and either the average log likelihood per frame or the recognition result such as recognition word accuracy can be used for selecting the best cluster. However, this may involve the application of a language model which will generally affect the log likelihoods and seems unlikely to perform better.

Once again in this approach, the role of the adaptation data is very important and naturally more adaptation data can lead to a more reliable choice.

As described in Section 4.6, the resultant ICM can be used to find the prior parameters needed for the MAP estimation of the current speaker. An iterative procedure as described in that section might even lead to a better prior set for MAP estimation.

4.7.5 Speaker Phone Clustering

The above clustering algorithm may be improved by use of speaker phone clustering. Here, broad phonetic groups have been used during the clustering process in order to improve the clustering performance.

In this case, the same concept of clustering is used in a somewhat wider sense. In speaker clustering the assumption is that certain speakers may share some features among each other. Hence, it might be useful to divide them into certain groups according to their similarities. Here, the idea is that among different speakers there may be only some common acoustic features while their other features may not even be close to each other. Hence it might be useful to group speakers according to their phonetic similarities. This can be carried out by dividing the phones into a number of broad phonetic groups such as vowels, stop consonants, fricatives, etc., and cluster speakers in each of these phonetic groups, hoping that better clusters of speakers with acoustic similarities could be found within these phonetic groups. The resulting phonetic group clusters can then be used for the purpose of speaker clustering.

In order to implement this algorithm, some changes have to be made to the previous one. The block diagram for this approach has been shown in Figure 4.3. The new algorithm works as follows:

1. The speaker-specific model training is carried out exactly as described in Section 4.7.1.

2. Using the model sets generated in the last step, and dividing the phone models into a number of groups, the speakers are clustered for each group individually according to the similarity of their models in that group, using the same algorithm described in Section 4.7.2. This will lead to L_i clusters for any phonetic group i. Note that the number of clusters L_i and also the number of speakers in any cluster might be different to those of other phonetic groups. Moreover, different clustering thresholds, T_d's, might be necessary for different phonetic groups.

3. The appropriate Intra-Cluster Phone Model sets (ICPMs) are trained for all the clusters. Thus, L_i ICPMs are trained for any broad phonetic group i. This step, for any cluster l_i in the broad phonetic group i is carried out by usual MAP estimation of the whole set of acoustic models using all the data available from the speakers in that cluster. Then the resultant models for the same phonetic group are extracted from the set of trained models and are added to the rest of the models from the basic SI system to form the ICPM.

4. For any phonetic group i, the ICPM with the highest average log likelihood per frame for the utterances of the new speaker is found. This specifies the most suitable phonetic group cluster for this speaker. Thus P such clusters are specified for a new speaker by selecting P ICPMs.

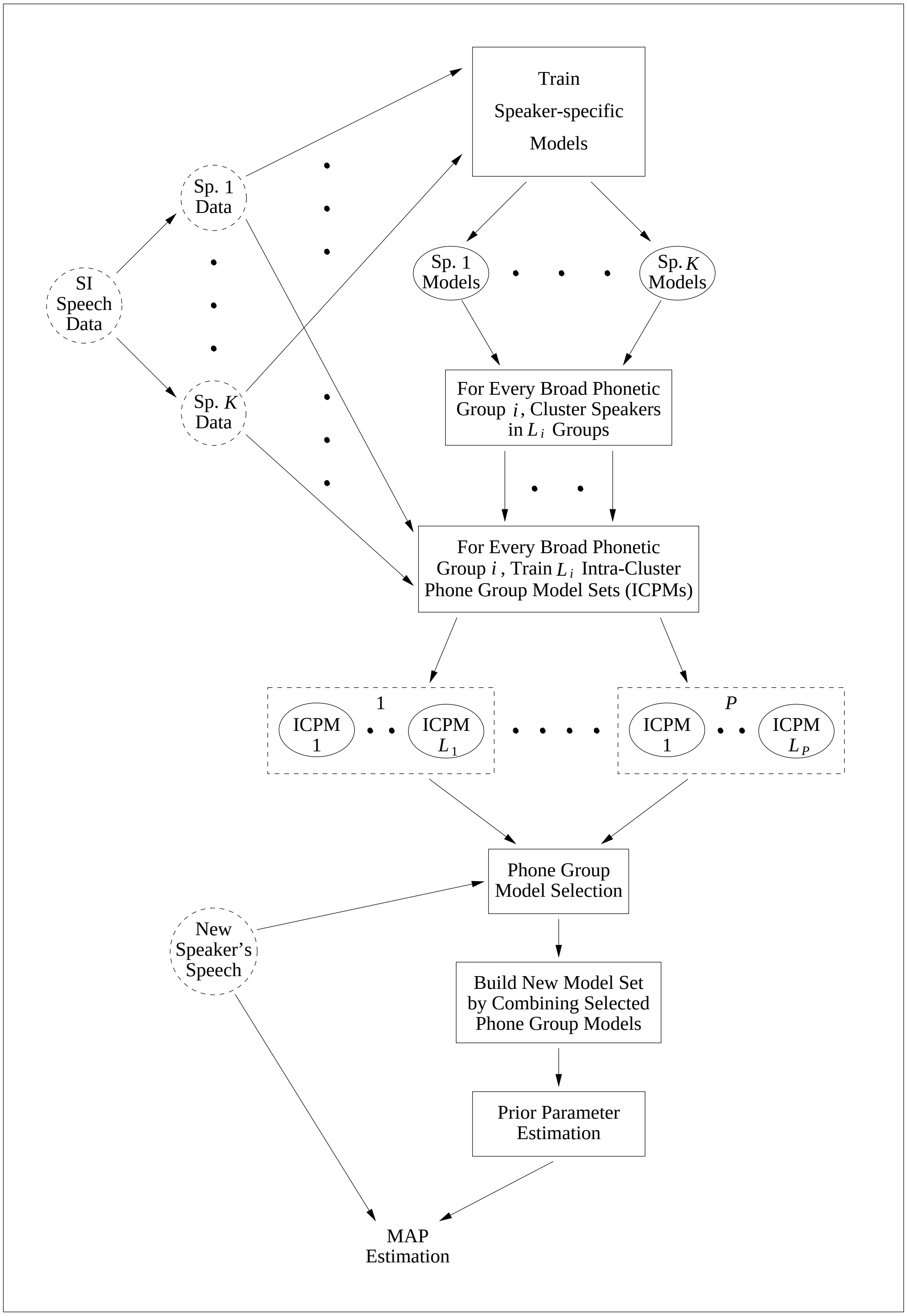


Figure 4.3: Block diagram for the speaker phone clustering approach.

5. The final model set for prior estimation of the new speaker is built by combining the set of acoustic models of any broad phonetic group i from the appropriate ICPM chosen in the last step.

The above algorithm, due to the use of broad phonetic groups, looks much more complicated, when compared to the previous clustering algorithm. However, since the clustering is focused on more detailed specifications of speakers, a better result could be expected. A further improvement in this algorithm may be achieved by using a larger number of less broad phonetic groups. However, this may result in even more complexity in implementation.

4.8 Summary

Throughout this chapter, the concept of MAP estimation which is used for the Bayesian adaptation of CDHMMs was introduced. This approach can be counted as an important systematic approach to speaker adaptation, since it can be implemented using the Forward-Backward algorithm which is the well-known training algorithm for HMM parameters. The application of this technique to HMMs with non-emitting states was discussed and the important issue of prior parameter estimation in this technique was focused on.

Since prior parameters play the main role in the MAP estimation process, a speaker clustering approach to improve the prior parameters of a CDHMM has also been introduced in this chapter. MAP estimation itself was heavily used in this approach to improve the results in the case of limited available training data for each speaker.

In order to reduce the amount of computation required for this approach, simpler models can be used to find the clusters and allocate each speaker to his own appropriate cluster, while for better recognition results, the same clustering results obtained by simpler models can be applied to more detailed acoustic modelling. This greatly reduces the amount of computation, memory and disk space needed for clustering procedures and also, due to the lower number of parameters, better-trained simpler models can be obtained which could also help in clustering. However, whilst it is true that more detailed models can better represent a speaker's acoustic characteristics, whether they can perform properly under the sparse training data conditions and lead to proper speaker separation is doubtful.

The evaluation of the MAP estimation of CDHMM parameters for speaker adaptation, using the proposed HMM structure and priors estimated from a speaker independent system, and also in the case of prior parameter estimation using the speaker clustering approach, will be given in Chapter 6.

Chapter 5

Regression-Based Model Prediction

In this chapter, a very fast speaker adaptation technique called *Regression-based Model Prediction* (RMP) is introduced. This technique uses MAP estimation as a first stage of adaptation, but unlike MAP, features a very fast improvement in the recognition performance, while for large amounts of data, retains the good asymptotic performance of MAP estimation. Altogether, as will be shown in Chapter 6, this system can outperform MAP estimation for any amount of adaptation data before reaching the speaker dependent saturation point.

5.1 The Problem of Undertrained Parameters

A major problem in HMM-based ASR system adaptation is encountered when only small amounts of training data is available. This is particularly a problem for systems with large numbers of parameters and can even lead to system performance deterioration. In such cases, the number of the system parameters which have received enough training data in order to be reliably estimated is very low and the rest of system parameters are either poorly trained or have received no training data at all. This problem often arises in speaker adaptation since the amount of training data from the new speaker can be very low. MAP estimation as introduced in Chapter 4, is a possible solution to this problem and has been used as a potential adaptation algorithm and shown improvement for this case of sparse training data.

However, when the quantity of adaptation data is lower than a certain amount, even MAP estimation cannot give reliable estimations of the system parameters. In these cases, although a small amount of adaptation data is available from a new speaker, the system performance is less likely to improve. This emphasises the problem of undertrained parameters, which in the case of MAP estimation can still be encountered. In such a case, it is evident that some acoustic information is available from the new speaker within the adaptation utterances which MAP estimation has not been able to exploit. With the above belief in mind, in the next few sections a new technique is presented that tries to make better use of the adaptation information available.

5.2 Model Parameter Relationships

The differences between speakers, as discussed in Section 3.1, can be the result of many phonetic or acoustic variations. However, similar utterances from different speakers are recognised as being the same by a human listener. Also, there are similarities among different utterances from a certain speaker which enables a human listener to distinguish between different speakers. This implies that for every listener some similarities exist among all his utterances. As a simple example, if a human listener hears a few words from unknown speakers, he/she, in most of the cases, due to these acoustic/phonetic similarities, will be able to distinguish any of those speakers, even if that speaker utters different words. Hence, the belief is that the acoustic level model parameters of a speaker will be affected by these similarities. In this case, the similarities could be extended to the important parameters of his/her phone models, such as output distribution mean vectors. Such relationships for the mean parameters of some HMMs have been used by Cox [16, 17] in an isolated-word HMM-based speech recognition system. Stern and Lasry [96] have also made use of such relationships in a MAP estimation approach for speaker adaptation called Extended MAP (EMAP).

Linear regression can be used to help find these relationships between the acoustic models of speakers. If a 2 dimensional space is created with each dimension representing a model parameter (e.g. a state output Gaussian distribution mean vector element), then a point would represent a certain speaker in this space. If K speakers are available, then there are K points in this 2 dimensional space. Assuming that a simple linear relationship exists between these 2 parameters, a line can be fitted to these set of points which can approximate the relationship of these parameters among all the speakers. For this purpose, the straightforward method is that of least squares. Since in practice, not all the points will lie on this line, the actual relationship, written over only a single element of the vector of mean parameters is of the type

$$y = b_1 x + b_0 + \epsilon \tag{5.1}$$

where y and x represent the two parameters whose relationship is to be found, ϵ is the error associated with this approximation and b_1 and b_0 are the regression parameters relating these two parameters and found by minimising

$$\sum_{k=1}^{K} \epsilon_k^2 = \sum_{k=1}^{K} (y_k - b_1 x_k - b_0)^2, \tag{5.2}$$

where K is the total number of regression points (speakers). One of the parameters (here y) will be called the *target* parameter, or an element of the parameter vector which is to be predicted later and the other (x) will be called *source* parameter. Note that the above formula can also be written for parameter vectors, but this requires that b_1, in the case of simple linear regression, be replaced by a diagonal matrix whose diagonal elements should be individually calculated based upon regression of the corresponding source and target vector elements. This type of representation implies that a transformation approach on the whole vector is used, while in fact, regression is applied individually to all vector elements using least squares method and hence, scalar notation is more appropriate. The application of a multiple linear regression will be discussed later in this chapter.

For simple linear regression, a correlation coefficient can be calculated between the target and source parameters, which could be interpreted as an index for the linearity of this relationship. The square of this parameter is given by [12]

$$\rho_v^2 = \frac{\left[\sum_{k=1}^{K}(x_{kv}-\bar{x}_v)(y_{kv}-\bar{y}_v)\right]^2}{\sum_{k=1}^{K}(x_{kv}-\bar{x}_v)^2\sum_{k=1}^{K}(y_{kv}-\bar{y}_v)^2} \tag{5.3}$$

where ρ_v is the sample correlation coefficient for the vth element of the $\mathbf{y}$ vector and the vth element of the $\mathbf{x}$ vector and $\bar{x}_v$ and $\bar{y}_v$ represent the average of x's and y's over all the regression points.

As an example of the application of linear regression, Figures 5.1(a) and 5.1(b) display the realisations of mean vector element relationships between two potential source and target model components in a monophone single Gaussian HMM system, with 5 states per model (including entry and exit states), trained on the ARPA RM1 continuous speech corpus. Twelve speaker model sets are used to construct the scattergrams displayed in these figures. In Figure 5.1(a), the relationships are shown for the first mean vector elements of the third states of phone models of /th/ and /f/, and the relationships in Figure 5.1(b) are for the first mean vector elements of the second states of phone models of /ao/ and /aa/. As can be seen, in Figure 5.1(a), the absolute value of the correlation coefficient among these parameters is small, i.e. the parameters are far from forming an exact linear relationship, while in Figure 5.1(b), the absolute correlation coefficient is much higher and the points are closer to the fitted line. This example shows that whenever high correlations are available among source and target parameters, a line is a good approximation of the relationship of these parameters. Obviously, the number of regression points (speakers) is very important for constructing these relationships, since a larger number leads to a better regression.

A more complete survey on the parameters of the same monophone HMM systems of 12 speakers reveals attractive facts related to correlation coefficients between the mean parameters of different models. Table 5.1 displays the best correlated states of these systems together with their correlation coefficients (ρ). In order to find these correlations, the best correlated HMM state distribution for every state distribution of every HMM in the system is found, i.e., for any state of a model, s_j, a state s_i among the other HMMs is found such that

$$s_i = \arg\max_i \frac{\sum_{v=1}^{V}|\rho_{vij}|}{V} \tag{5.4}$$

where ρ_{vij} is the correlation coefficient between element v of these two states. Note that

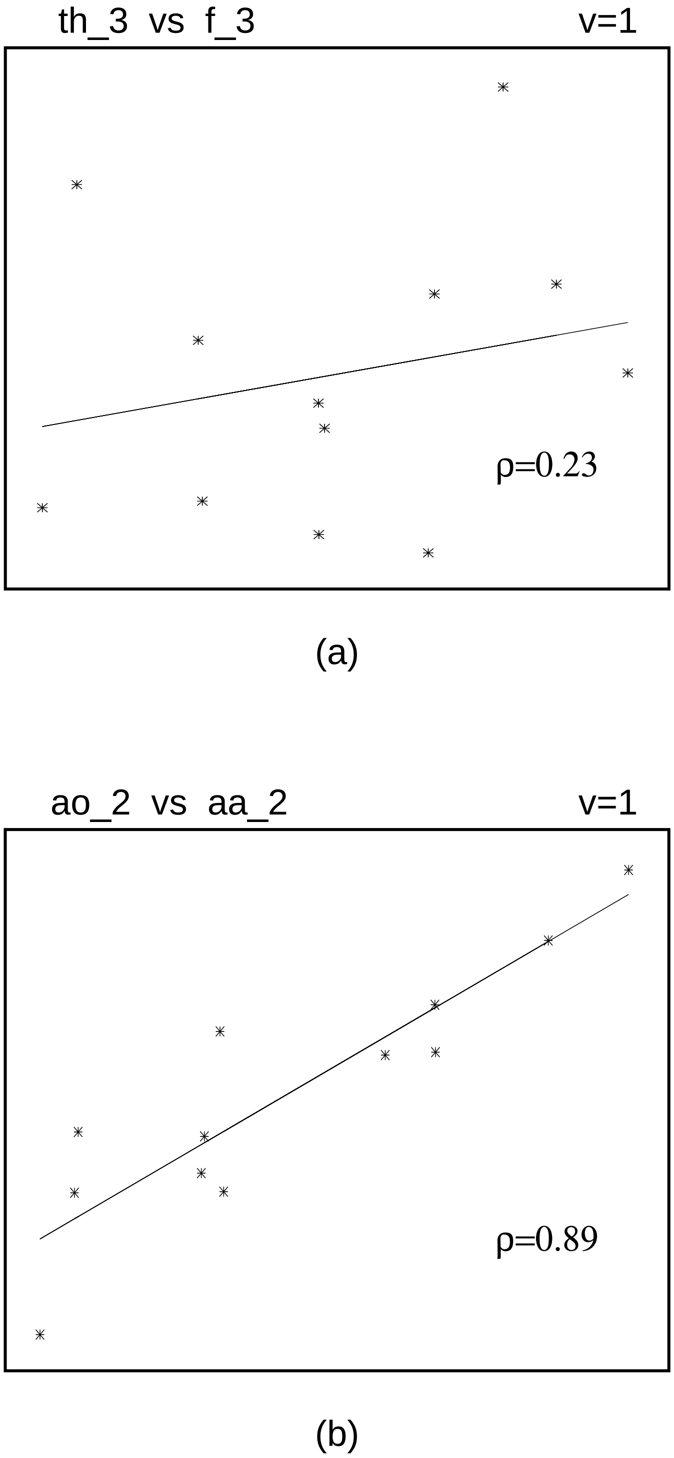


Figure 5.1: Scattergrams of mean vector element relationships between the first vector elements of (a) the third states of monophone models for /th/ and /f/ with a correlation coefficient of 0.23 and (b) the second states of monophone models for /ao/ and /aa/ with a correlation coefficient of 0.89.

state	2			3			4		
	Model	St.	$\bar{\rho}$	Model	St.	$\bar{\rho}$	Model	St.	$\bar{\rho}$
aa	ay	2	0.72	ay	3	0.56	ah	3	0.56
ae	eh	2	0.80	eh	3	0.72	eh	4	0.83
ah	dh	4	0.71	eh	3	0.57	ax	3	0.77
ao	aa	2	0.67	oy	2	0.48	oy	3	0.52
aw	ae	2	0.76	aa	3	0.43	aa	4	0.55
ax	dh	4	0.78	eh	4	0.77	ix	4	0.73
ay	aa	2	0.72	aa	3	0.56	oy	4	0.66
b	v	3	0.63	d	3	0.45	d	4	0.58
ch	jh	2	0.72	jh	3	0.75	sh	4	0.71
d	dd	2	0.69	t	3	0.52	dh	3	0.59
dd	td	2	0.73	d	2	0.57	td	4	0.42
dh	v	3	0.54	d	4	0.59	ax	2	0.78
dx	td	2	0.72	td	3	0.56	d	4	0.54
eh	ae	2	0.80	ae	3	0.72	ae	4	0.83
en	ax	2	0.55	n	2	0.67	n	4	0.81
er	r	2	0.69	r	3	0.61	r	4	0.80
ey	eh	2	0.68	iy	2	0.62	iy	4	0.80
f	th	2	0.60	p	3	0.47	p	4	0.67
g	d	2	0.64	k	3	0.58	y	2	0.60
hh	p	4	0.42	p	4	0.61	ae	2	0.51
ih	eh	2	0.78	eh	3	0.71	eh	4	0.80
ix	ax	2	0.78	ax	3	0.75	ih	4	0.75
iy	ey	2	0.68	ey	3	0.62	ey	4	0.80
jh	ch	2	0.72	ch	3	0.75	sh	4	0.68
k	kd	2	0.78	kd	3	0.67	t	4	0.59
kd	k	2	0.78	k	3	0.67	td	4	0.32
l	uh	2	0.47	w	4	0.43	ow	4	0.45
m	n	2	0.65	en	4	0.57	n	4	0.72
n	en	3	0.67	en	3	0.61	en	4	0.81
ng	iy	3	0.56	n	2	0.67	n	4	0.77
ow	aa	2	0.70	ey	3	0.46	ax	3	0.66
oy	ao	3	0.48	ao	4	0.52	ay	4	0.66
p	t	2	0.64	f	3	0.47	f	4	0.67
pd	b	2	0.59	d	3	0.29	b	4	0.34
r	er	2	0.69	er	3	0.61	er	4	0.80
s	z	2	0.78	z	3	0.71	z	4	0.79
sh	z	2	0.51	ch	3	0.67	ch	4	0.71
t	p	2	0.64	d	3	0.52	ch	4	0.65
td	dd	2	0.73	dx	3	0.56	dd	4	0.42
th	f	2	0.60	f	3	0.38	f	4	0.59
ts	td	2	0.64	s	2	0.48	s	4	0.67
uh	w	4	0.68	ih	3	0.61	ih	4	0.69
uw	ih	2	0.68	iy	3	0.51	ow	4	0.58
v	b	2	0.59	b	2	0.63	b	4	0.56
w	p	4	0.49	ao	2	0.58	uh	2	0.68
y	g	4	0.60	ih	2	0.46	eh	3	0.56
z	s	2	0.78	s	3	0.71	s	4	0.79

Table 5.1: Best matching states and corresponding average correlation coefficients in a single Gaussian monophone system.

the absolute correlation coefficient, $|\rho_{vij}|$, is averaged over all the mean vector elements to give an averaged correlation coefficient between the proposed state and any state of other HMMs in the system, and V is the dimension of the state mean vectors.

By inspecting Table 5.1, correlations as high as 0.8 can be seen. It should be noted that these correlations are averaged over all the vector elements, which in our experiments, as will be pointed out in Chapter 6, consist of cepstral and log-energy coefficients and their differential and second differential coefficients. As an example, the fourth state (note that states are numbered 2 to 4, since in our HMM structure states 1 and 5 are non-emitting states) of the phone /ae/ (like the vowel pronunciation in the word *cat*) has a correlation of 0.83 with the fourth state of the phone /eh/ (like the vowel pronunciation in the word *west*). These two phones also have high correlations of 0.80 and 0.72 on their second and third state distribution means. Other well-correlated phones presented in this table are /en/ (like the sound after the sound /z/ in *isn't*) and /n/ which have very high correlations between their third and fourth states, but while the second state of /n/ is well correlated to the third state of /en/, the highest correlated state to the second state of /en/ is the second state of /ax/. This might have been predictable since the first part of the phone /en/ has specifications close to a vowel, while the second part (starting from state 3) has similar specifications to phone /n/. Similar relationships exist among state parameters of other phones, some more examples of which are given in Table 5.1. It has been noted that the correlations among the differential and second differential elements in the above model distributions are usually less than those for instantaneous coefficients. However, the transitional coefficients have a less significant effect on the performance of a recogniser, since they are usually used to better represent the details of a speech signal. Hence, their exact prediction is believed to be of less importance in such an approach.

The basic idea behind the RMP technique is as follows. If enough regression data points are available, a linear approximation can be found for the relationships between different parameters of a model set. Once these relationships are found, if only some of the parameters can be estimated directly from the data, the remaining parameters may be estimated using these relationships. The ultimate goal is to adapt a real large vocabulary speech recognition system to a new speaker, using a very limited amount of adaptation data from that speaker, by means of predicting untrained or undertrained model parameters, using better trained ones. During the next sections, this technique and its implementation will be discussed in detail.

5.3 Use of Parameter Relationships in Continuous Speech Recognition

While the above discussion unveils a set of important relationships between model parameters, the remaining problem is how these relationships can be used in a real continuous speech recogniser. One approach followed by Cox [16, 17, 19] was to apply this technique to a limited vocabulary of isolated words (English alphabet) by allocating one model with

10 states to each word in the vocabulary and dividing the whole set of vocabulary (models) into two phonetically balanced sets and using one set as the source models for the prediction of the parameters of the other set's models. This approach, has been very successful in adapting the recognition system to new speakers. However, this technique was designed to cope with isolated word task and very limited vocabulary. Also, very hard constraints are applied to adaptation data by dividing the models into two groups so that only the adaptation data for source models (a fixed subset of models) is useful and only the other fixed subset of models are predictable. The above constraints severely limit the use of this predictive technique for speaker adaptation in large vocabulary continuous speech recognition systems.

For the model prediction technique to be applicable to continuous speech recognition, a new approach for speaker adaptation is needed. Such an approach must take into account the following problems of medium to large vocabulary continuous speech recognisers:

- The technique should be applicable to CDHMMs with mixture Gaussian output distributions, usually used for such tasks.
- Due to the nature of continuous speech, modelling is usually at a phone level and the adaptation data cannot be limited to certain phones since finding proper words and sentences including only those phones, or at least sufficient data for training them, could be very difficult or in some cases even impossible.
- Due to the nature of the adaptation data a hard division of the models into source and target models is not possible.

The above discussion suggests that in this case, not always the best correlated source state distribution may be used to predict a target state distribution parameters since adaptation data may not include data for that particular source distribution. Also, due to the nature of such adaptation data, there may be only very sparse data available for a particular distribution but this should not exclude the distribution from the list of target distributions.

A preliminary approach to the use of model relationships to predict model parameters in a continuous speech recognition system is shown in Figure 5.2. In this figure the simplified idea is shown by displaying darker models as the models that have a larger amount of adaptation data, hence receiving better parameter estimation by e.g. standard MAP adaptation technique. The diagram is simplified to show how the models can be related, while the real relationships are usually at a lower level such as state output distributions.

In order to discriminate between source and target models, a threshold is applied according to the amount of adaptation data received by each particular model. The models with larger amounts of adaptation data are considered as source models, while the rest (with smaller amounts of adaptation data) are considered as target models. Then the relationships between these 2 sets of models can be used for prediction purposes. Several drawbacks are associated with this approach, but seem to be inevitable if the technique is to be applied in a real situation: source parameters may not be fully adapted before relationships are used, while the targets are not fully unadapted parameters either; setting a proper threshold is difficult; the source and target models are not necessarily the best matching models; in order to calculate the regression parameters, which should be carried out before adaptation starts, the source and target parameters should be known.

The problems mentioned above are the result of coping with a real speech recognition system. In the next sections, it will be shown that in spite of the above problems, such a system is still capable of improving adaptation results in a continuous speech recognition system.

5.4 Basic RMP Technique

The RMP approach as described in this section is the basic approach for model prediction-based adaptation for continuous speech recognition. However, some changes have been made to this basic technique so as to either make it applicable to more improved modelling of speech or achieving a more refined speaker adaptation. These changes and their implementation in the system will be discussed in the next sections of this chapter.

5.4.1 Overall Approach

Figure 5.3 shows the basic system block diagram of the RMP-based speaker adaptation system. The first step in the implementation of RMP technique is finding the model parameter relationships. A number of speaker-specific model sets are needed for this purpose. A larger number of speakers in this case means a better regression, which can eventually lead to better model prediction via estimation of more reliable regression parameters. On the other hand, the best way to find these speaker-specific model parameters is to train speaker dependent systems of the same system architecture for all available speakers, which needs a large amount of training data to be available from each of these speakers. Hence, in the first step, a number of SD model sets should be trained for as many speakers as possible.

In the second step, the adaptation utterances from a new speaker must be applied to the system. This is done in a model adaptation framework (e.g. MAP) and all the model parameters, for which enough adaptation data is available, will be updated using the appropriate adaptation data. Normally, such an approach will leave a number of model parameters untrained or undertrained compared to some others which receive a large amount of adaptation data. Some other data related to the amount of adaptation each model state has received is also recorded during this process to help us later in separation of source and target parameters.

In the third step, using the information from the second step, the correlation parameters between all source and target state distributions of interest (using SD parameters) are calculated and the best correlated source distribution for each target is found. Then,

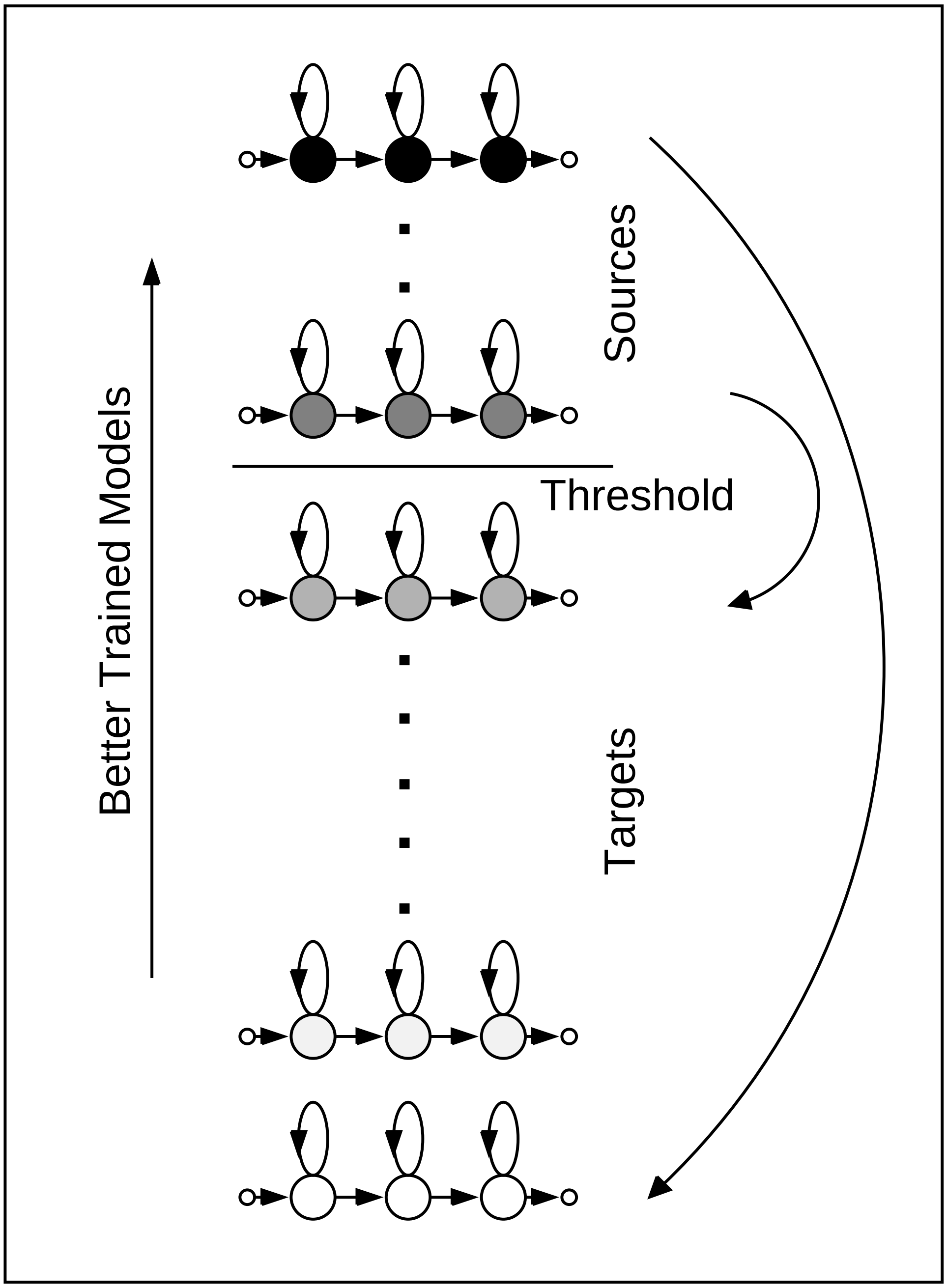


Figure 5.2: A simplified approach to the use of parameter relationships for model prediction in a continuous speech recognition system. The darker models are better trained ones and for the sake of simplicity relationships are shown for the whole models in place of state distribution parameters.

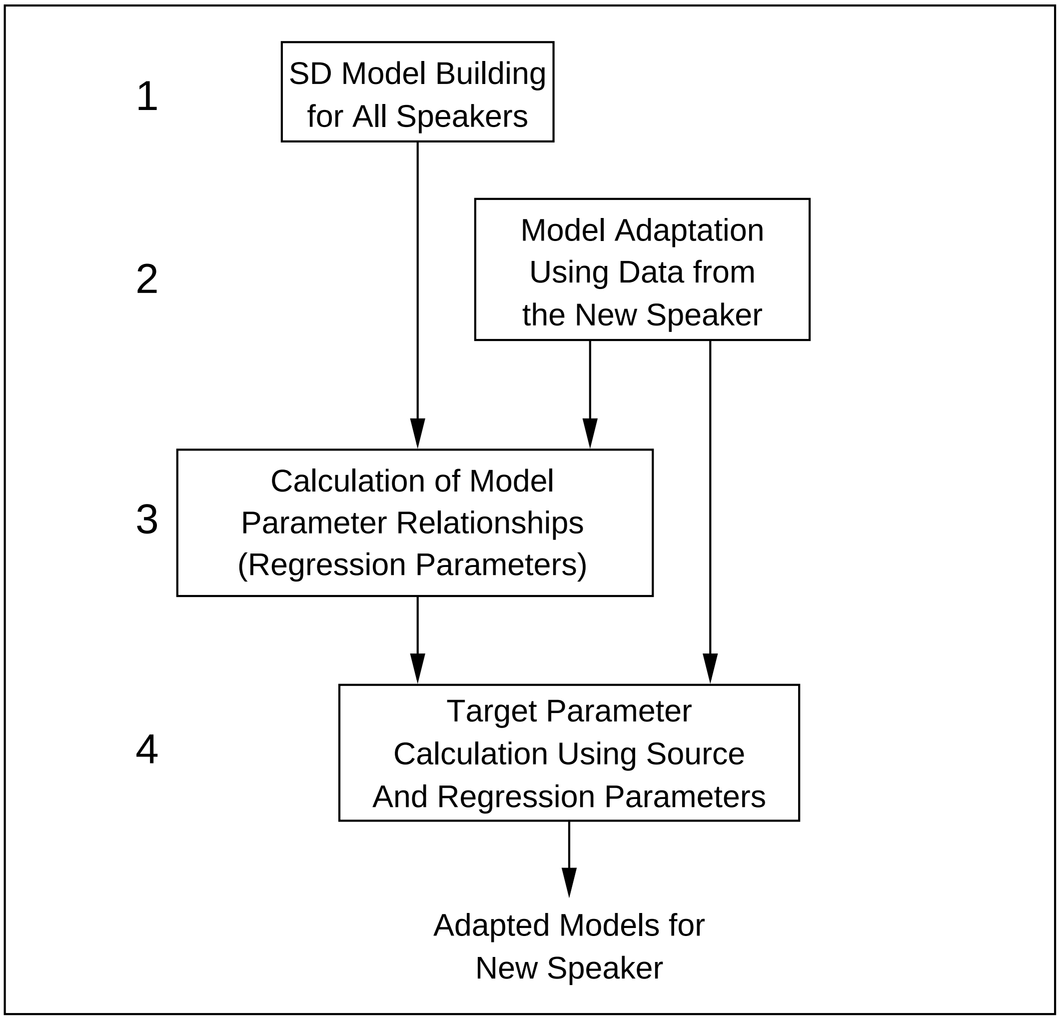


Figure 5.3: Basic system block diagram for an RMP-based speaker adaptation system

for all these best correlated sources and targets, all required regression parameters together with some other parameters needed for later calculations are calculated and saved. During the fourth step, all the regression-related parameters calculated in step 3 and the adapted models of step 2 are used to further adapt target state distribution parameters using source distribution parameters, giving a new set of target parameters in the system adapted to recognise speech from that new speaker.

The above steps will be discussed in detail during the next sections. Some changes to the basic concept and above block diagram will be covered subsequently.

5.4.2 SD Model Building

A number of speaker dependent systems must be available before regression parameters can be calculated. Baum-Welch re-estimation is usually used to train such systems in a maximum likelihood framework, using as much training data as is available from each speaker. Gauvain and Lee [35] achieved similar results using both MAP and ML estimation approaches for training SD systems. Huang et al. [42] have reported better performance compared to standard SD training, using MAP estimation, unless the number of training sentences grows very large. Although theoretically MAP estimation results must asymptotically approach ML estimation results with very large amounts of training data, it still remains to be decided just what amount of data should be considered large enough. This is mostly a matter of system architecture and number of system parameters. In the systems used in these experiments, as will be shown in Chapter 6, it has been found that MAP estimation, under certain circumstances, gives slightly better results in the case of speaker dependent training, for some of the system architectures used. Hence, a MAP estimation approach is used in this stage to train K speaker dependent systems.

5.4.3 Preliminary Model Adaptation

Model adaptation for the new speaker involves applying the available adaptation data to the models, so that all the model parameters are updated according to the amount of data available for them in the adaptation utterances. A good candidate for this job is MAP estimation which can give good output distribution parameter estimates, especially in the case of sparse training data, using the priors obtained from a speaker independent system. Since the RMP approach adapts only the mean vectors, the adaptation of other model parameters using MAP estimation is not important to us in this stage, thus only the estimation of the means is carried out.

In order to be able to separate source and target states the amount of adaptation each state distribution has received needs to be known. In Chapter 4, it was shown that the MAP estimation of mean parameter of the output distribution of state i of an HMM can be written as

$$\tilde{\boldsymbol{m}}_i = \frac{\tau_i \boldsymbol{\mu}_i + \sum_{t=1}^{T} c_{it} \boldsymbol{o}_t}{\tau_i + \sum_{t=1}^{T} c_{it}} \tag{5.5}$$

where $\boldsymbol{o}_t$ represents samples of the adaptation data, $\boldsymbol{\mu}_i$ represents the prior mean vector and c_{it} is the probability of being in state i at time t and the model generates the sequence $\boldsymbol{o}$. This probability can also be called *state occupation probability* at time t. Thus, $\sum_{t=1}^{T} c_{it}$ will be the total state occupation probability which gives the amount of adaptation each HMM state has received during this model adaptation process. These state occupation probabilities are also calculated and saved during this MAP estimation stage.

5.4.4 Regression Parameter Calculation

A flow diagram for the calculation of regression and other associated parameters is shown in Figure 5.4. For these calculations, K speaker dependent sets of models, trained in step 1 of Figure 5.3, as explained in Section 5.4.2, are used. Also, the state occupation probabilities, calculated in the second step (preliminary model adaptation), are used to determine the suitable state distributions of the models which could be used as sources and targets by applying a threshold. After partitioning the whole set of state distributions into two, the squared-correlation coefficients between all mean vector elements of the source distributions with the corresponding vector elements of the target distributions are found using (5.3) and averaged over all the vector elements for each pair of distributions. This provides a set of averaged squared-correlation coefficients for each target element with all the source elements. The best of these squared-correlations for each target component are found and the corresponding source component is marked as the best matching source component.

At this stage, to prevent weakly correlated matches between source and target state distributions taking place, a correlation coefficient threshold, T_c, is also applied. Thus, any matches with averaged squared-correlations less than the threshold are rejected and not used in the regression. This way, the best correlated source and target components are known.

The next step is the calculation of the regression parameters. For the case of simple linear regression, the regression parameters are calculated by applying the least squares method which leads to the following equations

$$b_1 = \frac{\sum_{k=1}^{K}(x_k - \bar{x})(y_k - \bar{y})}{\sum_{k=1}^{K}(x_k - \bar{x})^2} \tag{5.6}$$

and

$$b_0 = \bar{y} - b_1\bar{x}. \tag{5.7}$$

The above parameters are then calculated for each mean vector element of each target state distribution with the corresponding source state distribution and later saved individually for that target component. The other parameters calculated at this stage will be introduced and discussed in the next section.

5.4.5 Target Parameter Prediction

The last stage in the application of the RMP technique to speaker adaptation is the prediction of target parameters. The flow diagram for this stage is shown in Figure 5.5. A simple application of regression parameters, calculated in the last stage, to the basic simple regression formula, (5.1), will lead to a regression-based estimate of the target parameters. Meanwhile, for a more robust set of target parameters, a better estimate may be necessary.

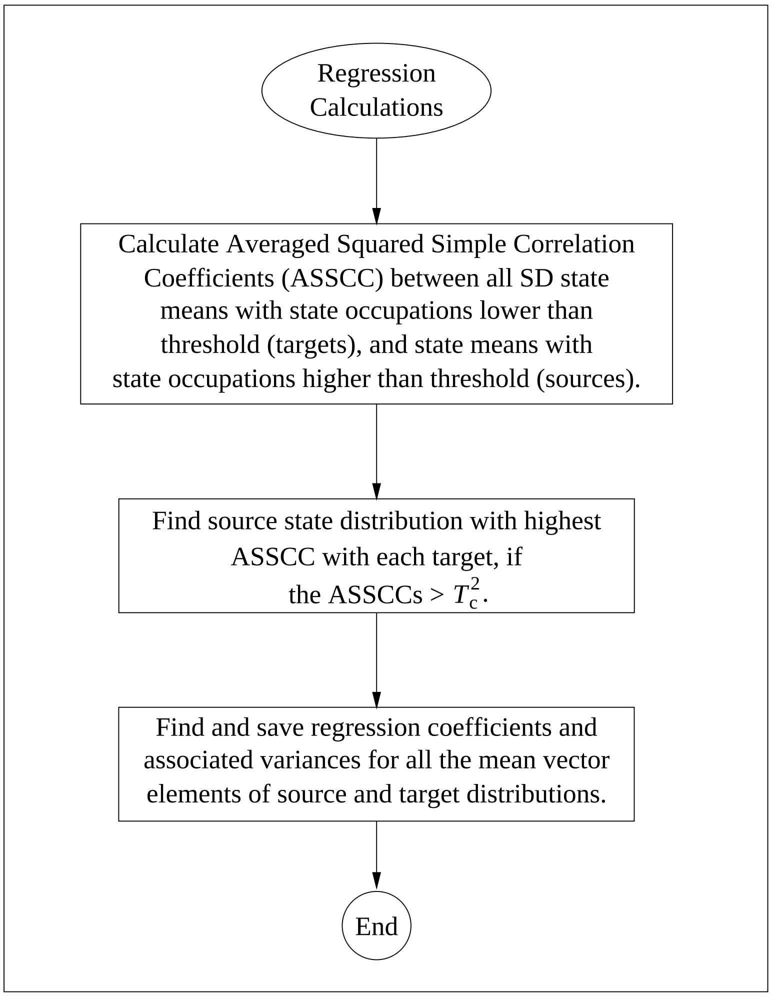


Figure 5.4: Flow diagram for the calculation of regression and other associated parameters in RMP.

If μ is considered to be a random sample from a normal distribution, in a maximum likelihood framework, this problem involves maximising a likelihood function, or calculating

$$\arg\max_{m} p(\mu|m) \tag{5.8}$$

where m is the mean of the final distribution of the target means. However, a Bayesian framework seems to be a more suitable candidate for this purpose due to the presence of a prior parameter, to combine the parameter estimated from linear regression with this prior. The MAP estimated target parameters can be considered as sources of *a priori* knowledge about the final estimation of target mean parameters. Hence the problem is to calculate the posterior probability $p(m|\mu)$, taking into account the prior density. Note that in this case, μ consists of a single observation, which is the regression-predicted target mean, and the mean of the normal prior density is considered to be the MAP estimated target mean parameter, ζ.

Such an approach leads to estimations of the type

$$\hat{m} = \mu \frac{s_\zeta^2}{s_\zeta^2 + s_\mu^2} + \zeta \frac{s_\mu^2}{s_\zeta^2 + s_\mu^2} \tag{5.9}$$

if the distribution of the means is assumed Gaussian [27]. Equation (5.9) gives the estimate for any element of a target state distribution mean vector in the RMP approach according to the values of regression estimated mean μ, and its associated variance s_μ^2, and the prior (MAP estimated) mean ζ, and its associated variance s_ζ^2.

The variance parameter associated with the regression predicted mean consists of two parts. The first part is the variance due to the application of linear regression. If the distribution of mean elements over different SD speakers during the regression parameter calculation section was assumed to be normal, then the estimated sample variance for the target parameters due to linear regression, s_e^2, can be calculated as

$$\begin{aligned} s_e^2 &= s_y^2(1-\rho^2) \\ &= \frac{1}{K-1}\sum_{k=1}^{K}(y_k - \bar{y})^2 - \frac{b_1}{K-1}\sum_{k=1}^{K}(y_k - \bar{y})(x_k - \bar{x}) \end{aligned} \tag{5.10}$$

where s_y^2 is the sample variance for the target element in the set of SD systems. There also exists another part in the total variance of the target parameters, which is due to the fact that the source distribution values used to find target ones using regression, are not true SD values, but the MAP estimates based on a small amount of adaptation data. This additional variance, due to error in the source distribution estimates, can be calculated by finding the sample variance of the MAP estimated parameters compared to the true SD parameters. Then, for each element of the state distribution mean vector, this can be computed as follows

$$s_\nu^2 = \frac{1}{K-1}\sum_{k=1}^{K}\left[(x_k - \nu_k) - \frac{1}{K}\textstyle\sum_{j=1}^{K}(x_j - \nu_j)\right]^2 \tag{5.11}$$

where ν_k stand for the MAP estimated source elements for the SD speakers, compared to x_k which are the actual SD source elements for these speakers. Obviously, these errors in the source parameter elements and the variance due to them will decrease with an increase in the amount of adaptation data, which leads to a better MAP estimation of source parameters.

The variances calculated in (5.10) and (5.11) can therefore be used to find the total variance of the target parameter due to regression. This can be written as

$$s_\mu^2 = s_e^2 + b_1^2 s_\nu^2, \tag{5.12}$$

and can therefore be used in (5.9).

Next, it is necessary to compute s_ζ^2 which is the variance of the MAP estimated target parameters. The same approach used to find the variance of the source element, s_ν^2, in

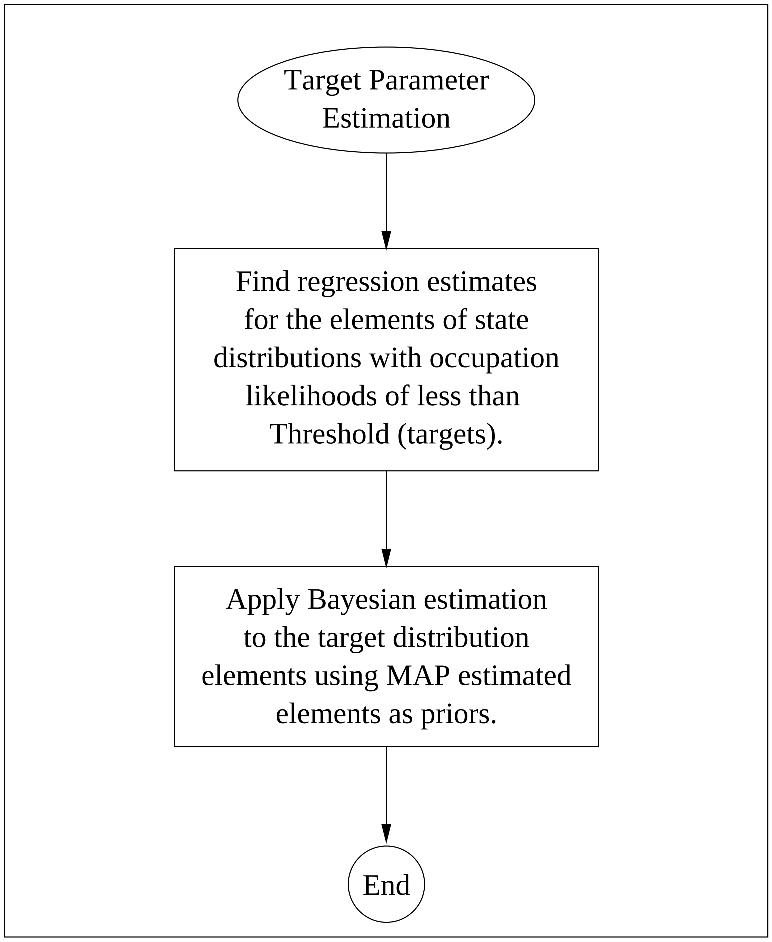


Figure 5.5: Flow diagram for the final adaptation of the target model parameters in RMP.

(5.11), can also be applied to find the variance for a target element, leading to

$$s_\zeta^2 = \frac{1}{K-1} \sum_{k=1}^{K} \left[(y_k - \zeta_k) - \frac{1}{K} \textstyle\sum_{j=1}^{K} (y_j - \zeta_j) \right]^2 . \tag{5.13}$$

Finally, using the variances s_μ^2 and s_ζ^2 and the target component mean values found by regression and the MAP estimated element as a prior, the final estimation of the target parameter vector elements can be found by (5.9).

The variances required above are found using SD speakers. The parameters needed for these calculations are not only the speaker dependent mean parameters, but also the mean parameters of the MAP estimated models for all the SD speakers with similar adaptation data. These calculations should be carried out in the previous stage of RMP (step 3 in Figure 5.3), together with the calculation of the regression parameters. It should also be evident from the discussions so far, that although not shown in the figure, a MAP estimation stage is also needed for all K SD speakers with the same adaptation data used from the new speaker in order to enable the variance calculations to be carried out. The resultant variance values for all the elements of the target state distributions are saved, together with the regression parameters, to be used in the final adaptation stage.

5.5 Mixture Gaussian Modelling

Mixture Gaussians are widely used in modern HMM-based continuous speech recognisers for detailed modelling of speech and have resulted in great improvements in the performance of speech recognisers [87, 106]. The approach discussed so far has not included mixture distributions in its scope. However, the inclusion of mixture distributions does not seem to be troublesome and the above approach can easily be generalised to the case of mixture distributions.

A simple extension to the case of mixture distributions is to treat all the mixture components of a single HMM state as parts of that state and still continue to find inter-state relationships. In this case, a target HMM state of M mixture components of V vector elements each would have $M \times V$ elements and the regressions are formed using all these elements and the corresponding ones from the states of other HMMs. Although such an approach can satisfy the basic idea of RMP, it is restricted by inter-state relationships, while in practice, as will be discussed in Chapter 6, it has been found that basing the relationships at a mixture component level can lead to superior performance, i.e. each Gaussian distribution is individually related to another one based on the average correlation between their mean vector elements.

It is also worth noting, for the purpose of separating source and target parameters, that the state occupation count, calculated and saved during the initial application of MAP model parameter estimation using adaptation data, was used in the basic technique, while its use in this case would not give an exact ranking of the best MAP-estimated mixture components. Hence, in this case the mixture component occupation counts are saved individually during the initial MAP estimation phase and used for the purpose of source and target distribution separation in later stages.

5.6 Multiple Regression

In the above discussion on the RMP technique, simple linear regression has been used for all parameter predictions and correlation calculations. Multiple regression can be more useful if enough data has been provided for robust estimation of regression parameters. In a multiple linear regression, (5.1) changes to

$$y = b_0 + \sum_{l=1}^{P} b_l x_l + \epsilon \tag{5.14}$$

where P is the order of regression (number of source distributions for each target) and b_l are regression coefficients.

One issue in this case is the appropriate regression order. One possibility could be the use of a full regression matrix of the size V, so that

$$\boldsymbol{y} = \boldsymbol{b}_0 + \boldsymbol{b}\boldsymbol{x} + \boldsymbol{\epsilon}. \tag{5.15}$$

The above equation can be interpreted as a transformation relating each element of the target distribution mean vector to all the elements of the source. Such transformations, as introduced in [66] and [32], can be successful in certain applications, but, specially in this approach, need larger amounts of data for parameter estimation due to the larger numbers of parameters. The application of multiple regression in the case of RMP is highly dependent on the number of SD speakers (regression points) available, which is an important limiting factor. However, it has been found that even with relatively small number of SD speakers, multiple regression can still be useful and improve the system performance (see Chapter 6).

In this case, (5.14) can be used to find a linear relationship between the target component vector elements and multiple source elements. The regression coefficients in this case can be calculated by solving the following matrix equation [12]

$$\boldsymbol{U}\boldsymbol{b} = \boldsymbol{w} \tag{5.16}$$

where $\mathbf{U}$ is a $P \times P$ matrix and $\boldsymbol{b}$ and $\boldsymbol{w}$ are $P \times 1$ vectors. The elements of $\boldsymbol{U}$ and $\boldsymbol{w}$ are given by

$$U_{nl} = \sum_{k=1}^{K} (x_{nk} - \bar{x}_n)(x_{lk} - \bar{x}_l) \tag{5.17}$$

$$w_l = \sum_{k=1}^{K} (y_k - \bar{y})(x_{lk} - \bar{x}_l) \tag{5.18}$$

and the value of b_0 is found by

$$b_0 = \bar{y} - \sum_{l=1}^{P} b_l \bar{x}_l. \tag{5.19}$$

The squared multiple correlation coefficient between the target y, and the regression predicted target value y', found using the regression formula, is given by

$$R^2_{yy'} = \frac{\sum_{l=1}^{P} b_l \sum_{k=1}^{K} (y_k - \bar{y})(x_{lk} - \bar{x}_l)}{\sum_{k=1}^{K} (y_k - \bar{y})^2}. \tag{5.20}$$

In this case, the variances introduced in Section 5.4.5 should be calculated taking into account multiple source distributions. Thus, the estimated sample variance due to multiple regression, assuming once again that the mean element values are normally distributed over different speakers, can be written as [28]

$$\begin{aligned} s_e^2 &= s_y^2 (1 - R^2_{yy'}) \frac{K-1}{K-P-1} \\ &= \frac{1}{K-P-1} \left[\sum_{k=1}^{K} (y_k - \bar{y})^2 - \sum_{l=1}^{P} b_l \sum_{k=1}^{K} (y_k - \bar{y})(x_{lk} - \bar{x}_l) \right]. \end{aligned} \tag{5.21}$$

Also, the additional sample variance in target parameters due to the errors in the source distribution parameter estimates, for each element of each source distribution is calculated as follows

$$s_{\nu_l}^2 = \frac{1}{K-1}\sum_{k=1}^{K}\left[(x_{lk}-\nu_{lk}) - \frac{1}{K}\sum_{j=1}^{K}(x_{lj}-\nu_{lj})\right]^2 \tag{5.22}$$

where ν_{lk} stand for the MAP estimated source elements. Therefore the total variance for the regression predicted target value, assuming for the sake of simplicity that the source distributions are independent, can be written as

$$s_{\mu}^2 = s_e^2 + \sum_{l=1}^{P} b_l^2 s_{\nu_l}^2. \tag{5.23}$$

In this case, the sample variance for the initial MAP estimated target parameters is the same as in (5.13) and finally (5.9) should be used to find the final estimate for any target mean vector element.

5.6.1 Choosing Source Components

The regression parameter calculation stage should be changed in this case since every target component is related to more than one source component. This can complicate the sequence of this stage since the application of multiple correlation coefficients, as given by (5.20), necessitates the availability of P source components for any target, before any calculation can be carried out. This means that for a system with S source mixture components and T target mixture components, for any potential target, a combination of P from S correlation calculations should be carried out. This means $T \times S!/[(S-P)!P!)]$ correlation calculations are required for all the mixture components. The number for the case of simple linear regression is $T \times S$. Even for a regression order of 2, the number of correlation calculations in this case would be $(S-1)/2$ times that of the case of simple regression. While the total number of mixture components could be of the order of several thousands in a typical modern speech recognition system, this factor, although related to the number of source components, could be a very significant one. Furthermore, the calculation of multiple regression correlation coefficients, depending on the order of regression, may need orders of magnitude more computation compared to a simple correlation coefficient calculation.

In order to alleviate this problem, a simpler approach to this task has been chosen by finding the averaged squared simple correlation coefficients between each target and all source components, as before, and finding the best P correlated components with each target (in place of one) to be used as sources for this target. This reduces the number of calculations for finding correlations to the same level as simple regression case, but still the best correlated sources with each target can be chosen, although individually, for multiple regression.

5.6.2 Dynamic Setting of Regression Order

In applying multiple regression to RMP, the order of regression is also left free to change dynamically. This is due to the fact that as a result of applying a correlation threshold to the process of finding matching targets and sources, in some cases, there may not be P source components available with averaged squared-correlation coefficients higher than the correlation threshold for a single target. In such cases, that target component is allowed to use a lower order of regression equal to the number of source components available for that target, satisfying the above condition. This approach leaves targets free to have any number of source components up to a maximum of P, hence increasing the chance for all targets to find at least one source component for the purpose of prediction.

On the other hand, this requires the order of regression to be set for each target individually, which means that this should be recorded and saved for each target. This option makes maximum use of any possible relationships between source and target parameters, which can lead to a better prediction, while the computation overhead and the extra memory and disk usage is small compared to the improvement in performance.

5.7 Multiple Thresholds for Distribution Separation

The application of a state or mixture occupation threshold for dividing distributions into two groups of sources and targets has been discussed. Although this technique has worked reasonably well for model adaptation, one small problem may still be encountered. Since the threshold divides the whole set of components into two sets of source and target distributions, it is very likely that components close to the threshold have very similar occupation counts, while some of them are counted as source and some others are counted as target distributions. Also, there is no limit during the whole adaptation process as to which source component is used to estimate the parameters of which component. Consequently, some source components with the occupation counts very close to the threshold might be used to estimate some target components which are themselves not too far from the threshold. This may result in source-target pairs with similar occupation counts which will not lead to reliable target estimates.

To overcome this problem and in order to have more reliable estimates, an occupation count gap is introduced between source and target parameters by using two thresholds known as higher threshold (T_H) and lower threshold (T_L). Therefore, some of the components between these two thresholds, for the sake of reliability, are left unused in RMP and the components with lower occupation counts than T_L are used as targets and those with higher occupation counts than T_H as sources, guaranteeing that there exists a minimum difference of occupation counts between source and target components. This approach is shown for symbolic Gaussian distributions in Figure 5.6. Note that the parameters of the components between the two thresholds are only re-estimated by MAP estimation.

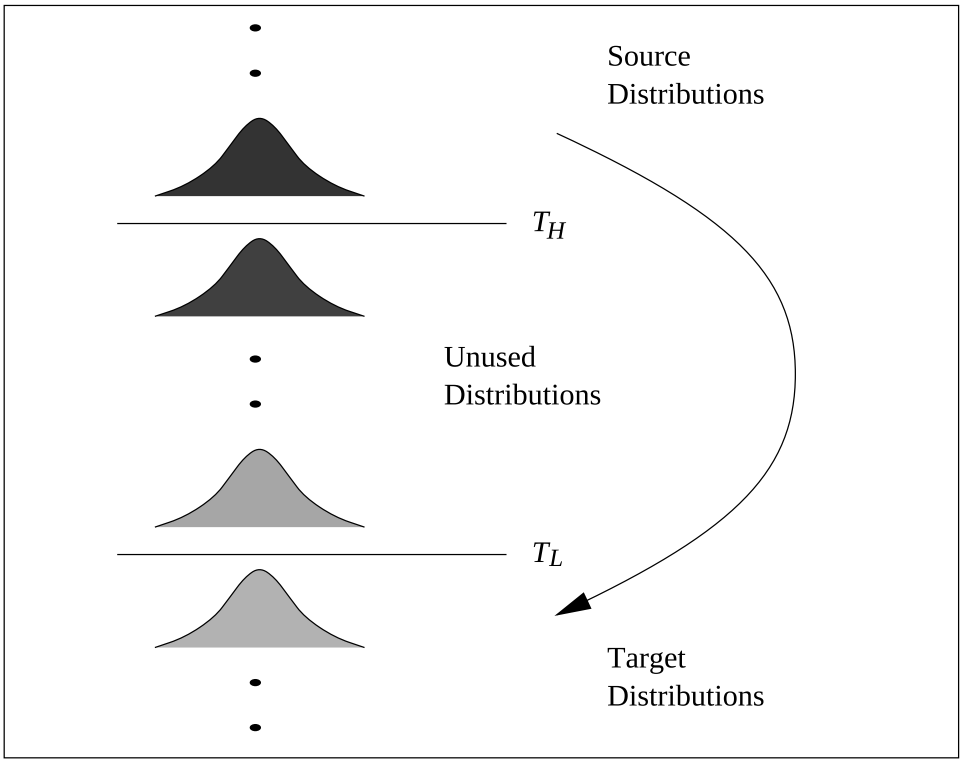


Figure 5.6: Application of multiple thresholds to Gaussian components in RMP. Note that the darker Gaussians represent better trained components and each component symbol should be considered as representing a multivariate Gaussian.

5.8 Application to State-clustered and Tied-mixture Systems

The use of RMP technique at the mixture distribution level, makes it suitable to be applied to any tied parameter system down to the mixture level. Hence the application to the case of tied-mixture, or state-clustered systems is straightforward. The only important change for this case is that the algorithm should work on the set of real mixture components or states available in the system, i.e. the tied parameters should be counted once in order to prevent duplicate mixture or state components from being used in the system.

5.9 Adaptation of Other HMM parameters

The RMP approach uses the relationships between the mean vector elements of the HMM output distributions to predict undertrained or untrained mean parameters. Hence, the total approach is based on the relationships between these mean parameters. However, attempts have been carried out to also adapt other parameters of the HMM. Other HMM output distribution parameters such as covariance matrices and mixture weights have been investigated.

5.9.1 Mixture Weight Adaptation

The estimation of the mixture weights in a maximum likelihood framework can be carried out by applying the Forward-Backward algorithm [53, 46]. In this case, for a K component mixture system, the mixture weights are estimated as follows

$$\hat{\omega}_{ik} = \frac{\sum_{t=1}^{T} c_{ikt}}{\sum_{t=1}^{T} c_{it}} \tag{5.24}$$

where c_{ikt} is the occupation probability for the mixture component k of the state i at time t and c_{it} is the total occupation probability for the state i at time t. Applying the Forward-Backward algorithm to MAP estimation, as discussed in Chapter 4, will result in an estimation of the mixture weight parameters as

$$\tilde{\omega}_{ik} = \frac{(\psi_{ik} - 1) + \sum_{t=1}^{T} c_{ikt}}{\sum_{k=1}^{K}(\psi_{ik} - 1) + \sum_{k=1}^{K}\sum_{t=1}^{T} c_{it}}. \tag{5.25}$$

Where $\psi_{ik} > 0$ are the parameters of the prior Dirichlet distribution of the mixture weights.

As the above equations show, in both cases of MAP and ML estimation of the mixture weight parameters, the estimation is derived by the use of the mixture occupation probability scaled by the state occupation probability, so that the condition $\sum_{k=1}^{K} \omega_{ik} = 1$ would be true for all i's. Hence the estimated mixture weight, for example for a speaker dependent system, whether trained by MAP or ML estimation, is representative of the amount of training data used for training that mixture component relative to the true state. Therefore these parameters, unlike means, cannot be expected to show a linear relationship among certain model distributions, once a regression is applied to them.

However, there still exists a chance of adaptation for these parameters when the adaptation on the mixture mean parameters has been carried out. In such cases, the target occupation count may be derived from the occupation counts of the sources by

$$\varsigma_{ik} = \frac{1}{V}\sum_{v=1}^{V} \frac{\sum_{l=1}^{P} |b_{lv}|\kappa_{ikl}}{\sum_{l=1}^{P} |b_{lv}|} \tag{5.26}$$

where κ_{ikl} is the occupation count of the lth source component for the current target component and ς_{ik} is the regression-derived target occupation count for mixture k of state i. Note that the occupation counts addressed here are the sum of occupation probabilities discussed earlier over all time frames.

A similar Bayesian approach to that used for the final stage of the adaptation of the mean components can be used to estimate a target component's final occupation count. In this Bayesian framework, the occupation count of the target found during the initial stage of MAP estimation is used as the prior parameter, while the regression-derived target occupation count is used as the single observation. Also the same variances calculated during the adaptation stage of the means are averaged and used here. Thus, the final occupation count for the target mixture would be

$$\hat{c}_{ik} = \frac{1}{V} \sum_{v=1}^{V} \left[\varsigma_{ik} \frac{s^2_{\zeta ikv}}{s^2_{\zeta ikv} + s^2_{\mu ikv}} + c_{ik} \frac{s^2_{\mu ikv}}{s^2_{\zeta ikv} + s^2_{\mu ikv}} \right] \tag{5.27}$$

where c_{ik} is the target occupation count, found during the initial MAP estimation.

Finally, once all the target occupation counts are calculated, they can be used in the MAP estimation mixture weight calculation formula, i.e. (5.25), to find the new mixture weights. Note that in this case, the new mixture weights should be calculated for all the mixtures of all states for all models, whether these mixtures have been used as targets or not, unless no target mixture is present in that state, where the mixture weights would remain untouched.

5.9.2 Covariance Matrix Adaptation

Unlike mixture weight parameters, the adaptation of the covariance matrices is more complicated due to the dependency of covariance estimation to several parameters in either ML or MAP approaches and also due to the fact that covariance matrices, like mixture weights, cannot be treated within a linear regression framework. Hence, the extension of this technique to adaptation of the covariance matrices is left to future work.

5.10 Other Changes to the Basic Approach

The RMP approach explained above consists of a large number of calculations, especially in the stage of regression parameter calculation, where $T \times S$ averaged squared-correlation calculations should be carried out before finding out which sources are appropriate for each target. Then about T regression and variance calculations are needed, if all the targets are to be estimated using available sources. While the second set of calculations are almost unavoidable, the first set, which usually constitute the main part of calculations, can be reduced.

Investigations on the relationships between the distributions belonging to different phones, and/or different states of similar phones, reveal that the number of relationships with high correlations between distributions from different states are few and far between. Thus, a constraint can be set to limit the correlation calculations to only those distributions of similar states from different models. This condition can reduce the amount of computation required at this stage to a third, without causing a major degradation to the overall result.

Another point to consider is that most of the relationships usually take place within fairly broad phonetic groups. Hence, there is usually no need to look for inter-group relationships. This can also be very helpful in reducing the number of computations. A very broad classification could be the division of the phones into vowel and consonant groups. Also, more specific phonetic groups can be used, but increasing the number of such groups may lead to performance degradation to some extent.

5.11 Summary

The RMP technique has been discussed in detail in this chapter. This is a technique developed to be used for the purpose of speaker adaptation using only limited adaptation data. The approach relies on a primary model adaptation technique, such as MAP estimation, to apply the small amount of adaptation data to the HMMs, and then uses better trained distributions, together with relationships found from a regression analysis on SD model sets, to predict the parameters of poorly trained distributions.

The relationships found for the mean parameters of model distributions have been discussed, together with the use of simple and multiple linear regression for finding these relationships. Problems associated with the separation of source and target components and the calculation of correlation coefficient between distributions have also been addressed. Also, a Bayesian approach for final estimation of the model parameters has been explained during this discussion.

The RMP approach is designed to be applicable to modern CDHMM-based continuous speech recognisers and can be applied to single or mixture Gaussian distribution means. Some refinements in the adaptation procedure which have led to improvements either in performance or in the amount of computation have also been discussed. The application of this technique to a continuous speech database and the results obtained will be given in Chapter 6.

Chapter 6

Experimental Evaluation

This chapter is dedicated to the experimental work and evaluation carried out using the techniques discussed so far. Details of the databases used and the acoustic modelling specifications in our systems will be given before discussing the specific evaluation procedures followed. The results, where appropriate, will be compared to similar systems and techniques.

6.1 Databases Used

The evaluation of MAP adaptation, with and without speaker clustering for prior improvement, and RMP adaptation is performed for the ARPA Resource Management task (RM1). Also, an evaluation of the RMP technique, which also includes a basic evaluation of MAP, is carried out on the ARPA Wall Street Journal (WSJ) task. A simple description of the above databases follows, with a more detailed description of the parts used in these experiments given in Appendix B.

6.1.1 The RM Database

The ARPA RM1 task [84] is the main database used for the evaluations of the MAP estimation, speaker clustering and RMP techniques. The SI-109 part of this database is used for building the baseline speaker independent system, which is frequently used as the base system, or the system used in finding the prior parameters in the case of MAP estimation. This part of the database consists of the speech from 109 speakers with 30-40 sentences from every speaker.

The adaptation and speaker dependent model training has been carried out using the speaker dependent (SD) section of the database. This part consists of speech data from 12 speakers with 600 sentences from each speaker. For adaptation purposes, the number of sentences needed are selected in order from 600 sentences of each speaker. The reason for this is that the number of adaptation sentences provided for each speaker in the RM1 SD database is only 10; this is not enough for some purposes and besides, the length of these sentences is inconsistent with the average sentence length of the speaker dependent section of the database. Also, having access to a larger number of adaptation sentences provides a more accurate way to analyse performance improvements with an increase in number of adaptation sentences.

For test purposes, in all cases, all 100 test sentences available from each SD speaker were used. These include all the speaker dependent evaluation test sentences from February 89, October 89, February 91 and September 92 RM evaluations. All results use the standard RM word-pair grammar (perplexity 60).

6.1.2 The Wall Street Journal Database

The ARPA Wall Street Journal Database [83, 57] is used for part of the evaluation of the MAP and RMP techniques. For building the baseline speaker independent system, the SI-284 data from both WSJ0 and WSJ1 sets were used. This system is used as the initial system for adapted system training and also as the speaker independent system for estimating prior parameters of the MAP estimation process. As the name implies, this part of the database consists of data from 284 speakers, with about 90-150 sentences per speaker.

For building the SD models for RMP evaluations, the data from both the WSJ0 and WSJ1 data sets was used. The portion of the data used consists of the long-term speaker dependent data from the WSJ0 set, including three speakers with about 2400 sentences per speaker, and long-term speaker independent data from WSJ1 set, including 25 speakers with about 1200 sentences per speaker. In order to enable later evaluation of the SD systems, 20 sentences in the 5K word vocabulary domain from each speaker are removed from his/her training data set to become his/her test sentences. The 5K word vocabulary is the one defined by MIT Lincoln Laboratories and used in the ARPA evaluations of 1992, 1993 and 1994.

The adaptation and test sentences for the WSJ experiments were not chosen from the set of SD speakers, but a set of SI test speakers were used for this purpose. This is possible since 40 adaptation sentences are available for all the speakers in the database. Another possible source of adaptation speakers is the ARPA 1994 spoke S3 set which consists of 11 non-native speakers with 40 adaptation and 20 test utterances for the development test section, and 10 speakers with a similar number of utterances for the evaluation section. All the test results for this case use a trigram grammar.

6.2 Data Parametrisation and Acoustic Modelling

Due to the differences in the parametrisation and modelling techniques used for the RM and WSJ databases, this is explained in two distinct sections.

6.2.1 Parametrisation and Modelling for RM data

All the data used from the RM database is parametrised using 12 cepstral coefficients plus log energy coefficient and first and second order derivatives of these values, making vectors of dimension 39 [107]. The parametrisation is carried out using the HTK tool HCode and the details are given in Appendix B.

The speaker independent systems used as both the initial system and the base for estimating the prior parameters in Bayesian adaptation are the ones already available at the CUED speech group and were constructed starting from a simple monophone system and extending to single and mixture Gaussian monophone and triphone systems [106].

The phone set and dictionary used for this purpose were those produced by CMU and listed in [63]. The basic phone set consists of 47 phone symbols plus silence and the dictionary defines a single pronunciation for each of the 991 words used in RM task. Since word-level SNOR[1] transcriptions are available for RM data, the word-level label files for the test data and phone-level label files for the training data are found using the available tools to convert the SNOR format files to label files with the HTK format.

For each of the above mentioned 48 phones, a five state HMM (including the non-emitting entry and exit states) with the left-to-right topology similar to one shown in Figure 2.1, but mostly without skip transitions, were defined. The single Gaussian monophone models were trained, using the RM SI-109 data coded as indicated above, by several iterations of embedded Baum-Welch re-estimation procedure. The resultant system was what is known as the RM single Gaussian monophone SI system.

For building the mixture Gaussian monophone system, starting from the single Gaussian models, mixture splitting is carried out by converting any k component p.d.f. to a $(k+1)$ component one and performing two iterations of embedded Baum-Welch re-estimation for updating all the HMM parameters at every stage and repeating this procedure in several stages to reach the desired number of mixture components[110].

The word-internal context-dependent models were built using the state-tying approach [107]. The concept of state-tying is introduced to improve the system performance by providing a better distribution of the training data among the HMMs, especially those which do not receive much training data. This also reduces the total number of states and leads to a more manageable system [110]. The clustering approach developed for this purpose is described in [108]. After this stage, the clustered states were tied and a system of tied-state word-internal triphones was generated. This system includes a total of 1704 HMMs and 1581 tied states. Finally the parameters of this system of HMMs were re-estimated once again using the Embedded Baum-Welch re-estimation procedure.

The increment of the number of mixture components per p.d.f. was carried out in the same way as described for single Gaussian monophone systems.

[1]Standard Normal Orthographic Representation (SNOR) format is a standard format used in such cases for representing transcriptions.

6.2.2 Parametrisation and Modelling for WSJ data

The parametrisation of the WSJ data is performed in almost the same way as for the RM data. The only difference is that here, the cepstral features are normalised on a sentence basis by subtraction of the mean of the cepstral coefficients.

The baseline SI systems used for WSJ evaluations were once again the systems already available at the CUED speech group. These systems used the 1993 LIMSI WSJ Lexicon and phone set and the HMM structure used is similar to the one used for RM evaluations. The SI system was trained with the SI-284 (WSJ0+WSJ1) data set and the training procedure was similar to RM's. The triphone system used, however, was a cross-word triphone system generated using the state tying process carried out by a decision tree based state-clustering approach, basically designed to tackle the problem of unseen triphone models [108, 105]. The basic system used consisted of 23364 HMMs with 6399 tied states [103].

It is worth noting here that the phone level transcription files in this case are generated by applying a Viterbi forced alignment procedure to the data, using an available set of models and dictionary. For the RM database, since the dictionary only contains a single pronunciation of each word, forced alignment was not necessary. However, in this case, different pronunciations for any single word may exist. Hence a forced alignment is inevitable, although, since the available models are not trained for the speaker whose data is being coded, the resultant transcriptions would be consistent with the current (SI) model parameters but not necessarily with the one which would be obtained from an SD parameter set.

6.3 Evaluation of Bayesian Adaptation

In MAP estimation one of the important parts is the determination of the prior parameters. For the case of HMMs with diagonal covariance matrices, the parameters of the normal-gamma prior distributions may be calculated using Equations (4.46) to (4.49). For the rest of the parameters, i.e. those with Dirichlet prior distributions, Equations (4.38) to (4.41) could be used. However, this would become rather complicated, since it initially involves the use of several different observation sets to estimate several sets of parameters, $\hat{\boldsymbol{\lambda}}_i$, in order to be able to estimate the parameters needed to calculate the above prior parameters, i.e. $E(a_{1i})$, $\mathrm{Var}(a_{1i})$, $E(a_{iN})$, $\mathrm{Var}(a_{iN})$, $E(a_{ij})$, $\mathrm{Var}(a_{ij})$, $E(\omega_{ik})$, $\mathrm{Var}(\omega_{ik})$, $E(r_{ikv})$, $\mathrm{Var}(r_{ikv})$, $E(m_{ikv})$ and $\mathrm{Var}(m_{ikv})$.

An alternative to this approach, as stated before, could be the use of an already trained speaker independent CDHMM system. This approach has the advantage of having a set of system parameters that have been estimated by the data polled from a large set of speakers. However, because of the lack of data, the calculation of above equations can not be fully carried out and hence some *ad hoc* constraints, or even parameter settings, need to be applied to the prior parameter estimation formulae. In spite of this, this approach is very appealing since normally when speaker adaptation is used a speaker independent system is readily available which can be used as a base for prior parameter estimation and saves a large amount of time and effort. Also, the application of the MAP estimation approach is more straightforward because the same speaker independent system can also be used as the initial system for Forward-Backward re-estimation procedure.

In this section, first an implementation of the latter case of MAP estimation for the purpose of speaker adaptation will be explained, and later, the improvements achieved by applying improved priors to MAP estimation will be discussed.

6.3.1 Bayesian Adaptation Using SI-Based Priors

This approach can be carried out by applying a set of constraints to the prior parameter estimation procedure by using subjective knowledge of the prior parameters. Examples of such constraints were given in Section 4.6 for the case of full covariance matrix HMMs. Similar constraints can also be set for the case of diagonal covariance matrix HMMs. This approach has been used by Gauvain and Lee [34, 60] in a Segmental MAP estimation approach and some simplifying constraints have been applied to the prior parameters to facilitate the estimation of these parameters. The resultant *ad hoc* prior parameters used in the experiments reported in [34, 60] were calculated as follows:

$$\psi_{ik} = \hat{\omega}_{ik} \sum_{k=1}^{K} \tau_{ik} \tag{6.1}$$

$$\mu_{ik} = \hat{m}_{ik} \tag{6.2}$$

$$\alpha_{ik} = \frac{1}{2}(\tau_{ik} + 1) \tag{6.3}$$

$$\beta_{ik} = \frac{1}{2}\tau_{ik}\hat{r}_{ik}^{-1} \tag{6.4}$$

where $\hat{\omega}_{ik}$, $\hat{m}_{ik}$ and $\hat{r}_{ik}^{-1}$ are the mixture weights, means and precision parameters extracted from the SI system. From the above equations all the prior parameters calculated depend on the values of τ_{ik} set for each mixture component. However, in [34, 60] τ_{ik} is set to a fixed value of 2 for all the mixture components.

As can be seen, apart from the mean parameter value, the parameters are estimated by some *ad hoc* constraints. Nevertheless, this approach has been used in [34, 60] due to its simplifying characteristics, which allows the use of SI parameters. It should also be noted that the state transition probability parameters have been assumed to be fixed in this approach, which means that no MAP estimation for these parameters is carried out.

Due to the convenience of estimating the prior parameters from the parameters of an SI system, a similar approach has also been carried out here, but in our case, a Forward-Backward MAP estimation procedure has been followed. Using a system of HMMs with diagonal covariance matrices, the estimation of the prior parameters has been carried out using similar formulations, while different fixed values have been used for τ_{ik} to assess the effect of its change on the adaptation process. The adaptation in most of the cases is only carried out for means, variances, mixture weights or combinations of these three. Since the

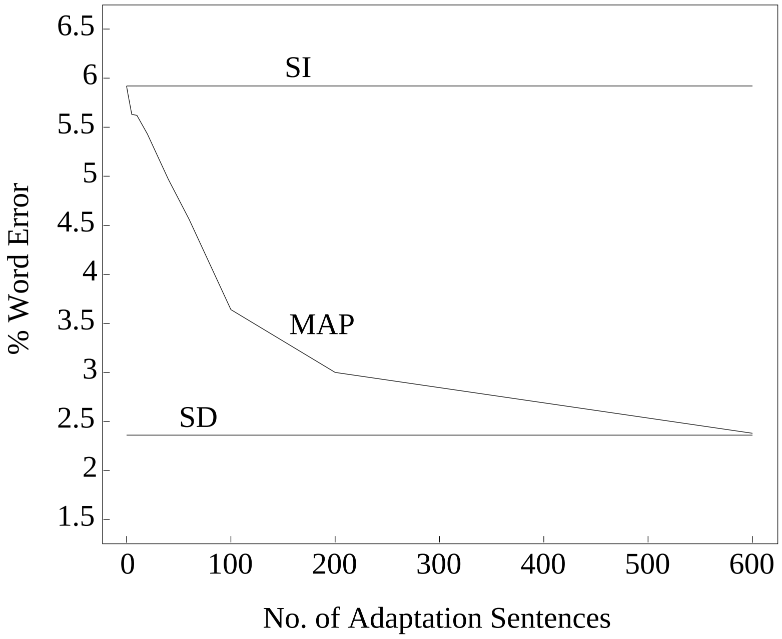


Figure 6.1: The results of speaker adaptation using MAP estimation on a 6 component mixture triphone system. The Gaussian means, variances and mixture weights have been MAP estimated for these experiments. All results are averaged over all 12 speakers available.

implementation of the adaptation of state transition probabilities is reported by very few researchers, and because the output distribution parameters are known to be numerically more important than transition parameters in specifying an HMM (for example see [55]), either the values of η_i, η_i' and η_{ij} can be set to one during these experiments, to disable the MAP adaptation according to Equations (4.26) to (4.28), or, more preferably, the state transition parameters are not updated at all.

The results of applying the above MAP estimation approach to the parameters of a set of 6 component mixture Gaussian context-dependent HMMs is shown in Figure 6.1, together with the results of the tests carried out on SI and maximum likelihood estimated SD systems. The results shown are the average of the results obtained on all the 12 speakers of the SD part of RM database. In these tests, the value of τ_{ik} in Equations (6.1), (6.3) and (6.4) is set to 10 and the adapted parameters are means, variances and mixture weights of the Gaussians.

As is shown in Figure 6.1, the important characteristic of the MAP estimation is that starting from speaker independent system performance, it gradually improves with the amount of adaptation data received from the new speaker and asymptotically leads towards the SD system performance. Two important features of the Bayesian approach are shown in this figure: the ability to make use of small amounts of adaptation data for system performance improvement, i.e. sparse data conditions; and the asymptotic

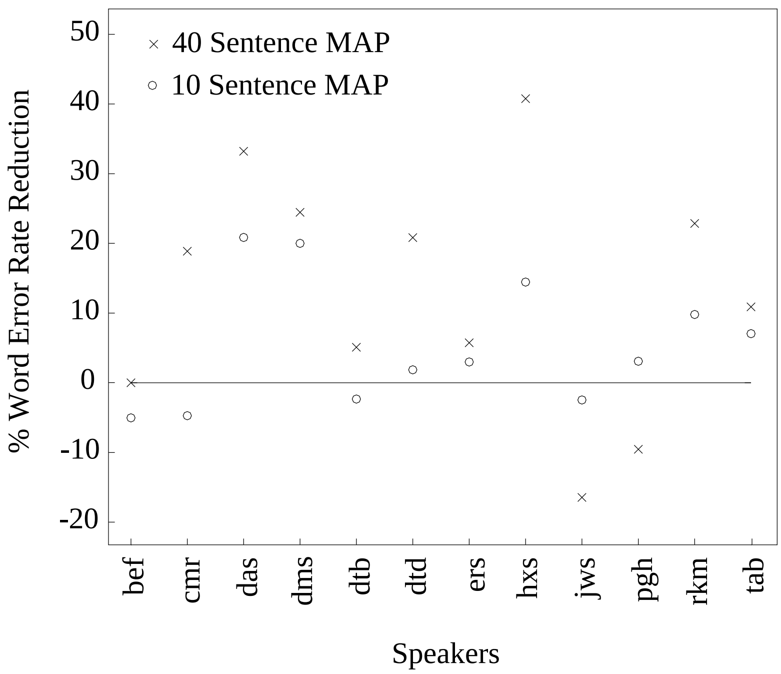


Figure 6.2: The percentage word error rate reduction of individual RM SD speakers with 10 and 40 adaptation sentences, compared to the SI system performance. The baseline word error rate is 5.92%.

improvement of the adapted system towards the SD system performance.

Figure 6.2 displays the percentage improvement in the word error rate for each of the 12 individual speakers in the RM SD database for 10 and 40 adaptation sentences, using the 6 mixture state-clustered triphone system discussed above. For most of the speakers, some improvement is obtained with both 10 and 40 sentences. However, in four cases, the system performance deteriorates with 10 sentences, while with 40 sentences, this problem is faced for 2 of the 12 speakers. This is mostly believed to be due to the sparseness of adaptation data. In other words, for such speakers, the SI system performs better and much more adaptation data is needed before any improvements compared to the baseline SI system performance are obtained.

Similar experiments have been carried out on single Gaussian context-independent, single Gaussian triphone and 2 component mixture Gaussian triphone systems and the results are shown in Figure 6.3. Similar conclusions can be drawn from these results. This shows that a consistent improvement, according to the amount of adaptation data available, can be obtained by the application of MAP estimation in the cases of context-dependent and context-independent modelling with differing complexities of mixture Gaussian output distributions.

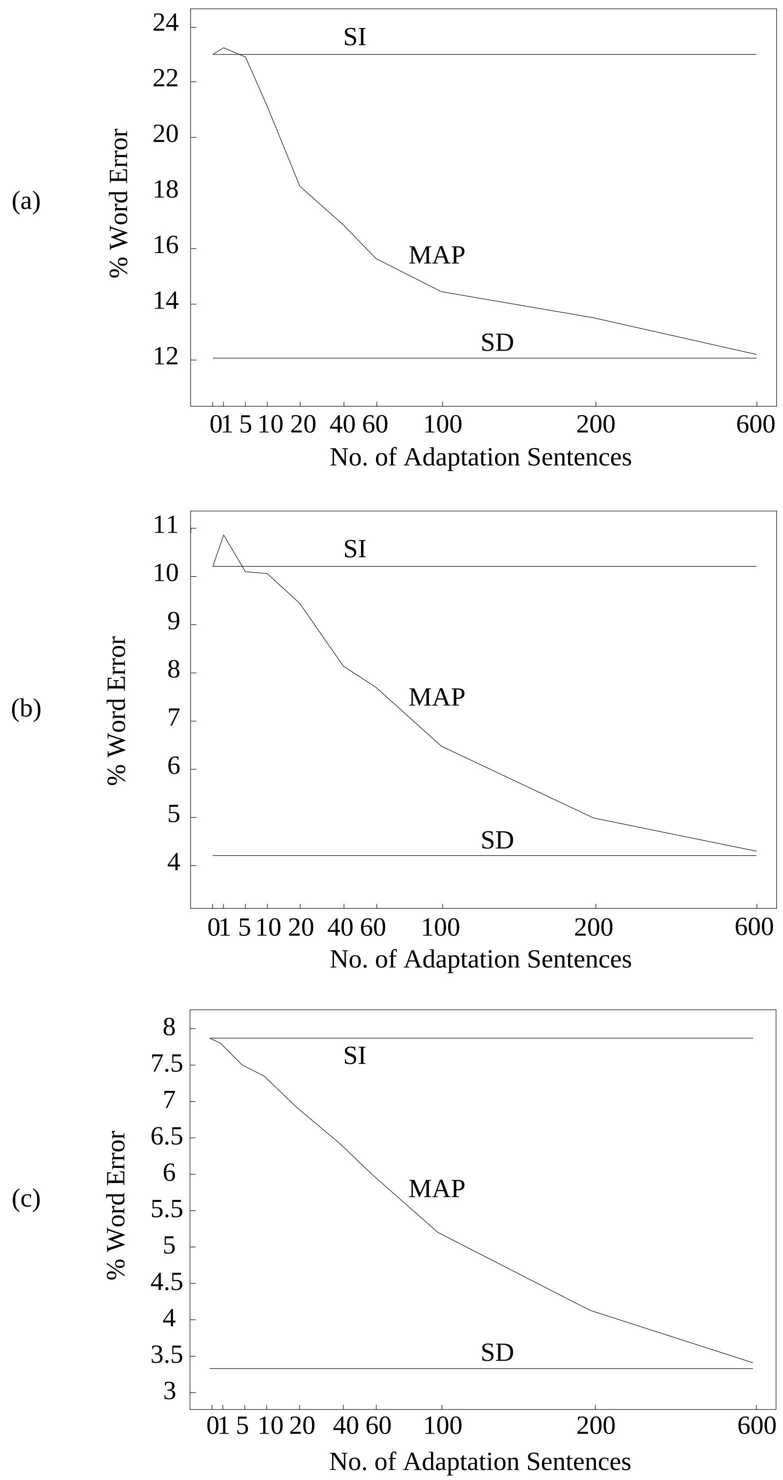


Figure 6.3: The results of the application of MAP estimation technique to different HMM systems for speaker adaptation; (a) single Gaussian monophone system, (b) single Gaussian triphone system and (c) 2 component mixture Gaussian triphone system. All results are obtained by applying MAP estimation to means, variances and mixture weights of the Gaussians and averaged over all 12 speakers available.

6.3.1.1 Effect of Adapting Different Parameters on System Performance

Figure 6.4 displays the improvement obtained by the application of MAP estimation to Gaussian mixture weights and means individually for the case of the 6 component mixture Gaussian system discussed before. The results are compared to that of the case of full adaptation using means, variances and mixture weights. Note that from Equation (4.31), due to the use of the MAP estimated mean values in the estimation of variance, MAP estimated variances are not shown without mean estimation. As can be seen, the improvement in the overall system performance due to the adaptation of mixture weights is limited, while Gaussian means, as expected, have the largest influence in the improvement of the system performance. Also, the improvement due to the adaptation of all three (mixture weights, means and variances) gives the best overall system performance, although it is only slightly better than the performance obtained only by adapting the means. The performance of the baseline SI system and the ML estimated SD system with means, variances and mixture weights training is given for comparison purpose.

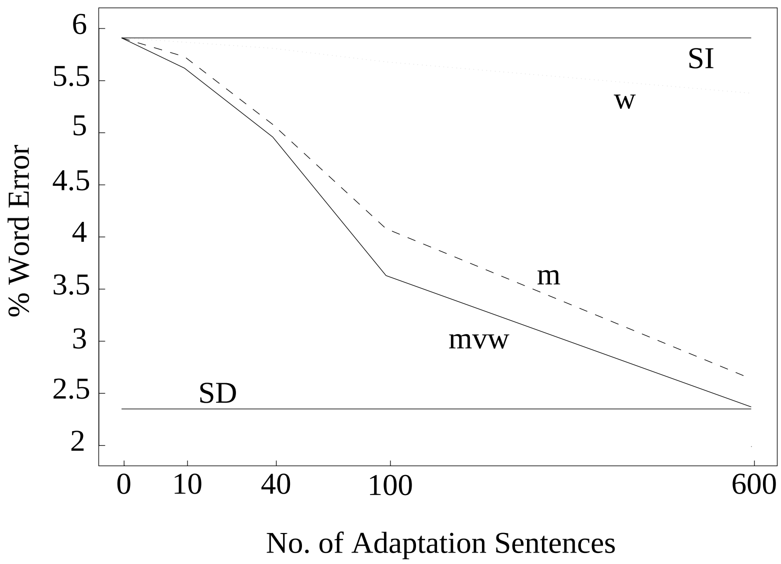


Figure 6.4: A comparison between the performances of the systems with MAP estimations of different parameters. In this figure, 'w' denotes the adaptation of mixture weights, 'm' denotes the adaptation of means and 'mvw' denotes the adaptation of means, variances and mixture weights. The SD performance given is from a similar system, with means, variances and mixture weights re-estimated using Baum-Welch re-estimation technique. All results are averaged over all 12 speakers available.

6.3.1.2 Comparison with ML Training

To show the difference between the use of MAP estimation approach for adapting the system to a new speaker and the maximum likelihood approach, i.e. Baum-Welch re-estimation, the results of the adaptation of the system means, variances and mixture weights by these two methods have been compared in Table 6.1. Note that in this case, both systems used the baseline SI system as their initial system.

While MAP estimated system, by introduction of adaptation sentences, starts to improve from the baseline SI system performance towards the SD system performance, even when small amounts of adaptation data are available, the performance of the ML-estimated system, again starting from the SI performance, greatly deteriorates for the first few utterances of adaptation. This is usually known as the effect of sparse training data in ML-based training. Depending on the size of the system, the number of parameters to be adapted and the material included in the adaptation utterances, the situation may get even worse by the introduction of a few further adaptation sentences. It should be noted that the MAP and ML tests were carried out in similar conditions, i.e. the models were updated with as little as 1 observation and a variance floor was used during adaptation. For the case of the system tested here, i.e. a tied-state triphone system with 6 component Gaussian mixture output distributions, there are 9486 mixture components, each a 39 dimensional multivariate Gaussian, which makes a total of about 750,000 parameters to be adapted.

Ad. Sent.s	0	1	5	10	20	40	60	100	200	600
SI	5.92	-	-	-	-	-	-	-	-	-
MAP	5.92	5.86	5.62	5.63	5.43	4.97	4.56	3.64	3.00	2.38
ML	5.92	39.76	63.05	65.88	57.90	43.75	39.25	25.32	7.32	2.36
SD	-	-	-	-	-	-	-	-	-	2.36

Table 6.1: Results of the application of Forward-Backward method for MAP and ML estimation of the parameters of the tied-state triphone HMM system with 6 component mixture Gaussian output distributions, using different amounts of adaptation data. The results reported are the average of the error rates obtained by each method among all 12 speakers under test. In both cases, means, variances and mixture weights of the Gaussians are trained and for MAP estimation, a value of $\tau = 10$ is used.

As Table 6.1 shows, in this case, the deterioration of the performance of the ML-trained system reaches its maximum at some point around 10 adaptation sentences, after which the system performance starts to improve. However, a few hundred adaptation sentences need to be used before even the same SI system performance is achieved, although eventually the system acquires the speaker dependent performance.

6.3.1.3 Iterative MAP Estimation

As pointed out in Section 4.6, an iterative approach to prior parameter estimation may be used for improving the MAP approach efficiency. Although this was suggested for the empirical Bayes approach, it can also be applied to the current approach.

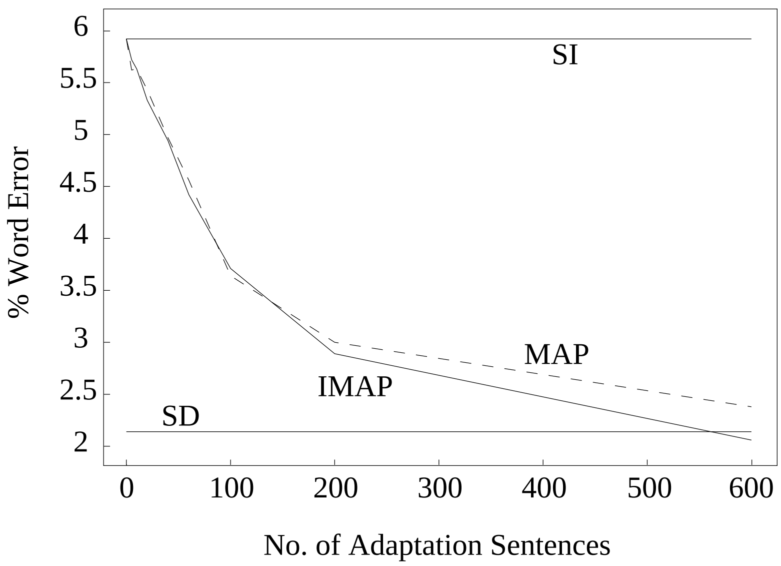


Figure 6.5: The results of applying multiple iterations of Forward-Backward MAP estimation for improving the prior parameter estimation. The MAP estimation results with 4 iterations are marked IMAP and the single iteration estimation results are marked MAP. The SD result is also obtained running 4 passes of ML estimation.

For this purpose, as before, first the baseline SI system is used as both the initial system and the system for prior parameter calculation. After applying the Forward-Backward MAP estimation procedure to that, an initial estimate of the HMM parameter set, $\hat{\boldsymbol{\lambda}}$, is obtained. Further iterations of this process are performed in a similar way to the first iteration, with the exception that in the next iterations, new estimates of HMM parameters are used in place of the SI system parameters for prior parameter calculations, while the same SI system is still used as the initial system for parameter estimations.

The results of applying this technique to the estimation of the parameters of a 6 component mixture Gaussian triphone HMM system are shown in Figure 6.5, together with our previous results obtained using a single run of MAP estimation. This figure shows that with this system configuration, with smaller amounts of adaptation data, both techniques perform similarly, while with a larger number of adaptation sentences, the newer approach outperforms both the single iteration MAP estimation technique discussed earlier, and the ML estimation approach with the maximum available amount of adaptation data. Note that in this case, the ML estimation is also carried out using 4 iterations of Baum-Welch algorithm.

6.3.1.4 Effect of τ on Adaptation Results

Equation (4.30) indicates that the parameter τ plays an important role in the process of adaptation, especially for the mean parameter. According to (4.30), the larger this

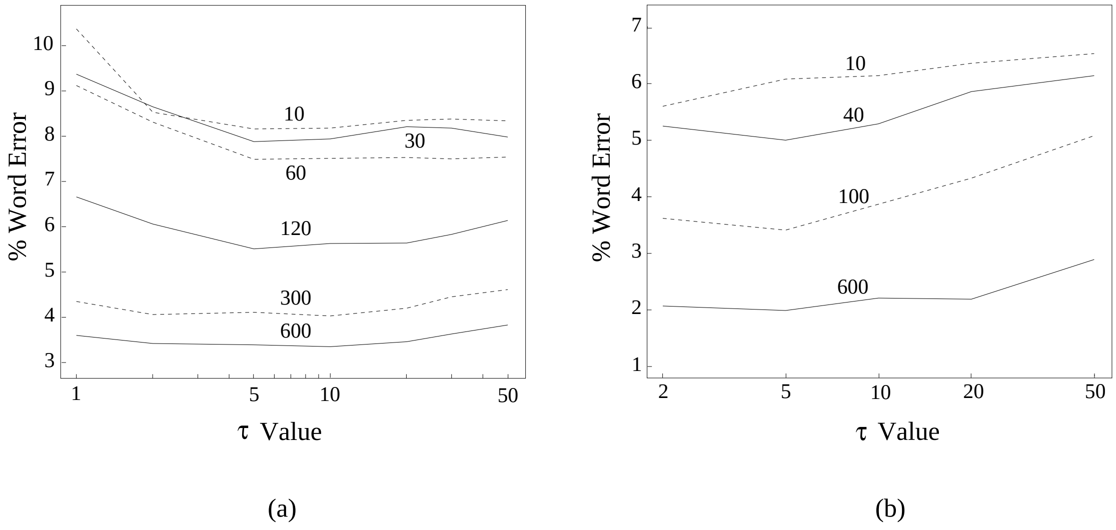


Figure 6.6: The change in the system performance due to the change in the value of τ for MAP estimation with different number of sentences for a) A 2 component mixture triphone system and b) A 6 component mixture triphone system. For these experiments, 4 iterations of MAP estimation are carried out in each run. The results are again averaged over all speakers.

parameter, the more dependent becomes the estimated mean to the SI mean, while a small τ can result in a value of estimated mean very close to the ML estimated one. This means that τ acts as a weight for the effect of prior parameter in the MAP estimation process. Setting τ to a small value, therefore, speeds up the adaptation process by giving more weight to observation samples at the expense of poorer system performance, while a larger τ, giving more weight to prior parameters, slows down the adaptation process so that little adaptation is performed until a considerable amount of adaptation data is received. Thus, finding proper values for this parameter is crucial to the adaptation process.

However, the *ad hoc* prior parameter setting discussed earlier in this section uses a constant and pre-specified τ value in all the cases. In order to find the effect of τ on the performance of the adapted system, a set of experiments have been carried out which will enable us to decide on the value of τ to be used. The results of two sets of experiments carried out for this purpose are shown in Figure 6.6. These results are obtained on a 2 component and a 6 component mixture Gaussian state-clustered triphone systems, built in a slightly different way compared to those used in the triphone evaluations so far [107]. The experiments were performed with several different τ values and number of adaptation sentences, with 4 iterations of MAP estimation in each run updating means, variances and mixture weights of the Gaussians. As usual, the reported results are the average of the results obtained for 12 speakers under test.

These experiments show that the best τ values in these cases are the values between 2 and 10. However, more experiments on different types of systems showed that values closer to 10 perform better in a range of different system architectures. Hence in most of the experiments, a τ value of 10 is used.

6.3.2 Prior Estimation by Moments Approach

As mentioned in Section 4.6, the method of moments can be used as a systematic approach to prior parameter estimation. In this approach, a better estimation of prior parameters may be expected due to the use of a systematic approach which lets the subjective constraints applied in the previous approach be relaxed. This can be carried out by the application of Equations (4.38) to (4.41) and (4.46) to (4.49) to estimate the prior parameters in a system of HMMs using output Gaussian distributions with diagonal covariance matrices. However, for the case of Gaussians with full covariance matrices some constraints are still needed.

Although the moments method should be preferred over the SI-based method, the calculation of the parameters needed can be cumbersome. One possible method for these calculations is to use the parameters from several different speaker dependent systems to calculate these sample moments. This, compared to the previous approach of using SI system models in an *ad hoc* fashion, is somewhat costly since several extra SD systems must be constructed, and extra computation to extract the sample moments from the parameters of these systems should be carried out.

In the application of the moments method for prior parameter estimation, two different approaches can be followed: a strictly systematic approach which uses the sample moments found from SD system parameters in calculating the prior parameters and a somewhat simpler approach which uses the SI parameter values in place of the sample moment estimates, while for variance parameters, the variances are found from sample moments. This can be useful in the cases where due to the limited number of SD systems, non-realistic moment estimates might be expected. However, the results of the application of a full moments approach is presented here.

During the prior parameter calculations using the moments method, in order to prevent the undesired conditions from happening, the following limitations have been applied:

1. Wherever the values of $\mathrm{Var}(r_{ikv})$, $\mathrm{Var}(m_{ikv})$ and $\mathrm{Var}(\omega_{ik})$ in (4.41), (4.46), (4.47) and (4.49) are very close to zero, the parameters ψ_{ik}, τ_{ikv}, α_{ikv} and β_{ikv} are set to a large value.

2. Whenever ψ_{ik} is found to be less than 1.0, it is set to 1.0, which in fact disables the prior effect on mixture weight adaptation according to (4.29).

3. According to (4.18), whenever $2\alpha_{ikv} \leq 1.0$, the adaptation of that variance vector element is cancelled and it is replaced by the corresponding SI parameter value.

Table 6.2 compares the results of MAP estimation with priors estimated using the moments approach and the results of the SI-based approach to prior parameter estimation. As expected, the moments method performs better during speaker adaptation experiments, especially with smaller amounts of adaptation data. Nevertheless, the *ad hoc* method of prior parameter estimation, despite its simplicity in calculation of prior parameters and implementation, performs reasonably well in comparison to the more systematic one and might be the choice in cases where system complications and larger amounts of calculations are to be avoided.

Ad. Sent.s	0	10	40	100	600
SI	5.92	-	-	-	-
MAP	5.92	5.63	4.97	3.64	2.38
MAP-MM	5.92	5.25	4.61	3.52	2.29
SD	-	-	-	-	2.36

Table 6.2: Comparison of the percentage recognition word error rates obtained using MAP estimation with SI-based prior parameter estimates (MAP) and MAP estimation with priors estimated using moments method in an empirical Bayes framework (MAP-MM) for different amounts of adaptation data. Gaussian means, variances and mixture weights were updated.

6.3.3 Prior Improvement Using Speaker Clustering

Speaker clustering, as discussed in Section 4.7, has been used to provide better estimated prior parameters for MAP estimation. The basic approach uses the algorithm described in that section and shown in the block diagram of Figure 4.1.

The data used for the purpose of speaker clustering in these experiments is the training data from SI-109 section of the RM database, which is described in Appendix B. This is believed to provide a better opportunity for having several distinguishable clusters because of the larger number of speakers available. However, the limited amount of training data available from each speaker could be a problem in training the speaker-specific models.

The acoustic models used for this stage of the clustering process are the single Gaussian monophone models which, due to their simplicity and smaller number of parameters, are believed to lead to better training by the application of the limited available training data for each speaker. This can also speed up the speaker-specific model training process. However, once the speaker clusters are determined, any other type of acoustic models may be used for the MAP estimation process. The training procedure, as described before, is carried out by the application of the MAP estimation technique using each speaker's training data together with a common SI model set to be used as both the initial set of models and the one used in prior parameter calculation. The prior parameters are estimated as described in Section 6.3.1, using a τ parameter value of 10, and only the

Gaussian means are trained during this process.

The resultant 109 speaker-specific model sets are used for the purpose of speaker clustering. The clustering algorithm shown in Figure 4.2 is then used to divide the speaker-specific model sets into clusters. The distance threshold T_d, introduced in this algorithm, is set to 0.5 in these experiments. This value was chosen experimentally to give the desired number of clusters. The data from all the speakers found to be in the same cluster are then used to train the intra-cluster model sets of that cluster in a MAP estimation framework training the Gaussian means, variances and mixture weights. The original SI model sets are once again used for prior parameter estimation.

Since the tests are to be carried out on the speakers from the RM SD section, the first few training sentences from each speaker, considered as adaptation sentences, are forced aligned with all the available intra-cluster model sets, to find the intra-cluster model set with the highest average log likelihood per frame for that speaker. In practice, any number of utterances from a speaker can be used for this purpose. During these tests, usually the first 10 sentences are used. However, a comparison between the results obtained using 10 sentences and only 1 sentence revealed that the difference is insignificant. Finally, the appropriate ICM is used in place of the general SI model set for the calculation of the prior parameters, with a τ value of 10 and adaptation of Gaussian means, variances and mixture weights.

Figure 6.7 displays the results of the application of the speaker clustering algorithm to the single Gaussian monophone, single Gaussian triphone and 6 component mixture Gaussian triphone systems. The results of MAP estimation with SI-based priors are also shown for comparison. All the results are obtained with final adaptation of the means, variances and mixture weights of the Gaussians.

As can be seen, the speaker clustering process has led to improvements of about 16%, 8.5% and 4.5% in single Gaussian monophone, single Gaussian triphone and 6 component mixture Gaussian triphone SI error rates respectively. This indicates that the speaker-clustered system is potentially a better system to be used for prior parameter estimation in a MAP estimation process. It should also be expected that with the introduction of further adaptation sentences, the error rate improvement reduces gradually, due to the asymptotic properties of MAP estimation. Hence, a certain amount of improvement in the MAP-based speaker adapted performance with smaller number of adaptation sentences is expected, while with larger amounts of adaptation data this amount is reduced. Although this amount is considerable for monophones and single Gaussian triphones, for mixture Gaussian triphones the error rate improvement is fairly small, which may be due to the larger number of parameters in this case.

For all three sets of results reported here, the same speaker clusters obtained by applying the algorithm discussed before, using a basic single Gaussian monophone system, were used. This algorithm divides the 109 SI speakers into 6 clusters as shown in Table 6.3. As can be seen, all of the female speakers, except two, are allocated to the same cluster, but the male speakers are divided between four different clusters, which means that the clustering algorithm is working correctly at least in separating females and males. How-

(a)

(b)

(c)

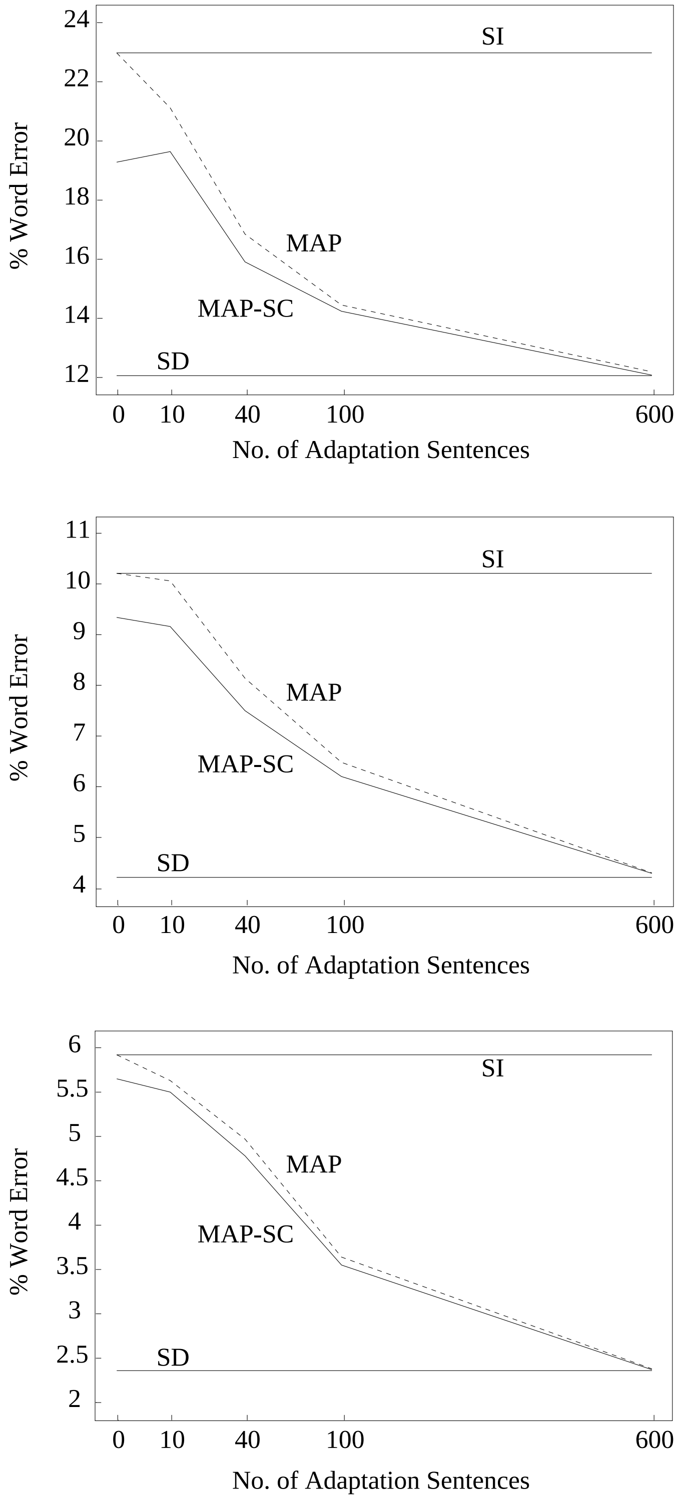


Figure 6.7: The results of the application of speaker clustering to prior parameter estimation in MAP estimation-based speaker adaptation: a) Single Gaussian monophone system, b) Single Gaussian triphone system and c) 6 component mixture Gaussian triphone system. The standard MAP results are also given for the purposes of comparison. The speaker cluster-based MAP results are marked MAP-SC and as usual, all results are averaged over all 12 available speakers.

Cluster	1	2	3	4	5	6
No. of male speakers	-	27	21	25	5	-
No. of female speakers	29	-	-	1	-	1

Table 6.3: The number of SI speakers determined by the clustering algorithm to be used to build clusters in the speaker clustering procedure.

ever, two of the six clusters have very few speakers in them which means that they may have not been trained properly. Nevertheless, as Table 6.4 shows, during the allocation of new speakers to the clusters, none of the 12 speakers were allocated to these 2 clusters. However, such clusters could be omitted from the set of clusters or merged with closer clusters in order to prevent them from being used by new speakers.

Speaker	bef (m)	cmr (f)	das (f)	dms (f)	dtb (m)	dtd (f)	ers (m)	hxs (f)	jws (m)	pgh (m)	rkm (m)	tab (m)
Cluster	3	1	1	1	3	1	4	1	2	3	3	2

Table 6.4: The allocation of new speakers to clusters in the above experiments. The speakers' gender is also shown with 'm' for male and 'f' for female speakers.

As shown in Table 6.4, all the 5 female speakers used for adaptation purposes used cluster 1, i.e. the females' cluster, as their cluster of choice, while for male speakers the other three clusters, i.e. 2, 3 and 4, were used.

6.3.3.1 Gender Dependent Based Clustering (MAP-MFSC)

The above approach to speaker clustering has also been used together with gender dependent models. Having the above clustering results in mind, it is believed that the application of a speaker clustering algorithm to gender dependent models can be more efficient. Here, the male and female models are already separated and in place of applying a certain threshold to the whole set, which could result, like in the previous approach, in having a large female cluster and several male clusters, male and female SI speakers could be clustered separately resulting in better clusters for each sex.

Referring to Figure 4.1, this approach has been carried out by building speaker-specific model sets using separate initial SI models for males and females and clustering the male and female speakers separately. In this approach, once again the problem of having clusters with a very small number of speakers is faced, so these are removed from the list of clusters manually. This leaves a total of 3 male and 2 female clusters. The rest of the approach is continued as before, but separately, for male and female speakers. A comparison between the improvements in the MAP estimation results obtained for single Gaussian monophone

Adaptation Sentences		0	10	40	100	600
1 Gaussian monophone	MAP-SC	16.1	7.0	5.6	1.5	0.9
	MAP-MFSC	17.8	9.2	5.5	2.5	2.0
6 Gaussian triphone	MAP-SC	5.2	5.1	4.5	3.4	2.3
	MAP-MFSC	12.0	9.1	9.5	7.1	2.5

Table 6.5: Comparison of the percentage improvements over standard MAP estimation obtained by speaker clustering (MAP-SC) and male/female based speaker clustering (MAP-MFSC). The baseline SI recognition word error rates were 22.98% for the single Gaussian monophone system and 5.92% for the 6 component mixture Gaussian triphone system.

and 6 component mixture triphone systems in this approach with those of the previous approach is given in Table 6.5.

The results of MAP-MFSC outperform those obtained with MAP-SC. The main improvement is once again obtained by the initial SI models which prove to be better sets of models for use in prior parameter estimation. Thus, by using these better models as priors, better results may also be expected from the MAP estimated models. However, with larger numbers of adaptation sentences, once again, due to asymptotic properties of MAP estimation, the improvements obtained become marginal.

6.3.3.2 Speaker Phone Clustering

Speaker phone clustering, as discussed in Section 4.7, is carried out as an extension to the speaker clustering procedure, to provide still better models for prior parameter calculations.

Vowels	aa ae ah ao aw ax ay eh en er ey ih ix iy ow oy uh uw
Stops	b d dd dh dx g k kd p pd t td th ts
Nasals	m n ng
Fricatives	ch f jh s sh v z
Glides	hh l r w y

Table 6.6: The five broad phonetic groups used in speaker phone clustering.

This is done by dividing the phone models in the speaker-specific model sets into 5 broad phonetic groups. These phonetic groups are introduced in Table 6.6. Then the clustering is carried out within these phonetic groups, i.e. the clustering algorithm only uses the models belonging to the same group for finding the speaker clusters within that phone group. Also different thresholds could be used for clustering within different groups.

The thresholds, in practice, are chosen so that the desired number of clusters are obtained within each phonetic group. However, this has proved to be rather difficult, especially with some clusters remaining with very few speakers. Once again, these clusters are omitted from the set of generated clusters to provide us with a number of clusters per phone group with enough speakers. Table 6.7 shows the thresholds used and the specifications of the clusters obtained in these experiments. Between 1 to 3 clusters with very small number of speakers are omitted from some of these phone groups.

Phone group	vowels		stops		nasals		fricatives		glides	
Thresh. used	0.59		0.48		0.68		0.49		0.60	
	m	f	m	f	m	f	m	f	m	f
Cluster 1	-	30	-	30	-	23	-	30	-	30
Cluster 2	52	-	76	1	56	-	13	-	48	-
Cluster 3	26	1	-	-	14	-	63	1	17	-

Table 6.7: Specification of the clusters generated for each of the 5 broad phonetic groups.

The results of the application of speaker phone clustering-based MAP estimation to a set of single Gaussian monophones are shown in Table 6.8. The results obtained show a marginal improvement over the previous speaker clustering approach in most of the cases, while the improvements are less than those obtained using gender dependent based clustering. The major problem here is the complicated clustering procedure and the difficulty of setting proper distance thresholds. For these reasons, this approach has not been applied to the case of mixture Gaussian triphones, since a better, or at least similar result, might be expected from the simpler technique of gender dependent based clustering.

Adaptation Sentences	0	10	40	100	600
Word Error	19.3	19.4	15.9	14.3	11.9
% Improvement	16.0	8.2	5.8	0.9	2.4

Table 6.8: Word error rate and the percentage improvements over standard MAP estimation obtained by speaker phone clustering on a single Gaussian monophone system.

6.3.4 Unsupervised Bayesian Adaptation

Two scenarios for the unsupervised application of Bayesian adaptation, where the transcriptions of the utterances are not available, are introduced here. These are batch and incremental unsupervised adaptation.

6.3.4.1 Unsupervised Batch Adaptation

For unsupervised batch adaptation, usually a high performance SI system is used to transcribe the adaptation utterances. These transcriptions are used later, together with the utterances, to adapt the system. Hence the improvement of the system performance is dependent to the performance of the SI system in two ways. Firstly, by using an SI system to estimate the prior parameters for the MAP estimation and secondly due to its use in finding the training labels. Thus, the application of a high performance SI system is vital for the success of this approach.

The adaptation results obtained by the application of unsupervised Bayesian adaptation are shown in Figure 6.8 and compared with the performance of supervised MAP adaptation. Note that the SI system used for this purpose has an average word error rate of 5.92% on the test speakers. Hence, the results obtained using the unsupervised method are inferior to those obtained using the supervised method. However, this experiment shows that MAP estimation can also be used for unsupervised speaker adaptation purpose, by only partially sacrificing the expected word accuracy, if a high performance baseline SI system is already available.

6.3.4.2 Unsupervised Incremental Adaptation

Unsupervised adaptation is also used in an incremental fashion in different approaches, where recognition labels generated by the system for the speaker utterances are used for adapting the system to the new speaker, hence improving the system performance during the course of recognition of a test speaker utterances. This can be carried out in several consecutive stages, where at the end of each stage, the resultant recognition labels are used to further adapt the system.

The implementation of MAP incremental adaptation consisted of the use of transcriptions obtained from the recognition of 10 speaker test sentences (out of a total of 100) at each adaptation pass. Figure 6.9 illustrates the improvements obtained, relative to the SI system performance, at each step of the incremental adaptation procedure, based on the recognition results of the whole test data set encountered so far. The improvement obtained at the second pass indicates the large contribution of the first 10 sentences to the adapted system improvement. The changes in the percentage improvement are related to many factors such as the nature of the last set of adaptation sentences, the contribution of previous passes to system improvement and the amount of similar data available in different sentences. However, since this is a cumulative index, the fluctuations tend to decrease as the number of passes increase. Note that once again, all the results reported

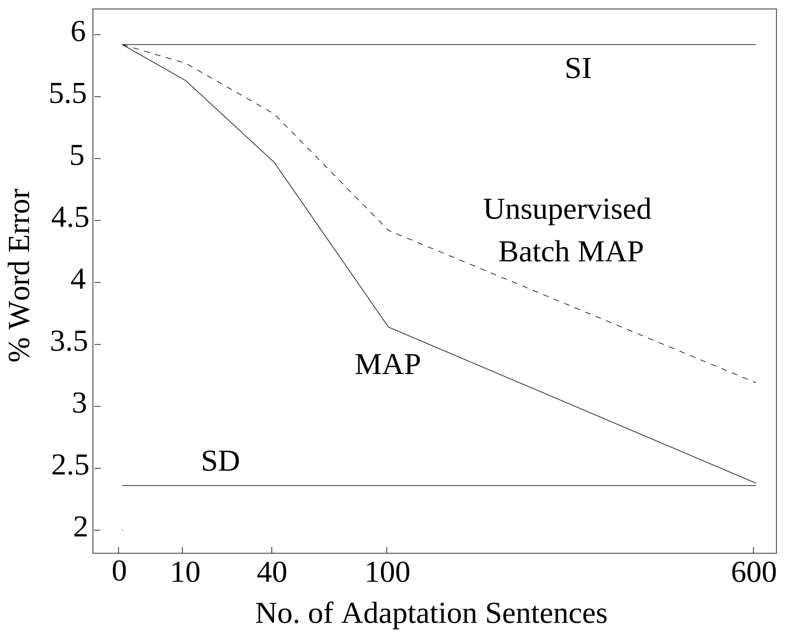


Figure 6.8: A comparison of the averaged results obtained using unsupervised batch MAP estimation and supervised MAP estimation on the 6 component triphone system. Note that the adaptation was applied to Gaussian means, variances and mixture weights.

are averaged results over all 12 SD speakers.

Table 6.9 includes two figures indicating cumulative system performance and adapted system performance in the above approach compared to that of the SI system. The cumulative performance is the same as the number indicated in the stage 10 of Figure 6.9 and is the total recognition result found by adding up the recognition results of all the passes of algorithm. The adapted system performance is found by running another complete pass of recognition on the whole test data available using the adapted system of the final stage of the algorithm. As can be seen, an error rate improvement of nearly 21% can be obtained using unsupervised incremental MAP adaptation, which is equal to adapting the system with about 40 to 60 adaptation sentences, while in fact no explicit adaptation sentences are needed.

The application of such an approach, in conjunction with standard supervised Bayesian adaptation is also possible. In this case, an initial performance improvement is obtained by adapting the system to the new speaker using the available transcribed adaptation sentences. During recognition, using a higher performance initial system, incremental unsupervised adaptation can be carried out in the same manner explained earlier, using the recognition utterances. However, such an approach has not been implemented in these experiments.

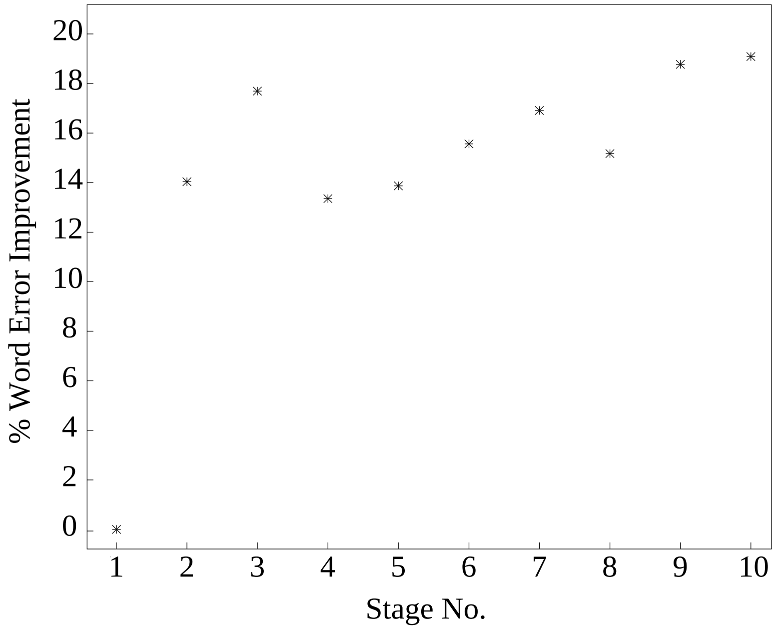


Figure 6.9: The total improvement obtained by unsupervised incremental MAP adaptation at the end of each stage, relative to SI performance.

	Error Rate	% Improvement Rel. to SI
SI	5.92	-
Cumulative Unsup. MAP	4.79	19.1
Adapted Unsup. MAP	4.69	20.8

Table 6.9: The performance improvements obtained by unsupervised MAP adaptation.

6.3.5 Summary and Discussion

Several applications of the MAP estimation approach to the problem of speaker adaptation were explained during this section. The effectiveness of this approach has been demonstrated by its application to CDHMM-based speech recognition systems. An important point regarding MAP estimation is that at the start and with even very small amounts of adaptation data the system performance does not deteriorate much, compared to SI system performance, while this could happen with many other speaker adaptation approaches. As an example, such deterioration is reported in [66], although this can be prevented by disabling adaptation for very small amounts of adaptation data.

Another interesting specification of the MAP estimation approach, which is again not available in many other adaptation approaches (as an example see [73]), is its ability to converge to SD performance asymptotically. The above two characteristics of MAP estimation make it an ideal candidate for many applications, especially due to its similarity of implementation to the widely used Baum-Welch re-estimation algorithm.

In this section, it has also been shown that although using SI models in an *ad hoc* manner is a simple method of prior parameter estimation, the use of other more accurate techniques may lead to larger improvements. The method of moments can lead to about 11% reduction in error rate, compared to just 5% for the SI-based approach, with 10 adaptation sentences. However, this is obtained with a large overhead in computation and storage, due to the need for a number of speaker dependent systems in order to perform the required calculations.

Speaker clustering can also help by providing better models for several groups of speakers to be used for prior parameter calculation in MAP estimation. This approach in fact provides better baseline systems which in turn will help in improving MAP results. However, such improvements are not achieved without computation and storage overheads, but these overheads may well pay off with improved system performance.

The effectiveness of the MAP estimation approach is also shown in the case of unsupervised batch and incremental adaptation, where particularly in the case of incremental adaptation, MAP can prove to be very useful, compared to similar model parameter estimation techniques, both in computation efficiency and performance.

6.4 Evaluation of Regression-Based Model Prediction using RM Data

As pointed out in Section 5.11, the RMP method is designed to improve the performance of an HMM system already adapted to a speaker, using a Bayesian or another similar model parameter adaptation technique, especially in extremely sparse data conditions. As noticed in the previous section, the asymptotic convergence of the performance of MAP estimation, together with its ability to provide improvements starting from small amounts of training data, makes it a desirable technique for speaker adaptation. However, RMP is designed to further improve the performance of such MAP estimated systems.

As an example, the percentage of system parameters receiving any amount of adaptation in a MAP estimated 6 component mixture Gaussian tied-state triphone system is presented in Table 6.10. This shows that a large number of model parameters do not receive any adaptation with small numbers of adaptation sentences. Furthermore, even among the small number of adapted parameters in such cases, the largest amount of adaptation a certain parameter receives is far less, compared to the average amount of adaptation for the cases of larger amounts of training data such as 100 or 600 adaptation utterances[2]. Thus, in the case of small number of adaptation utterances, only a very small percentage of the system parameters receive the minimum amount of adaptation data needed for

[2]Here, the amount of adaptation a parameter receives is measured by the occupation count for the mixture component to which the parameter belongs.

improvement. This points to the major problem which prevents MAP estimation from achieving larger improvements with a small number of adaptation sentences.

Adaptation Sentences	1	10	40	100	600
Percent Adapted Means	5.6	31.8	64.3	82.4	97.0

Table 6.10: Percentage of the system mean parameters receiving any amount of adaptation with different number of adaptation sentences. The reported data correspond to 6 component mixture Gaussian triphones.

Two different implementations of RMP, with respect to the system structure, have been carried out: one on the monophone system and the other one on state-clustered triphones. These two approaches will be discussed in the next sections.

6.4.1 Monophone system

A very simple single Gaussian monophone system has been used for this purpose. Since this system does not use any type of tying among system parameters, the application is straightforward, because the parameter relationships can be found among untied model parameters rather than tied ones.

In this approach, RMP is applied with a correlation coefficient threshold of 0.4, a maximum regression order of 2 and different higher and lower thresholds for different numbers of sentences, which, on average, leaves about 12 Gaussian components, out of the total of 141, as source, and about 115 as targets which are mostly adapted. The results of these experiments, together with the MAP results are averaged over all 12 speakers available and reported in Table 6.11.

Adaptation Sentences	0	1	10	40	100	600
SI	22.98	-	-	-	-	-
MAP	22.98	22.05	19.49	16.31	15.52	14.05
RMP	22.98	21.45	17.70	15.91	15.35	14.05
SD	-	-	-	-	-	14.04

Table 6.11: The comparison of performance of the baseline single Gaussian monophone system with MAP and RMP adapted and SD systems.

Due to the simplicity of the monophone system the baseline error rate is relatively high. The results obtained show that although some performance improvement is obtained by the application of RMP technique, which is more significant for a smaller number of adaptation sentences, the overall performance improvement is not as much as hoped for. This could be mainly due to the relatively small number of parameters in this system.

Hence, totally unadapted parameters is less of a problem. The improvement obtained by RMP, compared to MAP performance, for the case of context-dependent models might be more significant since in that case, the similarities between the models of the same phone in different contexts, which is not used by MAP estimation, could lead to large improvements due to RMP. Thus, in the next step, the application of the RMP approach to context-dependent models is considered.

6.4.2 State-Clustered Triphone system

The RMP adaptation algorithm has been applied to several context-dependent systems to assess its performance in several different conditions. These include single, 2 and 6 component mixture Gaussian tree-based state-clustered triphone systems.

The difference between the applications of RMP to monophone and triphone systems is that unlike in the case of monophones, for the case of triphones, due to the tying of the HMM states, the prediction algorithm is applied to the tied parameters in place of the physical HMM parameters.

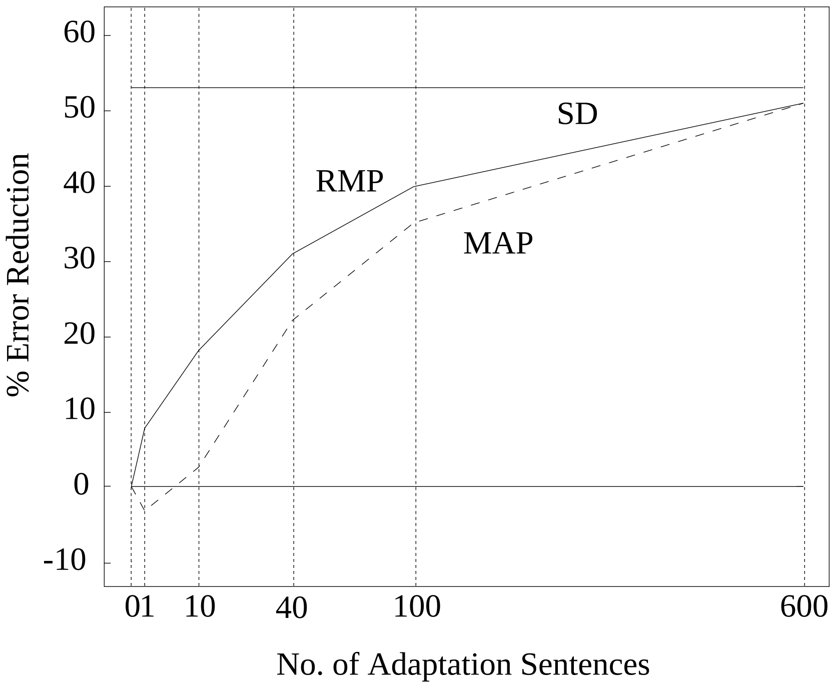
For these experiments, once again, a correlation coefficient threshold of 0.4 and a maximum regression order of 2 is used. The 12 SD speakers available are used in a "leave one out" manner, so that in any one pass, one speaker could be used for test purposes and the remaining 11 for regression parameter calculations. This procedure which is utilised due to the limited number of speakers available which should be used for both regression parameter calculations and adaptation and test purposes, has been repeated for all 12 speakers so that an average of the results could be obtained. The reasons for using above regression order and correlation threshold will be given in Sections 6.4.3 and 6.4.4.

The results of the application of RMP to single and 6 component mixture Gaussian triphone systems are shown in Figure 6.10, together with the corresponding MAP, SI and SD results. Note that all MAP and SD results are obtained by running a single iteration of either Forward-Backward MAP estimation (for MAP) or Baum-Welch algorithm (for ML), updating only the Gaussian mean parameters.

For MAP estimation purposes, for each system, the prior parameters are estimated using the corresponding baseline SI system parameters in the same manner introduced in Section 6.3.1, using a value of $\tau_{ik} = 10$ in all the experiments. The thresholds, T_H and T_L, used in RMP evaluations are set to different values, depending on the number of adaptation sentences and the type of system in use, which have a direct influence on the mixture occupation values obtained during MAP estimation. Usually, the thresholds were set so that the RMP uses only about 5-15% of the distributions as source and about 75-90% of them as targets. Further details of the implementation are given in [1].

The results reported in Figure 6.10 indicate that a worthwhile improvement can be obtained with the application of RMP to context-dependent models. It is shown that in the case of only one adaptation utterance, where no improvement is expected from MAP estimation, the application of RMP results in a worthwhile improvement in performance. This is believed to be due to the use of parameters with larger amounts of adaptation

(a)



(b)

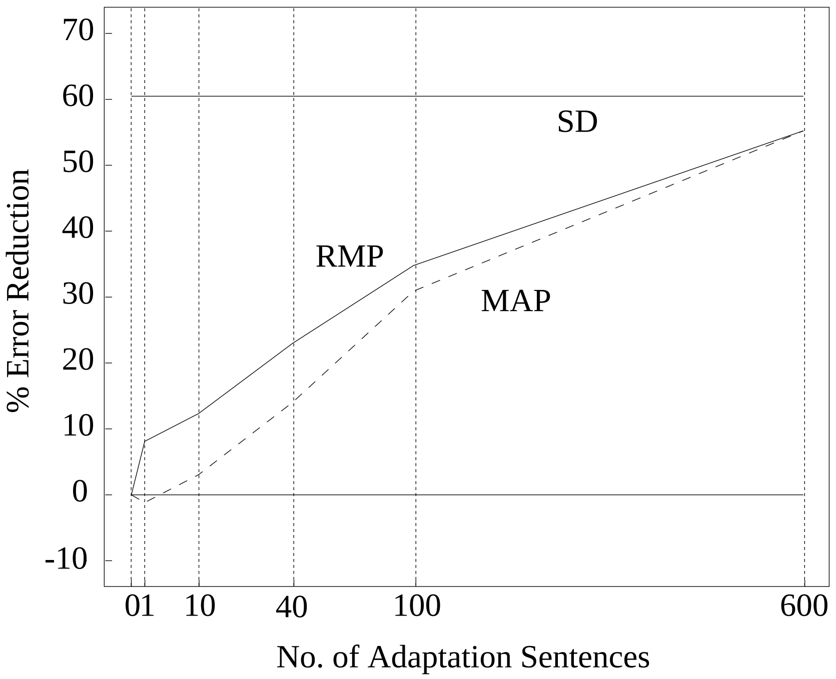


Figure 6.10: A comparison between percentage error rate reductions, relative to SI, obtained by the application of RMP and MAP methods for speaker adaptation of (a) single Gaussian and (b) 6 component mixture Gaussian triphone systems.

data to update other system parameters. Moreover, the RMP performance always remains superior to that of MAP estimation when more adaptation sentences become available, although, the amount of improvement, relative to that of MAP, reduces gradually. This could be expected, since with more adaptation data, more parameters are either trained or better trained, reducing the number of untrained or poorly trained parameters.

This leads to a performance equal to MAP estimation with very large number of adaptation sentences, which can be seen from Equations (5.9) and (5.13), since in this case s_ζ^2 would be zero. In fact, RMP displays two desirable characteristics: the asymptotic performance of the MAP estimation with a large number of adaptation sentences and very fast speaker adaptation performance. It also continuously outperforms MAP estimation with different numbers of adaptation utterances.

Another measure of improvement is to find out how close the RMP adapted Gaussians are to the SD Gaussians of the same speaker. A Mahalanobis distance measure can be used in this case, since the Gaussians share the same variances. A comparison between the distances of the Gaussian distributions of the SI system, the RMP adapted system and the MAP estimated system of the single Gaussian triphone type, with those of the corresponding SD system are shown in Table 6.12. The RMP adapted Gaussians are apparently closer to SD ones, compared to the MAP ones, for both 1 and 10 adaptation utterances. Also, comparing the individual Gaussians, about 74% of them in the case of 1 adaptation sentence and 67% of them in the case of 10 adaptation sentences are closer to their corresponding SD counterparts in RMP adapted models than in MAP ones, while only about 19% and 20% of them are farther in RMP adapted models, respectively. This shows that the number of Gaussians adapted by RMP in the right direction is about 4 times those adapted in the wrong direction for 1 adaptation sentence and more than 3 times for 10 adaptation sentences[3].

No. of Ad. Sentences	SI	MAP	RMP
0	0.296	0.296	0.296
1	-	0.313	0.277
10	-	0.270	0.225

Table 6.12: The average Mahalanobis distance between SD Gaussians and SI, MAP and RMP adapted ones using 1 and 10 adaptation utterances. The values are averaged over all SD speakers.

Figure 6.11 demonstrates the improvements in the 6 component state-clustered triphone system performance for individual speakers achieved by the application of MAP and RMP techniques to 12 SD speakers of the RM database with 1 and 10 adaptation sentences. This figure shows that with RMP, compared to SI system, most of the speakers

[3]By right and wrong directions we mean getting closer to, or farther from SD Gaussian means. However, since our SD models are not ideal SD models it is not clear that these models are always moving towards or away from the true SD parameters.

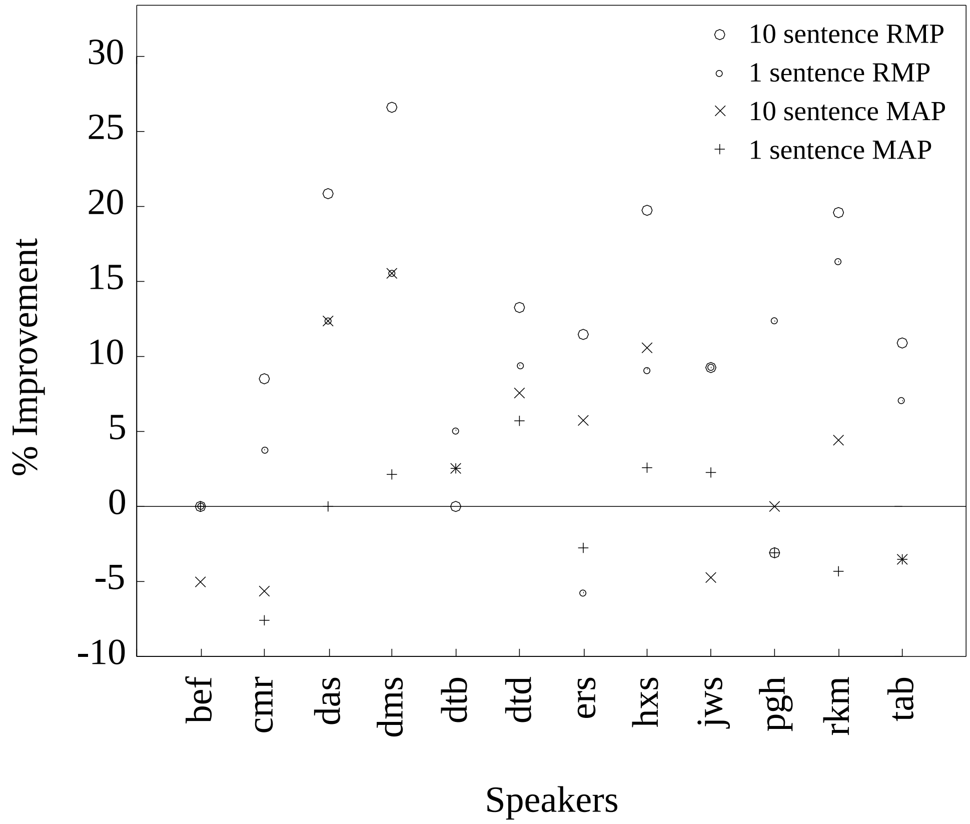


Figure 6.11: Percentage change in the performance of speaker adapted system for individual speakers, relative to the SI system performance, using the 6 component state clustered triphone system.

achieve better recognition results even with 1 adaptation sentence, while this is not true for MAP estimation. Also, with 10 adaptation sentences, although some speakers' MAP adapted models perform better, RMP results are still much better for most of the speakers. This emphasises the robustness of the RMP adapted systems with a very small number of adaptation sentences.

6.4.3 Setting Adaptation Thresholds and Regression order

A practical issue in the implementation of the RMP technique is the setting of adaptation thresholds. The two thresholds T_H and T_L described in Chapter 5, are defined to separate the better trained distributions, which can be used as source distributions, from those which have not received much adaptation data but would benefit from adaptation and those which have received a reasonable amount of data and should remain untouched. Since there is no way of exactly calculating optimal threshold points, these are to be set experimentally.

However, it has been found that the adaptation process is not very sensitive to these thresholds within a certain range. A few reasons can be counted for this such as

- Due to the usually large number of target parameters, a small change in their number, i.e. changing T_L, should not have a large effect on the adaptation results, especially since this change does not affect the number of untrained or very poorly trained parameters.
- The best trained distributions, which are believed to have the largest effect on performance improvement, are still used as source distributions, even if T_H is changed slightly.
- Due to the use of multiple regression, most of the target parameters depend on more than one source distribution for adaptation. Hence, the effect of change in one of the source distributions, either from a more correlated one to a less correlated one or vice versa, for only a fraction of target distributions, is believed not to have too much effect on the success of the adaptation process.

Thus, the setting of the adaptation thresholds can be carried out more freely, which makes RMP implementation somewhat easier. However, it is obvious that a different number of adaptation sentences, and also different HMM systems, require different thresholds. A rule of thumb could be applied to threshold setting by allowing a certain number of source and target distributions in each case.

The regression order setting is another important factor in RMP performance. While as discussed in Section 5.6, a larger regression order would be beneficial, statistically it is not recommended to increase the regression order much unless the number of regression points are quite high [11]. For these reasons, it has been found during these experiments that a regression order of 2 performs best with the available number of speakers. Tests performed with orders of 1 and 3 showed inferior performance in comparison with an order of 2. Hence in all experiments reported here, this regression order is used.

6.4.4 Effect of Correlation Coefficient Threshold on Adaptation

As stated in section 5.4.4, a correlation coefficient threshold (T_c) is set during the regression parameter calculations to be used as a lower limit in finding a proper source for each target. This parameter is one of the factors which have important effects on RMP adaptation results. The main role of this parameter is to prevent distributions with low correlations being used as source-target pairs. Hence setting it to a very low value can lead to performance degradations since less correlated distributions might be used as source distributions for the prediction of target distribution parameters. On the other hand, a high T_c, in spite of allowing only very well correlated distributions to be used as source-target pairs, can limit the number of adapted parameters which in turn can reduce the amount of improvement expected from the technique. Thus, a careful selection of this parameter is of great importance in this approach.

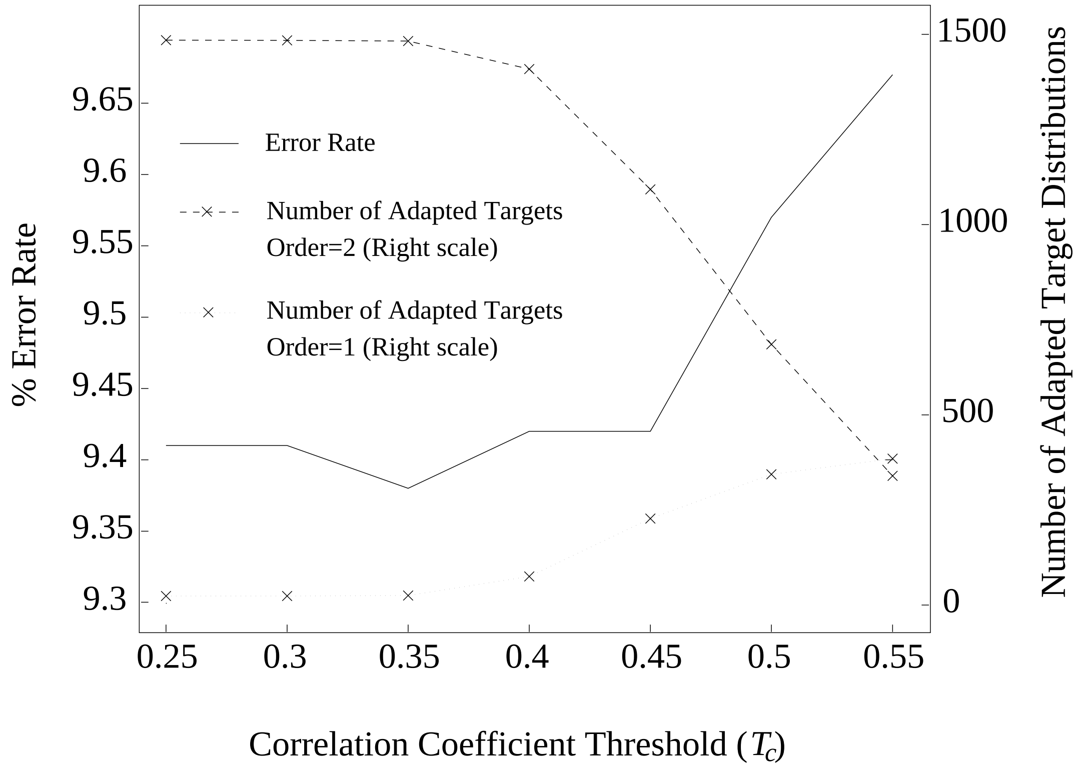


Figure 6.12: Effect of change in the correlation coefficient threshold (T_c) on the word error rate of a single Gaussian triphone system, adapted with one sentence using the RMP method. The change in the number of adapted target distributions with orders of 1 and 2 are also displayed.

A series of experiments were carried out to inspect the effect of this parameter on the result of adaptation. Figure 6.12 illustrates the effect of the change of the correlation coefficient threshold on the adaptation result, on a single Gaussian triphone system adapted with 1 sentence and with a maximum regression order of 2. The results reported are averaged over all available speakers. As can be seen, the best system performance is obtained with correlation coefficients smaller than 0.45. The reason is that further increase in the value of T_c will cause the problem discussed above, i.e. reduction in the number of adapted parameters. However, as can be noticed, further decrease in its value does not have an important effect on the adaptation process. This is mainly because by reducing the parameter's value to less than 0.4, two proper source parameters have been found for almost all the available target parameters, which means that sources with lower correlations are not of interest anymore. Hence, for lower correlation coefficient thresholds, the system performance remains almost constant.

The question of the reason for the existence of a correlation coefficient threshold arises at this point. Although the effectiveness of this parameter is not emphasised by this example, it is obvious that if higher orders of regression on systems with larger number of parameters are used, this parameter will play a more important role during the adaptation process. This is especially true when there are very few adaptation sentences.

6.4.5 Reducing Adaptation Computation

The RMP technique, as discussed above, is computationally expensive. The main drawback for this approach is the need to calculate the regression parameters for every new speaker. This, regarding the size of the system used, the experimental parameters set and the number of SD models used can be very costly. The need for the calculation of the regression parameters for every new speaker has arisen due to the use of the speaker's mixture occupation counts for deciding on the allocation of the source and target distributions. In fact, for any new speaker, in a supervised adaptation scheme, given the number of adaptation sentences to be used, the calculation involves the use of the same basic SD and MAP model sets for SD speakers. The only difference would be the difference in the mixture occupation counts of Gaussian components for different speakers.

A somewhat simpler approach would be to use a general set of mixture occupation counts for all the speakers. This could be found, say, by averaging the mixture occupation counts from several speakers. Then, for a given number of adaptation utterances, a global regression parameter calculation can be carried out using all the SD models available, plus all the corresponding MAP estimated models and the global mixture occupation counts. In this case, the models of the speaker under test are not included in the set of SD models. These globally calculated parameters can then easily be used to adapt the models of any new speaker. Note that, once the regression parameters are available, the updating of model parameters for a new speaker is a simple and fast task which does not involve much computation.

Adaptation Sentences		0	1	10	40	100	600
Single Gaussian Triphones	RMP	10.21	9.42	8.37	7.06	6.15	5.02
	RMP-AMO	10.21	9.42	8.31	7.13	6.04	5.02
6 Component Triphones	RMP	5.92	5.44	5.19	4.56	3.86	2.34
	RMP-AMO	5.92	5.60	5.31	4.62	3.98	2.34

Table 6.13: Comparison of the results of RMP and RMP-AMO for single Gaussian and 6 component mixture triphone systems.

Table 6.13 includes a comparison between the results obtained on the single component and 6 component mixture triphone systems, using the standard RMP method and this method, identified as *RMP with Averaged Mixture Occupations* (RMP-AMO). While for single Gaussian models, the results are almost the same as before, the amount of improvement obtained on the 6 component mixture Gaussian system is slightly smaller. While the total state occupation counts might be comparable for different speakers and hence the use of the averaged values does not change the adaptation process significantly in a single Gaussian system, in a mixture Gaussian one the state occupation counts might be spread differently among mixture components for different speakers. In other words, in a mixture Gaussian system, speaker differences might lead to different mixture occupations within similar states, making the use of an averaged set of mixture occupations more approximate.

However, on such systems, a reduction of more than 90% in the adaptation computation time results from the application of RMP-AMO in comparison to RMP, if the adaptation is to be carried out on 10 new speakers. This is quite a worthwhile saving in computation time, when such reductions in the adapted system performance is acceptable.

6.4.6 Iterative RMP Adaptation

Due to the use of MAP estimation in the RMP method, an approach based on the ideas behind those explained in 6.3.1.3 and 6.3.4.2 can also be implemented here. The basic idea is again the use of adapted parameters to provide new prior parameters for another step of MAP estimation. However, in this case the adapted parameters are extracted from the already adapted RMP systems. The whole idea is shown in Figure 6.13.

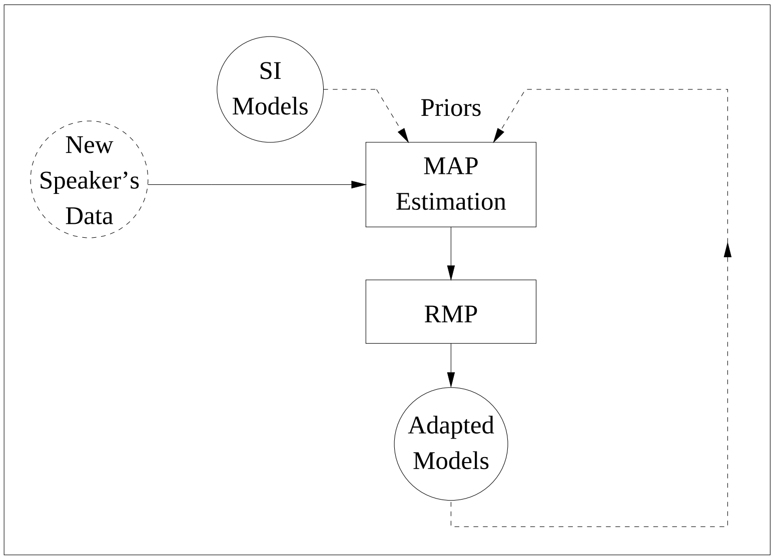


Figure 6.13: The basic block diagram for iterative RMP adaptation.

Here, initially a set of adaptation utterances from the new speaker are used to MAP estimate the system parameters using the SI model parameters to estimate priors. Then RMP is applied to further adapt the parameters. The adapted model parameters can be used as improved system parameters to estimate the prior parameters needed for MAP estimation of the system parameters after receiving new adaptation data from the speaker. This can be done in several iterations, depending on the amount of data used in every iteration.

In our experiments, this has been applied to a single Gaussian triphone system in 4 stages. First the usual MAP estimation and RMP method are applied with 10 adaptation sentences. Then the resultant system is used to estimate prior parameters for a MAP estimation with a further 30 sentences. RMP is run once more and again the resultant system parameters are used for prior estimation of another MAP adaptation with a further 60 sentences, increasing the total number of adaptation sentences to 100. This procedure is continued with a further 500 sentences, which increases the total number of adaptation sentences to 600. However the final pass of RMP, is not necessary, since as discussed earlier, it cannot further improve the MAP estimated performance.

Total Ad. Sentences		10	40	100	600
Standard Approach	MAP	9.95	7.96	6.64	5.02
	RMP	8.37	7.06	6.15	5.02
Iterative Approach	MAP	9.95	7.28	6.10	4.94
	RMP	8.37	6.84	5.90	4.94

Table 6.14: Comparison of the results of standard and iterative RMP adaptation.

The results obtained from the application of the above procedure to the single Gaussian triphone system are shown in Table 6.14. In comparison to standard MAP and RMP, the current approach has resulted in a noticeable improvement in the results using 40, 100 and 600 sentences. However, the relative improvement of RMP over MAP is somewhat smaller. The reason for this may be that some of the model relationships have already been used in previous steps and might not help in further steps. These experiments show that an initial RMP can provide MAP estimation with better priors which could boost its performance for further adaptation, if the adaptation is to be carried out in several stages using MAP estimation.

6.4.7 RMP Adaptation with Moment-Based MAP

It was shown in Section 6.3.2 that the application of the empirical Bayes method to prior estimation can lead to some improvements in the overall performance of the adapted system. Since in RMP, the MAP estimated system is used as the basic system, it is believed that having better priors, which could lead to better adaptation of the models for which adaptation data is provided, can also lead to better estimation of other model parameters which is performed during the application of RMP, due to the existence of better source models.

This has been the motivation for applying the RMP method to the MAP estimated models with moment estimated priors. The results of these experiments, together with the previous results of MAP and RMP on our 6 component mixture Gaussian triphone system are shown in Figure 6.14.

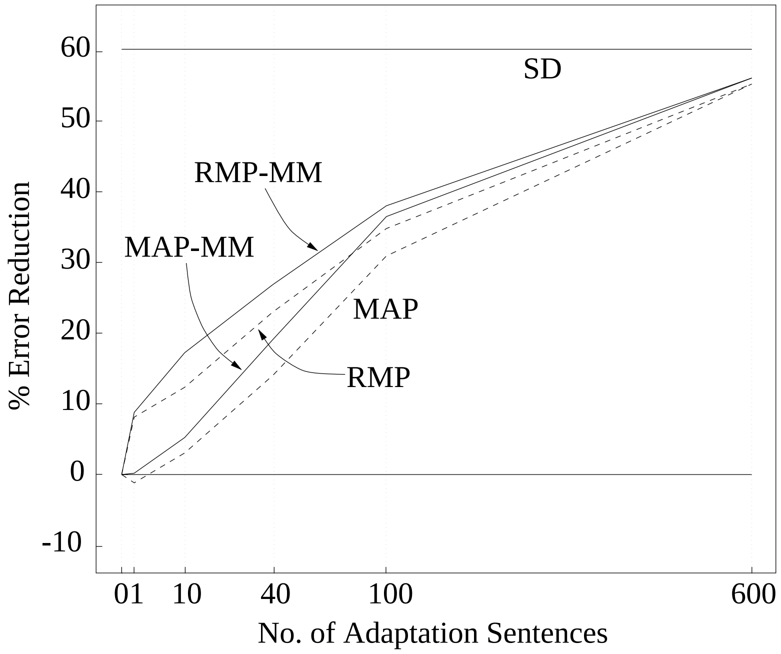


Figure 6.14: A comparison of the improvements obtained over SI system performance using the moments method for estimating the prior parameters of MAP estimation and corresponding RMP adaptation (marked MAP_MM and RMP_MM) with previous MAP and RMP results on a 6 component mixture Gaussian triphone system.

The results of MAP estimation, as expected, are improved in comparison with the previous approach and accordingly, almost the same effect can be observed on the RMP results. Note that the improvement in the MAP result for one adaptation sentence is insignificant and hence, the RMP result is not changed much too. However, for larger number of adaptation sentences, better MAP models have helped to improve RMP results. For very large numbers of sentences, the difference between MAP and RMP results is reduced. This could be due to the fact that in the recent case, because of using exactly the same adaptation thresholds, some better trained targets have also been adapted which can adversely affect the adaptation process, especially due to the existence of presumably better prior parameters for these targets, which should have led to better estimation of their parameters.

It should also be noted that in comparison to the results reported in Section 6.3.2, these results are obtained updating only the means during MAP estimation.

6.4.8 Adaptation of other HMM parameters

The issue of adaptation of other HMM parameters in the RMP framework was also addressed in the experimental evaluations. As pointed out in Section 5.9, the adaptation of variance parameters has not been addressed in this work, while the problem of mixture weight adaptation has been tackled.

The adaptation of mixture weights has been carried out according to Equations (5.26) and (5.27), alongside the adaptation of Gaussian means, for different numbers of adaptation sentences on our 6 component mixture Gaussian triphone system. However, the results obtained did not show any improvement over the adaptation of means alone. The reasons for this can be counted as:

1. The adaptation of mixture weights, when carried out besides mean adaptation, according to our previous experiences with MAP estimation, does not provide a significant contribution to the overall performance improvement.

2. A criticism might be applicable to the target component occupation count estimation in Equation (5.26). In this equation, although one might justify the use of source component occupation counts for the estimation of the target occupation counts, the use of the relationships between corresponding mean parameters in a linear regression framework for this estimation has yet to be justified. Hence, Equation (5.26) should only have been considered as a (perhaps improper) suggestion for this purpose and the derivation of a more appropriate equation is left for further work in the field.

6.4.9 Comparison with Other Approaches

The most important characteristic of the RMP approach is its ability to perform very fast speaker adaptation, i.e. improving system performance with as little as 1 adaptation sentence or about 3.7 seconds of speech data. Although there have been some reported works on fast speaker adaptation, e.g. [56, 115], reports of the improvements with very small amounts of adaptation data on similar continuous speech recognition systems are quite rare.

As an example of a successful speaker adaptation procedure, Leggetter and Woodland [65] have reported that using a regression-based transformation approach to speaker adaptation, called MLLR, at least 3 adaptation sentences are needed in similar conditions before any improvement in system performance could be obtained. However, using very similar model sets and the same adaptation and test data, better system performance is obtained with medium amounts of adaptation data, i.e. the largest difference is at around 40 adaptation sentences where a 38.2% improvement over baseline SI system is obtained by MLLR, where RMP only achieves a 23.0% improvement, while with 100 sentences, only a difference of 4.9% exists between the two techniques. Once again with larger amounts of adaptation data, RMP, inheriting the desirable asymptotic performance of MAP estimation, outperforms MLLR, or similar transform-based adaptation schemes by performing close to an SD system.

6.4.10 Summary and Discussion

In this section the experiments using the RMP method for speaker adaptation were introduced. This adaptation algorithm is designed to overcome the problem of not adapting the unseen HMM parameters encountered with MAP estimation. It has been shown that using linear regression, RMP has been able to predict some model parameters which have not received any or enough adaptation during the MAP estimation process. The improvements, especially in the case of very small adaptation sentences, have been worthwhile.

RMP has also improved the results of MAP estimation for a range of adaptation sentences. For a large number of sentences RMP converges towards the MAP results and SD performance.

It has also been shown that RMP could be more computationally efficient by using averaged mixture occupations in regression calculations. This eliminates the need for running the regression calculations every time adaptation is to be carried out and substantially reduces the amount of computations. To achieve this, however, for mixture Gaussian models, part of the improvement should be sacrificed, while for single Gaussians almost the same performance could be obtained. The iterative adaptation scheme using RMP has also been discussed in this chapter and under certain circumstances, it has been found useful.

Although no successful adaptation of variances or mixture weights of the Gaussians has been carried out in the RMP framework, these, as shown previously, are easily applicable to the initial MAP estimation process and are believed to boost the overall system performance. Also, other techniques to provide better prior parameters for MAP estimation can be used to improve the initial MAP estimation process, which would eventually lead to a better performance obtained from RMP.

6.5 Evaluation of Regression-Based Model Prediction using WSJ Data

A brief application of RMP approach to WSJ data has been carried out. This also involves an initial application of MAP.

The triphone system used in these experiments, as described in Section 6.2.2, consists of 6399 tied states for 23364 models. A single Gaussian and a 12 component mixture Gaussian versions of this system were used as baseline systems for these experiments. For RMP, in order to save in computations, a modified version of the RMP-AMO approach described in Section 6.4.5 was followed. This allows the use of the same regression parameters for all the speakers with similar adaptation sentences. The modification consists of disabling adaptation in the cases where, for any target parameter in the MAP estimated system of the speaker under test, either of the following happens

1. One of the source parameters has a mixture occupation less than the lower mixture occupation threshold (T_L).

2. No source parameter has a mixture occupation larger than the higher mixture occupation threshold (T_H).

3. Target mixture occupation is larger than that of any of the sources.

This prevents further adaptation of the parameters which have been assigned as target parameters during regression parameter calculations according to the mixture occupations of SD speakers' MAP estimated models but do not possess appropriate adaptation conditions according to test speaker's mixture occupations during MAP estimation.

Another improvement to RMP approach in this case has been the use of separate gender dependent regression parameters for male and female speakers. This was done by dividing the SD set of speakers to two male and female groups, each with 14 speakers. The regression parameters were then calculated separately for male and female speakers. It was aimed at improving the system performance as the acoustic differences within gender groups are known to be less, compared to the differences across gender groups. Some preliminary investigations confirmed this conclusion.

It is also worth noting that all the results reported in this section are generated using available transcriptions of the corresponding development test data.

6.5.1 Single Gaussian Triphones

Initially, it was decided to apply the adaptation algorithms to single Gaussian triphones for the reasons of simplicity and manageability of the models.

Based on the baseline system, 28 speaker dependent systems were generated using the data from WSJ0 and WSJ1 long-term speakers[4]. The training was carried out by applying one pass of MAP estimation, updating only the means. A similar procedure was followed for generating MAP estimated sentences with available adaptation sentences for each SD and new speaker. The MAP estimation approach was similar to the one used in RM experiments, as reported in Section 6.4.2.

The results of an initial investigation of the improvement of system parameters towards SD model parameters for the case of single Gaussians is shown in Table 6.15. The results illustrate the averages of the Mahalanobis distances of SI, MAP estimated and RMP-AMO adapted models with SD models over all 28 SD speakers. Forty adaptation sentences have been used in this case and RMP regression is carried out using the "leave one out" method. This shows that both MAP and RMP have succeeded in reducing the speakers' models distance with SD models.

In the next step, the MAP and RMP techniques were used to adapt the model parameters to a set of new speakers, i.e. the WSJ 1993 H2 development test speakers. The specifications of adaptation and test data for these speakers are given in Appendix B. The averaged error rates for SI models and MAP and RMP adapted models and percentage improvements over SI performance obtained from MAP and RMP adaptation with 40

[4] Further details are given in Appendix B.

System	SI	MAP	RMP
Distance	0.189	0.146	0.134

Table 6.15: The averaged Mahalanobis distance between SD model Gaussians and SI, MAP estimated and RMP adapted model Gaussians for single Gaussian state-clustered triphone system.

adaptation sentences for ten 1993 H2 development test speakers are given in the upper rows of Table 6.16. It should be noted that the modification in RMP-AMO approach mentioned above prevented about 15% of the target parameters from being further adapted in this case. For these experiments a regression order of 2 and correlation coefficient threshold of 0.4 were used. The other thresholds (T_H and T_L) were set according to the values of mixture occupation counts. While MAP estimation has resulted in 9.0% improvement in word error rate, RMP has increased this value to 19.2%, i.e. the contribution of the RMP in system performance improvement with 40 adaptation sentences is more than that of MAP.

		SI	MAP	RMP
40 Adaptation Sentences	Error Rate (%)	10.02	9.12	8.10
	Improvement (%)	-	9.0	19.2
1 Adaptation Sentence	Error Rate (%)	10.02	9.89	9.29
	Improvement (%)	-	1.3	7.2

Table 6.16: The averaged Error rates and percentage improvements for MAP and RMP adapted single Gaussian triphone systems using 40 and 1 adaptation sentences.

Further to the above test, a set of tests using one adaptation sentence was performed to assess the performance of RMP for very fast speaker adaptation. These results are also reported in Table 6.16. While MAP estimation shows only a very small improvement over SI system performance with one adaptation sentence, a 7% improvement is obtained from RMP.

A final set of experiments were directed on WSJ 1994 Spoke S3 non-native speakers development data set. The RMP in this case was carried out using the same general regression parameters used in the above H2 40 sentence adaptation experiments. Hence, all the RMP parameters remained the same. For recognition purposes, however, in order to prevent large number of insertions and deletions, the grammar scale factors and word insertion penalties had to be tuned. This is due to the fact that non-native speakers show many mispronunciations and phonetic differences in their speech, in comparison to natives, which leads to large increases in the system error rate. This can be partly overcome by changing the above parameters, i.e. by giving more weight to the language model and discouraging word insertion during recognition. In this work, however, a coarse tuning of the above parameters was employed.

Non-Natives			
	SI	MAP	RMP
Error Rate (%)	28.39	20.61	17.80
Improvement (%)	-	27.4	37.3

Table 6.17: The averaged Error rates and percentage improvements on non-native speakers for MAP and RMP adapted single Gaussian triphone systems using 40 adaptation sentences.

The recognition results for non-native speakers are given in Table 6.17, which show a 27.4% improvement in the error rate after the application of MAP estimation and a 37.3% improvement as a result of RMP. Although, this seems to be a considerable improvement, unlike the previous case, the contribution of RMP in performance improvement of the system in this case is not as high as MAP's. This could partly be due to the fact that the relationships between different phone models found using regression calculations may not be strictly correct for non-native speakers. Hence, their application to non-native speakers' MAP estimated models might not result in the same percentage improvement over MAP estimation error rates as obtained on native speakers.

6.5.2 Mixture Gaussian Triphones

The baseline mixture Gaussian system chosen for the application of RMP method was the 12 component state-clustered triphone system described in [103]. This consists of a total of nearly 77000 Gaussian components. A procedure similar to the one described in the previous section was followed to build Speaker dependent and MAP estimated models. The use of RMP-AMO approach was essential here for computational efficiency and the same adaptation prevention strategies in the cases of unacceptable adaptation were implemented.

In spite of the use of a gender dependent regression algorithm, besides RMP-AMO, which is believed to provide better regression parameter calculation, on average, only about 32% of the mixture components were updated during adaptation. This is believed to be due to the use of averaged mixture occupations and will be discussed later in this chapter.

The results of the application of MAP and RMP adaptation techniques using 40 adaptation sentences on native WSJ H2 development test speakers and non-native Spoke S3 development test speakers are shown in Table 6.18. The results reported in this section are not the result of full decoding, but have been obtained only by re-scoring. The same regression order and correlation threshold values used in previous experiments have also been used here, but the mixture occupation thresholds (T_H and T_L), as before, have been set according to the averaged mixture occupation counts calculated during MAP estimation for SD speakers. The same regression parameters have been used in both cases.

		SI	MAP	RMP
Native Speakers	Error Rate (%)	5.58	5.20	5.03
	Improvement (%)	-	6.8	9.9
Non-Native Speakers	Error Rate (%)	19.59	16.56	16.37
	Improvement (%)	-	15.5	16.4

Table 6.18: The averaged Error rates and percentage improvements on native and non-native speakers for MAP and RMP adapted 12 component mixture Gaussian triphone systems using 40 adaptation sentences.

The results of these experiments show about 10% improvement for native speakers as the result of application of RMP. However, larger improvements might have been expected given the previous experimental results with RM models. Several reasons may have caused this problem. A major cause is the small amount of improvement obtained from MAP estimation. This is due to the large number of system parameters in this case. Also better results from MAP estimation can be expected if better tuning of prior parameter setting is performed. The prior parameters used in this case were those found to perform reasonably well for the case of RM model sets, while they may not necessarily perform as well on WSJ models.

It should also be noted that in this case the parameters used in performing RMP-based regression and adaptation are either inherited from RM experiments, or set by a rule of thumb. Further optimisation and fine tuning of these parameters could lead to better performance. Another reason could be the use of RMP-AMO technique. Although this technique has performed well for the case of single Gaussians, due to the large number of Gaussians per state, the distribution of mixture occupation counts among Gaussians may not be similar for different speakers. Thus, the use of averaged mixture occupation counts may lead to choosing one component as source and another one as target while for the speaker under test, the component which must act as source might not have received enough training. The opposite could also happen to the target component. In fact, this is the reason why only 32% of the components have been adapted. Furthermore, even this small percentage might not have been adapted optimally since the sources allocated for each target, for the same reason, are not necessarily the best available sources.

In addition to the points raised in the previous section, the fact that the parameter relationships found on native speakers is being used for non-natives is the major reason preventing larger improvements for the case of non-natives.

6.5.3 Comparisons and Discussion

A comparison between the results obtained from the application of WSJ data and those of the RM experiments can be carried out in order to investigate the performance of the RMP method with WSJ data.

While for single Gaussian triphone models about a 30% reduction in the error rate was obtained during RM experiments with 40 adaptation sentences, WSJ experiments show an improvement of about 19% for native speakers. Taking into account that the number of parameters for WSJ models is about 4 times that of RM models (6399 versus 1581), while the length of the enrolment utterances is about twice that of RM's, about twice as many parameters should be adapted with the same amount of data. Hence, the performance of RMP on single Gaussian WSJ models is comparable to RM ones. A similar conclusion can be derived from 7.7% and 7.2% improvements with one adaptation sentence for RM and WSJ models respectively. The non-native improvements, however, can not be assessed in this way since there is not any outlier speaker available in RM database.

For the case of mixture Gaussian models, RM experiments showed an improvement of about 23% with 40 adaptation sentences, while for WSJ models this figure was about 10%. However, in this case the number of system parameters for WSJ system is about 8 times that of RM system (6399×12 versus 1581×6), while the length of the utterances is also about double. Thus, the ratio of the number of parameters per sentence for WSJ system is about 4 times that of RM's. As discussed in the last Section, many factors can affect the performance of the RMP in this situation. The number of mixture components, especially, seems to be very important in this case.

Another comparison could be with the performance of other speaker adaptation schemes. Leggetter [68], although has not reported static adaptation experiment results on similar native speakers' data, but has reported a best improvement of about 41% on the same non-native speakers' data set. However, for the reasons stated earlier, the improvement obtained with RMP on non-native speakers in this case is not large.

Further research seems to be necessary to obtain a better improvement from this approach in such cases. In order to avoid the problem associated with mixture Gaussians and RMP-AMO, regression parameter calculation can be performed on every single speaker. However, this can impose large computation overheads. Alternatively, one can use a clustering approach to obtain different regression parameters for different clusters and use them for the speakers allocated to each cluster. This would be a further improved version of the approach followed above in performing gender dependent regression.

Chapter 7

Conclusion and Further Work

The main emphasis in this thesis has been on the use of Maximum *a posteriori* (MAP) estimation for speaker adaptation of CDHMMs for continuous speech recognition. A number of approaches to this problem have been introduced and their implementation and results have been discussed.

7.1 Adaptation Algorithms

Two main speaker adaptation approaches have been dealt with throughout this thesis, MAP estimation and Regression-based Model Prediction (RMP). The important points regarding their implementation follow.

7.1.1 MAP Estimation

The history and use of the Bayesian techniques for speaker adaptation was discussed in Chapter 3. The theory, implementation and results are presented and discussed in Chapters 4 and 6.

In Chapter 4, the theory for MAP estimation, originally developed by Gauvain and Lee [36], was extended to cover the HMMs with non-emitting states used in HTK and implemented using the Forward-Backward algorithm within the HTK framework. Also, the problem of prior parameter estimation was discussed and both the *ad hoc* estimation of prior parameters using the SI model parameters, as devised by Gauvain and Lee, and an empirical Bayes approach to prior parameter estimation were discussed. Also, the method of improving the prior parameters by the use of a speaker clustering approach was introduced and a few possible different implementations of this technique explained.

The evaluations of the above approaches were carried out in Chapter 6. Using the ARPA Resource Management (RM) database, the evaluation of MAP estimation with SI-based priors was found to be effective in reducing the error rate with increasing amounts of adaptation data and that the results converge to SD results with large amounts of adaptation data. The evaluations were carried out on context-dependent and context-independent models with different numbers of Gaussian mixture components and similar results were obtained. The effects of adapting different model parameters and changes in prior parameters on the performance of the system were also assessed.

It was shown that an iterative approach to MAP estimation could improve the performance, especially for large amounts of training data, leading to performances superior to that of SD in some cases.

It was also found that the estimation of priors by the method of moments can lead to better MAP estimation results, although the computation and storage overheads are high. Hence, in the application of such an approach, extra attention should be paid to the trade-off between the performance gain and the computation and storage overheads.

Speaker clustering has also been found to be advantageous in estimating prior parameters. However, the advantages were not as great as hoped. This is believed to be due to the small number of available sentences (between 30 to 40) for each SI speaker used for clustering purposes, which made speaker model parameter estimation difficult. Other problems with speaker clustering, pointed out in Chapter 3, are also present in this approach. Thus, one can conclude that speaker clustering for this purpose could only be useful when certain conditions are met, such as larger number of sentences available per speaker, larger number of speakers and large variety of speakers.

MAP estimation was also used to perform unsupervised Bayesian adaptation in both batch and incremental modes. In the unsupervised batch mode, MAP estimation, while, as expected, did not perform as well as in supervised mode, still showed reasonable levels of improvement over SI result. In unsupervised incremental adaptation, when tested with the same 100 sentences and with no additional adaptation sentences, MAP estimation gave about 20% improvement over the SI result. These results demonstrate the abilities of MAP estimation to be used in a number of different adaptation scenarios.

7.1.2 RMP for Speaker Adaptation

Regression-based Model Prediction was developed to overcome the problem of untrained and undertrained models in a model parameter adaptation approach, usually caused by sparse training data. RMP can be applied to any model parameter adaptation approach suffering from this problem. In this case, MAP estimation has been chosen due to its special characteristics as a speaker adaptation technique and because, although designed to partly overcome the problem of sparse training data in a maximum likelihood estimation approach to HMM model training, itself still suffers from the same problem since only models with available training data are updated in this method.

The basic idea and the theory behind this approach are discussed in Chapter 5, while its evaluation has been addressed in Chapter 6. The RMP technique is based on the idea of using the relationships among phone model parameters in a speech recognition system to update model parameters of the phones with insufficient amounts of training data, using well-trained phone models. The relationships were found via multiple regression on sets of SD models trained before performing adaptation.

The results of the application of this approach to sets of models already MAP adapted to new speakers are given in Chapter 6. The results are obtained on different acoustic model structures including context-independent and context-dependent tied-state models. Also models with different numbers of Gaussian mixture components in the output distribution are used. Results have shown that RMP is particularly effective for very small amounts of adaptation data. The RMP method has shown an 8% improvement in the SI error rate with the introduction of only one sentence with a duration of about 3.7 seconds in a system with about 9500 mixture components. The importance of this improvement is clear considering that with the same adaptation data the MAP estimated system performance, compared to SI, deteriorates.

Another important point about RMP is that it always outperforms the MAP estimated system with any number of adaptation sentences and eventually converges to the MAP performance. However, as expected, with larger amounts of adaptation data the improvements obtained by RMP, compared to MAP, reduce.

Another comparison showed that for the case of one adaptation sentence, about 75% of the Gaussians of an RMP-adapted system are closer to their SD counterparts in comparison to MAP-adapted system Gaussians, while only about 20% in the latter system are closer to SD Gaussians, which shows that RMP adapts in the right direction in most cases.

The higher computational cost of the RMP compared to a MAP estimation approach has been shown to be considerably reduced by the use of averaged mixture occupations. The performance degradations which result when such an approach is applied to mixture Gaussian systems are found not to be very large for the case of RM models, while for the case of WSJ models this can result in larger performance degradations. This is believed to be due to the larger number of mixture components per state in the latter case. This effect might be reduced by the application of some grouping techniques such as speaker clustering or male/female grouping for regression purpose. However, its application to single Gaussians (called averaged state occupations in this case) has been found to incur almost no performance degradation.

Altogether, RMP has been found to be an effective approach to very fast supervised speaker adaptation in the case of medium to large vocabulary continuous speech recognition using HMMs. Its convergence towards SD system performance and outperforming MAP with any number of adaptation sentences are other advantages of RMP.

7.2 Further Work

There are several areas for improvement in speaker adaptation using HMMs, many of which have already been addressed by researchers. As far as MAP and RMP adaptation techniques are concerned, the following areas are still potentially open for further research

- Prior parameter estimation is still an area for research and remains the most important issue faced in the application of MAP estimation techniques. The importance of the prior parameter values in MAP estimation results has already been shown by comparing the results obtained from different methods. However, there is a general belief that available methods do not lead to the best possible priors [71, 48].

- The speaker clustering approach for prior parameter estimation in MAP adaptation needs more research and is expected to have a larger influence on the system performance improvement if the obstacles stated earlier, such as very limited amounts of data for SI speakers or limited numbers of speakers, could be removed.

- Adaptation of other HMM parameters besides means, such as variances and mixture weights, in RMP still need further attention. As stated in Chapter 6, further work in this area is still needed to provide more appropriate adaptation techniques for these parameters.

- The threshold settings have not been optimised in RMP. Although the adaptation process has been found to be fairly insensitive to adaptation thresholds, their effect on the system performance improvement cannot be disregarded.

- The RMP technique has been applied to a state-clustered triphone system. Nevertheless, the distribution relationships have been found at mixture component level. This is the main reason for obtaining a smaller improvement when averaged mixture occupations are used for reducing computation cost. Hence, the application of this technique to tied-mixture systems might further improve the performance.

- In the application of RMP to WSJ data, the two last items mentioned above are believed to degrade the performance of the adaptation technique, especially in the mixture Gaussian case, due to the fairly large number of mixture components per state. Hence, improvements in any of these fields could yield larger improvements.

- No attempt has been made to obtain the optimal number of system parameters in any stage of adaptation. However, it is generally known that for any adaptation approach the adaptation performance is very dependent on the number of the system parameters. Hence, such an attempt might lead to better adaptation results.

It is believed that the above points are important in the MAP and RMP techniques and further research in these fields could improve the adaptation capabilities of these techniques.

Appendix A

Derivation of the MAP Estimation Formulations

A complete derivation of the MAP estimation formulae presented in Chapter 4 is given here. Note that the procedure followed during these derivations is similar to the one followed by Gauvain and Lee in [36].

A.1 Derivation of MAP Formulations for Gaussian Mixture Densities

The ultimate goal, here, is to maximise the posterior distribution of parameter vector $\boldsymbol{\theta}$, as defined in (4.6). For the observation vector $\boldsymbol{O} = (\boldsymbol{o}_1, ..., \boldsymbol{o}_T)$ of T independent and identically distributed (i.i.d.) observation samples, the joint p.d.f. is given by (4.7).

Defining the prior density of the mixture weight parameter vector to be a Dirichlet density of the form given in (4.9), and the joint prior density of $(\boldsymbol{m}_k, \boldsymbol{r}_k)$ to be a normal-Wishart density of the form given in (4.10), the joint prior density $g(\boldsymbol{\theta})$, as indicated before, can be given as the product of these prior p.d.f.s, i.e.

$$g(\boldsymbol{\theta}) = g(\omega_1, ..., \omega_K) \prod_{k=1}^{K} g(\boldsymbol{m}_k, \boldsymbol{r}_k). \tag{A.1}$$

MAP estimation of the parameters using the EM algorithm consists of maximising an auxiliary function $Q(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}})$ at each iteration [4, 24, 89]. This function is defined as the expectation of the *complete data* log-likelihood log $h(\boldsymbol{U}|\boldsymbol{\theta})$, given the incomplete data $\boldsymbol{O} = (\boldsymbol{o}_1, ..., \boldsymbol{o}_T)$ and the current fit $\hat{\boldsymbol{\theta}}$, i.e.

$$Q(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}) = E[\log h(\boldsymbol{U}|\boldsymbol{\theta}) \ |\boldsymbol{O}, \hat{\boldsymbol{\theta}}]. \tag{A.2}$$

Note that for the complete data, $\boldsymbol{U} = (\boldsymbol{O}, \boldsymbol{l})$, where $\boldsymbol{l} = (l_1, ..., l_T)$ is the vector of unobserved mixture component labels. Also note that [24, 89]

$$\log\ f(\boldsymbol{O}|\boldsymbol{\theta}) = Q(\boldsymbol{\theta},\hat{\boldsymbol{\theta}}) - H(\boldsymbol{\theta},\hat{\boldsymbol{\theta}})$$

where

$$H(\boldsymbol{\theta},\hat{\boldsymbol{\theta}}) = E[\log h(\boldsymbol{U}|\boldsymbol{O},\boldsymbol{\theta})\ |\boldsymbol{O},\hat{\boldsymbol{\theta}}]$$

and

$$H(\boldsymbol{\theta},\hat{\boldsymbol{\theta}}) \leq H(\hat{\boldsymbol{\theta}},\hat{\boldsymbol{\theta}}).$$

Thus if for any $\boldsymbol{\theta}$, $Q(\boldsymbol{\theta},\hat{\boldsymbol{\theta}}) > Q(\hat{\boldsymbol{\theta}},\hat{\boldsymbol{\theta}})$, then

$$f(\boldsymbol{O}|\boldsymbol{\theta}) > f(\boldsymbol{O}|\hat{\boldsymbol{\theta}}).$$

Therefore the same iterative procedure can be followed to maximise the function $\Omega(\boldsymbol{\theta},\hat{\boldsymbol{\theta}}) = Q(\boldsymbol{\theta},\hat{\boldsymbol{\theta}}) + \log g(\boldsymbol{\theta})$ for MAP estimation, in place of $Q(\boldsymbol{\theta},\hat{\boldsymbol{\theta}})$ in conventional ML estimation [24].

The auxiliary function Q in the case of a mixture of K densities $\{f(\cdot|\boldsymbol{\theta}_k)\}_{k=1,\ldots,K}$ will become [89]

$$Q(\boldsymbol{\theta},\hat{\boldsymbol{\theta}}) = \sum_{t=1}^{T}\sum_{k=1}^{K} \hat{\omega}_k f(\boldsymbol{o}_t|\hat{\boldsymbol{\theta}}_k)\ f^{-1}(\boldsymbol{o}_t|\hat{\boldsymbol{\theta}})\ \log \omega_k f(\boldsymbol{o}_t|\boldsymbol{\theta}_k). \tag{A.3}$$

Let the function $\Upsilon(\boldsymbol{\theta},\hat{\boldsymbol{\theta}}) = \exp\left[\Omega(\boldsymbol{\theta},\hat{\boldsymbol{\theta}})\right]$ be the function to be maximised, and also let

$$\begin{aligned}
c_{kt} &= \hat{\omega}_k f(\boldsymbol{o}_t|\hat{\boldsymbol{\theta}}_k)\ f^{-1}(\boldsymbol{o}_t|\hat{\boldsymbol{\theta}}) = \hat{\omega}_k \mathcal{N}(\boldsymbol{o}_t|\hat{\boldsymbol{m}}_k,\hat{\boldsymbol{r}}_k)\ f^{-1}(\boldsymbol{o}_t|\hat{\boldsymbol{\theta}}) && \text{(A.4)}\\
c_k &= \sum_{t=1}^{T} c_{kt} && \text{(A.5)}\\
\bar{\boldsymbol{o}}_k &= \frac{\sum_{t=1}^{T} c_{kt}\boldsymbol{o}_t}{c_k} && \text{(A.6)}\\
\boldsymbol{W}_k &= \sum_{t=1}^{T} c_{kt}(\boldsymbol{o}_t - \bar{\boldsymbol{o}}_k)(\boldsymbol{o}_t - \bar{\boldsymbol{o}}_k)'. && \text{(A.7)}
\end{aligned}$$

One can also write [23]

$$\sum_{t=1}^{T}(\boldsymbol{o}_t - \boldsymbol{m}_k)'\boldsymbol{r}_k(\boldsymbol{o}_t - \boldsymbol{m}_k) = \sum_{t=1}^{T}(\boldsymbol{o}_t - \bar{\boldsymbol{o}}_k)'\boldsymbol{r}_k(\boldsymbol{o}_t - \bar{\boldsymbol{o}}_k) + \sum_{t=1}^{T}(\boldsymbol{m}_k - \bar{\boldsymbol{o}}_k)'\boldsymbol{r}_k(\boldsymbol{m}_k - \bar{\boldsymbol{o}}_k). \tag{A.8}$$

Also since

$$tr(\boldsymbol{W}_k\boldsymbol{r}_k) = \sum_{t=1}^{T} c_{kt}(\boldsymbol{o}_t - \bar{\boldsymbol{o}}_k)'\boldsymbol{r}_k(\boldsymbol{o}_t - \bar{\boldsymbol{o}}_k) \tag{A.9}$$

Equation (A.8) can be rewritten as

$$\sum_{t=1}^{T} c_{kt}(\boldsymbol{o}_t - \boldsymbol{m}_k)'\boldsymbol{r}_k(\boldsymbol{o}_t - \boldsymbol{m}_k) = c_k(\boldsymbol{m}_k - \bar{\boldsymbol{o}}_k)'\boldsymbol{r}_k(\boldsymbol{m}_k - \bar{\boldsymbol{o}}_k) + tr(\boldsymbol{W}_k\boldsymbol{r}_k). \tag{A.10}$$

Using (A.3), it is also possible to write

$$\begin{aligned}\Omega(\boldsymbol{\theta},\hat{\boldsymbol{\theta}}) &= Q(\boldsymbol{\theta},\hat{\boldsymbol{\theta}})+\log\ g(\boldsymbol{\theta})\\ &= \sum_{t=1}^{T}\sum_{k=1}^{K}\hat{\omega}_k f(\boldsymbol{o}_t|\hat{\boldsymbol{\theta}}_k)\ f^{-1}(\boldsymbol{o}_t|\hat{\boldsymbol{\theta}})\ \log\ \omega_k f(\boldsymbol{o}_t|\boldsymbol{\theta}_k)+\log\ g(\boldsymbol{\theta}).\end{aligned} \tag{A.11}$$

Then

$$\begin{aligned}\Upsilon(\boldsymbol{\theta},\hat{\boldsymbol{\theta}}) &= \exp\ [\Omega(\boldsymbol{\theta},\hat{\boldsymbol{\theta}})]\\ &= g(\boldsymbol{\theta})\prod_{k=1}^{K}[\omega_k f(\boldsymbol{o}_t|\boldsymbol{\theta}_k)]^{c_k}\\ &\propto g(\boldsymbol{\theta})\prod_{k=1}^{K}\omega_k^{c_k}|\boldsymbol{r}_k|^{c_k/2}\ \exp[-\sum_{t=1}^{T}\frac{c_{kt}}{2}(\boldsymbol{o}_t-\boldsymbol{m}_k)'\boldsymbol{r}_k(\boldsymbol{o}_t-\boldsymbol{m}_k)].\end{aligned} \tag{A.12}$$

Using (A.10), one will have

$$\Upsilon(\boldsymbol{\theta},\hat{\boldsymbol{\theta}})\propto g(\boldsymbol{\theta})\prod_{k=1}^{K}\omega_k^{c_k}|\boldsymbol{r}_k|^{c_k/2}\ \exp[-\frac{c_k}{2}(\boldsymbol{m}_k-\bar{\boldsymbol{o}}_k)'\boldsymbol{r}_k(\boldsymbol{m}_k-\bar{\boldsymbol{o}}_k)-1/2tr(\boldsymbol{W}_k\boldsymbol{r}_k)]. \tag{A.13}$$

Comparing (A.13) and (A.1) it can be verified that $\Upsilon(\cdot|\hat{\boldsymbol{\theta}})$ and $g(\cdot)$ belong to the same family of distributions. The parameters of $\Upsilon(\cdot|\hat{\boldsymbol{\theta}})$, i.e. $\{\psi'_k,\tau'_k,\boldsymbol{\mu}'_k,\alpha'_k,\boldsymbol{u}'_k\}_{k=1,\ldots,K}$ can be derived as follows. For the parameters ψ'_k and α'_k, direct comparison will result

$$\psi'_k = \psi_k+c_k \tag{A.14}$$
$$\alpha'_k = \alpha_k+c_k. \tag{A.15}$$

For the other parameters, the combined exponential term of (A.13) would become

$$\exp[-\frac{\tau_k}{2}(\boldsymbol{m}_k-\boldsymbol{\mu}_k)'\boldsymbol{r}_k(\boldsymbol{m}_k-\boldsymbol{\mu}_k)-\frac{c_k}{2}(\boldsymbol{m}_k-\bar{\boldsymbol{o}}_k)'\boldsymbol{r}_k(\boldsymbol{m}_k-\bar{\boldsymbol{o}}_k)-1/2tr(\boldsymbol{u}_k\boldsymbol{r}_k)-1/2tr(\boldsymbol{W}_k\boldsymbol{r}_k)]. \tag{A.16}$$

Referring to [23], this can be rewritten as

$$\begin{aligned}\exp\{-\frac{\tau_k+c_k}{2}(\boldsymbol{m}_k-\boldsymbol{\mu}'_k)'\boldsymbol{r}_k(\boldsymbol{m}_k-\boldsymbol{\mu}'_k)-1/2tr[\frac{\tau_k c_k}{\tau_k+c_k}(\boldsymbol{\mu}_k-\bar{\boldsymbol{o}}_k)(\boldsymbol{\mu}_k-\bar{\boldsymbol{o}}_k)'\boldsymbol{r}_k]\\ -1/2tr(\boldsymbol{u}_k\boldsymbol{r}_k)-1/2tr(\boldsymbol{W}_k\boldsymbol{r}_k)\}\end{aligned} \tag{A.17}$$

where

$$\boldsymbol{\mu}'_k=\frac{\tau_k\boldsymbol{\mu}_k+c_k\bar{\boldsymbol{o}}_k}{\tau_k+c_k}. \tag{A.18}$$

Also comparing this exponent with Equations (4.11) and (4.10) one can conclude

$$\tau'_k=\tau_k+c_k \tag{A.19}$$

and

$$\boldsymbol{u}'_k=\boldsymbol{u}_k+\boldsymbol{W}_k+\frac{\tau_k c_k}{\tau_k+c_k}(\boldsymbol{\mu}_k-\bar{\boldsymbol{o}}_k)(\boldsymbol{\mu}_k-\bar{\boldsymbol{o}}_k)'. \tag{A.20}$$

Therefore it is understood that the prior p.d.f. defined in (4.11) as $g(\cdot)$ can be considered as a conjugate family for $\Upsilon(\cdot,\hat{\boldsymbol{\theta}})$, i.e. the complete-data density.

The estimates of the posterior parameters $(\tilde{\omega}_k, \tilde{\boldsymbol{m}}_k, \tilde{\boldsymbol{r}}_k)$ can then be derived using Dirichlet, normal, and Wishart distributions' properties. Referring to the definitions of these distributions [23], the mode of the posterior distribution will be obtained as

$$\begin{aligned} \tilde{\omega}_k &= \frac{\psi'_k - 1}{\sum_{k=1}^{K}(\psi'_k - 1)} && \text{(A.21)} \\ \tilde{\boldsymbol{m}}_k &= \boldsymbol{\mu}'_k && \text{(A.22)} \\ \tilde{\boldsymbol{r}}_k &= (\alpha'_k - V)\boldsymbol{u}'^{-1}_k. && \text{(A.23)} \end{aligned}$$

The resultant MAP estimate formulae can then be derived. Using (A.21), (A.14) and (A.5) will result in

$$\tilde{\omega}_k = \frac{\psi_k - 1 + \sum_{t=1}^{T} c_{kt}}{\sum_{k=1}^{K}\psi_k - K + \sum_{k=1}^{K}\sum_{t=1}^{T} c_{kt}}. \tag{A.24}$$

From Equations (A.22), (A.18), (A.6) and (A.5) it follows that

$$\tilde{\boldsymbol{m}}_k = \frac{\tau_k \boldsymbol{\mu}_k + \sum_{t=1}^{T} c_{kt}\boldsymbol{o}_t}{\tau_k + \sum_{t=1}^{T} c_{kt}}, \tag{A.25}$$

and finally using (A.23), (A.18), (A.20), (A.7) and (A.6)

$$\tilde{\boldsymbol{r}}_k^{-1} = \frac{\boldsymbol{u}_k + \sum_{t=1}^{T} c_{kt}(\boldsymbol{o}_t - \tilde{\boldsymbol{m}}_k)(\boldsymbol{o}_t - \tilde{\boldsymbol{m}}_k)' + \tau_k(\tilde{\boldsymbol{m}}_k - \boldsymbol{\mu}_k)(\tilde{\boldsymbol{m}}_k - \boldsymbol{\mu}_k)'}{\alpha_k - V + \sum_{t=1}^{T} c_{kt}}. \tag{A.26}$$

The Equations (A.24), (A.25) and (A.26) are similar to (4.12), (4.13) and (4.14) and are the MAP estimation formulae for the parameters of mixture Gaussians. The extension of the above equations to the case of diagonal covariance matrices, pointed out in (4.17) and (4.18), is trivial and hence is not repeated here.

A.2 Derivation of MAP Formulations for HMM Parameters

Consider a hidden Markov model with parameter vector $\boldsymbol{\lambda} = (\boldsymbol{a}_1, \boldsymbol{a}_N, \boldsymbol{A}, \boldsymbol{\theta})$ where the parameters are as defined in Section 4.5. For a sample $\boldsymbol{O} = (\boldsymbol{o}_1, ..., \boldsymbol{o}_T)$, the complete data is $\boldsymbol{U} = (\boldsymbol{O}, \boldsymbol{s}, \boldsymbol{l})$. The joint p.d.f. for the complete data is defined as in (4.19). As stated before, if the parameters $\boldsymbol{A}$, $\boldsymbol{a}_1$ and $\boldsymbol{a}_N$ are assumed fixed and known, the prior density would be

$$G(\boldsymbol{\lambda}) = \prod_{i=2}^{N-1} g(\boldsymbol{\theta}_i) \tag{A.27}$$

where $g(\boldsymbol{\theta}_i)$ is defined by (4.11). In a more general case, a Dirichlet density can be used for both the initial and final probability vectors $\boldsymbol{a}_1$ and $\boldsymbol{a}_N$, and also each row of the transition probability matrix $\boldsymbol{A}$. Thus for all the HMM parameters, the relationship stated in (4.21) exists.

In order to apply the forward-backward algorithm to MAP estimates in this case, the auxiliary function Q of the EM algorithm should first be defined as

$$Q(\boldsymbol{\lambda}, \hat{\boldsymbol{\lambda}}) = E[\log\ h(\boldsymbol{U}|\boldsymbol{\lambda})|\boldsymbol{O}, \hat{\boldsymbol{\lambda}}]. \tag{A.28}$$

In this case, the above function can be decomposed into the sum of four auxiliary functions, using (4.19) [53]. These four functions can be independently maximised and have the following forms

$$Q_{a_1}(\boldsymbol{a}_1, \hat{\boldsymbol{\lambda}}) = \sum_{i=2}^{N-1} \gamma_{i1} \log a_{1i} \tag{A.29}$$

$$Q_{a_N}(\boldsymbol{a}_N, \hat{\boldsymbol{\lambda}}) = \sum_{i=2}^{N-1} \gamma_{iT} \log a_{iN} \tag{A.30}$$

$$Q_A(\boldsymbol{A}, \hat{\boldsymbol{\lambda}}) = \sum_{t=1}^{T} \sum_{i=2}^{N-1} \sum_{j=2}^{N-1} P(s_{t-1} = i, s_t = j|\boldsymbol{O}, \hat{\boldsymbol{\lambda}}) \log a_{ij} = \sum_{i=2}^{N-1} Q_{a_i}(a_{ij}, \hat{\boldsymbol{\lambda}}) \tag{A.31}$$

$$Q_\theta(\boldsymbol{\theta}, \hat{\boldsymbol{\lambda}}) = \sum_{t=1}^{T} \sum_{i=2}^{N-1} \sum_{k=1}^{K} P(s_t = i, l_t = k|\boldsymbol{O}, \hat{\boldsymbol{\lambda}})\ \log \omega_{ik} f(\boldsymbol{o}_t|\boldsymbol{\theta}_{ik}) = \sum_{i=2}^{N-1} Q_{\boldsymbol{\theta}_i}(\theta_i|\hat{\boldsymbol{\lambda}}), \tag{A.32}$$

where

$$Q_{a_i}(a_{ij}, \hat{\boldsymbol{\lambda}}) = \sum_{t=1}^{T} \sum_{j=2}^{N-1} \xi_{ijt}\ \log\ a_{ij} \tag{A.33}$$

$$Q_{\theta_i}(\boldsymbol{\theta}_i|\hat{\boldsymbol{\lambda}}) = \sum_{t=1}^{T} \sum_{k=1}^{K} \gamma_{it} \frac{\hat{\omega}_{ik} f(\boldsymbol{o}_t|\hat{\boldsymbol{\theta}}_{ik})}{f(\boldsymbol{o}_t|\hat{\boldsymbol{\theta}}_i)}\ \log\ \omega_{ik} f(\boldsymbol{o}_t|\boldsymbol{\theta}_{ik}). \tag{A.34}$$

and where $\xi_{ijt} = P(s_{t-1} = i, s_t = j|\boldsymbol{O}, \hat{\boldsymbol{\lambda}})$ and $\gamma_{it} = P(s_t = i|\boldsymbol{O}, \hat{\boldsymbol{\lambda}})$. Both of these probabilities can be computed at each EM iteration.

Once again the auxiliary function $\Omega(\boldsymbol{\lambda}, \hat{\boldsymbol{\lambda}}) = Q(\boldsymbol{\lambda}, \hat{\boldsymbol{\lambda}}) + \log\ G(\boldsymbol{\lambda})$ is to be maximised. Referring to (4.21) it can easily be understood that independent maximisation of the parameters is possible. Thus $\Omega(\boldsymbol{\lambda}, \hat{\boldsymbol{\lambda}})$ is written as $\Omega_{a_1}(\boldsymbol{a}_1, \hat{\boldsymbol{\lambda}}) + \Omega_{a_N}(\boldsymbol{a}_N, \hat{\boldsymbol{\lambda}}) + \sum_i \Omega_{a_i}(a_i, \hat{\boldsymbol{\lambda}}) + \sum_i \Omega_{\boldsymbol{\theta}_i}(\boldsymbol{\theta}_i, \hat{\boldsymbol{\lambda}})$. Then one can write

$$\Omega_{a_1}(\boldsymbol{a}_1, \hat{\boldsymbol{\lambda}}) = Q_{a_1}(\boldsymbol{a}_1, \hat{\boldsymbol{\lambda}}) +\ \log\ (\prod_{i=2}^{N-1} a_{1i}^{\eta_i - 1}) \tag{A.35}$$

hence

$$\begin{aligned} \Upsilon_{a_1}(\boldsymbol{a}_1, \hat{\boldsymbol{\lambda}}) &= \exp[\Omega_{a_1}(\boldsymbol{a}_1, \hat{\boldsymbol{\lambda}})] \\ &= \exp[Q_{a_1}(\boldsymbol{a}_1, \hat{\boldsymbol{\lambda}})] \prod_{i=2}^{N-1} a_{1i}^{\eta_i - 1} \\ &= \prod_{i=2}^{N-1} a_{1i}^{\gamma_{i1} + \eta_i - 1}, \end{aligned} \tag{A.36}$$

also

$$\Omega_{a_N}(\boldsymbol{a}_N, \hat{\boldsymbol{\lambda}}) = Q_{a_N}(\boldsymbol{a}_N, \hat{\boldsymbol{\lambda}}) +\ \log\ (\prod_{i=2}^{N-1} a_{iN}^{\eta_i - 1}) \tag{A.37}$$

hence

$$\begin{aligned}\Upsilon_{a_N}(\boldsymbol{a}_N, \hat{\boldsymbol{\lambda}}) &= \exp[\Omega_{a_N}(\boldsymbol{a}_N, \hat{\boldsymbol{\lambda}})] \\ &= \exp[Q_{a_N}(\boldsymbol{a}_N, \hat{\boldsymbol{\lambda}})] \prod_{i=2}^{N-1} a_{iN}^{\eta_i - 1} \\ &= \prod_{i=2}^{N-1} a_{iN}^{\gamma_{iT}+\eta_i-1},\end{aligned} \tag{A.38}$$

and

$$\begin{aligned}\Omega_{\theta_i}(\boldsymbol{\theta}_i, \hat{\boldsymbol{\lambda}}) &= Q_{\theta_i}(\boldsymbol{\theta}_i, \hat{\boldsymbol{\lambda}}) + \log\ g(\boldsymbol{\theta}_i) \\ &= \sum_{t=1}^{T}\sum_{k=1}^{K} \gamma_{it} \frac{\hat{\omega}_{ik} f(\boldsymbol{o}_t|\hat{\boldsymbol{\theta}}_{ik})}{f(\boldsymbol{o}_t|\hat{\boldsymbol{\theta}}_i)}\ \log\ \omega_{ik} f(\boldsymbol{o}_t|\boldsymbol{\theta}_{ik}) + \log\ g(\boldsymbol{\theta}_i)\end{aligned} \tag{A.39}$$

Then, similar to the previous discussion

$$\Upsilon_{\theta_i}(\boldsymbol{\theta}_i, \hat{\boldsymbol{\lambda}}) \propto g(\boldsymbol{\theta}_i) \prod_k \omega_{ik}^{c_{ik}} |\boldsymbol{r}_{ik}|^{c_{ik}/2}\ \exp[-\frac{c_{ik}}{2}(\boldsymbol{m}_{ik} - \bar{\boldsymbol{o}}_{ik}) - 1/2tr(\boldsymbol{W}_{ik}\boldsymbol{r}_{ik})] \tag{A.40}$$

where

$$c_{ik} = \sum_{t=1}^{T} c_{ikt} \tag{A.41}$$

and

$$c_{ikt} = \gamma_{it} \frac{\hat{\omega}_{ik} f(\boldsymbol{o}_t|\hat{\boldsymbol{\theta}}_{ik})}{f(\boldsymbol{o}_t|\hat{\boldsymbol{\theta}}_i)}, \tag{A.42}$$

and finally

$$\Omega_A(\boldsymbol{A}, \hat{\boldsymbol{\lambda}}) = Q_A(\boldsymbol{A}, \hat{\boldsymbol{\lambda}}) + \log(\prod_{i=2}^{N-1}\prod_{j=2}^{N-1} a_{ij}^{\eta_{ij}-1}) \tag{A.43}$$

hence

$$\begin{aligned}\Upsilon_A(A, \hat{\boldsymbol{\lambda}}) &= \exp[\Omega_A(A, \hat{\boldsymbol{\lambda}})] \\ &= \exp[\sum_t\sum_i\sum_j \xi_{ijt}\ \log\ a_{ij}].\prod_i\prod_j a_{ij}^{\eta_{ij}-1} \\ &= \prod_i\prod_j a_{ij}^{\sum_t \xi_{ijt}}.\prod_i\prod_j a_{ij}^{\eta_{ij}-1} \\ &= \prod_i\prod_j a_{ij}^{\sum_t \xi_{ijt}+\eta_{ij}-1}.\end{aligned} \tag{A.44}$$

It should be noted that in the above equations, compared to the previous discussion, c_{kt} in (4.15) is replaced by c_{ikt} as defined by (A.42).

Since the posterior distribution of a_{1i} has been assumed to be a Dirichlet distribution, using Equation (A.36) above will lead to

$$\begin{aligned}\tilde{a}_{1i} &= \frac{\gamma_{i1} + \eta_i - 1}{\sum_{j=2}^{N-1}(\gamma_{j1} + \eta_j - 1)} \\ &= \frac{\eta_i - 1 + \gamma_{i1}}{\sum_{j=2}^{N-1}(\eta_j - 1) + \sum_{j=2}^{N-1}\gamma_{j1}}.\end{aligned} \tag{A.45}$$

Also for the posterior distribution of a_{iN}, the same procedure will lead to

$$\tilde{a}_{iN} = \frac{\eta'_i - 1 + \gamma_{iT}}{\sum_{j=2}^{N-1}(\eta'_j - 1) + \sum_{j=2}^{N-1}\gamma_{jT}}. \tag{A.46}$$

For a_{ij}, the same Dirichlet distribution exists which using (A.44) leads to

$$\begin{aligned}\tilde{a}_{ij} &= \frac{\eta_{ij} - 1 + \sum_{t=1}^{T}\xi_{ijt}}{\sum_{j=2}^{N-1}(\sum_{t=1}^{T}\xi_{ijt} + \eta_{ij} - 1)} \\ &= \frac{\eta_{ij} - 1 + \sum_{t=1}^{T}\xi_{ijt}}{\sum_{j=2}^{N-1}(\eta_{ij} - 1) + \sum_{j=2}^{N-1}\sum_{t=1}^{T}\xi_{ijt}}.\end{aligned} \tag{A.47}$$

To get the estimate formulae for $\boldsymbol{\theta}$, since the posterior distributions for the parameters of it are the same, the same procedure followed in the previous discussion should be followed. Using (A.21) and extending notations results in

$$\tilde{\omega}_{ik} = \frac{\psi'_{ik} - 1}{\sum_{k=1}^{K}(\psi'_{ik} - 1)} \tag{A.48}$$

by extending the subscripts in (A.14) and using (A.41)

$$\begin{aligned}\tilde{\omega}_{ik} &= \frac{\psi_{ik} + \sum_{t=1}^{T}c_{ikt} - 1}{\sum_{k=1}^{K}(\psi_{ik} + \sum_{t=1}^{T}c_{ikt} - 1)} \\ &= \frac{\psi_{ik} - 1 + \sum_{t=1}^{T}c_{ikt}}{\sum_{k=1}^{K}\psi_{ik} - K + \sum_{k=1}^{K}\sum_{t=1}^{T}c_{ikt}}.\end{aligned} \tag{A.49}$$

From (A.22)

$$\tilde{\boldsymbol{m}}_{ik} = \boldsymbol{\mu}'_{ik}$$

and from (A.6), (A.18) and (A.41)

$$\tilde{\boldsymbol{m}}_{ik} = \frac{\tau_{ik}\boldsymbol{\mu}_{ik} + \sum_{t=1}^{T}c_{ikt}\boldsymbol{o}_t}{\tau_{ik} + \sum_{t=1}^{T}c_{ikt}}, \tag{A.50}$$

and finally from (A.23)

$$\tilde{\boldsymbol{r}}_{ik} = (\alpha'_{ik} - V)\boldsymbol{u}'^{-1}_{ik},$$

using (A.15), (A.20), (A.7) and (A.6)

$$\tilde{\boldsymbol{r}}_{ik}^{-1} = \frac{\boldsymbol{u}_{ik} + \sum_{t=1}^{T}c_{ikt}(\boldsymbol{o}_t - \tilde{\boldsymbol{m}}_{ik})(\boldsymbol{o}_t - \tilde{\boldsymbol{m}}_{ik})' + \tau_{ik}(\tilde{\boldsymbol{m}}_{ik} - \boldsymbol{\mu}_{ik})(\tilde{\boldsymbol{m}}_{ik} - \boldsymbol{\mu}_{ik})'}{\alpha_{ik} - V + \sum_{t=1}^{T}c_{ikt}}. \tag{A.51}$$

The Equations (A.45), (A.46), (A.47), (A.49), (A.50) and (A.51) are similar to (4.26), (4.27), (4.28), (4.29), (4.30) and (4.31) and are the MAP estimation formulae for the HMM parameters.

Appendix B

Continuous Speech Data

Two continuous speech databases are used in this work. These are the ARPA Resource Management database (RM1) and the ARPA Wall Street Journal CSR corpus (WSJ). The specifications of the parts used from both of these databases are given in this appendix, together with the parametrisation technique used for coding the data.

B.1 Resource Management Database

The ARPA RM1 database [84] consists of two parts: speaker independent (SI) and speaker dependent (SD). Both parts consist of training, development and evaluation test sections. The database is a set of American English utterances of a naval resource management task consisting of about 1000 words. The speakers are chosen from a variety of U.S. dialects and materials such as dialect sentences and adaptation sentences are provided in the database.

B.1.1 Speaker Independent Training Data

The training section of the speaker independent database consists of the speech from 109 speakers with an even distribution over four U.S. geographic regions. The ratio of male/female speakers in this part of this database is about 70/30. Each subject in the training section reads about 30 to 40 sentences, leading to a total of 3990 sentences.

About 40 speakers are dedicated to each of the development and evaluation tests in this database, with specifications similar to that of the training section, leading to nearly 1200 sentences per test. Moreover, each test speaker has also recorded 10 adaptation sentences.

B.1.2 Speaker Dependent Data

The speaker dependent database consists of 12 speakers, from which 11 are common between SD and SI databases. There are 7 male and 5 female speakers in this part of database, with various geographical representations.

The training section consists of 600 common sentences read by each of the 12 speakers plus 10 adaptation sentences. There are 4 evaluation and development test sets available for the RM1 SD database. These are the February 1989, October 1989, February 1991 and September 1992 test sets, usually known as Feb 89, Oct 89, Feb 91 and Sep 92 test sets respectively. Each of these test sets consists of 25 sentences per speaker, comprising a total of 100 test sentences per speaker for each of the evaluation and development sets.

B.2 Wall Street Journal Corpus

The ARPA Wall Street Journal Database is designed for general purpose, large vocabulary continuous speech recognition. This CSR corpus consists of several sections with different vocabulary sizes, different perplexities and short-term and long-term speaker dependent and speaker independent training material. The parts of the corpus which we are concerned with are explained below.

B.2.1 Short-Term Speaker Independent Training Data

This part, which is also referred to as SI-284, consists of the speech data from two releases of the above corpus: WSJ0 and WSJ1. Both these sets are used for training the SI system, making up a total of 36493 sentences, after discarding 707 sentences due to mispronunciations and alignment problems.

The WSJ0 short-term speaker independent training data consists of the utterances from 84 speakers. These include both verbalised and non-verbalised pronunciations of the words. For the training purposes, some of the utterances with mispronunciations of the words were removed and in total 7185 sentences were used during the training process which make up about 14 hours of speech data.

The WSJ1 short-term speaker independent training data comprises the utterances from 200 speakers. Every speaker has recorded about 150 sentences and in total 29308 sentences from this part of the corpus were used for training the speaker independent system. This is about 52 hours of speech data. It is worth noting that this part of the corpus also includes verbalised and non-verbalised pronunciations of the words.

It should also be noted that only the channel 1 data of the corpus (recorded by a Sennheiser microphone) was used for this work.

B.2.2 Speaker Dependent Training Data

The speaker dependent training data used is made up of two parts. The first part consists of the data from the long-term speaker dependent section and part of the short-term speaker dependent section of the WSJ0 database. The long-term speaker dependent section consists of about 1800 sentences from each of a set of 3 speakers. About 600 sentences per speaker from the short-term speaker dependent section of the database have been added to these sets to make up a total of 2400 training sentences for each of these 3 speakers. There were also another 9 speakers with 600 sentences per speaker available in the WSJ0 short-term speaker dependent section which were not used in the current work.

The second part of the speaker dependent training data consists of the so called long-term speaker independent section of the WSJ1 database which includes 25 speakers with around 1200 sentences per speaker. This makes the total number of speakers used for speaker dependent model training equal to 28, with three of them having about twice the amount of training data compared to the others.

It should be noted that the actual number of sentences used for training of the above models was slightly less than the numbers quoted above as about 20 sentences for each speaker were taken away from his/her set of training sentences to be used for testing purposes. Also, in the case of WSJ0 speakers, due to the large size of the vocabulary, a number of sentences from each speaker which included words out of the 1994 65k test vocabulary were also removed from the training set. This leaves more than 2300 sentences per speaker for the 3 WSJ0 speakers and about 1180 sentences per speaker for the 25 WSJ1 ones.

B.2.3 Adaptation and Test Data

For adaptation and test purposes, two parts of this database were used which are:

- **1994 NAB Spoke S3 Non-native Speakers** This consists of about 20 test sentences for each of the 10 evaluation test or 11 development test non-native speakers of American English.
- **WSJ1 1993 H2 5K SI Test Material** This consists of about 20 test sentences for each of the 10 speakers.

Note that for all the speakers mentioned above, 40 adaptation sentences are also available which are used for adaptation purposes.

B.3 Parametrisation of Speech Data

The parametrisation of the speech data is carried out using the HTK tool HCode [112]. In all cases, data is parametrised to 12 Mel Frequency Cepstral Coefficients (MFCC) plus a log energy coefficient. For this, first the speech waveform, $s(n)$, is pre-emphasised by a first-order difference with factor of $a = 0.97$, i.e.

$$\tilde{s}(n) = s(n) - as(n-1). \tag{B.1}$$

A Hamming window with a duration of 25.0 msec is then applied to the speech waveform every 10.0 msec. The window has a typical form of

$$w(n) = 0.54 - 0.46\cos(\frac{2\pi n}{N-1}), \qquad 0 \leq n \leq N-1 \tag{B.2}$$

where N is the number of samples per frame. A set of 24 triangular mel-filters are used in the next step, where the Mel-scale used is

$$\mathrm{Mel}(f) = 2595 \log_{10}(1 + \frac{f}{700}) \tag{B.3}$$

and each frame is coded into 12 cepstral coefficients using the following discrete cosine transform

$$c_i = \sum_{j=1}^{P} m_j \cos\left[\frac{\pi i}{P}(j - 0.5)\right], \qquad 1 \leq i \leq N \tag{B.4}$$

where P is the order of analysis and N is the number of output parameters. The log energy coefficient is also added to this set of frame cepstral coefficients.

A cepstral-liftering with a coefficient of $L = 22$ is then applied to the cepstral and log energy coefficients. The liftering function is defined as

$$l(n) = 1 + \frac{L}{2}\sin(\frac{\pi n}{L}). \tag{B.5}$$

The first and second differential coefficients are later added to this set to make a 39 element vector of coefficients.

Bibliography

[1] S.M. Ahadi and P.C. Woodland. Rapid Speaker Adaptation Using Model Prediction. In *Proc. IEEE Int. Conf. Acoust. Speech, Signal Processing*, volume 1, Detroit, May 1995.

[2] B.S. Atal. Automatic Recognition of Speakers from Their Voices. *Proc. IEEE*, 64(4):460–475, 1976.

[3] J.K. Baker. The DRAGON System - An Overview. *IEEE Trans. Acoust., Speech, Signal Processing*, ASSP-23(1):24–29, February 1975.

[4] L.E. Baum. An Inequality and Associated Maximization Technique in Statistical Estimation for Probabilistic Functions of Markov Processes. *Inequalities*, 3:1–8, 1972.

[5] 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. *Bull. Amer. Meteorol. Soc.*, 73:360–363, 1967.

[6] L.E. Baum, T. Petrie, G. Soules, and N. Weiss. A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains. *The Annals of Mathematical Statistics*, 41(1):164–171, 1970.

[7] J.R. Bellegarda, P.V. DeSouza, A. Nadas, D. Nahamoo, M.A. Picheny, and L.R. Bahl. The Metamorphic Algorithm: A Speaker Mapping Approach to Data Augmentation. *IEEE Trans. Speech and Audio proc.*, 2(3):413–420, July 1994.

[8] J.R. Bellegarda, P.V. DeSouza, A.J. Nadas, D. Nahamoo, M.A. Picheny, and L.R. Bahl. Robust Speaker Adaptation Using A Piecewise Linear Acoustic Mapping. In *Proc. IEEE Int. Conf. Acoust. Speech, Signal Processing*, volume 1, pages 445–448, 1992.

[9] J.O. Berger. *Statistical Decision Theory and Bayesian Analysis*. Springer-Verlag, New York, 2nd edition, 1985.

[10] P.F. Brown, C-H. Lee, and J.C. Hooper. Bayesian Adaptation in Speech Recognition. In *Proc. IEEE Int. Conf. Acoust. Speech, Signal Processing*, pages 761–764, 1983.

[11] C. Chatfield. *Statistics for Technology.* Chapman and Hall, London, 3rd edition, 1983.

[12] S. Chatterjee and B. Price. *Regression Analysis by Example.* John Wiley and Sons, New York, 2nd edition, 1991.

[13] H.C. Choi and R.W. King. Direct and Joint-Space Approaches to the Use of Spectral Transformation for Speaker Adaptation in Continuous Speech Recognition. In *Proc. Eurospeech*, pages 1151–1154, Madrid, September 1995.

[14] F. Class, A. Kaltenmeier, P. Regel, and K. Trottler. Fast Speaker Adaptation for Speech Recognition Systems. In *Proc. IEEE Int. Conf. Acoust. Speech, Signal Processing*, volume 1, pages 133–136, 1990.

[15] F. Class, A. Kaltenmeier, P. Regel-Brietzmann, and K. Trottler. Fast Speaker Adaptation Combined with Soft Vector Quantization in An HMM Speech Recognition System. In *Proc. IEEE Int. Conf. Acoust. Speech, Signal Processing*, volume 1, pages 461–464, 1992.

[16] S.J. Cox. Speaker Adaptation in Speech Recognition using Linear Regression Techniques. *Electronic Letters*, 28(22):2093–2094, October 1992.

[17] S.J. Cox. Speaker Adaptation Using A Predictive Model. In *Proc. Eurospeech*, volume 3, pages 2283–2286, September 1993.

[18] S.J. Cox. A Speaker Adaptation Technique Using Linear Regression. In *Proc. IEEE Int. Conf. Acoust. Speech, Signal Processing*, volume 1, pages 700–703, Detroit, May 1995.

[19] S.J. Cox. Predictive Speaker Adaptation in Speech Recognition. *Computer Speech and Language*, 9:1–17, 1995.

[20] S.J. Cox and J.S. Bridle. Unsupervised Speaker Adaptation by Probabilistic Spectrum Fitting. In *Proc. IEEE Int. Conf. Acoust. Speech, Signal Processing*, volume 1, pages 294–297, Glasgow, May 1989.

[21] S.J. Cox and J.S. Bridle. Simultaneous Speaker Normalisation and Utterance Labelling Using Bayesian/Neural Net Techniques. In *Proc. IEEE Int. Conf. Acoust. Speech, Signal Processing*, volume 1, pages 161–164, 1990.

[22] S.B. Davis and P. Mermelstein. Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences. *IEEE Trans. Acoust., Speech, Signal Processing*, ASSP-28(4):357–366, August 1980.

[23] M.H. DeGroot. *Optimal Statistical Decisions.* McGraw-Hill, New York, 1970.

[24] 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, Ser. B:1–38, 1977.

[25] V. Digalakis and L. Neumeyer. Speaker Adaptation Using Combined Transformation and Bayesian Methods. In *Proc. IEEE Int. Conf. Acoust. Speech, Signal Processing*, volume 1, pages 680–683, Detroit, May 1995.

[26] V.V. Digalakis, D. Rtischev, and L.G. Neumeyer. Speaker Adaptation Using Constrained Estimation of Gaussian Mixtures. *IEEE Trans. Speech and Audio proc.*, 3(5):357–366, September 1995.

[27] R.O. Duda and P.E. Hart. *Pattern Classification and Scene Analysis.* John Wiley and Sons, New York, 1973.

[28] O.J. Dunn and V.A. Clark. *Applied Statistics: Analysis of Variance and Regression.* John Wiley and Sons, New York, 1987.

[29] B.S. Everitt. *Cluster Analysis.* Edward Arnold, London, 3rd edition, 1993.

[30] F.Fallside and W.A. Woods. *Computer Speech Processing.* Prentice-Hall, London, 1985.

[31] K. Fukunaga. *Introduction to Statistical Pattern Recognition.* Academic Press, New York, 1972.

[32] S. Furui. A Training Procedure for Isolated Word Recognition Systems. *IEEE Trans. Acoust., Speech, Signal Processing*, ASSP-28(2):129–136, April 1980.

[33] S. Furui. Speaker-Independent Isolated Word Recognition Using Dynamic Features of Speech Spectrum. *IEEE Trans. Acoust., Speech, Signal Processing*, ASSP-34(1):52–59, February 1986.

[34] J-L. Gauvain and C-H. Lee. Bayesian Learning of Gaussian Mixture Densities for Hidden Markov Models. In *Proc. DARPA Speech and Natural Language Workshop*, February 1991.

[35] J-L. Gauvain and C-H. Lee. Bayesian Learning for Hidden Markov Model with Gaussian Mixture Observation Densities. *Speech Communication*, 11:205–213, 1992.

[36] J-L. Gauvain and C-H. Lee. Maximum *A Posteriori* Estimation for Multivariate Gaussian Mixture Observations of Markov Chains. *IEEE Trans. Speech and Audio Proc.*, SAP-2(2):291–298, April 1994.

[37] H. Gish, M-H Siu, and R. Rohlicek. Segregation of Speakers for Speech Recognition and Speaker Identification. In *Proc. IEEE Int. Conf. Acoust. Speech, Signal Processing*, volume 1, pages 873–876, 1991.

[38] Y. Gong, O. Siohan, and J-P Haton. Minimization of Speech Alignment Error by Iterative Transformation for Speaker Adaptation. In *Proc. Int. Conf. Spoken Lang. Processing*, pages 377–380, Alberta, 1992.

[39] Y. Hao and D. Fang. Speech Recognition Using Speaker Adaptation by System Parameter Transformation. *IEEE Trans. Speech and Audio proc.*, 2(1):63–68, January 1994.

[40] H. Hattori and S. Sagayama. Vector Field Smoothing Principle for Speaker Adaptation. In *Proc. Int. Conf. Spoken Lang. Processing*, pages 381–384, Alberta, 1992.

[41] A.J. Hewett. *Training and Speaker Adaptation in Template-Based Speech Recognition.* PhD thesis, Cambridge University Engineering Department, 1989.

[42] X. Huang, F. Alleva, H-W. Hon, M-Y. Hwang, K-F. Lee, and R. Rosenfeld. The SPHINX-II Speech Recognition System: An Overview. *Computer Speech and Language*, 7(2):137–148, April 1993.

[43] X.D. Huang. A study on Speaker-Adaptive Speech Recognition. In *Proc. DARPA Speech and Natural Language Workshop*, pages 278–283, February 1991.

[44] X.D. Huang. Speaker Normalization for Speech Recognition. In *Proc. IEEE Int. Conf. Acoust. Speech, Signal Processing*, volume 1, pages 465–468, 1992.

[45] X.D. Huang, F. Alleva, S. Hayamizu, H-W. Hon, M-Y Hwang, and K-F Lee. Improved Hidden Markov Modeling for Speaker-Independent Continuous Speech Recognition. In *Proc. DARPA Speech and Natural Language Workshop*, pages 327–331, June 1990.

[46] X.D. Huang, Y. Ariki, and M.A. Jack. *Hidden Markov Models for Speech Recognition.* Edinburgh University Press, Edinburgh, 1990.

[47] X.D. Huang and M.A. Jack. Semi-continuous Hidden Markov Models for Speech Signals. *Computer Speech and Language*, 3:239–251, 1989.

[48] Q. Huo and C. Chan. Bayesian Adaptive Learning of the Parameters of Hidden Markov Model for Speech Recognition. Technical report, Department of Computer Science, University of Hong Kong, September 1992.

[49] Q. Huo, C. Chan, and C-H. Lee. Bayesian Learning of the SCHMM Parameters for Speech Recognition. In *Proc. IEEE Int. Conf. Acoust. Speech, Signal Processing*, volume 1, pages 221–224, 1994.

[50] A. Imamura. Speaker-Adaptive HMM-Based Speech Recognition with A Stochastic Speaker Classifier. In *Proc. IEEE Int. Conf. Acoust. Speech, Signal Processing*, pages 841–844, 1991.

[51] J. Jaschul. Speaker Adaptation by a Linear Transformation with Optimised Parameters. In *Proc. IEEE Int. Conf. Acoust. Speech, Signal Processing*, volume 3, pages 1657–1660, Paris, 1982.

[52] F. Jelinek. Continuous Speech Recognition by Statistical Methods. *Proc. IEEE*, 64(4):532–556, April 1976.

[53] B-H. Juang. Maximum-Likelihood Estimation for Mixture Multivariate Stochastic Observations of Markov Chains. *AT&T Technical Journal*, 64(6):1235–1249, July-August 1985.

[54] B-H. Juang, S.E. Levinson, and M.M. Sondhi. Maximum Likelihood Estimation for Multivariate Mixture Observations of Markov Chains. *IEEE Trans. Inform. Theory*, IT-32(2):307–309, March 1986.

[55] B-H. Juang and L.R. Rabiner. A Probabilistic Distance Measure for Hidden Markov Models. *AT&T Technical Journal*, 64(2):391–408, 1985.

[56] T. Kosaka and S. Sagayama. Tree-Structured Speaker Clustering for Fast Speaker Adaptation. In *Proc. IEEE Int. Conf. Acoust. Speech, Signal Processing*, volume 1, pages 245–248, 1994.

[57] F. Kubala. Design of the 1994 CSR Benchmark Tests. In *Proc. ARPA Spoken Language Technology Workshop*, Barton Creek, 1995. Morgan Kaufmann.

[58] F. Kubala, R. Schwartz, and C. Barry. Speaker Adaptation From A Speaker-Independent Training Corpus. In *Proc. IEEE Int. Conf. Acoust. Speech, Signal Processing*, volume 1, pages 137–140, 1990.

[59] M.J. Lasry and R.M. Stern. *A Posteriori* Estimation of Correlated Jointly Gaussian Mean Vectors. *IEEE Trans. Pattern Anal. Machine Intell.*, PAMI-6(4):530–535, July 1984.

[60] C-H. Lee and J-L. Gauvain. A Study on Speaker Adaptation for Continuous Speech Recognition. In *Proc. ARPA Cont. Speech Rec. Workshop*, pages 59–64, Stanford, September 1992.

[61] C-H. Lee, E. Giachin, L.R. Rabiner, R. Pieraccini, and A.E. Rosenberg. Improved Acoustic Modeling for Large Vocabulary Continuous Speech Recognition. *Computer Speech and Language*, 6:103–127, 1992.

[62] C-H. Lee, C-H. Lin, and B-H. Juang. A Study on Speaker Adaptation of the Parameters of Continuous Density Hidden Markov Models. *IEEE Trans. Sig. Proc.*, SP-39(4):806–814, April 1991.

[63] K-F. Lee. *Automatic Speech Recognition: The Development of the SPHINX System.* Kluwer Academic Publishers, Boston, 1989.

[64] K-F. Lee. Context-Dependent Phonetic Hidden Markov Models for Speaker-Independent Continuous Speech Recognition. *IEEE Trans. Acoust., Speech, Signal Processing*, 38(4):599–609, April 1990.

[65] C.J. Leggetter. *Improved Acoustic Modelling for HMMs using Linear Transformations.* PhD thesis, Cambridge University Engineering Department, 1995.

[66] C.J. Leggetter and P.C. Woodland. Speaker Adaptation of Continuous Density HMMs Using Linear Regression. In *Proc. Int. Conf. Spoken Lang. Processing*, Yokohama, 1994.

[67] C.J. Leggetter and P.C. Woodland. Flexible Speaker Adaptation for Large Vocabulary Speech Recognition. In *Proc. Eurospeech*, volume 2, pages 1155–1158, Madrid, September 1995.

[68] C.J. Leggetter and P.C. Woodland. Flexible Speaker Adaptation using Maximum Likelihood Linear Regression. In *Proc. ARPA Spoken Language Technology Workshop*, Barton Creek, 1995.

[69] C.J. Leggetter and P.C. Woodland. Maximum Likelihood Linear Regression for Speaker Adaptation of Continuous Density Hidden Markov Models. *Computer Speech and Language*, 9(2):171–185, April 1995.

[70] L.A. Liporace. Maximum Likelihood Estimation for Multivariate Observations of Markov Sources. *IEEE Trans. Inform. Theory*, IT-28(5):729–734, September 1982.

[71] J.S. Maritz and T. Lwin. *Empirical Bayes Methods.* Chapman and Hall, 2nd edition, 1989.

[72] L. Mathan and L. Miclet. Speaker Hierarchical Clustering for Improving Speaker-Independent HMM Word Recognition. In *Proc. IEEE Int. Conf. Acoust. Speech, Signal Processing*, pages 149–152, 1990.

[73] T. Matsuoka and K. Shikano. Speaker Adaptation by Modifying Mixture Coefficients of Speaker-Independent Mixture Gaussian HMMs. In *Proc. Int. Conf. Spoken Lang. Processing*, pages 373–376, Alberta, 1992.

[74] J. Nakahashi and E. Tsuboka. Speaker Adaptation Based on Fuzzy Vector Quantization. In *Proc. Int. Conf. Spoken Lang. Processing*, pages 467–470, Yokohama, 1994.

[75] S. Nakamura and K. Shikano. A Comparative Study of Spectral Mapping for Speaker Adaptation. In *Proc. IEEE Int. Conf. Acoust. Speech, Signal Processing*, volume 1, pages 157–160, 1990.

[76] B.F. Necioglu, M. Ostendorf, and J.R. Rohlicek. A Bayesian Approach to Speaker Adaptation For the Stochastic Segment Model. In *Proc. IEEE Int. Conf. Acoust. Speech, Signal Processing*, volume 1, pages 437–440, 1992.

[77] L. Neumeyer, A. Sankar, and V. Digalakis. A comparative Study of Speaker Adaptation Techniques. In *Proc. Eurospeech*, pages 1127–1130, Madrid, September 1995.

[78] H. Ney. Experiments on Mixture-Density Phoneme-Modelling for the Speaker-Independent 1000-Word Speech Recognition DARPA Task. In *Proc. IEEE Int. Conf. Acoust. Speech, Signal Processing*, volume 2, pages 713–716, 1990.

[79] F. Nolan. *The Phonetic Bases of Speaker Recognition.* Cambridge University Press, Cambridge, 1983.

[80] K. Ohkura, H. Ohnishi, and M. Iida. Speaker Adaptation Based on Transfer Vectors of Multiple Reference Speakers. In *Proc. Int. Conf. Spoken Lang. Processing*, pages 455–458, Yokohama, 1994.

[81] K. Ohkura, M. Sugiyama, and S. Sagayama. Speaker Adaptation Based on Transfer Vector Field Smoothing with Continuous Mixture Density HMMs. In *Proc. Int. Conf. Spoken Lang. Processing*, pages 369–372, Alberta, 1992.

[82] Y. Ono, H. Wakita, and Y. Zhao. Speaker Normalisation Using Constrained Spectra Shifts in Auditory Filter Domain. In *Proc. Eurospeech*, volume 1, pages 355–358, Berlin, 1993.

[83] D.B. Paul and J.M. Baker. The Design of the Wall Street Journal-Based CSR Corpus. In *Proc. DARPA Speech and Natural Language Workshop*, pages 357–362, February 1992.

[84] P. Price, W.M. Fisher, J. Bernstein, and D.S. Pallett. The DARPA Resource Management Database for Continuous Speech Recognition. In *Proc. IEEE Int. Conf. Acoust. Speech, Signal Processing*, volume 1, pages 651–654, 1988.

[85] L.R. Rabiner. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. In A. Waibel and K-F. Lee, editors, *Readings in Speech Recognition*, pages 267–296. Morgan Kaufmann, San Mateo, 1990.

[86] L.R. Rabiner and B-H. Juang. *Fundamentals of Speech Recognition.* Prentice Hall, New Jersey, 1993.

[87] L.R. Rabiner, B-H. Juang, S.E. Levinson, and M.M. Sondhi. Recognition of Isolated Digits Using Hidden Markov Models with Continuous Mixture Densities. *AT&T Technical Journal*, 64(6):1211–1234, July-August 1985.

[88] L.R. Rabiner and J.G. Wilpon. Some Performance Benchmarks for Isolated Word Speech Recognition Systems. *Computer Speech and Language*, 2:343–357, 1987.

[89] R.A. Redner and H.F. Walker. Mixture Densities, Maximum likelihood and the EM Algorithm. *SIAM Review*, 26(2):195–239, April 1984.

[90] H. Robbins. The Empirical Bayes Approach to Statistical Decision Problems. *Annals Math. Stat.*, 35(1):1–20, March 1964.

[91] W.A. Rozzi and R.M. Stern. Speaker Adaptation in Continuous Speech Recognition via Estimation of Correlated Mean Vectors. In *Proc. IEEE Int. Conf. Acoust. Speech, Signal Processing*, volume 1, pages 865–868, 1991.

[92] R. Schwartz, Y. Chow, O. Kimball, S. Roucos, M. Krasner, and J. Makhoul. Context-Dependent Modeling for Acoustic-Phonetic Recognition of Continuous Speech. In *Proc. IEEE Int. Conf. Acoust. Speech, Signal Processing*, April 1985.

[93] K. Shinoda, K-I. Iso, and T. Watanabe. Speaker Adaptation for Demi-Syllable Based Continuous Density HMM. In *Proc. IEEE Int. Conf. Acoust. Speech, Signal Processing*, pages 857–860, 1991.

[94] K. Shinoda and T. Watanabe. Speaker Adaptation with Autonomous Control Using Tree Structure. In *Proc. Eurospeech*, pages 1143–1146, Madrid, September 1995.

[95] F.K. Soong and A.E. Rosenberg. On the Use of Instantaneous and Transitional Spectral Information in Speech Recognition. *IEEE Trans. Acoust., Speech, Signal Processing*, ASSP-36(6):871–879, June 1988.

[96] R.M. Stern and M.J. Lasry. Dynamic Speaker Adaptation for Feature-Based Isolated Word Recognition. *IEEE Trans. Acoust., Speech, Signal Processing*, ASSP-35(6):751–762, June 1987.

[97] C.W. Therrien. *Decision, Estimation, and Classification: An Introduction to Pattern Recognition and Related Topics*. John Wiley & Sons, New York, 1989.

[98] M. Tonomura, T. Kosaka, S. Matsunaga, and A. Monden. Speaker Adaptation Fitting Training Data Size and Contents. In *Proc. Eurospeech*, pages 1147–1150, Madrid, September 1995.

[99] C. Tuerk and T. Robinson. A New Frequency Shift Function for Reducing Inter-Speaker Variance. In *Proc. Eurospeech*, volume 1, pages 351–354, Berlin, 1993.

[100] A.J. Viterbi. Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm. *IEEE Trans. Inform. Theory*, IT-13(2):260–269, April 1967.

[101] J.C. Wells. *Accents of English*. Cambridge University Press, Cambridge, 1982.

[102] J.G. Wilpon, C-H. Lee, and L.R. Rabiner. Improvements in Connected Digit Recognition Using Higher Order Spectral and Energy Features. In *Proc. IEEE Int. Conf. Acoust. Speech, Signal Processing*, volume 1, pages 349–352, 1991.

[103] P.C. Woodland, C.J. Leggetter, J.J. Odell, V. Valtchev, and S.J. Young. The 1994 HTK Large Vocabulary Speech Recognition System. In *Proc. IEEE Int. Conf. Acoust. Speech, Signal Processing*, volume 1, pages 73–76, Detroit, May 1995.

[104] P.C. Woodland, C.J. Leggetter, J.J. Odell, V. Valtchev, and S.J. Young. The Development of the 1994 HTK Large Vocabulary Speech Recognition System. In *Proc. ARPA Spoken Language Technology Workshop*, Barton Creek, 1995. Morgan Kaufmann.

[105] P.C. Woodland, J.J. Odell, V. Valtchev, and S.J. Young. Large Vocabulary Continuous Speech Recognition Using HTK. In *Proc. IEEE Int. Conf. Acoust. Speech, Signal Processing*, volume II, pages 125–128, Adelaide, 1994.

[106] P.C. Woodland and S.J. Young. Benchmark DARPA RM Results with the HTK Portable HMM Toolkit. In *Proc. DARPA Workshop*, Stanford, Sept. 1992.

[107] P.C. Woodland and S.J. Young. The HTK Tied-state Continuous Speech Recogniser. In *Proc. Eurospeech*, volume 3, pages 2207–2210, Berlin, September 1993.

[108] S.J. Young, J.J. Odell, and P.C. Woodland. Tree-Based State Tying for High Accuracy Acoustic Modelling. In *Proc. ARPA Human Language Technology Workshop*, pages 307–312, Plainsboro, March 1994. Morgan Kaufmann.

[109] S.J. Young, N.H. Russel, and J.H.S. Thornton. Token Passing: a Conceptual Model for Connected Speech Recognition Systems. Technical Report F_INFENG/TR38, Cambridge University Engineering Department, 1989.

[110] S.J. Young and P.C. Woodland. The Use of State Tying in Continuous Speech Recognition. In *Proc. Eurospeech*, volume 3, pages 2203–2206, Berlin, September 1993.

[111] S.J. Young and P.C. Woodland. State Clustering in Hidden Markov Model-Based Continuous Speech Recognition. *Computer Speech and Language*, 8:369–383, 1994.

[112] S.J. Young, P.C. Woodland, and W.J. Byrne. *HTK: Hidden Markov Model Toolkit V1.5*. Cambridge University Engineering Department & Entropic Research Laboratories Inc., September 1993.

[113] G. Zavaliagkos, R. Schwartz, and J. Makhoul. Adaptation Algorithms for BBN's Phonetically Tied Mixture System. In *Proc. ARPA Spoken Language Technology Workshop*, Barton Creek, January 1995. Morgan Kaufmann.

[114] Y. Zhao. A New Speaker Adaptation Technique Using Very Short Calibration Speech. In *Proc. IEEE Int. Conf. Acoust. Speech, Signal Processing*, volume 2, pages 562–565, 1993.

[115] Y. Zhao. An Acoustic-Phonetic-Based Speaker Adaptation Technique for Improving Speaker-Independent Continuous Speech Recognition. *IEEE Trans. Speech and Audio proc.*, 2(3):380–394, July 1994.