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 footnotes and appendices is approximately 52,000 words.

Summary

Hidden Markov models (HMMs) have been used successfully for speech recognition for many years. However, in some respects the assumptions behind HMM models are poor. HMMs model only the within-class data and no attempt is made at discriminating between classes. This is a problem, especially in speaker independent systems where a wide variety of speakers may be used. This thesis considers the problem of acoustic modelling in speaker independent systems in two ways: (a) by incorporating discrimination into the HMM framework; and, (b) by adapting the HMMs to the chosen speaker (speaker adaptation). In both cases, linear transformation methods are proposed which aim to tune the model parameters on a class-specific basis to improve the modelling. Particular emphasis is placed on applications in large vocabulary continuous speech.

The acoustic-class discrimination problem is addressed at the HMM state level by considering the feature space representation of each class. Confusable class distributions are identified and class-specific mappings in the form of linear transforms are used to separate the within-class data from confusable data. The transforms reduce the dimensionality of the feature space so that those elements in the feature space which are confusable are discarded. Two methods of identifying confusable distributions are considered, one data-driven using data from the training set, and the second based on the distances between class distributions.

For speaker adaptation, a new approach using transformations, termed maximum likelihood linear regression (MLLR), is derived. Transforms are associated with each component distribution within the HMM system and estimated using a maximum likelihood approach similar to standard HMM parameter estimation. The transforms capture the general speaker characteristics between the current system parameters and the new speaker. A flexible form of tying of transforms is derived to make efficient use of the available data, allowing adaptation on small amounts of example speech. The method can be implemented in supervised or unsupervised adaptation modes and the flexible framework can be extended for incremental adaptation.

The discriminative and speaker adaptation transformations have been evaluated on a 1000 word task (Resource Management), and the speaker adaptation has also been evaluated on a larger 5000 word task (Wall Street Journal).

Keywords: Hidden Markov models, speech recognition, discrimination, feature extraction, speaker adaptation, maximum likelihood linear regression.

Acknowledgements

First, I must thank the Engineering and Physical Sciences Research Council, for funding my research. I would also like to thank the Engineering Department, Emmanuel College and the Royal Academy of Engineering for subsidising my conference visit to Japan.

I would like to thank everyone who has helped me during the course of my studies. It has been a great privilege to be a member of the Cambridge University Speech, Vision and Robotics group. The camaraderie, advice and encouragement of people in the group have made my studying that much more enjoyable. Needless to say, none of this work could have been completed without the excellent computing facilities that have been maintained by various members of the group. Many thanks go to Richard, Patrick, Andy, Carl and whoever else has been involved over the years in the smooth running of the network.

Thanks to Steve Young, Phil Woodland and Julian Odell for providing the outstanding HTK toolkit and assorted software and model sets which saved me much time and effort.

Special thanks go to those people who have resided in the Speech Lab over the years and made it such a wonderful working atmosphere. In particular I would like to thank Valtcho for keeping me company right to the end. I must also thank those poor people who had to proof read my thesis - Kate, Rob, Kev, Mark and Graham.

I must single out Matthew Jones for very special thanks, for many things, but most of all for just being himself.

Finally, I save my biggest thank you for my supervisor Phil Woodland. Throughout my time in Cambridge he has been a constant source of inspiration. His ideas, wisdom, and (occasional) wit, but most of all his dedication to the cause have been an example to me and many others, and I will miss our regular meetings.

Contents

1 Introduction **1**

1.1 Large Vocabulary Speech Recognition 2

1.2 The Acoustic Modelling Problem 3

1.3 Speaker Independent and Speaker Dependent Modelling 3

1.4 Transformations Within HMMs 4

1.5 Organisation of Thesis 5

2 Hidden Markov Models for Speech Recognition **6**

2.1 Parameterisation of Speech 6

2.2 Hidden Markov Models 7

2.2.1 HMM Representation 8

2.2.2 Assumptions 9

2.3 Pattern Matching with HMMs 10

2.4 Estimation of the HMM parameters 11

2.4.1 Maximum Likelihood Estimation 11

2.4.2 MLE Parameter Estimation 12

2.5 Recognition with HMMs 13

2.6 Whole Word and Sub-word Modelling 14

2.7 Parameter Tying 15

2.8 HMMs for Continuous Speech 15

2.9 Grammars and Language Models 16

3 Discriminative Methods for HMMs **18**

3.1 The Discrimination Problem 18

3.2 Discriminitive Training Methods 19

3.2.1 Maximum Mutual Information 19

3.2.2 Minimum Discrimination Information 20

3.2.3 Optimisation of Error Rates 21
3.3 Discriminitive Feature Vectors 22
3.4 Class Separability Measures 23
3.5 Feature Selection 24
3.5.1 Use of Recognition Rate 24
3.5.2 Use of the Fisher ratio 24
3.5.3 Discriminative Feature Selection 25
3.5.4 Comparison of Feature Selection Methods 26
3.6 Feature Extraction 27
3.6.1 Principal Components Analysis 28
3.6.2 Linear Discriminant Analysis 29
3.6.3 The Use of LDA 31
3.7 Class-Specific Feature Vectors 32
3.7.1 Word Adaptive LDA 32
3.7.2 Individual Class Discriminants 33
3.7.3 Confusion Discriminants in HMMs 34
3.8 Summary 36

4 Transformations for Discriminative Feature Selection **38**
4.1 Model Sets for Evaluation 38
4.1.1 Baseline Results for R_{full} and R_{grand} 39
4.2 A Global Feature Selection Method 39
4.3 Global Feature Space Selection Using LDA 41
4.3.1 Analysis of Global LDA Assumptions 41
4.3.2 Global LDA on RM Models 42
4.4 State-Based Feature Space Selection Using Confusion Discriminant Analysis (CDA) 44
4.4.1 Derivation 44
4.4.2 Computational Considerations 45
4.4.3 Relationship to MMIE 46
4.4.4 Relationship to LDA 47
4.5 Applying CDA to Continuous Speech Tasks 47
4.5.1 Confusion Gathering Approach for Isolated Words 47
4.5.2 Problems of Extending to Continuous Speech Tasks 48
4.5.3 Simulating Isolated Confusions for Continuous Speech 49
4.6 State-Based CDA using Global Distributions 51

4.7 Summary 53

5 Evaluation of Confusion Discriminant Analysis Transforms 54

5.1 State-Specific Confusion Distributions for CDA 54
5.2 Data-Driven Confusions 55
5.2.1 Identifying Confusions 56
5.3 Data-Driven Confusion Methods 57
5.3.1 N-Best Recognition Confusions 57
5.3.2 Frame-by-Frame Recognition Confusions 58
5.3.3 Restricted Frame-by-Frame Recognition Confusions 59
5.3.4 Summary of Data-Driven Confusion Methods 61
5.4 Evaluation of Data-Driven Confusions 61
5.4.1 Problems with confusion gathering 62
5.4.2 Selecting Robust Confusion Estimates 64
5.4.3 Recognition Confusion Transform Results 65
5.4.4 Use of Thresholds with N-Best 66
5.4.5 Discussion of Results 67
5.5 Confusions from Model Parameters 68
5.6 Evaluation of PDF-based Confusions 70
5.6.1 Threshold for Confusions 71
5.6.2 Weights for Combination 72
5.6.3 Results 72
5.7 Comparison between Methods 74
5.7.1 Comparison with LDA Methods 74
5.7.2 CDA Confusions on Different Test Sets 75
5.8 Analysis of Discriminative Feature Selection 76
5.8.1 Improving PDF-based Confusions 77
5.8.2 Improving Data-Driven Confusions 77
5.8.3 Application to Component Mixture Distributions 77
5.9 Summary of Discriminative Feature Methods 78

6 Speaker Adaptation Techniques 80

6.1 Introduction to Speaker Adaptation 80
6.1.1 Speaker Variation 81
6.1.1.1 Linguistic Differences 81
6.1.1.2 Physiological Differences 82

6.1.2 Speaker Normalisation 82
6.1.3 Speaker Adaptation 83
6.2 Modes of Speaker Adaptation 84
6.3 Speaker Clustering for Adaptation 84
6.4 Spectral Transformation Approaches 85
6.5 Model Parameter Adaptation Approaches 89
6.6 Approaches Combining Parameter and Spectral Adaptation 91
6.7 Summary 91

7 A Transformation Approach to Speaker Adaptation 93

7.1 Maximum Likelihood Linear Regression 93
7.2 Estimation of MLLR Regression Matrices 94
7.2.1 Maximisation of Auxiliary Function 95
7.2.2 Re-estimation Formula for Tied Regression Matrices 96
7.2.3 Extension to Mixture Distributions 97
7.2.4 Multiple Observation Sequences 98
7.3 Incremental Adaptation 99
7.4 Computational Considerations 100
7.5 Diagonal Regression Matrices 100
7.6 Regression Classes 101
7.7 Adapting Other Parameters with MLLR 102
7.8 Alignment of Data 104
7.9 MLLR Matrices as Input Transformations 104
7.10 Comparison to Least Squares Regression 106
7.11 Summary 107

8 Evaluation of MLLR Transformations for Speaker Adaptation 109

8.1 Model Sets used for Adaptation Evaluation 109
8.1.1 Models for RM experiments. 109
8.1.1.1 Word Internal Triphones 110
8.1.1.2 Cross-Word Triphones - R_{cwt} 110
8.1.2 Wall Street Journal Models 111
8.1.2.1 Gender Independent Models - W_{a} 111
8.1.2.2 Gender Dependent Models W_{b} 111
8.1.2.3 WSJ Language Models 111
8.2 Implementation Issues 111

8.2.1 Alignment of the Data . . . 112
8.2.2 Generation of Class Definitions . . . 112
8.3 Class Definitions . . . 114
8.3.1 Fixed Class Definitions . . . 114
8.3.2 Regression Class Trees . . . 114
8.3.2.1 Traversing the Regression Class Tree . . . 115
8.3.2.2 Regression Class Tree Generation . . . 116
8.4 Initial MLLR Adaptation Experiments . . . 116
8.4.1 A Global MLLR Transform . . . 117
8.4.2 Application to Multiple Component Mixture Distributions . . . 118
8.4.3 Effect of Increasing the Number of Regression Classes . . . 118
8.4.4 Iterative Adaptation . . . 119
8.4.5 Adaptation of Specific Feature Elements . . . 120
8.5 Comparison with MAP adaptation . . . 121
8.6 MLLR with High Accuracy Models . . . 122
8.6.1 Very Small Amounts of Adaptation Data . . . 123
8.6.2 Varying the Number of Regression Classes . . . 124
8.6.3 Diagonal Regression Matrices . . . 124
8.6.4 Alignment Strategies . . . 126
8.7 Supervised vs. Unsupervised Adaptation . . . 127
8.8 Evaluation of Regression Trees . . . 129
8.9 Incremental Adaptation . . . 129
8.9.1 Evaluation of Incremental Adaptation . . . 130
8.10 Static Adaptation of Poor Speakers . . . 132
8.10.1 Non-Native Speakers . . . 133
8.10.2 Adaptation for Non-Native Speakers . . . 134
8.11 Incorporating MLLR into SI Recognisers . . . 134
8.11.1 The 1994 ARPA SI Evaluation . . . 135
8.12 Discussion . . . 136
8.13 Summary . . . 137

9 Conclusions **139**
9.1 Review of Work . . . 139
9.1.1 Discriminative Feature Transforms . . . 139
9.1.2 Transforms for Speaker Adaptation . . . 141

9.1.3 Summary 142

9.2 Ideas for Future Work 142

9.2.1 Discriminative Feature Transforms 142

9.2.2 Speaker Adaptation Transforms 143

9.3 Summary 144

A Databases **145**

A.1 Parameterisation of Databases 145

A.2 The TIMIT Database 145

A.3 The Resource Management Database 146

A.3.1 Speaker Independent RM Database 146

A.3.2 Speaker Dependent RM Database 146

A.3.3 RM Word-Pair Language Model 146

A.3.4 Function Words 147

A.4 The Wall Street Journal Database 147

A.4.1 WSJ Training Database Description 147

A.4.2 The 1994 NAB Database 148

B Distance Measures **149**

B.1 Derivation of Closed Form of Divergence Measures 149

B.2 Likelihood Measure 150

C Broad Phonetic Classes **152**

List of Figures

1.1 Schematic diagram for a large vocabulary speech recogniser 2

2.1 An example HMM. 8
2.2 Example of combining phone models into word models 14

3.1 Potential problem with PCA 29

4.1 Global feature selection using Bocchieri & Wilpon approach 40
4.2 Comparison of eigenvalues in grand and full covariance models 42
4.3 Results of LDA feature extraction 43
4.4 Effect of feature reduction using CDA with global measures 52

5.1 Illustration of the restricted Frame-by-Frame method. 60
5.2 Examination of robustness of distribution estimates 62
5.3 Interaction of default and confusion distributions 63
5.4 Comparison of recognition based confusion methods 65
5.5 N-Best confusions gathered using a fixed number of alternative hypotheses . 67
5.6 N-Best confusions gathered using likelihood thresholds 68
5.7 Number of confusable states at different divergence thresholds 71
5.8 Comparison of weights for PDF-based confusions 73
5.9 Comparison of different thresholds for PDF-based confusions 73
5.10 Comparison of the best confusion gathering approaches for CDA 74
5.11 Comparison of CDA confusion gathering and LDA approaches 75
5.12 CDA confusion on different test sets 76

6.1 Schematic diagram of speaker normalisation process 83
6.2 Schematic diagram of speaker adaptation process 84
6.3 Example of the canonical correlation approach 86
6.4 Example of spectral transformation approach 87

7.1 Example of assumption of similar relative acoustic distributions 102
7.2 Example regression class members in a 2-dimensional acoustic space 103
7.3 Alignment of the adaptation data, and allocation of data 105

8.1 Example of change in modality of distributions 113
8.2 A regression class tree 115
8.3 Supervised MLLR adaptation of cross-word triphones using global matrix . 123
8.4 Varying the number of regression classes for MLLR 125
8.5 Comparison of diagonal and full regression transforms 126
8.6 Unsupervised MLLR adaptation using a global transform 127
8.7 Varying the number of regression classes with the amount data 128
8.8 Comparison of unsupervised incremental adaptation error rates 131
8.9 Number of classes used for incremental adaptation. 132

List of Tables

3.1 Comparison of ordering of features using different feature selection methods 26
3.2 Woodland's E-set results with discriminant transform 36

4.1 Baseline results of models on RM speaker independent test sets 39
4.2 Comparison of eigenvalues for in-class distributions 41
4.3 Isolated confusions simulated on TIMIT using phone classification 50

5.1 Minimum number of confusions for robust estimates 65
5.2 Number of confusable frames identified using N-Best 66

6.1 Comparison of speaker dependent and speaker independent systems 81
6.2 Example MAP adaptation results 90
6.3 Incorporating prediction methods into MAP adaptation 91

8.1 Global MLLR adaptation of $R1_{\text{diag}}$ models 117
8.2 Changes in log likelihood per frame for each speaker on the test data 118
8.3 Effect of global MLLR transform on different model sets 119
8.4 Comparison of phone-base and distance-based regression classes 120
8.5 Iterative MLLR adaptation using $R2_{\text{diag}}$ models 121
8.6 MLLR adaptation of selected elements of the feature vector 121
8.7 Comparison of MLLR and MAP adaptation approaches 122
8.8 Global MLLR transform on cross-word triphones 124
8.9 Comparison of MLLR alignment strategies: Viterbi vs. forward-backward . 126
8.10 Best performance for different amounts of data using R_{cwt} models 128
8.11 Adaptation results using a regression class tree 129
8.12 Unsupervised incremental adaptation on WSJ NAB 1994 spoke S4 130
8.13 Error rates for individual speakers for Nov'94 S4. 132
8.14 Native and non-native recognition results 133
8.15 Supervised adaptation for non-native speakers 134

8.16 Results of 1994 ARPA CSR evaluation 135
8.17 Speaker breakdown of WSJ Nov'94 evaluation task H1 136

A.1 Words defined as function words in RM 147

C.1 Broad Phonetic Classes 152

Notation

The following notation has been used in this work.

$\boldsymbol{x}$	a vector (lower case bold)
$\boldsymbol{A}$	a matrix (upper case bold)
$\boldsymbol{x}'$	the vector transpose of $\boldsymbol{x}$
$\boldsymbol{A}^{-1}$	the inverse of matrix $\boldsymbol{A}$
$\lvert\boldsymbol{A}\rvert$	the determinant of matrix $\boldsymbol{A}$
$tr(\boldsymbol{A})$	the trace of matrix $\boldsymbol{A}$
$\hat{\boldsymbol{x}}$	an estimate of $\boldsymbol{x}$
$\boldsymbol{\mu}$	a Gaussian mean vector
$\boldsymbol{\Sigma}$	a Gaussian covariance matrix
λ	the parameters of a HMM model set
$\boldsymbol{\Theta}$	a set of model state sequences
$\boldsymbol{\theta}$	a single model state sequence
$\mathcal{F}(\boldsymbol{x})$	the likelihood of $\boldsymbol{x}$

Abbreviations

CDA	Confusion Discriminant Analysis
HMM	Hidden Markov Model
LDA	Linear Discriminant Analysis
MAP	Maximum *a posteriori*
MLE	Maximum Likelihood Estimation
MLLR	Maximum Likelihood Linear Regression
MMI	Maximum Mutual Information
PCA	Principal Components Analysis
RM	Resource Management Database
SI	Speaker Independent
SD	Speaker Dependent
WSJ	Wall Street Journal Database

Chapter 1

Introduction

Over the last twenty years much research has been performed in the field of speech recognition with the volume of new ideas expanding as rapidly as the power of computers. Although a perfect speech recogniser is still a distant dream, the technology available today has reached a level where applications may be considered. There is, however, still much research to be performed in many different areas to improve the recognition systems available. Many current systems are designed for very limited tasks in controlled environments, and perform poorly when extended to more general situations. These problems of robustness must be addressed before general real world applications can be considered.

A speech recogniser is a device which takes an utterance of speech and processes it to produce a hypothesis of the sequence of words in the utterance. The actual process is, however, extremely difficult due to the large amount of variability that appears in various stages of the process. There is a great deal of variability in the speech wave itself. No person is physically able to produce the same utterance in exactly the same acoustic form twice, and the same utterance by different speakers results in two completely different forms. The environment in which the recognition system is used can affect the acoustic forms due to the channel conditions, such as microphone used, and the background noise. Further variation comes from the number of speakers which use the system, the style of speaking, the vocabulary used and the grammar used.

Designing a speech recogniser to cope with the variations is a difficult problem, and as a result, research is focused on resolving only a small number of these problems at once. The problem of channel conditions is usually overcome by using the same environment for developing and testing a system, and ensuring that only clean speech (no background noise) is used. In the early years of speech research tasks were limited to considering only isolated words from specific speakers. Such systems had no grammar and the vocabularies used were very small. Gradually techniques developed allowing systems which were not limited to a specific speaker, so called speaker independent systems, and allowed continuous speech instead of isolated words.

At the present point in time, the methods have moved forward far enough to consider sentences of continuous speech made up from words of reasonably large vocabularies. Indeed, as the system in chapter 8 will demonstrate, recognition performances on tasks with supposedly unlimited vocabularies can be very good. However, the tasks are still being limited in many ways such as using only clean speech, and all speakers having fairly standard accents. Thus, much research effort is being directed at improving the underlying basics so that the techniques can be extended to less restricted tasks.

1.1 Large Vocabulary Speech Recognition

Hidden Markov models (HMMs) are the most successful and widely used speech recognition technique, although other techniques such as neural networks [94] give good recognition performance.

The HMM approach is an extremely effective recognition technique when the tasks are limited to very small vocabularies. For example, a word recognition rate of over 99% can be achieved with a connected digit task [83]. The extension to larger vocabularies is, however, non-trivial. The increased number of words leads to a much wider range of acoustic phenomena, especially when dealing with continuous speech. Much more variation in the speech occurs, leading to increased misclassification of the speech classes. To overcome this problem the acoustic modelling needs to be improved. Adding other knowledge sources to restrict the search for sentence hypotheses, for example grammatical constraints, can also aid recognition by reducing the number of possible classes considered at any one time.

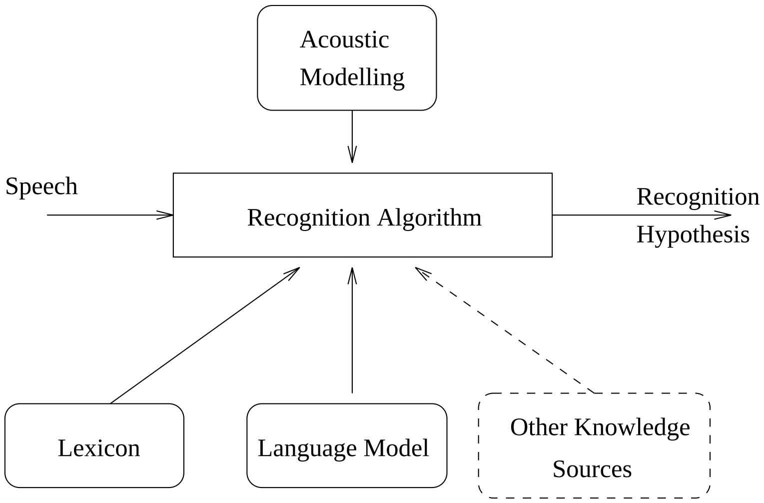

Figure 1.1: Schematic diagram for a large vocabulary speech recogniser

Figure 1.1 shows a schematic diagram for a large vocabulary recognition system, incorporating lexical constraints for word pronunciations, language models for constraining the word possibilities (due to grammatical constraints, syntax etc.), and the acoustic modelling component. Other knowledge sources such as domain dependent factors or semantic interpretation, can be incorporated, but, except for task dependent grammars, no examples of the use of such high level knowledge being incorporated into the basic recognition paradigm exist.

The definition of a large vocabulary is relative to the current perceptions of size. In the initial stages of the work for this thesis, the 1000-word task resource management task considered for much of the evaluation of ideas was considered large. However, by the completion of this work, recognition tasks on unlimited texts with a 65000 word vocabulary were being considered.

1.2 The Acoustic Modelling Problem

The core of any speech recognition system is the ability to accurately model the set of acoustic classes chosen as the basic units of recognition. If the acoustic modelling is poor, the effect of additional knowledge gained from language modelling or lexicons is limited. Ever since HMMs were first proposed, great effort has been put into trying to improve the acoustic modelling aspect. Recognition is performed by matching the observed unknown speech to the model which is most likely to have generated that speech. The acoustic modelling problem is thus twofold: (a) to maximise the likelihood of correctly identifying speech belonging to the correct model; (b) to minimise the chance of misclassifying speech by assigning it to the incorrect model.

Traditional approaches to estimating the models concentrate only on the first problem, and ignore the second. Improving the recognition rate of one model may also increase the chances of that model matching speech from other models, thus the overall recognition rate may not improve. This problem of discrimination has been addressed by several people (e.g. Brown [16]), mainly in terms of alternative methods for estimating the model parameters. The aim of many of these methods is to improve the poor assumptions that are made when using HMMs for speech recognition, since from a speech production point of view HMMs are notoriously poor models. Although the results of some of these attempts have been successful on small tasks, the extension to larger tasks has proved more difficult and few effective methods have been reported.

1.3 Speaker Independent and Speaker Dependent Modelling

When considering large vocabulary systems the training of the model parameters is of paramount importance. As with any pattern recognition system, the estimation of HMMs relies on the amount of data available for training. The more example speech data that is available the more detailed the acoustic modelling can be. For speaker independent systems the data can be drawn from many different sources and there is no onus on any one speaker to provide a substantial amount of spoken utterances. The resulting system must model both intra-speaker and inter-speaker variations in speech, and hence the acoustic modelling for large vocabulary systems needs to be extremely detailed.

If a system is required to recognise just one speaker, the situation is different. A large amount of example speech is still required to train the models for a sufficient degree of modelling accuracy in large vocabulary systems, but the data must be provided by the single speaker. This puts a large burden on the chosen speaker, although speaker dependent systems do typically perform much better than speaker independent systems [69].

In many applications it is not feasible for a single speaker to provide a substantial amount of training data, thus a speaker independent system must be used. In other applications where the system is nominally speaker independent, but a single speaker may use the recogniser for a sustained period, then speaker dependent performance is desirable. A partial solution to both of these instances is that of **speaker adaptation**, whereby a small number of example utterances from the new speaker are used to adjust the original system to improve recognition for the speaker. Several methods for such adaptation have been proposed, but the most successful approaches still require a reasonably large sample of speech to be effective.

1.4 Transformations Within HMMs

The two problems outlined above, namely that of discrimination and speaker adaptation, have been tackled in this thesis by using existing HMM theory to estimate model parameters, and then using linear transformations of the model parameters to improve the models. Only continuous density HMMs (CDHMMs) have been considered since these are widely accepted as being better at modelling speech than discrete density HMMs which use vector quantisation (VQ) to limit the variation in the representation of speech. However, the ideas presented are, in part, developed from ideas implemented on template matching with dynamic time warping (DTW) and discrete density HMMs.

In an aim to improve the discrimination within HMMs a technique based on selecting a discriminative subspace of the acoustic representation is considered. By considering each state within an HMM to represent a distinct class of acoustic sounds, the class-specific specific acoustic information which is best for discriminating between that class and all other classes can be chosen by transformation of the feature space. This discriminative feature selection can be used to improve the acoustic modelling by identifying the errors in classifying individual frames (confusion analysis), and selecting a subspace in which these confusions are minimised. The basic approach has previously been shown to work on a very limited task (the recognition of 8 isolated letters) [105]. The extension of this approach to large vocabulary continuous speech is the focus for the first part of this work.

Speaker adaptation techniques can also be viewed as approaches to improving the acoustic modelling. In this case the problem is that although the model parameters may be well estimated, they are sub-optimal for the new speaker. The aim of the adaptation is not necessarily to improve discrimination, but to make the class distributions better suited to the new speaker. This can be achieved by either transforming the parameters to move the classes in acoustic space, or by reestimating the parameters using the sample speech from the new speaker. A new transformation approach is presented which formulates the transforms in a parameter reestimation framework in an aim to combine the two methods. The new technique is structured so that it can be easily placed in a flexible framework, and adaptation can be successfully implemented on very large vocabulary tasks using a small amount of data.

Both of the transformation methods presented implement the transformations on a class-specific basis, thereby improving the acoustic modelling at the very lowest level of the recognition paradigm. It is hoped that the improved modelling at the lower level propagates through the other levels and results in improved word and sentence recognition. If this is the case then the limitations on the speech recognition tasks with respect to vocabulary size, language used etc. can be more easily resolved.

1.5 Organisation of Thesis

This thesis, as described above, contains two separate topics, based on the use of linear transformations. Chapters 3 to 5 investigate the use of transformations for discriminative feature selection, and chapters 6 to 8 investigate speaker adaptation techniques with linear transformations.

Chapter 2 describes HMMs and the notation used throughout the thesis, and reviews the basic HMM theory which is required for understanding and implementing the ideas proposed. The training and recognition algorithms for HMMs are presented and the issues of applying HMMs to continuous speech are reviewed.

A survey of previous methods for feature selection is given in chapter 3, along with other discriminative methods based on changing the standard HMM assumptions. Chapter 4 evaluates the existing global transformation methods and proposes a theory for applying transformations on a state-specific basis. The problems of applying such a method to continuous speech are discussed and a compromise using a similar technique to the global approaches is implemented on a state-specific basis. The problems of a full state-specific implementation are addressed in chapter 5, and two approaches to solving the problems are proposed and evaluated.

Chapter 6 introduces the topic of speaker adaptation with a brief outline of the problems involved. Past work in the area of adaptation is reviewed, and the problems of implementing such techniques in practical systems are discussed. A new method of adaptation, drawing on several ideas from the work reviewed is presented in chapter 7. This is a state based transformation approach which is placed within a standard HMM estimation framework. The efficacy of the method is evaluated in chapter 8 where successful application to a state of the art speech recognition system is demonstrated.

A summary of the ideas presented in the thesis is given in chapter 9 and future avenues of research to develop the ideas are suggested.

Chapter 2

Hidden Markov Models for Speech Recognition

Speech recognition using HMMs is a subject which has expanded rapidly in recent years. This chapter briefly reviews the theory of HMM speech recognition relevant to the work presented in this thesis. The assumptions behind the theory are discussed and methods of using HMM systems for recognition of continuous speech are presented.

2.1 Parameterisation of Speech

The aim of speech parameterisation is to represent the signal in a form suitable for use in a recognition system. The first stage in this process is to convert the speech to a digital representation by sampling the speech at an appropriate frequency. Due to the effects of sampling, the speech must be sampled at at least twice the frequency of the desired highest frequency component. Most speech information has been found to be in the 0-8 KHz frequency range and the speech wave is typically sampled at 16KHz to preserve this information.

Signal processing techniques are then used to reduce the sampled speech into a computationally more manageable form. The sampled signal is split into frames each representing a time slice short enough so that the speech wave can be considered stationary within the frame. This is achieved using a windowing function which windows part of the speech signal and weights those samples within the window so that the samples at the centre of the window are weighted more. The weighting is used to minimise spectral leakage. A common window function is the Hamming window:

$$w(n) = \begin{cases} 0.54 - 0.46\cos(2\pi\frac{n}{N-1}) & \text{if } 0 \leq n \leq N-1 \\ 0 & \text{otherwise} \end{cases} \tag{2.1}$$

where N is the window length.

By stepping the window along the sampled speech signal a sequence of frames representing the whole speech wave is obtained. A typical window length is 25ms, and the frames are stepped at shorter intervals (e.g. 10ms) so that the frames overlap.

The windowed speech frames are then processed further to capture the characteristic information in a compact form. Most current systems use frequency domain-based parameterisations, although time domain and autocorrelation approaches such as linear predictive coding (LPC) have been used. A common frequency domain-based approach is to use mel-frequency cepstral coefficients (MFCCs). A Fourier transform is applied to the windowed speech and a set of band-pass filters (filterbanks) used to determine the different frequencies in the frame. Using filterbanks spaced along the mel-scale gives mel-filterbank coefficients. These can be transformed into MFCCs by applying a discrete cosine transform to the logarithm of the filter-bank outputs [32]. The formula for producing M MFCCs is

$$c_i = \sum_{j=1}^{p} X_j \cos\left(\frac{i(j-0.5)\pi}{p}\right) \qquad i = 1, 2, \ldots M \tag{2.2}$$

where c_i is the i^{th} MFCC, p is the number of filter bank channels and X_j is the log of the j^{th} filter bank output.

Parameterising the speech using MFCCs is superior to using LPCs since there is no assumption of an all-pole speech production model, and a mel-scale frequency resolution can be made. Including time derivative information in the parameterisation has also been shown to be advantageous [68, 103]. This dynamic information is computed over several frames so that long term effects can be incorporated into the speech frame. The first order derivatives are termed $\Delta MFCCs$ and the second order derivatives are termed acceleration parameters $\Delta\Delta MFCCs$. Energy terms and their time derivatives can also be included in the parameterised speech vector (also termed a *feature* or *observation* vector).

2.2 Hidden Markov Models

If an utterance of speech is represented by a series of T parameterised speech frames (referred to as observation vectors) $\boldsymbol{O}$,

$$\boldsymbol{O} = \boldsymbol{o}_1, \boldsymbol{o}_2, \ldots, \boldsymbol{o}_T \tag{2.3}$$

the object of the pattern matching stage is to determine the most likely sequence of words which produce this observation sequence.

Assuming the observation vectors are produced by a Markov process the pattern matching can be achieved by using Hidden Markov Models (HMMs). A Markov model is a finite state machine which makes a state transition once every time unit, and each time a state is entered, an observation vector is generated according to a probability density function associated with that state. Probabilities are also associated with the transitions between states. Hence, the likelihood of generating $\boldsymbol{O}$ using any sequence of states can be computed.

2.2.1 HMM Representation

The definition of an HMM in this work corresponds to that used in the HTK Toolkit [113]. Note, only HMMs with continuous density output distributions are considered.

An HMM consists of a set of N states $S_1 \ldots S_N$ (Figure 2.1) with the properties:

- State S_1 is the left most (entry) state of the model (occupied only at time 0).
- State S_N is the right most (exit) state of the model (occupied only at time $T+1$).
- States $S_2 \rightarrow S_{N-1}$ are emitting states. Each emitting state j has an associated output probability density function $b_j(\boldsymbol{o}_t)$
- A transition matrix $\boldsymbol{A} = [a_{ij}]$ defines allowable transitions between states. a_{ij} is the probability of moving from state i to state j,

$$a_{ij} = p(s_{t+1} = j | s_t = i) \tag{2.4}$$

  (s_t is the state occupied at time t), with the constraint,

$$\sum_{j=1}^{N} a_{ij} = 1.$$

To correspond with other notations which do not use non-emitting entry and exit states, the initial state probability vector $\boldsymbol{\pi}$ (where π_i is the probability of occupying state i at time 0) in this case is given by

$$\pi_i = a_{1i}. \tag{2.5}$$

The non-emitting states allow simple construction of model sequences and incorporation of language models (see [113]). The output probability density functions $b_j(\boldsymbol{o}_t)$ can in theory

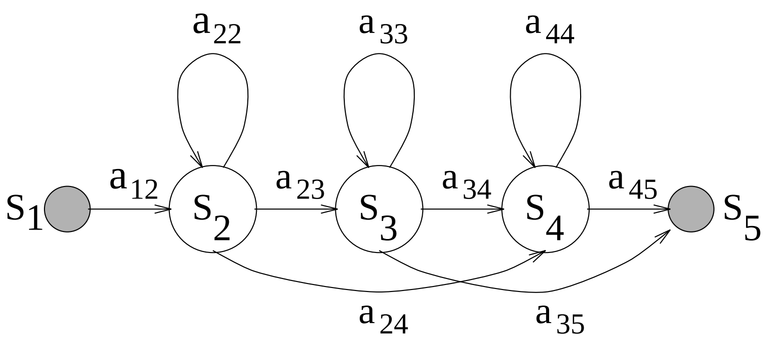

Figure 2.1: An example HMM. States 1 and 5 are non-emitting.

be any form of distribution, however it is standard practice to assume that they are made up of a mixture of Gaussian densities [92],

$$b_j(\boldsymbol{o}_t) = \sum_{k=1}^{K} c_{jk} b_{jk}(\boldsymbol{o}_t) \tag{2.6}$$

where K is the number of component densities in the mixture, c_{jk} is the mixture weight (with the constraints $0 \leq c_{jk} \leq 1$ and $\sum_{k=1}^{K} c_{jk} = 1$), and b_{jk} is a multivariate Gaussian density function,

$$b_{jk}(\boldsymbol{o}_t) = \frac{1}{(2\pi)^{n/2}|\boldsymbol{\Sigma}_{jk}|^{1/2}} e^{-\frac{1}{2}(\boldsymbol{o}_t - \boldsymbol{\mu}_{jk})' \boldsymbol{\Sigma}_{jk}^{-1} (\boldsymbol{o}_t - \boldsymbol{\mu}_{jk})} \tag{2.7}$$

where n is the dimension of the observation vector $\boldsymbol{o}_t$, and $\boldsymbol{\mu}_{jk}$ and $\boldsymbol{\Sigma}_{jk}$ are the mean and covariance respectively of the k^{th} Gaussian component. The covariance may be either a full $n \times n$ matrix modelling the correlations between elements of the feature vector, or restricted to a diagonal covariance, in which case the elements of the feature vector are assumed to be independent.

An HMM can be completely described by the parameters N, $\boldsymbol{A}$ and $\{b_i\}$, and this parameter set will be referred to as λ ($= \{N, \boldsymbol{A}, \{b_i\}\}$).

2.2.2 Assumptions

When using HMMs to model speech several assumptions are made:

> **(1)** The distribution of observation vectors is dependent only on the state and not on other vectors (independence assumption).
>
> **(2)** The speech is assumed stationary over a whole speech frame.
>
> **(3)** The observation vector distributions are adequately modelled by the number of component mixture Gaussians present.

There are certainly correlations between successive speech frames since they are produced by a continuous articulatory process. This is shown by the effectiveness of the inclusion of predictive processes in recognition where knowledge of the current frame is used to adjust the probability values of succeeding frames [60, 104]. However, these methods are computationally expensive and difficult to train so are not widely implemented.

Assumption (2) is a function of the front end speech processing and is a common assumption in many speech recognition systems. Reducing the size of the speech frames makes the stationarity assumption more reasonable, but introduces other problems such as increasing the variance of the spectral estimate.

Other output distributions have been investigated [42] but the consensus is that the computational and statistical qualities of Gaussian distributions are desirable.

2.3 Pattern Matching with HMMs

The aim of the recognition system is to take the observation sequence $\boldsymbol{O}$ and compute which words[1] were most likely to have produced this set of observation vectors. Considering the case where the observation sequence is known to represent a single word from a limited set of possible words (the vocabulary V), the task is actually to compute

$$\max_i \{p(w_i|\boldsymbol{O})\} \tag{2.8}$$

which is the probability of word w_i given the observation vector $\boldsymbol{O}$. Using Bayes' rule

$$p(w_i|\boldsymbol{O}) = \frac{p(\boldsymbol{O}|w_i)p(w_i)}{p(\boldsymbol{O})} \tag{2.9}$$

this task can be reduced to determining $p(\boldsymbol{O}|w_i)$ assuming that $p(w_i)$ can be determined from a language model using a priori knowledge, and that $p(\boldsymbol{O})$ does not affect the choice of w_i. Hidden Markov models are ideally suited to generating the class conditional $p(\boldsymbol{O}|w_i)$, assuming that an acoustic model, m_i, representing each w_i exists. This can be achieved by computing the likelihood, $\mathcal{F}(\boldsymbol{O}|m_i)$, of taking any path through the acoustic model m_i and producing the given sequence of observation vectors. $\mathcal{F}(\boldsymbol{O}|m_i)$ takes into account all possible state sequences,

$$\begin{aligned} \mathcal{F}(\boldsymbol{O}|m_i) &= \sum_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} \mathcal{F}(\boldsymbol{O}|\boldsymbol{\theta}, m_i)\mathcal{F}(\boldsymbol{\theta}|m_i) && (2.10) \\ &= \sum_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} a_{\theta_T N} \prod_{t=1}^{T} a_{\theta_{t-1}\theta_t} b_{\theta_t}(\boldsymbol{o}_t) && (2.11) \end{aligned}$$

where $\boldsymbol{\Theta}$ is the set of all K possible states sequences of length T in model m_i,

$$\Theta = \{\boldsymbol{\theta}^{(1)}, \ldots, \boldsymbol{\theta}^{(K)}\}$$

and θ_t is the state occupied at time t in $\boldsymbol{\theta}$ ($\theta_0 = 1$). To compute this likelihood efficiently it is convenient use a recursion involving forward and backward 'probabilities' [8]. The forward 'probability' is defined as

$$\alpha_j(t) = \mathcal{F}(\boldsymbol{o}_1, \boldsymbol{o}_2, \ldots, \boldsymbol{o}_t, s_t = j, m_i) \tag{2.12}$$

i.e. the likelihood that state j is occupied at time t and having generated the first t observations. The values of $\alpha_j(t)$ can be computed recursively. From the definition of the HMM, the initial conditions for $\alpha_j(0)$ are

$$\begin{aligned} \alpha_1(0) &= 1 \\ \alpha_j(0) &= 0 \quad \text{if } j \neq 1 \end{aligned}$$

[1] *words* is intended to cover the unit of speech being modelled, so could equally be phones, syllables, etc.

and the recursion proceeds for $1 \leq t \leq T$

$$\alpha_j(t) = \left[\sum_{i=1}^{N-1} \alpha_i(t-1) a_{ij} \right] b_j(\boldsymbol{o}_t) \quad \text{for } 1 < j < N \tag{2.13}$$

and terminates with the condition

$$\alpha_N(T) = \sum_{i=2}^{N-1} \alpha_i(T) a_{iN}.$$

In a similar way, the set of backward 'probabilities' $\beta_j(t)$ is defined as the likelihood that the model will generate the remaining observations $(\boldsymbol{o}_{t+1}, \ldots \boldsymbol{o}_T)$ whilst occupying state j at time t,

$$\beta_j(t) = \mathcal{F}(\boldsymbol{o}_{t+1}, \ldots \boldsymbol{o}_T | \theta_t = j, m_i). \tag{2.14}$$

This recursion starts at state N at time T with the conditions

$$\beta_j(T) \quad = \quad a_{jN} \qquad \text{for } 1 \leq j \leq N$$

and proceeds from $t = T - 1$ to $t = 0$,

$$\begin{aligned} \beta_j(t) &= \sum_{i=2}^{N-1} a_{ji} b_i(\boldsymbol{o}_{t+1}) \beta_i(t+1) \quad \text{for } 1 \leq j < N \qquad (2.15) \\ \beta_N(t) &= 0. \end{aligned}$$

The forward/backward probabilities can be used to compute the overall likelihood $\mathcal{F}(\boldsymbol{O}|m_i)$

$$\mathcal{F}(\boldsymbol{O}|m_i) = \alpha_N(T) = \beta_1(0) = \sum_{j=1}^{N} \alpha_j(t) \beta_j(t) \tag{2.16}$$

These recursive calculations are very useful in the estimation of HMM parameters.

2.4 Estimation of the HMM parameters

The parameters of the models must be estimated before the HMMs can be used for recognition. This is achieved by using estimation techniques which use a sequence of known utterances (the training set) to estimate the appropriate parameter values.

2.4.1 Maximum Likelihood Estimation

During the training phase all the parameters within the HMM system are chosen to optimise some criteria, given the training data. Common practice is to use Maximum Likelihood

Estimation (MLE) which attempts to maximise the likelihood of generating the the training data with the correct model. More formally, in MLE the aim is maximise $f_{\text{mle}}(\lambda)$,

$$f_{\text{mle}}(\lambda) = \mathcal{F}(\boldsymbol{O}|w_i, \lambda)p(\omega_i) \tag{2.17}$$

where $\boldsymbol{O}$ is the observed data when the word w_i is spoken.

It can be shown that MLE will lead to an optimal estimate of the model parameters satisfying equation 2.17 if: (a) all samples $(w_i, \boldsymbol{O})$ are from the assumed family of distributions; (b) the family of distributions is well behaved; (c) the sample size is large enough; and, (d) the performance of a system cannot get worse as its parameters get closer to the true estimates [80]. However, none of the above are valid.

It should be noted that the optimisation criteria for MLE only considers the acoustic vectors of the class under consideration during training. Hence there is no attempt to discriminate between different classes, only to optimise the likelihood of recognising utterances from the correct class. One of the aims of this work is to rectify this and add some discriminative information into the model parameters. Alternative training methods, for example maximum mutual information (MMI) [16], have shown that it is possible to improve on MLE in certain circumstances. This aspect is discussed further in chapter 3.

2.4.2 MLE Parameter Estimation

Maximum likelihood estimation of the model parameters is usually achieved by using Expectation-Maximisation (EM) techniques, an example of such a technique is the Baum-Welch algorithm [8]. Given an estimate of the parameter set λ the estimate is improved using a set of reestimation formulae to give a new set of parameters $(\bar{\lambda})$. These formulae are defined such that

$$f_{\text{mle}}(\bar{\lambda}) \geq f_{\text{mle}}(\lambda) \tag{2.18}$$

i.e. the new estimate is guaranteed to be at least as good as the previous estimate.

This allows an iterative approach to MLE. The reestimation formulae are derived by the use of an auxiliary function, $Q(\lambda, \bar{\lambda})$, defined as

$$Q(\lambda, \bar{\lambda}) = \sum_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} \mathcal{F}(\boldsymbol{O}, \boldsymbol{\theta}|\lambda) \log \mathcal{F}(\boldsymbol{O}, \boldsymbol{\theta}|\bar{\lambda}). \tag{2.19}$$

where Θ contains all possible state sequences leading to the recognition of $\boldsymbol{O}$.

It can be shown that finding the $\bar{\lambda}$ which maximises the auxiliary function leads to an increased value of $f_{\text{mle}}(\bar{\lambda})$, unless already at a maximum, so iterative application eventually converges to the MLE estimate. Baum first proved the convergence of the algorithm [8] and the proof was later extended to mixture distributions and vector observations [58, 77].

The auxiliary function (2.19) can be expanded

$$Q(\lambda, \bar{\lambda}) \quad = \quad \sum_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} \mathcal{F}(\boldsymbol{O}, \boldsymbol{\theta}|\lambda) \left\{ \log \bar{a}_{\theta_T N} + \sum_{t=1}^{T} \log \bar{a}_{\theta_{t-1}\theta_t} + \sum_{t=1}^{T} \log \bar{b}_{\theta_t}(\boldsymbol{o}_t) \right\}.$$

and separated into component auxiliary functions based on the individual parameter sets:

$$Q_a(\lambda, \bar{\lambda}) = \sum_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} \mathcal{F}(\boldsymbol{O}, \boldsymbol{\theta}|\lambda) \left\{ \log \bar{a}_{\theta_T N} + \sum_{t=1}^{T} \log \bar{a}_{\theta_{t-1}\theta_t} \right\} \tag{2.20}$$

$$Q_b(\lambda, \bar{\lambda}) = \sum_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} \mathcal{F}(\boldsymbol{O}, \boldsymbol{\theta}|\lambda) \left\{ \sum_{t=1}^{T} \log \bar{b}_{\theta_t}(\boldsymbol{o}_t) \right\}. \tag{2.21}$$

$Q_b(\lambda, \bar{\lambda})$ can be further decomposed into individual mixture components and weights. These auxiliary functions for individual parameter sets lead to closed form solutions for the reestimation of the model set parameters [58]. The use of the auxiliary function is discussed further in chapter 7.

2.5 Recognition with HMMs

To recognise an utterance with an HMM, MAP (maximum *a posteriori*) decoding is used. Instead of computing $p(\omega_i|\boldsymbol{O})$, which is the desired probability, the probability $p(\boldsymbol{O}|\omega_i)$ is computed and Bayes' theorem used to convert to the desired form (equation 2.9).

The recognition of an unknown utterance thus involves computing the likelihood of the observation sequence given each acoustic model (m_i), which can be achieved by computing the total likelihood using the forward-backward alignment. This considers every possible path through model m_i and gives the combined likelihood of all paths generating the observation vector. However, in practice the forward-backward likelihood is dominated by likelihoods from a small number of possible paths through the model. Thus to reduce computation, the likelihood of taking the most likely state sequence through the model is computed. The likelihood of the best state path to state j at time t is computed as,

$$\gamma_j(t) = \max_i \{\gamma_i(t-1) a_{ij} b_j(\boldsymbol{o}_t)\} \tag{2.22}$$

where $\gamma_1(0) = 1$, and $\gamma_j(0) = 0$ if $j \neq 1$ (c.f. the definition of $\alpha_j(t)$ equation 2.13). The likelihood for the whole utterance is,

$$\gamma_j(T+1) = \max_i \{\gamma_i(T) a_{iN}\}. \tag{2.23}$$

This is the basis for the Viterbi algorithm [100] which is commonly used for recognition. For computational efficiency, and to prevent underflow, the likelihoods are computed in the log domain.

The main advantage of using the Viterbi algorithm is that it can easily be extended to continuous speech. It also has the advantage of allowing disjoint paths within the same model to be considered separately. For instance if the model has two different state paths from one state to another which correspond to alternative pronunciations, the combined likelihood of both paths is not the desired likelihood. By using the Viterbi path the alternative pronunciations can be considered separately.

A computation saving may be made by using pruning algorithms to restrict the search. A typical pruning algorithm is the beam search whereby all hypotheses that fall below a certain threshold of the current most likely hypothesis are pruned out of the search. There is no need to compute state likelihoods for any state outside the current 'beam' and computation can be greatly reduced.

2.6 Whole Word and Sub-word Modelling

The theory presented in the previous section has assumed that there is an HMM for each word that can be recognised. In practice for continuous speech recognition tasks this is not the case, due to the problems with data sufficiency. To train a word model robustly several example utterances of the word are required. In many databases the amount of training data is limited and for many words there are insufficient numbers of examples. Indeed, in unconstrained recognition tasks where the test set has an unlimited vocabulary there will be words which have no examples in the training data (for the 65000 word WSJ tasks described in appendix A the majority of words are unseen during training). For this reason word modelling is not generally implemented and sub-word modelling is used instead.

Linguistics defines the most basic unit of speech as phonemes. These are abstract units and about forty of them occur in English speech. For the purpose of sub-word modelling, a set of basic units representing the realisations of phonemes are used. These representations are termed phones. Most large continuous speech recognition systems use phones as the unit of speech to model, and then use combinations of the sub-word models to build up whole word models (Figure 2.2).

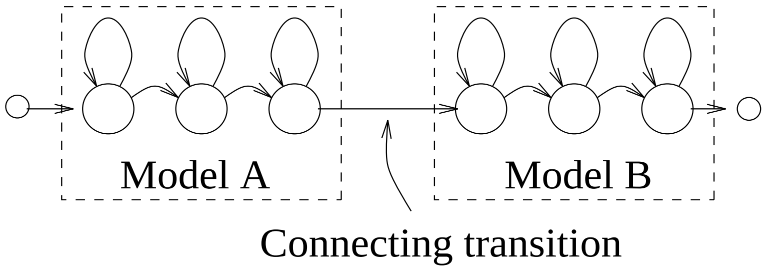

Figure 2.2: Example of combining phone models into word models. Phone A and phone B are merged into a word model by removing the final state of A and replacing it by state 2 of B.

The acoustic realisation of a phone can vary greatly depending on the preceding and following speech. Although using multiple component densities in the output distributions may help to model this variation, this leads to poorer class separation. A better solution is to use introduce context-dependent phone models. This can be achieved in a number of ways: (a) function word dependent phone models, where a limited number of frequently occurring words have separate phone models defined for them. For example if 'THE' is pronounced '/dh//ax/' there would be a model for /dh/ in normal speech and a separate one for /dh/ in 'THE'; (b) phone context dependent models - different models represent the same phone in each of the possible immediate left and/or right contexts. Different degrees of context can be used - models using a single left (or right) contexts are termed biphones, those using both immediate left and immediate right contexts are triphones. This may in theory be extended to any number of contexts but this then leads to problems with sufficient training data due to the large number of models involved; and, (c) incorporation of cross word effects. The pronunciation of phones at the beginning or end of words can be significantly affected by the end/start of the last/previous word.

2.7 Parameter Tying

Defining context dependent models can lead to a large number of models. However, the training data is not infinite and limits the number of parameters that can be robustly estimated for the model set. It is possible that some phone contexts which are necessary for the recognition tasks are not present in the training data and occurrences of others are limited. To overcome such problems the technique of parameter tying is widely used. When two distributions or models share the same set of parameters they are said to be 'tied' [11, 109].

If a phone in one context is thought to have a similar acoustic realisation as the phone in a different context then the two models may be tied. At a lower level similar states or even mixture components may be tied. The tying may be determined via linguistic knowledge or after an initial estimation of the parameters. Once the parameters are tied they may be reestimated using the data associated with all the tied components. Thus the use of tying can reduce the number of parameters in the system to a level where there is sufficient data for robust estimation [112].

The idea of using phonetic knowledge has also been extended to predict the pronunciations of unseen phone contexts [85].

2.8 HMMs for Continuous Speech

The theory presented so far has assumed that one model can match a whole utterance. For the application to continuous speech this is no longer the case. Each utterance will consist of a series of words which are not strictly separated. Thus the theory must be extended to allow for a sequence of models.

For the training algorithms, since the speech is labelled, a single sentence model may be built in a similar manner to building word models from phone models, and the parameter estimation can be performed as before.

The recognition procedure is more complicated since the number of combinations of words in an utterance is unknown. Thus a method such as the token passing algorithm [111] must be employed which utilises the non-emitting entry and exit states of each model to generate a Viterbi alignment. In this method a token is associated with each state in a model. The token records the likelihood of occupying that state at the current time having generated all the observation vectors up to that time ($\gamma_j(t)$ from section 2.5). The token also records the model sequence that has been taken to reach the current state. Within the model, the state transitions occur as in the standard case, and the individual state likelihoods in the token are updated according to the transitions and the observation vector. However, after each frame, token propagation takes place between models.

For each model, the token in the final state is computed from the current tokens in the other states of the same model. A new token is then placed in the first state of each model (m_i), the token being selected from the final state of model m_j such that,

$$\begin{aligned} \mathcal{F}(\boldsymbol{o}_1, \ldots, \boldsymbol{o}_t, \theta_t = N_j)T_{j,i} \quad > \quad & \mathcal{F}(\boldsymbol{o}_1, \ldots, \boldsymbol{o}_t, \theta_t = N_k)T_{k,i} \\ & \forall \quad \text{models} \quad m_k \neq m_j. \end{aligned} \tag{2.24}$$

where N_j and N_k are the final states of the models m_j and m_k, and $T_{j,i}$ represents the transition probability between model m_j and m_i which can be used to incorporate grammars and language models to constrain the search space. The likelihood of the selected token is also updated to include the transition.

After the last frame a similar propagation is made to the final states of all models which can terminate a sentence, and the most likely final model determined. The recognition hypothesis can then be found from the recorded history of the token to find the Viterbi word sequence.

Multiple hypotheses (the N-Best) can be generated by building a word lattice during recognition. The lattice contains alternative words for each section of speech. An A^* search can then be performed on the lattice to determine all possible word sequences [98]. Computing and storing such a word lattice is computationally expensive, and methods of pruning down the number of possible words must be used. Usually only very restricted implementations may be used to generate approximate solutions [95]. The alignment must consider the likelihood of the sequence of words and not states for the N-Best path score so that multiple pronunciations of the same word are not included in the same path [96].

2.9 Grammars and Language Models

During recognition, different word sequences can be weighted by grammars and more weight can be given to the more grammatically correct sequences. The search space for a sequence of words can grow extremely large if all possible word sequences are considered, and pruning out unlikely sequences can reduce the search effort.

- Deterministic Grammars
  The possible successors of a particular word can be restricted using deterministic grammars, $T_{i,j} = \{1|0\}$. By enforcing grammar rules on recognition the hypotheses should correspond more to valid sentences. However, natural language grammars are far from trivial and only basic rules may be implemented easily.

- Stochastic Grammars
  The probabilities of certain words following other words can be used to weight the more likely word sequences, and thus aid in pruning the less likely sequences from the recognition path. The $T_{i,j}$ probabilities are usually computed from a large corpus of example sentences. The grammars can be extended to consider more context. A bigram language model represents the probability of a word given the previous word, whilst trigram and 4-gram language models represent the probability of a word given the previous 2 and 3 words respectively.

These word based factors can aid the pruning and increase the recognition performance of recognisers.

Chapter 3

Discriminative Methods for HMMs

The basic HMM theory outlined in the previous chapter has addressed the problem of how to model specific classes of speech. However, no attempt was made to discriminate between different speech classes. This chapter examines the discriminative ability of HMMs and methods for improving discrimination between classes.

3.1 The Discrimination Problem

As discussed earlier (chapter 2) the use of maximum likelihood estimation (MLE) for training HMMs is widespread. MLE aims to find an estimate for the parameters such that the likelihood of the model generating the training data is maximised. MLE is both computationally efficient and gives a robust estimate when sufficient training data is available. Indeed, when the true forms of the distributions of the speech classes are known it can be shown that MLE leads to an optimal estimate if sufficient data is available. In general, however, the forms of the distributions are not known, hence MLE leads to sub-optimal estimates for the HMM parameters. Furthermore, the amount of data available for training the models is limited, and, unless it is completely representative of the test set, although MLE may lead to a good estimate for the training set data this is not necessarily the case for the recognition test set. A second problem with MLE is that the method does not take into account other classes when making class distribution estimates, there is no attempt to maximally separate the classes. This leads to cases where the likelihood of incorrect data being recognised by the class is also increased.

Therefore, by incorporating some discriminative aspect into the estimation of HMM parameters it should be possible to improve recognition performance. This can be achieved in two ways: (1) by changing the parameter estimation algorithm to incorporate discriminative optimisation criteria; and, (2) by ensuring the class characteristics of each sound class modelled (as represented by the feature vector) are maximally separated. Some such training techniques are briefly outlined in the next section, and in section 3.3 methods of improving discrimination via transformation of the feature space are investigated.

3.2 Discriminitive Training Methods

A number of alternative training methods have been proposed which aim to improve the estimation of the model parameters from the available training data. These methods vary in their approaches to resolving the problems of MLE, either using different optimisation criteria or resolving the poor assumptions underlying the optimisation. There are two broad categories for these training methods, one based on sound theoretic estimation of the parameters (e.g. using an information theoretic approach) and the other based on ad-hoc methods of maximising some recognition criteria, i.e. the aim is to directly reduce errors. Examples of both types of method are briefly reviewed in the following sections.

3.2.1 Maximum Mutual Information

Brown [16] viewed the HMM training problem from an information theoretic point of view via the maximum mutual information (MMI) approach. The aim of MMI is to maximise the joint information between the class samples and the model parameters. The mutual information measure is defined as

$$I(\boldsymbol{O},\omega) = \sum_{\omega,\boldsymbol{O}} \mathcal{F}(\boldsymbol{O},\omega|\lambda)\log\left(\frac{\mathcal{F}(\boldsymbol{O},\omega|\lambda)}{p(\omega)\mathcal{F}(\boldsymbol{O})}\right) \tag{3.1}$$

where $\boldsymbol{O}$ is the acoustic event, and ω is a word.

Assuming that the training data is a representative sample, $\mathcal{F}(\boldsymbol{O},\omega)$ is constant, so only the final term in equation 3.1 is of interest, and this is used as the MMI criteria f_{mmi},

$$f_{\mathrm{mmi}}(\lambda) \quad = \quad \log\left(\frac{\mathcal{F}(\boldsymbol{O},\omega|\lambda)}{p(\omega)\mathcal{F}(\boldsymbol{O})}\right)$$

Applying Bayes' theorem and re-ordering,

$$f_{\mathrm{mmi}}(\lambda) = \log\mathcal{F}(\boldsymbol{O}|\omega,\lambda)p(\omega) - \log\mathcal{F}(\boldsymbol{O})p(\omega). \tag{3.2}$$

The first term on the r.h.s. of equation 3.2 is exactly that of the MLE criteria f_{mle}, the difference between MLE and MMI lies in the second term.

Assuming that the language model is given, consider the derivative,

$$q_\lambda(\boldsymbol{O}|\omega) = \frac{\delta\mathcal{F}(\boldsymbol{O}|\omega,\lambda)}{\delta\lambda_i} \tag{3.3}$$

where λ_i is an arbitrary individual parameter. The differential of f_{mmi} w.r.t. λ_i is,

$$\frac{\delta f_{\mathrm{mmi}}}{\delta\lambda_i} = \left[\frac{1}{\mathcal{F}(\boldsymbol{O}|\omega,\lambda)} - \frac{p(\omega)}{\mathcal{F}(\boldsymbol{O})}\right] q_\lambda(\boldsymbol{O}|\omega) - \sum_{\hat{\omega}\neq\omega}\frac{q_\lambda(\boldsymbol{O}|\hat{\omega})p(\hat{\omega})}{\mathcal{F}(\boldsymbol{O})} \tag{3.4}$$

where $\hat{w}$ represents all words. Compare this to the differential of f_{mle} w.r.t. λ_i,

$$\frac{\delta f_{\mathrm{mle}}}{\delta\lambda_i} = q_\lambda(\boldsymbol{O}|\omega) \tag{3.5}$$

and it can be seen that f_{mmi} has contributions in the direction of MLE, but also has compensating components in the direction of $q_\lambda(\boldsymbol{O}|\hat{\omega})$ for incorrect word sequences. Thus, MMI can be seen to be incorporating discrimination into the training process.

One of the advantages of MMI is that there is no assumption that either the set of distributions modelling the speech, or the language model has to be correct. The disadvantage of MMI is that there is no direct optimisation formula as in MLE, and estimates have to be made using techniques such as gradient descent. Due to this problem there is no guarantee of convergence to an optimal parameter set.

It has been demonstrated both theoretically [81] and experimentally [7, 16] that MMI will outperform MLE when the assumption of model correctness is poor, and there is sufficient data for estimation. Although MMI has been used for small tasks such as digit recognition [18, 19, 83] the computation required for iterative estimation, and the problems of convergence have limited the use of MMI for larger tasks, although a few attempts have been made using discrete density HMMs [25].

3.2.2 Minimum Discrimination Information

The minimum discrimination information (MDI) [37, 38] approach attempts to generate a set of model parameters which has the minimum discrimination information (or cross-entropy) between the training data and the HMM parameters. This is another information theoretic approach which also removes the assumption that the true probability distribution is a Markov source.

The discrimination information between two probability distributions, U, the joint probability density of the acoustic signal $\boldsymbol{O}$ and word ω, and V, the joint probability density of the acoustic model λ and word model, is

$$D(U \parallel V) = \sum_{m=1}^{L} \sum_{\boldsymbol{o} \in \boldsymbol{O}} u(\boldsymbol{o}, m) \ln \frac{u(\boldsymbol{o}, m)}{v(\boldsymbol{o}, m)} \tag{3.6}$$

where $u(\boldsymbol{o}, m)$ and $v(\boldsymbol{o}, m)$ are the probability mass functions of U and V respectively, and m represents the L models in the model set. The aim is to minimise the discrimination information so that the model parameters produce a similar distribution to the acoustic source.

It can be shown that if the observations are produced by exactly the same source as that modelled by the HMMs (i.e. the assumption of true distributions for MLE is valid) then MDI produces the MLE estimate.

MDI can also be shown to incorporate the MMI estimate when estimating all acoustic models simultaneously using the empirical distribution of the acoustic signal from all words. Let U_ω be the p.d.f. U given word ω, and V_μ be the p.d.f. V given the word model μ then the discrimination information is,

$$D(U \parallel V) = D(U_\omega \parallel V_\mu) - I(\boldsymbol{O}, \omega). \tag{3.7}$$

Increasing the mutual information $I(\boldsymbol{O}, \omega)$ as in MMI reduces the discrimination information between the acoustic signal and the model. In this case MMI and MDI lead to the same estimate of the model parameters [37].

MDI suffers from the same problem as MMI in that no direct training algorithm is available to estimate the parameters, and using iterative gradient descent algorithms does not guarantee convergence. As a consequence MDI is rarely implemented.

3.2.3 Optimisation of Error Rates

Instead of attempting to optimise a theoretical measure to increase discrimination, attempts have been made to adjust the model parameters to directly increase the recognition performance. Bahl [6] uses a corrective training approach where the models are first trained using a standard technique, and then an iterative refinement of the model parameters is made. Each iteration involves a recognition of the training data to identify errors or confusable words. The model parameters are updated based on a simple shift in the (discrete) output probabilities for each state, so that within-state vectors are more likely. This approach has been extended to continuous speech by Lee & Mahajan [72], but again using discrete output distributions, and Mizuta shows that it is also viable with continuous density HMMs [79].

Chou [22, 23] implements a similar scheme, but aims at minimising the recognition error over a whole string. Using an N-Best framework confusable strings are generated and the model parameters adjusted according to a loss function. This scheme is refined by Chen [20] for implementation on continuous density output distributions and shows an improvement over MLE on a Chinese word task.

Franco [39] also uses an error criterion for optimising the parameters. Instead of identifying confusable strings, the state sequence is examined and the parameters are adjusted to ensure the correct state path is chosen during recognition.

Using the error rate on the training data as the optimisation criteria is intuitively a good idea, since the aim is low error rates on the test data. However, in terms of implementation and theory these methods have several disadvantages: errors on the training set may not be representative of errors on the test set; correction of errors may lead to new errors so convergence is not guaranteed; no true theoretical measure for the estimation of the parameters is used, so there is no guarantee of robustness; repeated recognition of the training data may be computationally expensive.

Other alternative training methods, such as placing HMMs in a neural network framework (e.g. competitive training [108]) and Bayesian estimation [42] have more theoretical basis, but are difficult to implement in large systems. Thus, at the present time, for large systems MLE training is generally used in spite of the poor assumptions.

3.3 Discriminitive Feature Vectors

It is clear that the choice of features for inclusion in the speech vector is very important. Ideally the feature vector should contain as much acoustic class specific information about the speech as possible, enabling good classification into individual sound classes. Unfortunately, it is very difficult to define what features are needed since different classes of sounds have different characteristics. The size of the feature vector must also be considered in relation to the amount of data available and the computational cost during recognition. Each mean vector and covariance matrix has the same number of dimensions as the feature vector. Using a large feature vector means that fewer class distributions may be robustly estimated from a fixed amount of data than using a small feature vector. The calculation of the observation vector likelihoods is the major computational overhead in most speech recognition paradigms, and this is also directly dependent on the number of elements in the feature vector.

Intuitively, increasing the number of elements in a feature vector should give more information about the speech wave, thus aiding classification. In practice however, this is not the case, since the extra features may contribute additional information which does not aid class separation but leads to more variation in within-class distributions. The features which are representative of the class contribute less to the class membership decision due to the contributions of the extra features. Addition of such features is unnecessary and adds to the computational cost without a corresponding improvement in discrimination. Features are generally selected on the basis of experimental evidence, but the best features for one task are not necessarily the best set for other tasks. The majority of large vocabulary continuous speech systems developed recently favour the use of MFCCs and incorporate an energy term along with first and second order time derivatives [32, 103].

Having selected a set of features to incorporate into the observation vectors, it is important to use the information contained within the vectors in the most effective way. Duplicated information can be reduced and any elements not contributing to class separation can be removed. There are several approaches for achieving this aim, some based on the properties of individual features, and others based on the interaction between features and their combination. Broad definitions for these groups are termed *feature selection* and *feature extraction*.

The purpose of both feature selection and extraction is twofold:

1. To optimise the recognition rate by selecting a set of elements which aid discrimination for the final speech feature vector.

2. To produce a speech vector of a size which is computationally tractable, and can be well estimated from the available training data.

Ideally each element should be independent of the other elements in the vector, and contribute to separating the different speech classes. Various methods for feature selection and extraction are described in section 3.5, but first methods of measuring class separation are examined.

3.4 Class Separability Measures

The problem of measuring class separability is well documented in pattern recognition literature [40]. Given a set of classes (ω_i), two measures of *scatter* can be defined:

a) Within-class scatter matrix - the scatter of samples around the expected vector for the class:

$$\boldsymbol{S}_w = \frac{1}{K}\sum_{i=1}^{K} E\{(\boldsymbol{x}-\boldsymbol{\mu}_i)(\boldsymbol{x}-\boldsymbol{\mu}_i)'|\boldsymbol{x} \in \text{class} \quad \omega_i\} \tag{3.8}$$

b) Between-class scatter - the distribution of the different class means (and therefore classes):

$$\boldsymbol{S}_b = \frac{1}{K}\sum_{i=1}^{K}(\boldsymbol{\mu}_i-\boldsymbol{\mu}_0)(\boldsymbol{\mu}_i-\boldsymbol{\mu}_0)' \tag{3.9}$$

assuming both distributions to be Gaussian, where $\boldsymbol{\mu}_i$ is the mean of class ω_i and $\boldsymbol{\mu}_0$ is the expected value of all the speech data regardless of class.

Having defined these scatter matrices, a measure of the class separability can be made by comparing the ratio of the within-class scatter to the between-class scatter.

Typical measures are:

$$\begin{aligned}
J_1 &= tr(\boldsymbol{S}_2^{-1}\boldsymbol{S}_1) \\
J_2 &= \ln|\boldsymbol{S}_2^{-1}\boldsymbol{S}_1| = \ln(|\boldsymbol{S}_1|/|\boldsymbol{S}_2|) \\
J_3 &= tr(\boldsymbol{S}_1) - \eta(tr(\boldsymbol{S}_2) - c) \\
& \text{where } \eta \text{ is a lagrange multiplier to add a constraint} \\
& \text{ and } c \text{ is a constant} \\
J_4 &= tr(\boldsymbol{S}_1)/tr(\boldsymbol{S}_2)
\end{aligned}$$

where $\boldsymbol{S}_1 = \boldsymbol{S}_b$, except for J_2 where $\boldsymbol{S}_1 = \boldsymbol{S}_w + \boldsymbol{S}_b$, and $\boldsymbol{S}_2 = \boldsymbol{S}_w$.

The scatter matrices are invariant under co-ordinate shifts. J_1 and J_2 are invariant under any non-singular transformation, but J_3 and J_4 are dependent on the co-ordinate system.

Other methods of estimating class separability are available, such as the divergence measure (see Appendix B), or Bhattacharyya distance which compare the actual distributions.

The aim is thus to determine an optimal vector of d dimensions from the original speech vector of n features from the speech coding. The vector is optimal in the sense that it maximises the class separation according to one of the measures above (for computational ease J_1 will normally be considered).

3.5 Feature Selection

Feature selection is a method of choosing from the available features a subset which is optimal for recognition. Each element is assessed for suitability in the vector by assigning it a score on some basis, and choosing the best ranked features to make up the vector.

Three possible approaches which have been investigated in the literature are

1. Use of recognition rate for individual features
2. Use of the Fisher ratio
3. Discriminative feature selection

These approaches are briefly examined.

3.5.1 Use of Recognition Rate

Paliwal [87] investigates a method in which a set of features were obtained by coding the speech wave. Each feature was used individually in a recognition task on the training set. The features were ranked according to the recognition performance obtained, and the best d features selected for use in the final recogniser.

This method has the advantage of assessing how effective each feature is within the recogniser. Whilst it is possible to rank the features in this way, the assumption that all the features are statistically uncorrelated means the selection is not guaranteed to be optimal. A further problem with this method is that it gives no indication as to the number of features to use for optimal recognition.

It is apparent from the results reported by Paliwal for recognition using individual features that a few features provide a lot of recognition information, but the majority of features give very closely matched recognition rates. With low recognition rates on individual features (the majority are $11 - 20\%$) ranking the features may be unreliable.

3.5.2 Use of the Fisher ratio

The Fisher ratio (F-ratio) is defined as the ratio of between-class variance and within-class variance. Making the assumptions:

1. The feature vectors within each class have Gaussian distributions.
2. The features are statistically uncorrelated.
3. The variances within each class are equal.

The F-ratio (F_i) for the i^{th} feature is defined as:

$$F_i = \frac{B_i}{W_i} \tag{3.10}$$

where

$$B_i = \frac{1}{K}\sum_{k=1}^{K}(\mu_i^k - \bar{\mu}_i)^2 \tag{3.11}$$

$$W_i = \frac{1}{K}\sum_{k=1}^{K} C_{ii}^k \tag{3.12}$$

where μ_i^k and C_{ii}^k are the mean and variance of the i^{th} feature of the k^{th} class, and $\bar{\mu}_i$ is the overall mean of the i^{th} feature. This is exactly the same principle as the class separation measure J_1 but applied to a single dimension. Larger F-ratios indicate that a feature has a small within-class variance compared to the between-class variance, so the feature should be good at discriminating between classes.

Calculating the Fisher ratio for each feature allows the features to be ranked and the best d selected. Since the third assumption is generally not valid, the method can only be implemented by using a pooled within-class variance.

Again, this method has been investigated by Paliwal [87]. A broad range of F-values is obtained ($0.03 \rightarrow 2.17$), however like the recognition rate measure, although the important features are very clear many features have very similar values making it difficult to order them properly. This is more obvious in the case of second order time differential parameters which generally have low magnitudes and have small between-class variances.

3.5.3 Discriminative Feature Selection

A third method of selecting features based on their discriminative ability has been proposed by Bocchieri & Wilpon [14, 15]. Instead of looking directly at the recognition rate achieved using individual features, a full feature vector is used in recognition of the training set. The contribution of each feature to the recognition errors is recorded.

A basic set of features is identified, and a set of models trained using those features on the whole set of training data. Recognition of the training data is performed using a Viterbi algorithm to generate a model sequence. The model sequence is compared with the reference sequence to identify substitution or insertion errors. For each error that occurs the erroneous frame, $\boldsymbol{f}_t$, and the state it is aligned with, s_t, are identified. An average distance measure for the i^{th} feature $D(i)$ is calculated from all the errors in the training set recognition.

$$D(i) = E\left[\frac{[f_t(i) - \mu_{s_t}(i)]^2}{\sigma_{s_t}(i)}\right] \tag{3.13}$$

where $f_t(i)$ is the index of the i^{th} feature in the observation vector $\boldsymbol{f}_t$, and $\mu_{s_t}(i)$ and $\sigma_{s_t}(i)$ are the mean and variance respectively of feature i in state s_t.

If the models have been trained using MLE, then the calculation of $D(i)$ for correctly classified data leads by definition to the result:

$$D(i) = 1 \qquad \forall i \tag{3.14}$$

Thus, the measure $D(i)$ for incorrectly recognised frames is effectively a measure of between-class and within-class ratio (c.f. J_1 and the Fisher ratio). The features with the d highest values are selected for use in the final recognition system.

In this case it is assumed that the within-class distribution is a Gaussian distribution and that all features are independent (diagonal covariance). Since the discriminant information measure has been produced by the recogniser with a full set of features, the features are specifically tuned for that recogniser on the particular database used.

3.5.4 Comparison of Feature Selection Methods

The different methods produce substantially different feature orderings (Table 3.1)[15, 87]. The initial feature vector contains 38 components made up of 12 linear prediction cepstral coefficients (c_i) together with the first (Δc_i) and second order ($\Delta\Delta c_i$) time differentials, and the first and second order time differentials of an energy term.

Rank	Indiv.	F-ratio	Disc.	Rank	Indiv.	F-Ratio	Disc.
1	c_1	c_2	ΔE	20	Δc_9	$\Delta\Delta c_1$	c_8
2	ΔE	ΔE	$\Delta\Delta E$	21	c_8	Δc_9	$\Delta\Delta c_6$
3	c_2	c_1	c_4	22	Δc_{10}	Δc_{11}	$\Delta\Delta c_4$
4	Δc_1	c_3	c_6	23	Δc_{11}	Δc_8	$\Delta\Delta c_1$
5	Δc_2	c_4	c_1	24	Δc_5	Δc_{10}	c_9
6	c_3	c_{10}	Δc_4	25	Δc_8	$\Delta\Delta c_3$	Δc_8
7	$\Delta\Delta E$	c_{11}	c_5	26	c_{12}	Δc_6	$\Delta\Delta c_5$
8	Δc_3	Δc_2	Δc_6	27	Δc_7	$\Delta\Delta c_2$	$\Delta\Delta c_7$
9	c_4	c_5	Δc_5	28	Δc_6	Δc_7	$\Delta\Delta c_9$
10	$\Delta\Delta c_1$	c_8	c_2	29	$\Delta\Delta c_7$	Δc_{12}	Δc_{10}
11	c_5	Δc_1	Δc_3	30	$\Delta\Delta c_9$	$\Delta\Delta c_4$	$\Delta\Delta c_8$
12	c_{10}	Δc_3	c_3	31	Δc_{12}	$\Delta\Delta c_{10}$	c_{11}
13	c_9	c_9	c_7	32	$\Delta\Delta c_8$	$\Delta\Delta c_9$	Δc_{11}
14	c_{11}	$\Delta\Delta E$	Δc_2	33	$\Delta\Delta c_4$	$\Delta\Delta c_5$	c_{10}
15	$\Delta\Delta c_3$	c_7	Δc_1	34	$\Delta\Delta c_6$	$\Delta\Delta c_{11}$	$\Delta\Delta c_{10}$
16	$\Delta\Delta c_2$	c_6	Δc_7	35	$\Delta\Delta c_{10}$	$\Delta\Delta c_8$	c_{12}
17	c_7	c_{12}	Δc_8	36	$\Delta\Delta c_{11}$	$\Delta\Delta c_7$	$\Delta\Delta c_{11}$
18	Δc_4	Δc_5	$\Delta\Delta c_3$	37	$\Delta\Delta c_5$	$\Delta\Delta c_6$	Δc_{12}
19	c_6	Δc_4	$\Delta\Delta c_2$	38	$\Delta\Delta c_{12}$	$\Delta\Delta c_{12}$	$\Delta\Delta c_{12}$

Table 3.1: Comparison of ordering of features using different feature selection methods (using information from [15, 87]). Indiv. is the ranking using recognition of individual features and Disc. is the ranking using the discriminative measure.

Equivalent performance to the full 38 feature vector is obtained using the top 16 features from either the individual recognition rate ranking or the discriminative analysis ranking, but for the F-ratio ranking at least 24 features are required. The best overall performance is achieved using the top 32 features ranked by discriminative analysis. This is interesting because the features discarded include the 10^{th} and 12^{th} cepstral coefficients, whilst c_{11} and Δc_{10} are retained. The second derivatives are ranked poorly in both the individual recognition rate and F-ratio rankings (except for those pertaining to c_1 and c_2) yet the inclusion of second derivatives has been shown to contribute significantly to recognition performance [103].

The rankings using different methods show that most of the features contribute to the recognition performance in some manner, however their relative importance is heavily dependent on the ranking measure used. Since all the measures have ignored correlations between features, different features may contribute similar information, thus the ranking may not be optimal. This is indicated by the use of an empirical combination of 24 features, made up of the energy terms, the best 10 ranked cepstral coefficients, the best 8 ranked first differential cepstras and the best 4 ranked second order differential cepstras (ranked by individual recognition rate). This gave a better performance than any ordered feature selection method [87].

Thus these feature selection methods have three fundamental problems:

- Different selection criteria produce different feature rankings.
- The number of features to use in the final vector must be determined empirically.
- Correlations between features are ignored.

In addition, the definition of an original feature set is important since only information apparent along the feature space axes can be used in the final vector.

3.6 Feature Extraction

Feature extraction differs from feature selection in that the new feature set is a linear combination of the original feature vector. Thus the reduced feature set contains components from all of the original features so that most of the information available is retained.

There are two basic approaches to feature extraction: principal components analysis and linear discriminant analysis, which are variations on the same idea. Both project the original feature vector into a new subspace, but with different intentions. Principal components attempts to select the subspace which is best for classification, while linear discriminant analysis attempts to preserve the discriminative information available whilst removing the confusable information.

3.6.1 Principal Components Analysis

Principal components analysis (PCA) is based on the assumption that the direction along which there is most variation in the speech vectors contains the most information about the classes of speech. This is a poor assumption as will be seen later.

The distribution of the original speech vectors is determined on a global basis. A rotation is then derived to project the distribution into a space where the directions of maximum variance are aligned along the axes, which represent the principal components. The first principal component is the direction along which there is the largest variance for the distribution. The dimensions which contain little variance (and thus little information) are discarded to select a subspace containing most of the variance of the features [91].

The principal component axes are determined by generating a total covariance matrix $\boldsymbol{T}$,

$$\boldsymbol{T} = \frac{1}{N}\sum_{\boldsymbol{x}}(\boldsymbol{x}-\boldsymbol{\mu})(\boldsymbol{x}-\boldsymbol{\mu})' \tag{3.15}$$

where $\boldsymbol{x}$ takes the value of all N speech vectors in the training data, and $\boldsymbol{\mu}$ is the mean vector of the whole training data.

The directions of maximum variance are those which correspond to the eigenvectors of the largest eigenvalues of $\boldsymbol{T}$. If the eigenvectors of $\boldsymbol{T}$, $\boldsymbol{v}_i$, have corresponding eigenvalues λ_i ($i = 1 \ldots n$) then a rotation matrix $\boldsymbol{R}_1$ can be defined:

$$\boldsymbol{R}_1 = [\boldsymbol{v}_1\boldsymbol{v}_2 \ldots \boldsymbol{v}_n] \tag{3.16}$$

$\boldsymbol{R}_1$ rotates the speech vectors into a new space where all features are statistically uncorrelated. The axes of the new space correspond to the eigenvectors of $\boldsymbol{T}$ and the variance along each axis is equivalent to the eigenvalue of the axis. The principal components axes can be selected by discarding those columns of $\boldsymbol{R}_1$ which have the smallest eigenvalues, so that $\boldsymbol{R}_1$ becomes a $(n \times d)$ matrix which can be applied to the original feature vector $\boldsymbol{x}$ to generate a principal components vector $\boldsymbol{y}$ with d components.

$$\boldsymbol{y} = \boldsymbol{R}_1'\boldsymbol{x} \tag{3.17}$$

Both the training and test data for a recognition system can then be transformed by the matrix $\boldsymbol{R}_1$ so that the system only uses the principal component axes.

There are several points to note about this method. First, the calculation of $\boldsymbol{T}$ depends only on the training data, and not on any class definitions. Secondly, there is no attempt to use class information so the between-class scatter and within-class scatter are not considered directly. However, as between-class scatter normally dominates the within-class scatter in the total variance calculation the transform usually preserves class separation.

The principal components are dependent on parameter scale. If one component of the feature vector is multiplied by a scaling factor, this does not increase the amount of information it contains, but does increase its variance. Hence, the principal component may become dominated by such scaled elements.

A further problem with PCA is that the directions of maximum variance are not necessarily the directions of maximum discrimination [16]. Consider two Gaussian distributions in two dimensions (Fig.3.1) which have the same variance along the 1^{st} principal component axis (axis A). Clearly, PCA would retain this axis, and discard the other axis (B) which provides all of the discriminant information.

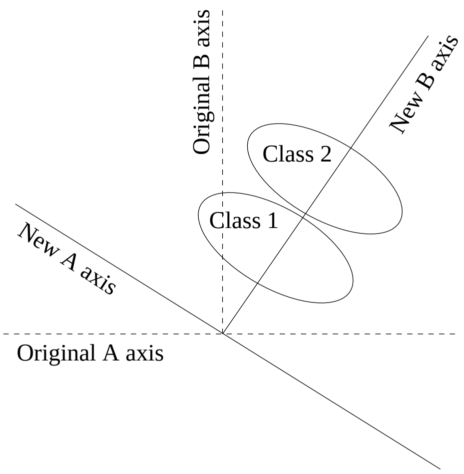

Figure 3.1: Potential problem with PCA - principal axes do not discriminate between classes.

3.6.2 Linear Discriminant Analysis

Linear discriminant analysis (LDA) can be used to increase the value of J_1 by using a linear transform to rotate and scale the original feature space into a new space. Note that since J_1 is invariant under non-singular linear transformations, it can only be changed by choosing a new space of different dimensions.

The transformation is expressed as,

$$\boldsymbol{y} = \boldsymbol{A}'\boldsymbol{x} \tag{3.18}$$

where $\boldsymbol{A}$ is the n x d transformation matrix (n is the original feature space dimensionality, d is the new space dimensionality).

LDA makes the following assumptions

1. Each class can be represented by a single Gaussian distribution,
2. All classes have an identical within-class covariance ($\boldsymbol{S}_w$),
3. The class centroids (means) can be represented by a Gaussian distribution with between-class covariance matrix ($\boldsymbol{S}_b$).

If it is assumed that the components of the feature vector are not statistically independent, the off-diagonal elements of the scatter matrices $\boldsymbol{S}_w$ and $\boldsymbol{S}_b$ are non-zero. This assumption is generally valid when the components of the feature vector are generated from a parameterisation of speech.

For LDA the aim is to directly minimise $tr(\boldsymbol{S}_w^{-1}\boldsymbol{S}_b)$, which can be achieved by an eigenanalysis of the matrices. The first stage in generating the transformation is to orthogonalise the within-class matrix by rotating it onto a set of axes in which the features become statistically independent of each other, using the Karhunen-Loeve transformation ($\boldsymbol{R}_1$). Rotating the within-class distribution $\boldsymbol{S}_w$ by $\boldsymbol{R}_1$ aligns the distribution along orthogonal axes corresponding to the eigenvectors of $\boldsymbol{S}_w$, with the within-class variance along each axis being equivalent to the corresponding eigenvalue (λ_i). By scaling after the rotation, the distribution of the within-class data in the new space can be made to be the identity matrix.

If the between-class scatter matrix is projected into this new space to give a distribution $\boldsymbol{G}$, an eigenanalysis can be performed to generate a further rotation so that the within-class and between-class distributions are projected into the same space and are both aligned along the same axes. Combining the rotations and scaling gives a single transformation, which is the desired linear transform $\boldsymbol{A}$. A full derivation of such a transform is given in section 4.4.1.

Applying this transformation to the set of within-class data, projects the data into the new space so that the all the features are uncorrelated and the variance of each feature for the within-class data is unity. The rotation of the between-class distribution by $\boldsymbol{A}$ decorrelates the features and aligns them along the axes of maximum discriminability. However, the measure J_1 is invariant under non-singular linear transformations (denoting transformed parameters by $\hat{\boldsymbol{S}}_w$,$\hat{\boldsymbol{S}}_b$, and $\hat{J}_1$),

$$\hat{\boldsymbol{S}}_w = \boldsymbol{A}'\boldsymbol{S}_w\boldsymbol{A} \qquad\qquad \hat{\boldsymbol{S}}_b = \boldsymbol{A}'\boldsymbol{S}_b\boldsymbol{A}$$
$$\hat{J}_1 = \quad tr(\hat{\boldsymbol{S}}_w^{-1}\hat{\boldsymbol{S}}_b) \quad = J_1$$

and the value of J_1 can only be altered by changing the dimensionality of the new space.

Rotating all data by $\boldsymbol{A}$ results in the within-class and between-class distributions being aligned along the same axes, with the within-class covariance known to be the identity matrix. By comparing the within-class and between-class variances in the new space (c.f. the F-ratio), a feature subspace can be selected to increase J_1. This is achieved by selecting features which maximise between-class scatter, whilst keeping within-class scatter well clustered. The between-class features which have a large variance are retained, and those with a small variance are discarded. The variances are directly related to the eigenvalues, γ_i, of the matrix $\boldsymbol{G}$, and these eigenvalues can be used to select an appropriate set of features. The trace of a matrix is the sum of its eigenvalues and denoting the measure J_1 in n dimensions as $J_1^{(n)}$,

$$J_1^{(n)} = tr(\boldsymbol{S}_w^{-1}\boldsymbol{S}_b) = \lambda_1 + \ldots + \lambda_n \tag{3.19}$$

The largest J_1 in d dimensions ($J_1^{(d)}$) is found by selecting the dimensions corresponding to the largest d eigenvalues.

$$J_1^{(d)} = tr(\boldsymbol{S}_w^{-1}\boldsymbol{S}_b) = \bar{\lambda}_1 + \ldots + \bar{\lambda}_d \tag{3.20}$$

where $\bar{\lambda}_i$ is the i^{th} largest eigenvalue of $\boldsymbol{S}_w^{-1}\boldsymbol{S}_b$. An n x d feature selection matrix $\boldsymbol{F}(=[f_{ij}])$ is defined to perform the reduction in dimensionality

$$f_{ij} = \begin{cases} 1 & \text{if } \gamma_i \text{ is } j^{th} \text{ largest variance and } j \leq d \\ 0 & \text{otherwise} \end{cases}$$

.

The LDA transform with dimensionality reduction is then applied to all speech vectors,

$$\boldsymbol{y} = \boldsymbol{F}'\boldsymbol{A}'\boldsymbol{x} \tag{3.21}$$

It can be seen that since LDA uses a scaling term to make the within-class variance of each feature unity in the new space, the selection of features is independent of scaling which is one of the problems with PCA. The feature selection criteria is based on the Fisher ratio, but in this case the features considered for selection are guaranteed to be independent.

3.6.3 The Use of LDA

LDA has emerged as a good method of producing a feature vector which is uncorrelated and of a computationally feasible size. The idea of using this technique for speech recognition was briefly mentioned by Hunt in a paper abstract in 1979 [51], but not actually implemented until much later when LDA was used with an auditory model [52]. This was extended to perform LDA on mel-scale filter bank outputs [53] and the resulting transform termed the IMELDA transform (Integrated Mel-scale representation using LDA), since it allowed heterogeneous sets of parameters to be combined into a single discriminant function. Initial experiments only considered template matching, and hence each class was assumed to have an identical within-class distribution with the template as the class centroid. The within-class covariance was found by comparing the examples of each word with the corresponding template and generating a pooled within-class distribution. The between-class distributions were calculated directly from the templates. These experiments showed that combining static and dynamic spectral information into a template of a fixed size performed better than using an equivalent number of static spectral parameters. This confirms the findings made in the implementation of feature selection that the dynamic spectral features can contribute important information to the class discrimination measure (this can also be demonstrated by recognition comparisons including/excluding dynamic spectral information in the feature vector [103]). Further work by Hunt [54, 55] confirmed that a similar approach is also viable for use in HMMs.

LDA is now used in many discrete [114] and continuous [16] density HMM systems. Most implementations follow a five step procedure:

1. Estimate models using original data,

2. Assign class labels to each frame (via recognition alignment),
3. Accumulate within and between-class statistics, and compute LDA transform,
4. Transform the data,
5. Retrain a new set of models on transformed data.

In general LDA is normally applied on a global basis using a single transform [115], but the definition of classes for accumulating statistics varies. Haeb-Umbach & Ney [48] investigated using different class definitions in a phoneme based recogniser and concluded that using states (phone-segments in their phraseology) as classes performed better than using basic phoneme classes. Parris & Carey [90] showed that reducing further to individual mixture components as classes could also produce good results. For computational reasons, e.g. accumulation of statistics, it is common to employ a class-independent transform [3, 47], but it may be beneficial to apply LDA on a more specific basis. This idea is examined in more detail in chapter 4.

3.7 Class-Specific Feature Vectors

In the previous sections it has been demonstrated that using appropriate feature selection/extraction methods the number of features can be reduced, often by a significant amount, without degrading recognition performance. All the methods examined considered the feature choice on a global level maximising the overall class separation. Attention is now focused on how such approaches can be implemented on a more specific basis so that highly confusable classes can be better separated.

3.7.1 Word Adaptive LDA

One of the limitations of global implementations of LDA is the assumption that all classes have a common covariance matrix. Ayer [4, 5] proposed a method of overcoming this limitation by applying a word-adaptive LDA transform (WALDA) which is computed using a gradient descent method on a word class basis. The set of states used to model a whole word (or set of words) is assigned to the same class and the transformation is derived by an iterative process based on maximising a measure of whole-word recognition performance. By weighting the training tokens according to their supposed confusability, the transform can be shown to perform the same function as LDA. Results of using WALDA on an alphabet database show a significant improvement over an IMELDA implementation. However, one of the limitations of WALDA is that it is vocabulary dependent. This limits its applicability to large scale tasks where word information is not always available in sufficient quantities in the training data. This is especially true for large vocabulary CSR systems which may have to recognise words which are never seen in the training data.

3.7.2 Individual Class Discriminants

Bocchieri & Doddington [13] proposed a method for applying an LDA approach on a more specific basis for template matching, which was later extended for use with HMMs. In standard template matching a Euclidean distance measure between the input speech $\boldsymbol{f}_{in}$ and the reference templates $\boldsymbol{f}_{ref}$ is used,

$$\| \boldsymbol{f}_{in} - \boldsymbol{f}_{ref} \|^2 \tag{3.22}$$

Initially Bocchieri investigated a global transform, computed by normalising (in the time domain) all the training data, and calculating an LDA transform using a global covariance matrix. The problems of LDA were noted, namely (1) all the frames were assumed to be statistically uncorrelated, and (2) the variability of all the input vectors is assumed independent of the acoustic event that produced it (global covariance). To overcome the correlation problem Bocchieri introduced frame stacking whereby after the global LDA transform, adjacent frames were combined to make a single frame. This approach is similar to the incorporation of time-differences into feature vectors now common in most HMM systems. Reference template specific transformations were used to attempt to resolve the second problem. After performing global LDA and frame stacking, a template specific transform ($\boldsymbol{T}_j$) was applied to both the input vector and reference template.

$$\| \boldsymbol{T}_j(\bar{\boldsymbol{f}}_{in} - \bar{\boldsymbol{f}}_{ref(j)}) \|^2 \tag{3.23}$$

where $\bar{\boldsymbol{f}}_{in}$ and $\bar{\boldsymbol{f}}_{ref(j)}$ are the input vector and reference template (for class j) respectively after global LDA and frame stacking. The template-specific transform was computed using eigen decomposition as in LDA, but using the covariance of the frames assigned to the class during training. i.e. the within class covariance for class j is

$$\boldsymbol{C}_j = \frac{1}{N_j}\sum_{k=1}^{N_j}(\boldsymbol{x}_k - \bar{\boldsymbol{f}}_{ref(j)})(\boldsymbol{x}_k - \bar{\boldsymbol{f}}_{ref(j)})' \quad \text{where } \boldsymbol{x}_k \in \text{class } j \tag{3.24}$$

The class-specific transform is chosen so that

$$\boldsymbol{T}_j\boldsymbol{C}_j\boldsymbol{T}'_j = \boldsymbol{I} \quad \text{where } \boldsymbol{I} \text{ is the identity matrix} \tag{3.25}$$

The actual derivation of such a transform will be detailed in chapter 4.

This frame-specific modelling improves the template matching by further decorrelating features for each class. This gives better recognition performance than using a global transform, even without a dimensionality reduction (note that, unlike HMMs, templates do not use covariance within the distance measure). A reduction in dimensionality may yield yet further improvement, but as the data is now statistically modelled on a template basis, using a standard approach of analysing between class distributions is no longer valid. Bocchieri proposes such a feature reduction by characterising the distributions of both correctly assigned frames and possible confusable frames, and feature reduction to maximally separate them. The confusable frames are identified via recognition of the training data to find frames which give a close match to the template, yet are not drawn from the same class. Having identified a set of within-class data $X = \{\boldsymbol{x}_1 \dots \boldsymbol{x}_n\}$ and a set of confusable vectors $Y = \{\boldsymbol{y}_1 \dots \boldsymbol{y}_m\}$ their corresponding distributions can be found. Due to the nature of the transformations used to compute X,

$$E(\boldsymbol{x}_k(i) - \bar{\boldsymbol{f}}_{ref(j)}(i)) = 1 \tag{3.26}$$

where $\boldsymbol{x}_k(i)$ and $\bar{\boldsymbol{f}}_{ref(j)}(i)$ are the i^{th} elements of the speech vector and reference template respectively, i.e. the expected individual contribution of each feature to the distance measure is unity. Thus, by computing the expected contribution of the confusable vectors, $E(\boldsymbol{y}_k(i) - \bar{\boldsymbol{f}}_{ref(j)}(i))$, those features which best contribute to separating the within-class and confusable vectors can be identified. This approach can be seen to be similar in principle to the discriminative feature selection discussed in section 3.5.3.

Using the global rotation, frame stacking and template specific feature selection is clearly advantageous. From a base of 16 features per frame, which gave 2.5% error on isolated digits, the global rotation and stacking reduced the error to 1.4%. Adding the final template specific selection further reduced the error to 0.6% using a final speech frame of 8 features. Using the template specific transform clearly improves the modelling capabilities of the templates through better identifying which components of the feature vector can best characterise a class whilst at the same time discriminating between similar classes. However, this gain is at a substantial computational cost, only 16 multiply-adds are needed for each distance measure calculation in the original system, but this rises to 132 multiply-adds in the best final system.

3.7.3 Confusion Discriminants in HMMs

The template specific model has been extended to continuous density HMMs by Doddington [35, 36]. The HMM system set up was aimed at simulating the template approach. The number of states was chosen to correspond to the average length of each speech event (digits) so that on average each state would match exactly one frame. Each state was modelled by a single Gaussian distribution, and the state likelihoods were computed using a Euclidean distance. The Euclidean distance measure can be used since a state-specific scaling of the feature vector ensures that the variance of each feature is unity. A global transform generated from the global covariance is first used to transform all the vectors to decorrelate features and reduce dimensionality. For the confusion discriminant (state-specific) transform within-class and confusable class distributions are computed. The within-class distribution is taken directly from the trained model states, and the confusable class distribution is determined using a three step procedure. First, all frames in the training data are allocated a “correct” state using a supervised maximum likelihood alignment. Second, an unsupervised alignment which is biased against the correct sequence is made. Finally, the two alignments are compared and where a frame is assigned to different states in the two alignments a confusion is identified. The frame from the supervised alignment is added to the confusion data for the state identified in the confusable alignment. Using the within-class and confusable distributions the confusion discriminant transform can be computed for each state in the same manner as the template transform used above. The transform decorrelates the features and those dimensions in which the confusion distribution contains less variance than the within-class distribution can be discarded. The distance measure used for the log likelihood calculation for input vector $\boldsymbol{x}_t$ state j is:

$$D = \| \boldsymbol{S}_j^{-\beta} \boldsymbol{T}_j (\boldsymbol{x}_t - \boldsymbol{\mu}_j) \|^2 \tag{3.27}$$

where $\boldsymbol{T}_j$ is the confusion discriminant, $\boldsymbol{\mu}_j$ is the state mean and $\boldsymbol{S}_j$ is a diagonal scaling matrix used to weight each feature by its standard deviation, and β is an arbitrary power. Although Doddington reported experiments on a connected digit database showing that confusion discriminant transforms reduced the word error rate from 0.8% without transforms to 0.5% with transforms it was later discovered that there was an error in the implementation [35]. In revised experiments it was found that a value of $\beta = 0.5$ was optimal and that dimensionality reduction using the confusion discriminant gave some reduction in error.

Two major problems can be identified in Doddington's approach. First, biasing the recognition pass to force errors for the confusion data can lead to the identification of 'false' errors, i.e. those which would not normally occur in recognition. Generating the transforms based on these errors may not lead to better class separation. Secondly, the distance measure used to calculate the likelihood is not consistent. It can be seen to be a corrupted form of the standard HMM probability density function. Woodland [105] noted that the normalisation term for a Gaussian p.d.f., namely

$$\frac{1}{(2\pi)^{d/2} |\boldsymbol{\Sigma}|^{1/2}} \tag{3.28}$$

(where d is the number of dimensions in the speech vector) makes a contribution to the log likelihood of

$$-\frac{d}{2} \log 2\pi - \log |\boldsymbol{\Sigma}|^{1/2} \; . \tag{3.29}$$

Since the covariance for the within-class data is diagonal (ensured by the transformations) the state dependent term can be written as:

$$-\sum_{i=1}^{d} \log \lambda_i^{1/2} \tag{3.30}$$

where λ_i is the i^{th} eigenvalue of the within-class covariance matrix. This is clearly not the same normalisation used by Doddington in which the standard deviation is used to scale the contribution of each feature. Each feature should be normalised by a fixed amount dependent on the variance of the feature in the transformed case. A comparison between the modified normalisation and the corrupted normalisation shows that significant gains are made by using the modified approach [105].

Woodland approached the gathering of confusion data in a different manner to Doddington. Instead of biasing an unsupervised recognition pass to create errors, possible confusable frames are identified by comparing the log likelihoods of frame/state pairings and including

No. of Dimensions	Frame Confusions	Viterbi Confusions
24	8.9	8.9
20	8.2	8.0
16	10.1	7.9
12	10.4	8.9
9	12.5	9.3

Table 3.2: Speaker Independent results quoted by Woodland on the E-set using a state based discriminant transform with correct Gaussian normalisation [105] (% error)

near misses. Two methods for accumulating confusable frames were used, a frame-based one where, similar to Doddington, incorrect frame assignments are identified, and a second where confusions are determined on a whole utterance basis using the utterance likelihoods.

For the frame based approach, the likelihood of the frame being generated by the best state in the correct model was identified. The likelihood of any other state generating the frame was compared and any which fall within a given threshold were deemed confusable. This allows confusable frames to be identified even if the whole frame sequence is not confusable. For the second approach, a Viterbi alignment of each isolated utterance with each potentially confusable model was generated. If the overall likelihood of the model matching the utterance is within a threshold of the likelihood of the correct model matching the utterance, the whole phrase is deemed confusable. The alignment of the utterance to the incorrect model is used to assign the frames to states to generate the confusion distributions.

An evaluation of this method on the British E-set (which consists of 8 letters B, C, D, E, G, P, T, V) using a 24 dimensional vector (12 MFCCs + first differentials) shows that both methods of gathering confusions can show improvements, with the Viterbi confusion collection being substantially more robust in lower dimensional feature spaces (Table 3.2).

3.8 Summary

Methods of improving discrimination in HMMs have been discussed in this chapter. Approaches using training methods have looked at correcting some of the invalid assumptions for HMMs, but have been seen to be limited in scope. These methods are either difficult to implement, for example estimating MMI parameters on large systems, or based on ad-hoc methods optimising an arbitrary measure. Approaches which manipulate the feature vector have shown that selecting an appropriate feature space can lead to significant improvements in recognition performance. Feature extraction methods based on transformations have been seen to be successful, both in terms of reducing computation and improving discrimination.

The state-specific transforms reported by Doddington and Woodland appear to be effective in selecting a more discriminative feature space, allowing better class separation.

These transforms incorporate many of the ideas from feature selection and training algorithms which are aimed at improving discrimination. The selection of feature dimensions based on the ratio of within-class and between-class variances is the basis for many feature extraction methods (e.g. the Fisher ratio), and the use of a recognition pass on the training data to identify errors is common amongst many of the discriminative training approaches.

Both implementations of the confusion discriminant transform have considered small scale recognition tasks and used simple algorithms to compute the confusion distributions. The extension to larger and more complex systems is certainly more difficult but given the success of the reported experiments, applying the transforms to larger tasks should be investigated. Such an investigation is the subject of chapters 4 and 5.

Chapter 4

Transformations for Discriminative Feature Selection

The selection of a good set of features for the observation vectors is crucial for high accuracy recognition. In this chapter, methods of feature extraction using linear transforms for application to continuous speech are investigated. The aims of selecting a smaller feature space are twofold :-

1. To improve recognition performance
2. To reduce computational cost during recognition.

Initially, methods of reducing the dimensionality of the observation feature vector on a global basis are examined. Both a feature selection and a feature extraction method are implemented on a continuous speech task, and the effects and problems of the methods discussed. In an aim to improve upon the global approach, a state specific approach termed confusion discriminant analysis (CDA) is presented and discussed in terms of application to continuous speech tasks.

4.1 Model Sets for Evaluation

The feature selection methods discussed in this chapter have been evaluated using HMMs with a single Gaussian density per state and a full covariance matrix.

Two model sets, R_{full} and R_{grand}, were defined for speaker independent Resource Management (RM) experiments (appendix A). These model sets are identical except that R_{full} has a separate covariance for each state, while R_{grand} is a grand covariance model set (a single covariance matrix is shared between all states).

For each model set, 48 phone models corresponding to the CMU phone set [71] were defined. Each model had 3 emitting states and a left to right topology. The number of models was increased to 130 by including function-word dependent phone models for the

32 most common function words in the RM training data (Table A.1, appendix A). A full covariance monophone system was trained using Baum-Welch re-estimation with all 3990 training utterances.

4.1.1 Baseline Results for R_{full} and R_{grand}

The baseline results for the R_{full} and R_{grand} model sets on the RM speaker independent test sets are shown in Table 4.1.

Test Set	R_{full} models	R_{grand} models
Feb'89	9.6	20.8
Oct'89	9.9	18.8
Feb'91	8.6	18.0
Sep'92	14.0	26.2

Table 4.1: Baseline results of models on RM speaker independent test sets (% word error).

4.2 A Global Feature Selection Method

Before the feature extraction methods are considered, a feature selection method is examined to assess the need for more complex methods of reducing the size of the feature vector.

The method proposed by Bocchieri & Wilpon [15] was implemented using the 39 element feature vector in the R_{full} model set. Errors were identified for the feature selection errors using an N-Best recognition pass with single mixture diagonal covariance models. For each error that occurred, the erroneous frame $\boldsymbol{f}_t$ and the state it was aligned with s_t were identified. The average distance measure $D(i)$ was calculated for each of the 39 features ($i = 1 \ldots 39$).

$$D(i) = E\left[\frac{[\boldsymbol{f}_t(i) - \boldsymbol{\mu}_{s_t}(i)]^2}{\sigma_{s_t}(i)}\right] \tag{4.1}$$

The features were rank ordered according to D(i) and the d best features selected (high D(i) are retained, low D(i) are discarded).

Diagonal covariance HMMs were used to calculate D(i) and select the features. The full covariance models were reduced to use the selected features (note that the reduction is simple selection, no scaling or rotation is involved).

A state-based selection using the same criteria has been examined, but this produced extremely poor results for the diagonal model set, and was not implemented on the full model set. The poor results were due to many states not being assigned sufficient confusion data to estimate the distance measure robustly.

The results of the global feature selection on the full covariance model R_{full} set (Fig. 4.1) show that no improvement in recognition rate can be achieved by reducing dimensionality (using 36 features gives equivalent performance to the baseline). For the diagonal model set it was possible to get a improvement at all dimensions down to 22, showing that the selection measure is model set dependent.

A possible problem with this approach is the identification of errors, and their relevance to the recognition. Here, errors have been identified on one model set, and used to correct another. The assumption for the discriminant feature selection theory was that all the features were independent, hence the diagonal models. This is a poor assumption and when correlations are modelled using full covariance models a different feature set should be selected.

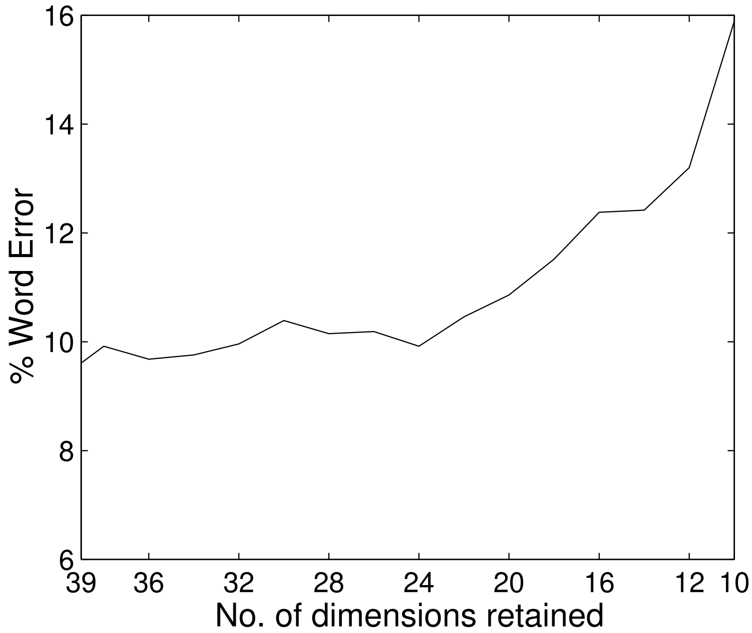

Figure 4.1: Simple feature selection using the approach proposed by Bocchieri & Wilpon. Results are % word error on Feb'89 using R_{full} models.

The results demonstrate that a simple feature selection cannot be implemented without a loss in performance. The selection does not try to separate classes, but chooses the set of features which will perform the best classification of the correct speech frames. In fact this method selects features based on a MMI measure between the state distributions and the frames assigned to the states during collection of statistics.

A reduction to 24 features is possible without a large fall in performance, suggesting that with more complex feature selection, a larger dimensionality reduction may be obtained.

4.3 Global Feature Space Selection Using LDA

Linear Discriminant Analysis (LDA) is widely used to generate reduced feature vectors. There are several assumptions made in the implementation of LDA which are not usually valid. These assumptions are examined in the following section and the effect of LDA on the selection a discriminative feature space from the standard observation vector coding is examined.

4.3.1 Analysis of Global LDA Assumptions

It is assumed that all classes share a common within-class covariance for a global LDA implementation. The validity of this assumption in terms of the R_{full} models was considered by performing an eigenanalysis to determine the relative shapes of the distributions for different classes. When considering the distributions of two classes ω_1 and ω_2, if the eigenvalues of distribution ω_2 are scalar multiples of ω_1, then the distributions may be of the same shape but different sizes and orientations. However, if the ratio of the largest eigenvalue to the next largest is different for the two classes then the distributions must have different shapes.

Model/State	λ_1	ratio λ_1/λ_2	Model/State	λ_1	ratio λ_1/λ_2
AE (state 2)	127.078	1.140	B (state 2)	90.941	1.181
AE (state 3)	142.148	1.121	B (state 3)	160.576	3.614
AE (state 4)	140.996	1.297	B (state 4)	81.569	1.096
OY (state 2)	173.434	1.542	CH (state 2)	111.929	1.298
OY (state 3)	149.898	1.222	CH (state 3)	72.721	2.022
OY (state 4)	157.268	1.570	CH (state 4)	107.850	1.739

Table 4.2: Comparison of eigenvalues for in-class distributions of states of R_{full} models. λ_1 is the largest eigenvalue and λ_2 is the next largest.

Table 4.2 shows these ratios for states belonging to a variety of models. It is clear that the shapes of the distributions are different even for distributions within the same model. An eigen analysis of the grand covariance distribution in R_{grand} gives a λ_1/λ_2 ratio of 1.123, significantly lower than the majority of individual states. Computing the average eigenvalues over all state distributions and comparing against the eigenvalues of the grand covariance shows significant differences in the values of the larger eigenvalues (Fig. 4.2) indicating that the distributions have different sizes as well as different orientations. Thus the LDA assumption of a common covariance for all classes is poor. This is confirmed by the baseline results using a grand covariance system (model set R_{grand}) which is considerably worse than the R_{full} model set (section 4.1.1). LDA also assumes a single Gaussian distribution per class, however the results using models with multiple mixture densities (section 8.1) shows that the classes represented by the HMM states are modelled better by multimodal distributions.

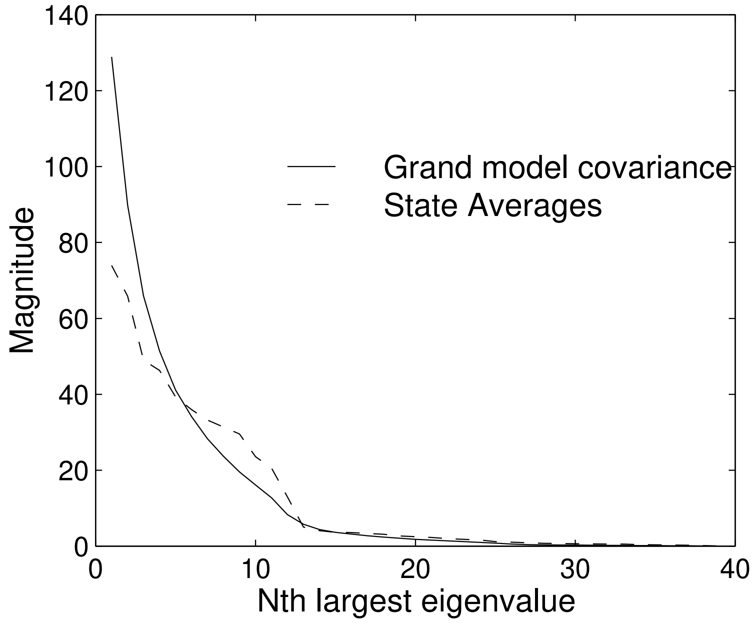

Figure 4.2: Comparison of eigenvalue distributions for grand covariance and individual model covariances (eigenvalues computed for each distribution and then averaged)

The results of this eigenvalues analysis indicate that the LDA assumptions are incorrect. To examine how this affects the performance a global LDA transform was used to transform the model sets.

4.3.2 Global LDA on RM Models

The grand covariance model set (R_{grand}) was used to compute the LDA distributions, using the grand covariance as the in-class distribution $\boldsymbol{C}_w$. The between class covariance ($\boldsymbol{C}_b$) of the HMM states was calculated by accumulating the individual state means,

$$\boldsymbol{C}_b = \sum_i \boldsymbol{\mu}_i \boldsymbol{\mu}_i' \tag{4.2}$$

where i indexes every HMM state.

The standard LDA transform was generated using $\boldsymbol{C}_w$ and $\boldsymbol{C}_b$ and feature reduction performed to generate an $n \times d$ transform $\boldsymbol{R}$. The parameters of each individual model state in the R_{full} model were transformed and reduced to a lower dimensionality,

$$\boldsymbol{\Sigma}_r = \boldsymbol{R}' \boldsymbol{\Sigma}_{full} \boldsymbol{R} \tag{4.3}$$

$$\boldsymbol{\mu}_r = \boldsymbol{R}' \boldsymbol{\mu}_{full} \tag{4.4}$$

where $\boldsymbol{\Sigma}_r$ is the reduced covariance and $\boldsymbol{\mu}_r$ the reduced mean, each of dimension d. The Gaussian normalisation term in each state was also reestimated using the value of the new covariance matrix.

The n dimensional speech observation vectors ($\boldsymbol{x}$) are projected into the d dimensional LDA space by the same transformation,

$$\boldsymbol{y} = \boldsymbol{R}'\boldsymbol{x} \tag{4.5}$$

and $\boldsymbol{y}$ is used as the observation vector in further processing.

Implementing the transform in this way removes the need to reestimate the reduced dimension models from scratch, however, it also relies on the assumption that the frame/state alignment for training is the same for both the full and reduced covariance models. The effect of reestimating the models using further training iterations shows only a small change in model performance so this assumption is reasonable (Fig.4.3).

The global LDA transform gives a substantial dimensionality reduction without loss of accuracy. A reduction to 30 dimensions results in the best performance (9.3% - without training), and further reductions gradually degrade the performance. All dimensions lower than 30 give worse performance than the full feature set.

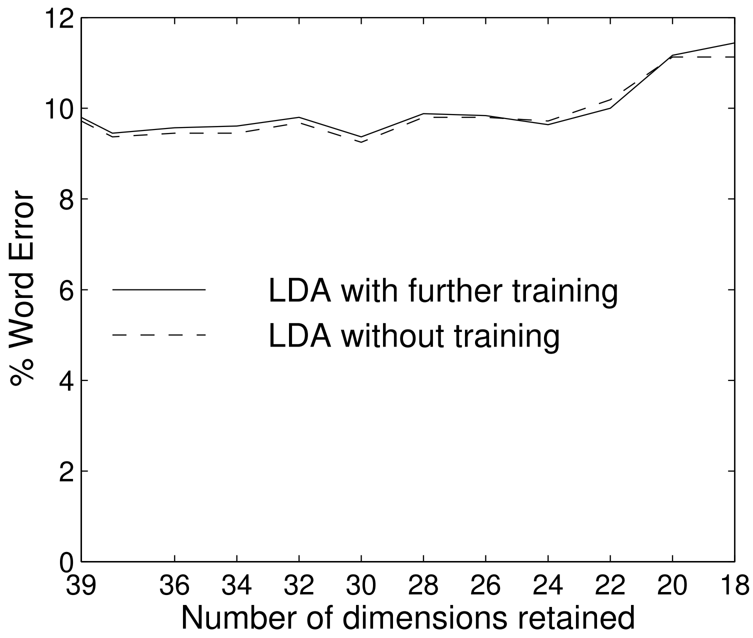

Figure 4.3: Results of global LDA feature extraction using R_{full} models on Feb'89 test set with and without further training

4.4 State-Based Feature Space Selection Using Confusion Discriminant Analysis (CDA)

Confusion Discriminant Analysis (CDA) performs much the same function as LDA in rotating the feature space to the best orthogonal axes to find the most discriminative features. The nature of the data used in deriving the transform, however, is ideally suited for application at the state level.

The transform is derived by identifying potentially confusable data and using this to estimate an out-of-class distribution for each class. In a similar manner to LDA, the original feature space is projected into a new space in which the in-class distribution is based on orthogonal axes. By projecting the confusable data onto the same axes a comparison of the suitability in terms of discriminative ability of each of the axes in the new space can be made. Those axes which are most representative of the in-class data, in comparison to the confusable data, are retained. The axes in which it is difficult to separate the in-class and out-of-class are discarded. As the out of class data is class dependent, the approach necessitates the use of class specific transforms.

4.4.1 Derivation

Following the notation of Woodland [105] the CDA transform is derived as follows. Given an in-class covariance matrix $\boldsymbol{C}$ and an out of class confusion covariance matrix $\boldsymbol{B}$ about the in-class mean, both matrices are diagonalised simultaneously.

The transformation $\boldsymbol{T}_1$ is used to decorrelate all the features and normalise the variances of all the features in the within class matrix to unity.

$$\boldsymbol{T}_1' \boldsymbol{C} \boldsymbol{T}_1 = \boldsymbol{I} \tag{4.6}$$

where

$$\boldsymbol{T}_1 = \boldsymbol{R}_1 \boldsymbol{\Lambda}^{-\frac{1}{2}} \tag{4.7}$$

and $\boldsymbol{R}_1$ and $\boldsymbol{\Lambda}$ are the eigenvalue and eigenvector matrices of $\boldsymbol{C}$ respectively.

The confusion data is also rotated using the transformation $\boldsymbol{T}_1$, giving the resulting matrix $\boldsymbol{S}$,

$$\boldsymbol{S} = \boldsymbol{T}_1' \boldsymbol{B} \boldsymbol{T}_1 \tag{4.8}$$

$\boldsymbol{S}$ is then diagonalised using $\boldsymbol{R}_2$ the eigenvector matrix of $\boldsymbol{S}$,

$$\boldsymbol{R}_2' \boldsymbol{S} \boldsymbol{R}_2 = \boldsymbol{D} \qquad (\boldsymbol{D} \text{ is diagonal}) \tag{4.9}$$

Under the same rotation $\boldsymbol{R}_2$, the normalised orthogonalised in-class matrix remains the identity matrix, thus the transformation

$$\boldsymbol{T} = \boldsymbol{R}_1 \boldsymbol{\Lambda}^{-\frac{1}{2}} \boldsymbol{R}_2 \tag{4.10}$$

transforms both in-class and out of class data to have uncorrelated features and be aligned along the same axes. These rotations do not affect the class separability measure J_1 so the discriminative ability of the classes is unchanged. By discarding those axes in which the confusable data and the in-class data are difficult to separate, the measure J_1 will be a maximum for the new reduced dimension space, and lead to better class separation.

Assuming that both the in-class and out-of-class distributions are Gaussian, features are selected on the basis of effectiveness in the Gaussian p.d.f. The features which will increase the likelihood difference between in and out of class data are those which have a confusion data variance greater than unity in the transformed feature space. The in-class variance is unity in the new space and the out-of-class variances are the eigenvalues of $\boldsymbol{S}$. Therefore the dimensions with the largest eigenvalues of $\boldsymbol{S}$ should be retained.

Features are selected by using an n x d selecting matrix $\boldsymbol{F}$ (n is the size of the original feature space and d the size of the reduced feature space). Element (i, j) of $\boldsymbol{F}$ has a value of one if the i^{th} dimension of the original space is to become the j^{th} dimension of the reduced space and zero otherwise. The transformation for rotation, scaling and feature selection becomes,

$$\boldsymbol{T} = \boldsymbol{R}_1\boldsymbol{\Lambda}^{-\frac{1}{2}}\boldsymbol{R}_2\boldsymbol{F} \tag{4.11}$$

If the speech vectors in the original feature space are $\{\boldsymbol{x}_1 \ldots \boldsymbol{x}_k\}$ the vectors after transformation (without scaling) into the subspace are $\{\boldsymbol{y}_1 \ldots \boldsymbol{y}_k : \boldsymbol{y}_i = \boldsymbol{F}'\boldsymbol{R}_2'\boldsymbol{R}_1'\boldsymbol{x}_i\}$. The covariance of the in-class data in the transformed subspace is $\boldsymbol{\Sigma}_t$,

$$\boldsymbol{\Sigma}_t = \frac{1}{N}\sum_i \boldsymbol{F}'\boldsymbol{R}_2'\boldsymbol{R}_1'(\boldsymbol{x}_i - \boldsymbol{\mu})(\boldsymbol{x}_i - \boldsymbol{\mu})'\boldsymbol{R}_1\boldsymbol{R}_2\boldsymbol{F} \tag{4.12}$$

where N is number of data samples in the class, and $\boldsymbol{\mu}$ is the mean of the in-class data. Noting that

$$\frac{1}{N}\sum_i (\boldsymbol{x}_i - \boldsymbol{\mu})(\boldsymbol{x}_i - \boldsymbol{\mu})' = \boldsymbol{R}_1\boldsymbol{\Lambda}\boldsymbol{R}_1' \tag{4.13}$$

and $\boldsymbol{R}_1$ is orthogonal, this simplifies to

$$\boldsymbol{\Sigma}_t = \boldsymbol{F}'\boldsymbol{R}_2'\boldsymbol{\Lambda}\boldsymbol{R}_2\boldsymbol{F} \tag{4.14}$$

The covariance $\boldsymbol{\Sigma}_t$ should be used in the constant term of the probability calculation.

4.4.2 Computational Considerations

Without transformations the probability density function of a particular observation vector $\boldsymbol{x}$ belonging to a Gaussian class is:

$$p(\boldsymbol{x}) = \frac{1}{(2\pi)^{\frac{n}{2}} \mid \boldsymbol{\Sigma} \mid^{\frac{1}{2}}} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_i)'\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_i)} \tag{4.15}$$

where $\boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma}$ are the mean vector and covariance matrix of the class distribution, and n is the dimensionality of the feature space.

In a straightforward implementation of the p.d.f. $n(n+1)+1$ multiplications and $\frac{n(n+1)}{2}+1$ additions are required to estimate the exponential expression (using the symmetry of the covariance matrix). However if $\mathbf{\Sigma}^{-1}$ is reduced by a Choleski decomposition,

$$\mathbf{\Sigma}^{-1} = \boldsymbol{L}\boldsymbol{L}' \tag{4.16}$$

the exponential part of the probability density becomes,

$$(\boldsymbol{x}-\boldsymbol{\mu}_i)'\mathbf{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_i) = ||\boldsymbol{L}'(\boldsymbol{x}-\boldsymbol{\mu}_i)||^2 \tag{4.17}$$

Only $\frac{n^2+n}{2}$ multiplications are required for the matrix multiplication, and n multiplications for the squared distance calculation.

When the state distribution is transformed by the CDA transform (without selection) the term $(\boldsymbol{x}-\boldsymbol{\mu}_i)'\mathbf{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_i)$ in the p.d.f. calculation can be rewritten as,

$$(\boldsymbol{x}-\boldsymbol{\mu}_i)'(\boldsymbol{R}_1\mathbf{\Lambda}\boldsymbol{R}_1')(\boldsymbol{x}-\boldsymbol{\mu}_i) \tag{4.18}$$

using $\boldsymbol{T} = \boldsymbol{R}_1\mathbf{\Lambda}^{-\frac{1}{2}}$ this becomes

$$\| \boldsymbol{T}'(\boldsymbol{x}-\boldsymbol{\mu}_i) \|^2 \tag{4.19}$$

Thus the feature space has been transformed so that all the components in the transformed feature space are linearly independent with respect to each other. The feature reduction can be included and the probability density for the new reduced feature space is given by,

$$p(\boldsymbol{x}) = \frac{1}{(2\pi)^{\frac{d}{2}} \mid \mathbf{\Sigma}_t \mid^{\frac{1}{2}}} e^{-\frac{1}{2}\|\boldsymbol{F}'\boldsymbol{T}'(\boldsymbol{x}-\boldsymbol{\mu}_i)\|^2} \tag{4.20}$$

For the transformed calculation, the selection of d features from the input vector requires n multiplications and $n+1$ additions per feature, and the vector norm calculation requires d multiplications, plus $d-1$ additions. To implement the method thus requires a total of $d(n+1)$ multiplications and $d(n+2)-1$ additions.

Assuming that multiplication is the most computationally expensive factor, the transformed calculation results in a ratio of $d:n$ saving on calculation over a direct implementation of the untransformed vector calculation, but for a saving over the Choleski decomposition approach $d \leq \frac{n}{2}$ is necessary.

In terms of storage the transform matrix requires $d \times n$ entries, as opposed to the $\frac{n^2}{2}$ entries required by the symmetric covariance matrix. A reduction to half the number of dimensions would result in the same storage requirements.

4.4.3 Relationship to MMIE

The method has strong similarities to the MMIE training approach. An attempt is made to maximise the probability of recognising the frames of the correct speech class, whilst reducing the probability of recognising frames from other classes. In effect the mutual information between the in-class acoustic events and the model state in the new subspace is maximised by discarding those dimensions which contribute little mutual information.

In section 3.2.1 it was shown that MMI optimisation effectively shifts the parameters in the direction of the MLE estimate but compensates in directions corresponding to confusable data. CDA starts with MLE estimates and selects a set of axes which least represent the confusable data. Hence, instead of compensating the parameter estimates for confusions, those directions which would require most compensation are discarded.

4.4.4 Relationship to LDA

The main differences between CDA and LDA are the generation of the out of class distribution, and that CDA is implemented on a class specific basis. For LDA the distribution may be found on a global basis by considering all of the available speech data and the classes to which they should be assigned. For CDA a distribution of confusable data needs to be generated separately for each class. Generating such distributions on an individual class basis is the major problem in implementing CDA.

4.5 Applying CDA to Continuous Speech Tasks

The reason for considering CDA is that it can be applied on a state basis to select particular characteristic traits of the class being modelled. Thus a state-based distribution which specifically selects the optimal features individually for each state should theoretically perform better than the global LDA approach. In order to apply the transform to maximum effect a distribution representing confusions with each state must be generated.

The term 'confusion' can be used to refer to many different concepts in the field of speech recognition, so a clarification of how it is defined here is necessary.

Given a distribution d representing an acoustic event (X), an acoustic observation $\boldsymbol{o}$ is deemed confusable with d if $\boldsymbol{o}$ is generated by an acoustic event other than X, yet may be assigned to d during recognition.

By identifying typical confusions within a state, the subspace of the feature space which best separates the in-class and confusable data can be determined. The problem examined in the remainder of this chapter is how to determine appropriate distributions for generating the state based CDA transforms. This will be approached as follows: first the confusion gathering methods used for isolated speech will be reviewed; the problems of extending such approaches to continuous speech will be detailed; further problems of identifying confusions are discussed; finally, an approach using global distributions at the state level is examined.

4.5.1 Confusion Gathering Approach for Isolated Words

As mentioned in chapter 3, CDA has previously been implemented by Woodland [105], on the isolated British E-set task which consists of 8 highly confusable letters (B, C, D, E, G, P, T, V). In those experiments a single 15-state HMM was used for each letter, with 6 states representing the consonant section and 9 states tied among all models to represent the common vowel sound. For the best confusion gathering method, each utterance in the training data was aligned against every model, and the Viterbi likelihood recorded. Those models which gave a likelihood within a threshold of the likelihood of the correct model were considered as confusable models, and the alignments of the speech frames to states was used to allocate confusion data. If a frame $\boldsymbol{f}$ was allocated to state s in the incorrect model, $\boldsymbol{f}$ was pooled into the confusion data for s.

After the alignment was complete, the pooled confusion data for each state was used to generate a confusion distribution for the state which was used in CDA to reduce the dimensionality of the feature vector. This proved effective and allowed the number of dimensions to be reduced whilst obtaining a reduction in error rate for the speaker independent test set. Reducing the original 24 dimensional feature vector to a 16 dimension feature vector resulted in the error rate dropping from 8.9% to 7.9%, and a 12 dimensional feature vector gave equivalent performance to the original vector. The E-set is a highly confusable database and if a reduction to half the number of original dimensions can be achieved on this task, other tasks may allow even greater reductions. However, the E-set is also a very small vocabulary task and thus may require only a small number of features to discriminate between classes.

There are two aspects to note in this confusion gathering process. First is the assumption that all models are confusable, all models are aligned and used for determining confusable data. The second is the use of confusability thresholds, which are used to ensure that enough data is present to generate a robust estimate of the distribution.

4.5.2 Problems of Extending to Continuous Speech Tasks

Extending the Viterbi confusion data collection method to continuous speech increases the complexity of the task by several orders of magnitude. The E-set task is small, requiring the use of only 8 different HMMs, and each utterance was known to consist of exactly one of the 8 E-set letters. Enumeration of all combinations of model/utterance was a simple and computationally feasible task. With continuous speech, not only are there more possible models (e.g. $40-50$ models just for the basic monophone set) and more speech frames in each utterance, but there is a variable number of words in the utterance. It becomes computationally infeasible to enumerate model sequences and utterances, even if the number of words in the utterance is predetermined.

For the extension to continuous speech there are several factors to consider:

- Computational feasibility,
- At what level to identify confusions (model level, word level, sentence level),
- How to incorporate insertion/deletion errors into the confusion framework,
- How to determine confusion distributions (what are confusable frames for a state),
- What other factors (e.g. grammar constraints) can restrict possible confusions.

The generation of the transform needs to be computationally feasible in terms of training time. As the calculation is a once only post-training step, absolute speed is not of great importance, but should be considered relative to the actual training time of the system.

Woodland investigated both frame level and word level approaches for identifying confusions and found that word level confusion gathering was significantly better in terms of robustness and recognition accuracy. Extending to continuous speech gives many more possible levels at which to determine confusions. Any point in the recognition hierarchy

$$\text{Frames} \Leftrightarrow \text{State} \Leftrightarrow \text{Phone Models} \Leftrightarrow \text{Words} \Leftrightarrow \text{Sentence}$$

can be used.

The decision is somewhat complicated by other factors which apply at recognition time, for example phone sequences must make up words, words must follow the language model, confusable sequences may be pruned out of a recognition search. Gathering confusions at the sentence level inherently incorporates all these factors, however, this leads to further problems in enumerating alternative sentences and in the type of confusions that can occur. Allowing large degrees of freedom for confusion comparisons leads to the problem of how to treat insertion and deletion errors in terms of confusion gathering. There is also the further problem that CDA requires confusion distributions on a state level, thus if confusions are identified at higher levels a mapping to assign data to individual states is required. The distributions must also be robustly estimated so any method should ensure that a sufficient number of confusable frames are identified for each state. For continuous speech the acoustic models have to model much more intra-class variability, which leads to many overlapping class distributions. The directions of maximum separability in these cases is more difficult to determine.

4.5.3 Simulating Isolated Confusions for Continuous Speech

One of the problems of extending to continuous speech is the larger number of models and increased size of the parameter set. This may limit the effect of the CDA implementation due to the difficulty of determining appropriate dimensions to separate the current class from all potentially confusable classes. To determine the extent of this problem the isolated word confusion gathering approach was examined in the context of continuous speech.

For the purposes of this experiment the TIMIT database (appendix A) was selected for a phone classification task. The detailed labelling of this database allows individual phone models to be aligned between the start and end points of each phone segment. By treating each segment as an isolated phone, Viterbi alignments can be generated and confusable models identified. Thus a direct correspondence to the confusion gathering on the E-set can be achieved. The effect of the confusion gathering is assessed by performing phone classification with the reduced dimensionality models. Confusions were gathered from the training data in a similar manner to Woodland.

To ensure a sufficient number of confusions, different thresholds for the Viterbi confusion scores were used. Observation sequence $\boldsymbol{O}$ is confusable for all alternative models m_a, if

$$\mathcal{F}(\boldsymbol{O}|m_t) - \mathcal{F}(\boldsymbol{O}|m_a) < \delta \tag{4.21}$$

where δ is the confusion threshold, $\mathcal{F}(\boldsymbol{O}|m_t)$ is the log likelihood of generating $\boldsymbol{O}$ using the true model m_t (according to the labelling), and $\mathcal{F}(\boldsymbol{O}|m_a)$ is the log likelihood of generating $\boldsymbol{O}$ with m_a.

number of	Training	Testing Set		
features	$\delta = 0$	$\delta = 0$	$\delta = 10$	$\delta = 25$
39	66.7	64.1	64.1	64.1
36	67.8	64.8	65.3	65.0
34	67.7	64.6	64.8	65.0
30	68.1	64.5	65.0	64.7
28	68.0	64.1	64.6	64.5
26	67.3	63.4	64.0	64.3
24	66.9	63.3	63.7	64.0
22	66.6	63.0	63.4	63.9
20	67.1	62.9	64.0	64.2
18	66.8	62.0		64.0
16	66.6	61.0		63.7
14	65.7	60.2		62.8
12	64.8	59.2		62.1

Table 4.3: Isolated confusions simulated on TIMIT database using phone classification. δ is the confusion threshold. Models are single Gaussian monophones with 39 element observation vectors. Results are percentage of phones classified correctly.

This simulation shows that gathering confusions in a similar manner to that of isolated words can improve the performance (Table 4.3) on both training and test data. The use of a threshold allowing near confusions as well as true confusions makes the estimation more robust and enables greater dimensionality reduction for the test set. A small reduction to 36 dimensions improves phone classification rates, and a reduction down to half the original number of dimensions maintains the same performance as the full feature set.

This result shows that CDA can determine good dimensionality reductions even with an increased number of models which is encouraging for the application to general continuous speech tasks. The confusion gathering approach used here is computationally very expensive, requiring an enumeration of all models and all labelled speech segments. In other databases, such segmentation is not available (e.g. databases labelled at word not phone level), and the size of the tasks (number of words, amount of training data) make the enumeration unmanageable.

4.6 State-Based CDA using Global Distributions

Before considering methods of attempting to identify specific confusion data for each state, an examination of the effect of using the same distribution for each state is considered. This gives an idea of the possible dimensionality reductions which may be achieved with appropriate state-specific confusion distributions.

Implementing CDA on a state basis using global distributions performs exactly the same function as LDA but the feature space is state specific since the in-class distributions used are state dependent. In theory this should improve classification on a state basis. The main problem is to determine which between-class distributions should be used to compute the transform. The within-class distribution of the state is known (from training). The global measures which can easily be computed are

(a) Distribution of class centroids (means)

(b) In-class distributions for all other classes

(c) Distribution of data as a whole

These measures give four possibilities for the distribution to use

1. The between-class distribution (a). This is a direct equivalent to LDA, but on a state-specific basis.

2. An average of the within-class distributions (b). E.g. A pooled (grand) state covariance. This is analogous to the no-confusion data implementation of Woodland, where the speech data was initially transformed to produce a grand within-class variance of unity for each feature. The identity matrix was used to represent a confusion distribution.

3. The total covariance of the whole of the training data (c). Here, the aim is to identify those features with large overall variance among other speech classes, but a small variance for the class under consideration. These features are intuitively representative of the class.

4. A fixed deterministic matrix not using any class/data information. E.g. the identity matrix.

The effect of using examples of each of these global distributions on the speaker independent RM Feb'89 test set using the R_{full} model set is shown in Figure 4.4. For the global covariance of the data and the between-class covariance, the distributions were first shifted to be around the state means before the CDA transform was implemented. This was found to give marginally better performance than using the distributions without the shift.

The between-class covariance is clearly superior to the other distributions and allows a dimensionality reduction to 16 dimensions without a degradation in performance. The best

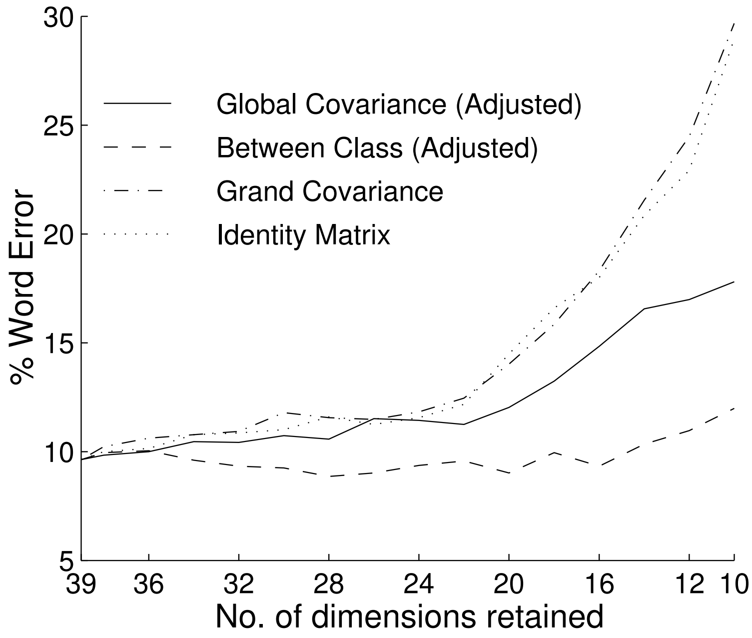

Figure 4.4: Effect of feature reduction using CDA with global measures. Results on RM SI feb'89 test set using R_{full} models.

performance of 8.9% word error is achieved using 28 dimensions. This result is better than the best global LDA result and a greater dimensionality reduction is achieved.

It is interesting to compare the results of transforms generated by the global covariance and the between-class distributions. Both represent similar measures of the data, however the between-class distribution gives equal weight to each class, whilst the global distribution gives more weight to the classes which occur more frequently in the training data. It appears that assuming each class is equally confusable is better than assigning confusability on frequency of occurrence of the classes.

The grand covariance and identity matrices have very similar effects, indicating that very similar rotations are generated. Looking at the case of the identity matrix, the rotated, scaled matrix for the out of class data is

$$\boldsymbol{S} = \boldsymbol{\Lambda}^{-\frac{1}{2}}\boldsymbol{R}_1'\boldsymbol{I}\boldsymbol{R}_1\boldsymbol{\Lambda}^{-\frac{1}{2}} = \boldsymbol{\Lambda}^{-1} \tag{4.22}$$

Since $\boldsymbol{S}$ is diagonal the second rotation has no effect and the features are selected based on a comparison of eigenvalues of the in-class data, and those dimensions with the largest eigenvalues (and therefore largest variances) discarded first. A similar argument can be made for the grand covariance case since the expected value of the class variance is the same as the grand variance. The second rotation has (on average) little effect so the selection criteria is very similar.

4.7 Summary

This chapter has compared global feature selection, global LDA and a state-specific transform approach. The feature selection was found to be poor in that no improvement on the recognition performance could be achieved. The global LDA assumptions were analysed and shown to be poor, although a dimensionality reduction can be achieved using LDA without loss of performance. This results in a computational saving since each observation vector is only transformed once.

The theory of a state based approach, CDA, has been presented and shown to be similar to LDA, but with better assumptions. Using global distribution measures on a state basis has been shown to be more effective than global LDA, both in terms of increasing recognition performance and in the dimensionality reduction which can be achieved. A computational saving over the original model set may be made without loss of performance.

Ideally the method should be applied with state-specific distributions which represent the possible confusions made during recognition. The problems of identifying such distributions when using continuous speech have been considered, and the problems involved identified. A simulation of the confusion gathering approach for isolated words on continuous speech has shown that generating state-specific confusion distributions is possible, however a direct implementation is not feasible. The problems that need to be considered when generating specific confusion data from continuous speech are

- Enumeration of all confusable models not feasible
- Model/word boundaries no longer fixed
- Incorporation of high level recognition constraints
- How to define confusion data (frames/models/words)
- How to ensure robust estimates of distributions

Methods for overcoming these problems are discussed in the next chapter.

Chapter 5

Evaluation of Confusion Discriminant Analysis Transforms

In chapter 4 the problems of generating confusion distributions on a state-specific level for use with CDA on continuous speech were identified. Methods for overcoming these problems are now proposed and investigated. Two approaches, one data driven and the other distribution based are examined and evaluated on the RM speaker independent database. The results achieved compare favourably with those obtained using global feature selection and extraction methods.

5.1 State-Specific Confusion Distributions for CDA

Gathering confusion data involves identifying and accumulating speech frames which are potentially confusable for each state. A distribution for the confusable frames needs to be estimated which is both accurate and robust. The distribution should represent all potentially confusable frames for the class, and sufficient numbers of confusions should be identified so that the estimate is not dominated by one particular type of confusion. Bearing these requirements in mind, two approaches to estimating the confusion distributions will be examined.

1. **Data-Driven Confusions**
   Identifying confusable vectors using the training data.

2. **Confusions from Model Parameters**
   Identifying potentially confusable regions of acoustic space using the parameters of the model distributions.

These approaches vary in the amount of computation required and the accuracy of the confusable distribution generated.

5.2 Data-Driven Confusions

If the training data is assumed to be completely representative of the test data, i.e. any modelling errors which will occur in the test data will also be present in the training data, a valid set of confusions may be identified from the training data. By performing recognition on the training data sections of speech which are poorly modelled can be identified by comparing the recognition output with the known transcription. The sections of speech which are assigned to incorrect classes can then be used as confusion data to try to improve the performance.

When considering possible approaches to confusion gathering from recognition of the training data, computational aspects must be considered. Since the tasks under consideration are based on continuous spoken words it is reasonable to restrict the search at the word level. This will ensure that only valid phone sequences corresponding to words are considered, and allows language models to be incorporated into the identification of confusions. If the phone model level is used the problem can become computationally expensive (phone models are more confusable than word models), whilst the sentence level is too broad. It may not seem intuitive that using higher level information as a basis to adjust individual state distributions is relevant, but the aim is to improve high level (word/sentence) recognition. Some training methods which use sentence performance to estimate state parameters [39, 93] provide justification for utilising higher level information to improve state modelling. Hence, algorithms based on word recognition will be considered.

Recognition on the test data is usually implemented as a Viterbi match to find the path through all model states which best matches the speech. The effectiveness of Viterbi confusions is limited for two reasons:

- The models are optimised on the training data, hence, the recognition performance on it is generally good. This can result in only a small proportion of the training utterances being incorrectly recognised.
- Confusions are only gathered from the single best matching model sequence. This has two consequences: first, it limits the number of confusions identified, and second it ignores the fact that the correct model sequence may be confusable with many other model combinations. Resolving the most likely confusion does not necessarily promote the correct transcription.

Using a standard Viterbi approach for gathering confusions is not ideal and may result in too few confusions to make a robust estimate of a confusion distribution. The obvious solution to this problem is to extend the one best match to generate multiple recognition hypotheses for each training utterance, or part of the training utterance. Any hypothesis which is more likely than the correct sequence is deemed confusable. Each incorrectly recognised utterance may contribute several hypotheses more likely than the correct one, and correctly recognised utterances may have segments which are confusable with partial hypotheses. In this way, the number of confusions gathered can be vastly increased, making estimate of the confusable distribution estimate more robust.

5.2.1 Identifying Confusions

The process for identifying confusions using recognition of the training data is straightforward. For each training data utterance there is a known word transcription W,

$$W = w_1 w_2 \ldots w_k$$

(transcription contains k words, each w_i is a word). Each w_i is represented by a phone model sequence of length l_i ($w_i = m_{i,1} m_{i,2} \ldots m_{i,l_i}$), hence the transcription at the model level can be determined:

$$W = m_{1,1} m_{1,2} \ldots m_{1,l_1} m_{2,1} \ldots m_{2,l_2} \ldots m_{k,1} \ldots m_{k,l_k}$$

The correct model level transcription can then be used to generate the best alignment of the frames of the training data utterance to the transcription. This is termed a forced alignment. Examining the frame/state alignments of the forced alignment each frame $\boldsymbol{f}_i$ is allocated to a 'true' state, $s_t(\boldsymbol{f}_i)$, deemed to be the class that $\boldsymbol{f}_i$ belongs to.

To identify confusions, each incorrect hypothesis generated by the recognition of the training data is frame/state aligned in a similar manner to the true state alignment to get a confusable state alignment ($s_c(\boldsymbol{f}_i)$).

The true and confusable state alignments are then compared. Any frame which is found to be assigned to different states in the two alignments is deemed to be confusable.

$$\begin{array}{lll} \text{If} & s_t(\boldsymbol{f}_i) = s_c(\boldsymbol{f}_i) & \text{then frame } \boldsymbol{f}_i \text{ is not confusable} \\ & s_t(\boldsymbol{f}_i) \neq s_c(\boldsymbol{f}_i) & \text{then frame } \boldsymbol{f}_i \text{ is confusable in state } s_c(\boldsymbol{f}_i) \end{array}$$

Having identified the frames that are confusable in a designated state, the confusion distribution (covariance $\boldsymbol{C}_c$, mean = state mean $\boldsymbol{\mu}$) for that state is calculated.

$$\boldsymbol{C}_c = \frac{1}{N_c} \sum_{i=1}^{N_c} (\boldsymbol{x}_i - \boldsymbol{\mu})(\boldsymbol{x}_i - \boldsymbol{\mu})' \tag{5.1}$$

where there are N_c confusable frames $\boldsymbol{x}_1 \ldots \boldsymbol{x}_{N_c}$ for the state. This is the confusion distribution to be used in CDA.

Using a standard recognition algorithm any factors which may affect the confusability on the test set, such as language models or model to word mappings, may be incorporated into the training data recognition. Problems may result from poor labelling of the database. If the correct transcription generated is a poor match in terms of modelling because the correct transcription is erroneous (maybe due to unusual pronunciation of words, or just poor labelling), the confusion data collected may be inaccurate. In all approaches it is assumed that the labelling is accurate.

A further problem that must be considered is the robustness of each confusion distribution estimate. If there are few recognition errors on the training set the number of confusions gathered for each state may be small, leading to problems in estimating distributions. In such cases, methods for making the distribution estimates more robust must be considered.

5.3 Data-Driven Confusion Methods

Two approaches for gathering confusion data from a recognition pass of the training data are proposed.

1. **N-Best recognition**
   A Viterbi based recogniser which generates multiple sentence hypotheses.

2. **Frame-by-Frame recognition**
   A Viterbi recogniser which uses information at the speech frame level to identify confusable sections of speech. A restricted case of frame by frame is also considered which attempts to incorporate lower level (e.g. model) restrictions into the confusions collected.

There is a trade off in these approaches between the specificity of confusions identified and the computation required. By using a word based recognition approach the search space for the recognition passes can be kept small helping to reduce the amount of computation.

5.3.1 N-Best Recognition Confusions

The N-Best recognition method of gathering confusions uses an N-Best recogniser with an additional state alignment procedure. The recogniser generates a lattice of word hypotheses, and a search of the lattice is used to find possible sentence hypotheses and corresponding likelihoods. Two definitions of confusable hypotheses may be used. One based on a comparison of likelihoods between the current hypothesis and the correct transcription alignment, and the other based on rank orders within the N-Best framework.

Confusions are gathered using the following algorithm:

Generate word hypothesis lattice

Search lattice for correct hypothesis, likelihood L_t

Search lattice for set of confusable hypotheses

For each hypothesis h *which is deemed confusable*

1. *Generate frame/state alignment* F_t *for correct hypothesis*
2. *Generate frame/state alignment* F_c *for confusable hypothesis* h
3. *Identify and assign confusable frames to states*

For the comparison of likelihoods the confusion criteria is

$$L_t - L_h < \delta$$

where L_h is the likelihood of hypothesis h and δ is a confusability threshold. $\delta = 0$ corresponds to the case where only confusable sentence transcriptions are considered. Relaxing this constraint so that $\delta > 0$ allows near confusions to be considered. For the comparison of rank orders, any hypothesis which is in the top n N-Best hypotheses is deemed confusable.

The identification of confusable frames is achieved in the manner described earlier, but for optimisation purposes, it is not necessary to generate complete frame/state alignments for the whole of the true and confusable utterances. Instead, the models used in each alignment can be compared, and state alignments only generated when different models (and not states) are used at the same time frames. Here the assumption is made that being in a different state in the correct model is not a confusion, but a relaxation of the time alignment within a model.

It should be noted that in this approach all confusions are first identified at the sentence level by comparison of sentence likelihoods or ranks, but the actual confusion analysis is performed at the word level to generate confusions for each state.

5.3.2 Frame-by-Frame Recognition Confusions

The N-Best recognition confusion gathering looks for errors at the sentence level. This may result in parts of the utterance which are highly confusable being ignored in the confusion analysis because over a whole utterance the confusability has been diluted. The Frame-by-Frame approach attempts to overcome this by looking for confusions at the model level.

In this approach, a standard (1-best) Viterbi recogniser is forced to recognise the correct model sequence. The same recogniser is then used in an unconstrained recognition to find the most likely model sequence. This gives two alternative state paths for recognition, a true path (P_t) and the most likely path (P_m). Confusions are gathered in a way in which aims to promote P_t in likelihood and degrade P_m in likelihood (unless $P_m = P_t$). At each frame, the model occupied in P_t is considered to be the correct model (i.e. the desired model to be in at that frame), and the model occupied in P_m should be considered to be a confusable model if different from that in P_t.

For the model comparisons, the best matching model over the immediately preceding frames are considered. Using the token passing recognition paradigm [111], after each frame has been processed by the recogniser the likelihood of being in the final state of each HMM represents how well that model has matched the immediately preceding speech frames. Assuming that the models used in the forced true recognition pass give the desired model sequence, the model with the highest final state likelihood at each frame is assumed to be the 'correct' model at that frame. It is known which frames this model matches (this information is recorded as part of the token passing algorithm), so any model in the free recognition which matched the same speech frames is potentially confusable. The potentially confusable models are tested by comparing the likelihoods. Confusions are gathered for each confusable model over the whole of that speech segment.

The outline algorithm is:

1. *Generate correct model sequence alignment M_t recording best models (models with best final state likelihoods) at each frame.*
2. *Generate a free model alignment M_f, but at each time t_b:*
   (a) *Identify the section of speech ($\boldsymbol{f}_{t_a} \rightarrow \boldsymbol{f}_{t_b}$) matched by the best model in M_t ending at time t_b*
   (b) *Find all models m_i matching $\boldsymbol{f}_{t_a} \rightarrow \boldsymbol{f}_{t_b}$ in free alignment M_f.*
   (c) *Check models m_i for confusability over $\boldsymbol{f}_{t_a} \rightarrow \boldsymbol{f}_{t_b}$ (compare likelihoods).*
   (d) *Generate frame/state alignment over $\boldsymbol{f}_{t_a} \rightarrow \boldsymbol{f}_{t_b}$ for models m_i*
   (e) *Assign confusable frames to confusion distributions.*

Although this method identifies confusions at the model level, the aim is to promote the likelihood of the correct sentence being recognised as the 1-best match. The recognition algorithm is identical to that which is used in recognition of the test set, with only the confusion statistic collection added, hence the confusions identified are directly relevant to the model parameters and recogniser.

This method is computationally very expensive as confusions at every single frame must be considered. Also, by considering the best models at every frame, models not actually in the true transcription can be deemed as correct models and thus inaccurate confusion data may be assigned to states.

5.3.3 Restricted Frame-by-Frame Recognition Confusions

The Frame-by-Frame method gathers confusion statistics at every single frame, with the aim of trying to make the most likely matching models at each frame in an unrestricted recognition the same as those produced in recognition of the correct utterance. Gathering confusions at every single frame is computationally expensive and unnecessary, so an adaptation of the approach is considered.

Examining the Viterbi recognition algorithm, it can be seen that although the best model finishing at each frame is recorded, the actual recognition sequence is determined via a traceback which looks at only a small percentage of the frames in the utterance.

The Restricted Frame-by-Frame method attempts to reduce computation by taking advantage of knowledge of the correct alignment. Only those models that are included in the traceback of the correct sequence need to be considered.

1. *Generate correct model sequence M_t alignment recording best models only at frames identified in the traceback.*

2. *Generate a free model alignment M_f.*

3. *For each time frame t_i locate the closest entry in the traceback list (at time t_b).*

4. *If $|t_i - t_b| < tol$, where tol is an arbitrary frame tolerance:*

   (a) *Determine the model m_t entered in the traceback at time t_b and the corresponding speech it successfully matched ($\boldsymbol{f}_{t_a} \rightarrow \boldsymbol{f}_{t_b}$).*

   (b) *Find any models m_c more likely than m_t at time t_i*

   (c) *Generate frame/state alignment over speech segment $\boldsymbol{f}_{t_a} \rightarrow \boldsymbol{f}_{t_b}$ for each m_c, and corresponding likelihood.*

   (d) *If alignment is confusable assign frames to confusion distributions.*

Figure 5.1 illustrates this idea. Model A is the best model finishing at time t_i, and the closest traceback entry is the previous frame. Assuming this is within the frame tolerance, confusion data for model A is gathered over the whole segment matched by the model in the traceback. By including a level of tolerance in the frame allocation, a degree of freedom

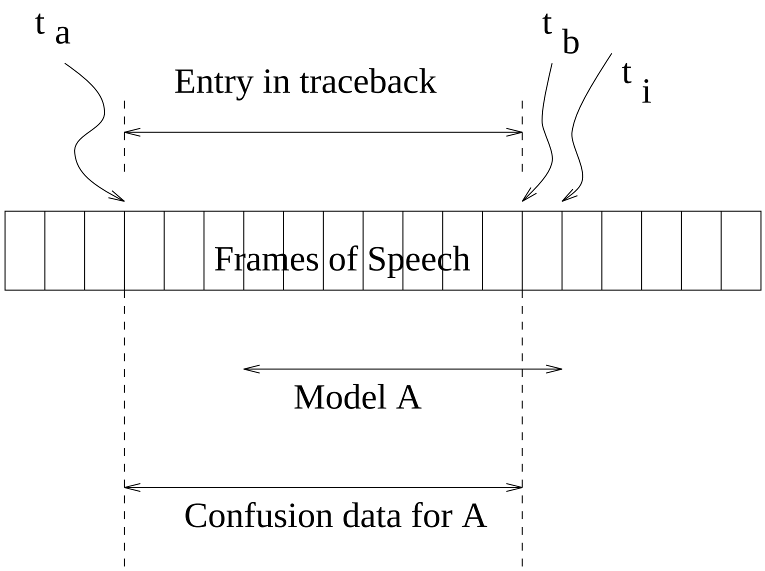

Figure 5.1: Illustration of the restricted Frame-by-Frame method.

in the time domain is allowed. This allows for slight deviations from the exact true model traceback, whilst not including all unnecessary confusion identifications for every model over every frame.

The amount of computation is vastly reduced in comparison to the Frame-by-Frame method since the restriction from the traceback means only a portion of the model/segment combinations need to be examined.

5.3.4 Summary of Data-Driven Confusion Methods

The data-driven confusion methods are all based on the assumption that errors in the recognition of the training data are representative of errors that will occur in the test set. All use a standard Viterbi recogniser, so any language models, model to word mappings, and recognition time constraints which are used in the recognition of the test set may be incorporated into the training data recognition.

If all the assumptions about the database and modelling accuracy are valid then the main problem with these methods is that they are computationally expensive in comparison to using the simpler global distributions for confusions. The training data set may be fairly large, and multiple passes of the data may be needed to generate all the alignments needed for gathering confusions. This is a once only cost, however, and may be considered as an extra stage in the training process.

5.4 Evaluation of Data-Driven Confusions

The approaches to gathering confusions by recognition were applied to the full covariance single mixture models (R_{full}) using the RM speaker independent training data. The specific experimental setups for each approach were:

1. N-Best recognition confusions
   Recognition of all 3990 utterances in the training data was performed using a word-dependent N-Best recognition algorithm [95], to produce transcriptions ranked in order of decreasing likelihood.

   For initial evaluation the rank of the correct transcription was identified, and all transcriptions more likely than the correct transcription were analysed for confusions (i.e. confusability threshold $\delta = 0$).

   More detailed evaluation examined N-Best confusions based on different thresholds for both comparison of likelihoods and comparison of ranks.

2. Frame by Frame confusions
   The algorithm was implemented exactly as described previously. The correct model sequence was forced on each utterance, however the optional inclusion of inter-word silence was allowed to ensure that true recognition conditions were satisfied. A standard Viterbi decoder was used for the free recognition. Only models matching the same speech segments were allowed to be confusable and only true confusions identified (confusability threshold $\delta = 0$).

3. Restricted Frame-by-Frame confusions
   The same forced recognition and free recognition algorithms as in Frame-by-Frame were used. The frame tolerance *tol* was set to 1, allowing confusions identified at $+/-1$ frames from the true phone boundaries. Likelihoods were compared over identical speech segments using an absolute confusability threshold ($\delta = 0$).

5.4.1 Problems with confusion gathering

The confusion gathering algorithms generally appeared to be successful in that reasonable confusions were being identified. However, the methods were all found to be computationally expensive, taking much longer than a single iteration of a standard Baum-Welch training pass.

A more important problem was that of ensuring that each state confusion distribution was well estimated. With a frame vector of size 39, at least 39 confusions were needed for an initial estimate, for a robust estimate many more would be needed. Figure 5.2 shows a breakdown of how many state confusion distributions would be poorly estimated if different minimum thresholds for the number of confusions required for robust estimates are specified. Even at low thresholds it can be seen that several distributions are poorly estimated, and as the threshold increases the number of poorly estimated distributions grows.

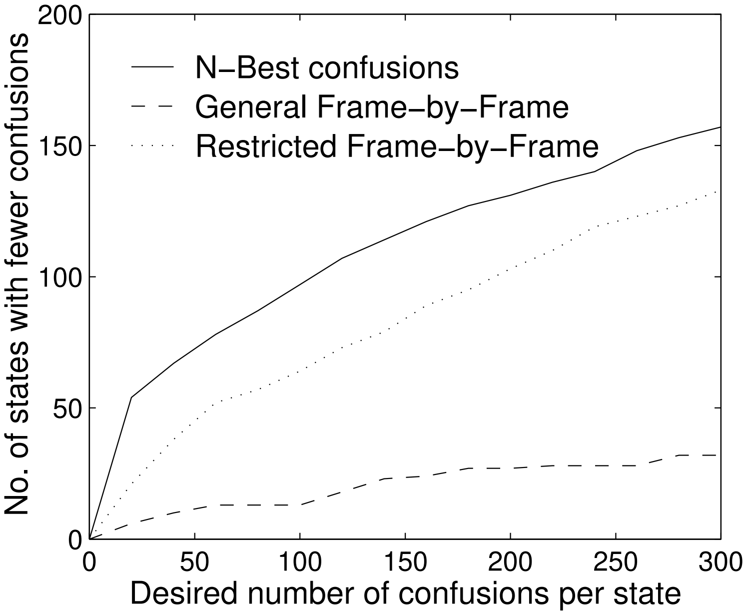

Figure 5.2: Examination of robustness of distribution estimates. All confusion data collected at threshold $\delta = 0$.

The N-Best confusion gathering had a particular problem with estimating robust confusion distributions due to very good recognition performance on the training data (69.5% of the training utterances were recognised as the most likely hypothesis). This limited the number of utterances which could be used to gather confusions, and of those utterances that were used, many had the correct transcription in the top 10 possibilities. Some states in the model set were actually deemed non-confusable, by the fact that zero confusions were collected for them. These were mainly states in function word dependent models, for example none of the states in all the models representing the word 'many' were confusable.

The Frame-by-Frame method is much better at generating sufficient data for robust confusion estimates with only a small number of states (24) lacking data at a threshold of 200.

To overcome this problem of robustness, poorly estimated distributions were replaced with a globally derived 'default' distribution. To ensure that the recognition confusions and global distributions could be interleaved in this way some brief experiments were carried out on a subset of the Feb'89 test set, combining the N-Best confusion distributions with the various globally derived distributions. The minimum number of confusions used to estimate the confusion distribution robustly was set at 200. Any states with fewer confusions identified were transformed by the default distribution, and any states with more confusions used the distribution estimated from the confusions gathered.

Figure 5.3 shows the results of these experiments, which confirm that the between class covariance is the best and most robust of the default distributions.

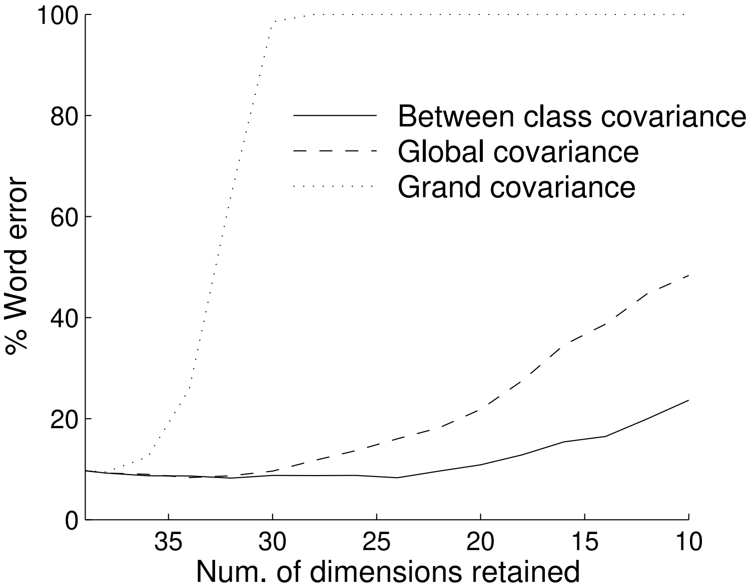

Figure 5.3: Examining interaction of default and N-Best confusion distributions on Feb'89 test set. Minimum no. of confusions for robustness = 200 per state.

The global covariance also appears to be robust when used in conjunction with the N-Best generated confusion distributions. However, using the grand covariance appears to combine in a destructive manner with the N-Best confusion estimates. The same effect occurs when using the identity matrix (not illustrated). This may be explained by looking at which features are discarded in each state. The grand covariance and the identity matrix both discard features with a high in-class variance, thus minimising the variance of the in-class features. The transforms from N-Best confusion distributions may discard any of the features, regardless of their in-class variance. An examination of the global variance of the individual features in the original feature space shows that 11 of the 13 second differential ($\Delta\Delta$s) features have variances smaller than unity, and the variances of the first differentials (Δs) are significantly less than the variances of the MFCCs. This implies that default confusions based on grand covariances or the identity matrix will select new dimensions based heavily on the differential features. These are less variant in the data as a whole, so the expected likelihood of a state transformed by these default confusions matching any speech frame is increased relative to those transformed using confusion distributions. This leads to the recognition process favouring those model strings with a high density of states transformed by the default confusions. Since only a minority of the states use default confusions, a small number of models dominate the output of the recogniser, most notably the function words. These words have a short duration, and the recogniser seems to prefer the use of two or more function words to the correct word in many cases, resulting in a large number of insertion errors, and a low recognition accuracy.

This problem is not as pronounced with the global covariance default confusions since the dimensions selected are those based on features with a large variance in the data set as a whole. The expected changes in likelihood after transformation by a transform generated from global covariance and a transform generated by N-Best confusion estimates are much more similar, and neither of the types of transformed states are heavily favoured.

The between-class covariance is derived on a state-basis, so is well tuned to the individual state characteristics, and allows good combination with the recognition based confusion distributions. Hence, the between-class covariance was chosen as the default distribution for all other experiments.

5.4.2 Selecting Robust Confusion Estimates

The effect of using different values for minimum confusions was evaluated using N-Best confusion estimates. The between-class covariance matrix was used as a default confusion distribution if there were deemed to be too few confusions for a robust estimate. Different values for minimum confusions were used to reduced the number of dimensions to 36 and 30. The results (Table 5.1) show that high values are mainly based on using the default matrix in most states, and performance is similar to that obtained using only the default matrices.

At lower minimum confusion thresholds the N-Best estimates dominate, and performance is maintained at a high level. As the threshold is increased, performance is impaired, due to either the interaction between default and N-Best confusion distributions, or just the increased use of the less specific default confusion distribution.

With 100 minimum confusions there are 289 out of the 388 states using the N-Best recognition confusion distributions, however some of these are poorly estimated, as can be seen from the fact that the 200 minimum confusion threshold performs better. There is a definite trade-off between the robustness of the confusion distribution estimates, and the number of states which use the estimates. A minimum confusion threshold of 200 apparently generates robust distributions and is used for all other recognition based confusion experiments.

Min no. confusions	No. states using N-Best estimates	% Word Error	
		36 features	30 features
100	289	9.0	8.8
200	254	8.7	8.8
300	231	8.7	9.1
400	209	9.1	9.7
500	200	9.1	9.9
1000	158	9.5	11.4

Table 5.1: Effect of choosing different values for minimum number of confusions for defining robust estimates, and reducing to 36 and 30 dimensions. Confusions generated using N-Best confusions with threshold $\delta = 0$. Default matrix is between-class covariance.

5.4.3 Recognition Confusion Transform Results

Each approach to gathering recognition confusions was tested on the Feb'89 test set using a minimum confusion threshold of 200, and transforming the feature space into dimensions as low as 10.

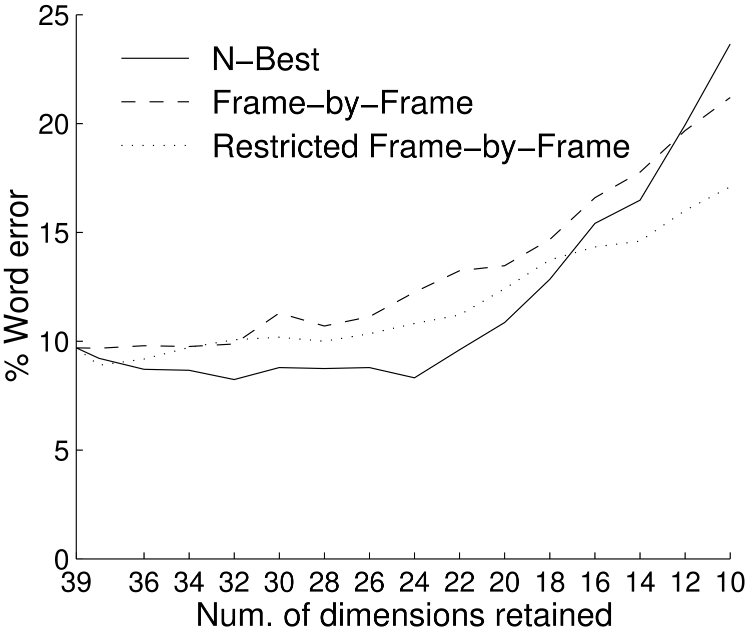

Figure 5.4: Comparison of recognition based confusion methods on Feb'89. All generated with threshold $\delta = 0$. Default matrix is between-class covariance.

The results (Figure 5.4) show that the different approaches have different effects on the performance on the test set. The N-Best confusion distributions initially increase performance substantially (from 9.7% word error to 8.3% word error, a reduction in error rate of 14.4%). However, at lower dimensionalities the performance falls off. The optimal performance is obtained at 32 dimensions, and all dimensions down to 24 features give better performance than the baseline. At lower than 24 dimensions the performance is substantially poorer than that of the original model set.

Both of the Frame-by-Frame methods are more robust at lower dimensionalities than the N-Best method, with a gradual decline in performance rather than a large fall. However, the performance is worse than the baseline for all dimensionality reductions except when discarding only one dimension. The general and restricted Frame-by-Frame methods both follow the same trends in performance, with the restricted version performing marginally better. There is very little change in recognition performance of the general Frame-by-Frame method until the dimensionality is reduced to less than 32. The restricted Frame-by-Frame gives an initial error reduction (to 8.9%) but then falls away with greater reductions in dimensionality. The restricted version gives better performance than the general Frame-by-Frame at lower dimensionalities.

The Frame-by-Frame approaches are computationally much more expensive than N-Best, and despite more confusions being identified perform poorly in comparison. The N-Best approach is concentrated on for the remainder of the evaluation.

5.4.4 Use of Thresholds with N-Best

The results reported above all used an absolute confusion threshold which for N-Best leads to problems of robustness for the estimated confusion distributions. By relaxing the definition of confusability the number of confusions identified can be increased. This is tackled in two ways: the top n from the N-Best list are treated as confusion data (Figure 5.5); the confusability threshold for the likelihood comparison is lowered (Figure 5.6).

As more confusions are identified the dimensionality reduction process becomes more robust. The best performance for the fixed rank confusion gathering is achieved when using the top 10 ranked sentences, this gives approximately 3 times the number of confusable frames as using the top 5 but less than half as many as using the top 20. This indicates that the transformations are well estimated using around 10 hypotheses for each utterance (one of the top 10 may actually be the correct hypothesis).

Rank Confusions		Likelihood Confusions	
Ranks	No. Conf. Frames	Threshold	No. Conf. Frames
1	29789	0.0	697177
5	383539	100.0	2193915
10	1006607	200.0	5748959
20	2512761	400.0	24380505

Table 5.2: Number of confusable frames identified using N-Best confusion gathering with different thresholds.

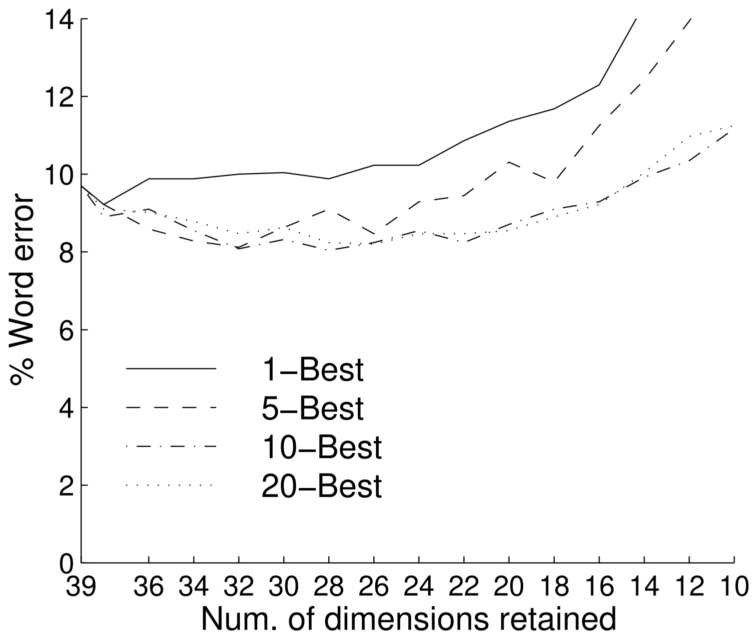

Figure 5.5: N-Best confusions gathered using a fixed number of alternative hypotheses. Results on Feb'89 test set.

For the likelihood comparison reducing the confusability threshold δ to below zero includes near confusions and increases the confusion data available (Table 5.2). Again, including more confusions increases the robustness of the method. Using a threshold of 100 results in a similar number of confusable frames to using the top 20 best ranked hypotheses, and increasing the threshold to 400 gives a ten-fold increase in the number of confusions.

The larger number of confusions enables a larger dimensionality reduction to be made and a similar performance to the original models can be achieved when using 14 features.

5.4.5 Discussion of Results

The N-Best and Frame-by-Frame methods both suffer from the problem of robustly estimating the matrices. Although the general Frame-by-Frame method results in a substantial amount of confusion data, the quality of the confusions identified is poor. This is demonstrated by the improvement gained by restricting the method. Neither the general or the restricted Frame-by-Frame methods allow a substantial dimensionality reduction without a drop in performance. The Frame-by-Frame method is also computationally much more expensive than the N-Best method.

The N-Best approach suffers from poor estimation of the confusion distributions when only absolute confusions are used, due to limited number of confusions being gathered. However, even with this method over 10% reduction in word error can be achieved by

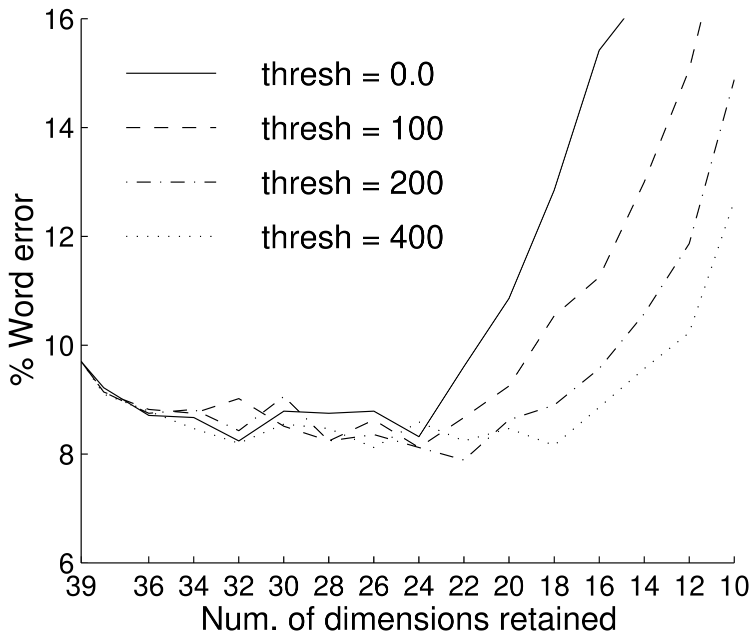

Figure 5.6: N-Best confusions gathered using different likelihood thresholds. Results on Feb'89 test set.

discarding a small number (7) of dimensions. By including more potentially confusable data by identifying near confusions, or simply the best matching hypotheses, the N-Best method can be made much more robust. This allows a dimensionality reduction down to 14 features, which is less than half the original number of features, and gives a saving in computation during recognition.

With N-Best confusions and a likelihood confusability threshold of 400, reducing to 18 dimensions not only results in a small computational saving, but also a 15% reduction in error (8.2% word error rate).

5.5 Confusions from Model Parameters

It was demonstrated earlier (section 4.6) that using the between class distribution successfully reduced the number of dimensions down to 16 without degrading performance. In that case, all classes were considered equally confusable. By considering only the closest classes it may be possible to improve this dimensionality reduction and get a further saving on computation.

During the training process, each frame of data in the training set is assigned to a particular state. Each state has an estimated distribution based on the data allocated to it. These distributions could be used to estimate which other distributions are confusable, and hence potential confusion data.

If a typical frame is drawn from a particular state distribution, by looking at the likelihood of other state distributions matching the typical frame a set of confusable state distributions can be identified. These confusable state distributions can be combined into a single confusable distribution for the state that the typical frame was drawn from. This method is termed *PDF-based confusion gathering*.

Assume there are N emitting states $(s_1 \ldots s_N)$ in the whole model set with each state s_i characterised by a mean vector $\boldsymbol{\mu}_i$ and covariance $\boldsymbol{\Sigma}_i$. The outline algorithm is:

1. *Consider state s_k which has a typical frame vector $\boldsymbol{f}$.*
2. *For each state s_i ($i = 1 \ldots N, \quad i \neq k$), if*
$$\mathcal{F}(\boldsymbol{f}|s_k) - \mathcal{F}(\boldsymbol{f}|s_i) < \delta_p \tag{5.2}$$
then distribution s_k is confusable with distribution s_i. $\mathcal{F}(\boldsymbol{f}|s_k)$ is the likelihood that frame $\boldsymbol{f}$ belongs to state s_k and δ_p is a confusability decision threshold.
3. *Combine all confusable state distributions into a single distribution.*

The confusability decision threshold is arbitrary and can be determined empirically. The combination of the state distributions into a single distribution is a weighted combination of the covariances after they have been adjusted around a common mean (the within-class mean for the state under consideration, $\boldsymbol{\mu}_k$).

$$\boldsymbol{\Sigma}_c = \frac{1}{N_{sum}} \sum_{i=1}^{N} g(s_i) N_i \left(\boldsymbol{\Sigma}_i + (\boldsymbol{\mu}_i - \boldsymbol{\mu}_k)(\boldsymbol{\mu}_i - \boldsymbol{\mu}_k)' \right) \tag{5.3}$$

where

$$g(s_i) = \begin{cases} 1 & \text{if } s_i \text{ is confusable with } s_k \\ 0 & \text{otherwise} \end{cases} \tag{5.4}$$

N_i is the number of vectors used to estimate $\boldsymbol{\Sigma}_i$ during training and

$$N_{sum} = \sum_{i=1}^{N} g(s_i) N_i. \tag{5.5}$$

The sampling of distributions for frame vectors and subsequent testing for belonging to other distributions can be simplified by considering the expected values of vectors belonging to each distribution. i.e. given $\boldsymbol{x} \in s_i$, what is the expected likelihood that $\boldsymbol{x} \in s_k$.

The comparison for confusability becomes,

$$E[\mathcal{F}(\boldsymbol{x}|s_i)] - E[\mathcal{F}(\boldsymbol{x}|s_k)] > \delta. \tag{5.6}$$

Such a measure can be translated into a comparison of the actual distributions, using a distance metric between distribution pairs which gives an idea of the relative confusability. Possible measures for confusability definition are the Kullback-Liebler measure [67] or Bhattacharyya distance [12], which are small for confusable distributions and large for non-confusable distributions.

If $H(s_k, s_i)$ is the measure between the distributions of state s_k and s_i then confusable distributions are those for which:

$$H(s_k, s_i) < \delta_h$$

where δ_h is an arbitrary threshold of confusability.

This approach is a compromise between the global approach based on the distribution of the classes, and the recognition approach based on actual errors identified. The distribution of the data in individual states is used, and confusable frames are assumed to be average state vectors instead of individually identified frames. There is no consideration of recognition run time factors such as the model sequences or grammar restrictions involved, and there is no tailoring to specific errors in the training data, which relaxes the assumption that the training data is completely representative of the test data.

5.6 Evaluation of PDF-based Confusions

For the implementation of PDF-based confusions, a measure of similarity between multivariate Gaussians distributions must be defined. The measure needs to be small when the distributions are very similar in shape and volume, and represent similar acoustic space, and increase as the distributions become more separated. For this purpose the divergence measure is considered. As the aim is to generate potential confusion data for a specific class, the directed divergence is more relevant than the full divergence measure.

Given two Gaussian distributions ω_1 and ω_2 with means $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$ and covariances $\boldsymbol{K}_1$ and $\boldsymbol{K}_2$ respectively, the log likelihood ratio Λ is defined as,

$$\Lambda(y) = \ln \frac{\mathcal{F}(\boldsymbol{y}|\omega_1)}{\mathcal{F}(\boldsymbol{y}|\omega_2)} \tag{5.7}$$

where $\mathcal{F}(\boldsymbol{y}|\omega_i)$ is the likelihood that $\boldsymbol{y}$ was drawn from distribution ω_i.

The directed divergence (also known as the Kullback-Liebler distance or relative information of class ω_1 with respect to class ω_2) is the likelihood that a vector from class ω_1 also belongs to class ω_2.

$$D_{dir} = E[\Lambda(\boldsymbol{y})|\omega_1] \tag{5.8}$$

with the closed form (appendix B),

$$D_{dir} = \frac{1}{2}tr(\boldsymbol{K}_2^{-1}\boldsymbol{K}_1 - \boldsymbol{I}) + \frac{1}{2}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T\boldsymbol{K}_2^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) + \frac{1}{2}\ln\frac{|\boldsymbol{K}_2|}{|\boldsymbol{K}_1|} \tag{5.9}$$

For the evaluation, two further factors are involved. A threshold (δ_h) is required for the directed divergence measure to identify similar distributions, and some means for determining the weights (N_i) used in combining state covariances is needed.

5.6.1 Threshold for Confusions

The threshold to use for a definition of confusability is dependent on the state distributions. If the threshold is set too low then very few state pairs will be confusable, and the confusion distributions will be biased. If the threshold is set too high, all states become confusable and the method will converge to the performance achieved using the global covariance distribution.

To get an idea of the values of a reasonable threshold, the divergence measures between all state pairs were generated, and the number of state pairs that would be confusable at various thresholds calculated. Figure 5.7 shows that the number of confusable states is similar to a cumulative Poisson distribution. There are 388 states present in the model set making 75078 state pairings. For each state it is desirable to have several different state distributions contributing to the confusion data, but including too many distributions reduces the relevance of each contribution.

Assuming a Poisson distribution, the mean over all state pairs gives a mid-point to base the threshold around, this is about 40.0 for the directed divergence measure. The mean values gives around 20000 confusable state pairs, which is an average of 50 confusable distributions per state, sufficient for a robust estimate of the distribution.

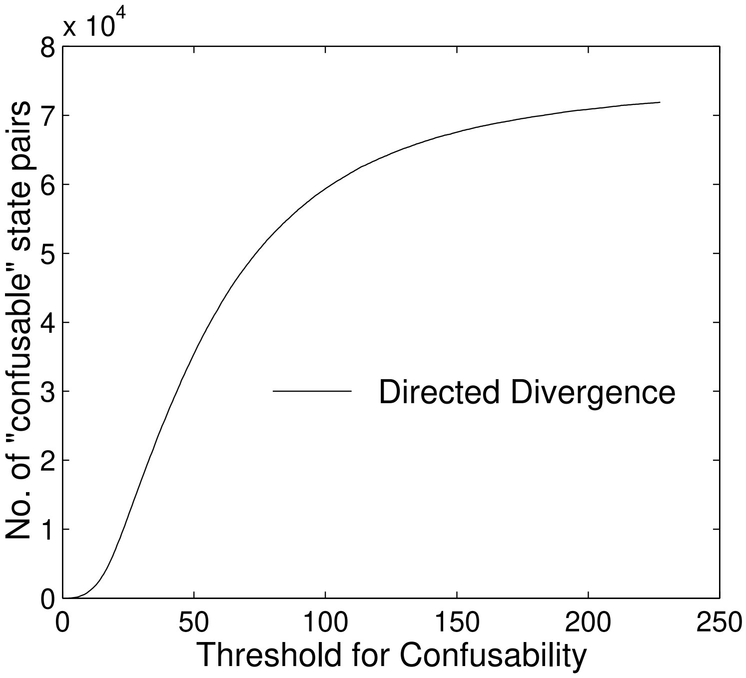

Figure 5.7: Number of confusable states at different divergence thresholds for R_{full} models.

5.6.2 Weights for Combination

The confusable distributions identified for a specific state s_k need to be combined into a single distribution about the state mean $\boldsymbol{\mu}_k$ (equation 5.3). From the definition of the covariance matrices the N_i values should represent the number of vectors used to estimate each state.

There are two approaches for choosing the weights:

1. Weights from state occupation counts during training
   During each training iteration the frame/state alignment may change, and the alignment of the final training iteration is needed. This requires an alignment to be generated after the final training iteration has been used to re-estimate the model parameters. Some training methods (e.g. Baum-Welch embedded re-estimation) which use a total likelihood approach to state alignment of the vectors, assign the same vector to several states, weighted with the probability of it being generated by each of the states. Hence, the state occupation counts are not integers.

2. Weights from a recognition pass
   In a similar way that confusable vectors are calculated using the N-Best recognition approach (see section 5.3.1), the confusable distributions can be counted. If a vector $\boldsymbol{x}$ is assigned to state s_t in the alignment of the true model sequence, but state s_c in a free recognition alignment, then s_t is a confusable distribution for s_c. For each state, a count of how many times every other state is confusable is made. These counts are used as the weights for combinations in states which are deemed confusable.

The second method amounts to a recognition of the training data which is computationally expensive, and one of the reasons that this approach is being examined is to try to negate the need to do a recognition pass on the data.

5.6.3 Results

Experiments have been performed using different thresholds with the two methods of generating the weights for combining the distributions on the Feb'89 test set (Figure 5.8).

It can be clearly seen from the results that using combination counts from the training pass is superior to using N-Best recognition counts. A test was made to investigate the effect of just using the weights from confusion recognition as both the confusability test (instead of using the D_{dir} measure) and combination weights, but this was found to produce poor results. This indicates that recognition errors identified on the training set are not completely representative of errors which will occur on the test set.

The results are very encouraging in two ways. First, there is an initial improvement in recognition accuracy when discarding a small number of features, for example with a threshold of 20.0, when reducing the dimensionality to 34 the error rate drops to 9.0% word error. Secondly, the method is robust, using as few as 18 dimensions the system achieves a word error of 10.2%.

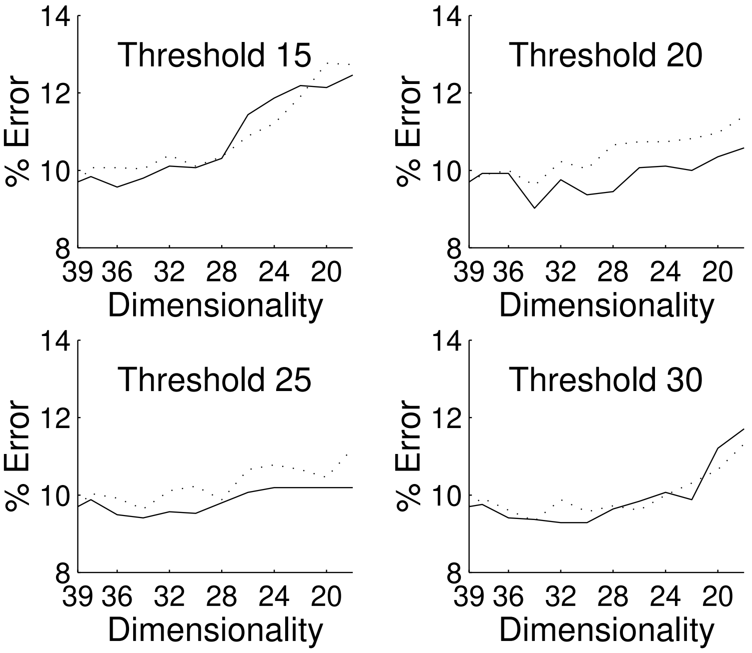

Figure 5.8: PDF-based confusions generated by combining distributions based on weights from training (solid line) and weights from recognition of training data (dotted line). % word error on Feb'89 test set.

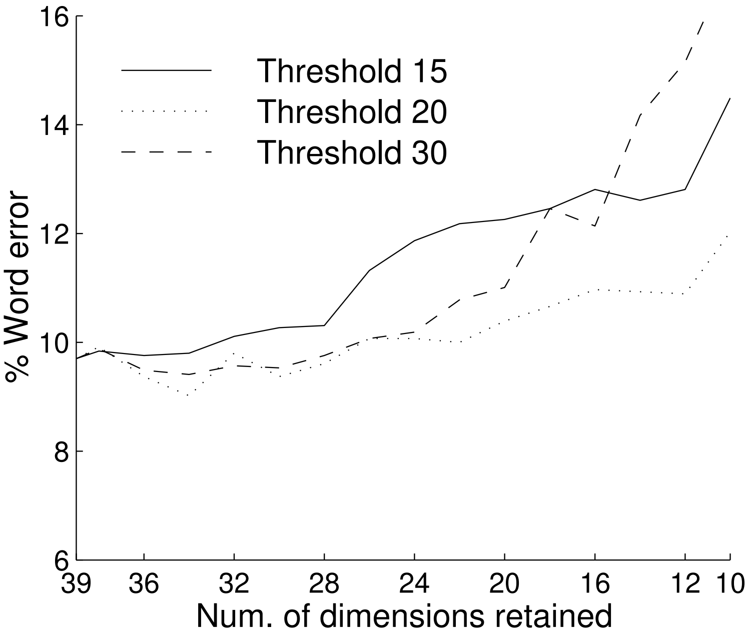

Figure 5.9: Comparison of different thresholds for PDF-based confusions with weights from training data. % word error on Feb'89 test set.

It also appears that the threshold value is fairly flexible in that over a range of values (20 to 30) similar recognition performance is achieved from the confusion transform. Closer examination of the thresholds at lower dimensionalities shows that a threshold of 20.0 is more robust and gives the best reduction in error for all the PDF-based confusions (Figure 5.9).

5.7 Comparison between Methods

The different methods of generating state-specific confusion data for CDA dimensionality reduction are compared in Figure 5.10.

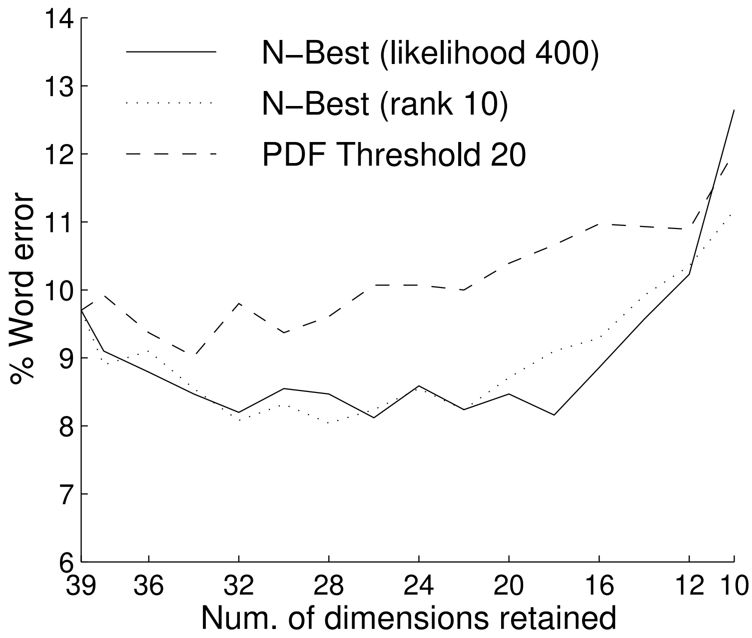

Figure 5.10: A comparison of the best confusion gathering approaches for CDA on Feb'89 test set.

The N-Best approaches give a much greater error reduction than the PDF-based confusion method, and allow more dimensions to be discarded. The fixed rank and likelihood comparison approaches to confusion gathering give very similar performance until half the original dimensions are discarded. With further reductions the confusions based on likelihood are slightly more robust and allow a more dimensions to be discarded without degrading performance.

5.7.1 Comparison with LDA Methods

The state-specific CDA confusion approach has been compared with state-specific CDA using global distributions and global LDA transforms (Figure 5.11).

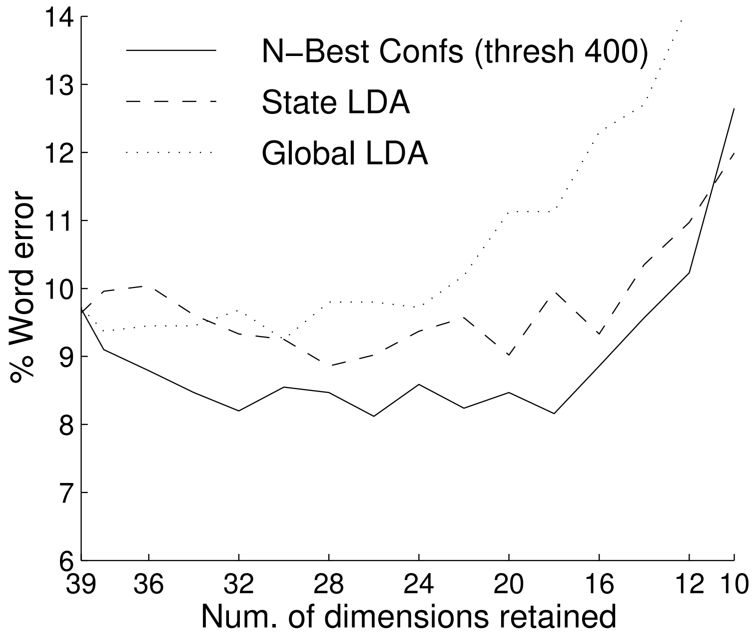

Figure 5.11: A comparison of the best confusion gathering approaches with global and state-specific LDA. Results on Feb'89 test set.

The CDA with confusions gathered using N-Best recognition is by far the most effective approach. A 15% reduction in error can be achieved and the number of dimensions dramatically reduced. Enough dimensions can be discarded to achieve a computational saving over the global LDA approach, and the equivalent dimensions give lower error rates than the state-specific LDA. Global LDA allows a reduction to 24 features while maintaining baseline performance, and state-specific CDA with global distributions allows a reduction to 20 dimensions before becoming slightly unstable. Using confusion distributions allows an even greater reduction, any dimension down to 14 features gives better performance than the baseline.

From a computational point of view, calculating the CDA transforms using confusion data is vastly more expensive than any of the global approaches. However, this is a once only cost, and the resulting models have a lower computation cost during recognition.

5.7.2 CDA Confusions on Different Test Sets

The best CDA confusion gathering approach was implemented on the different RM speaker independent test sets (Figure 5.12). Three of the four test sets allow reductions down to 14 dimensions with no great adverse effect on the recognition performance. The Oct'89 test set is the exception and the error rate increases with most dimensionality reductions, however even this test set allows a dimensionality reduction to 24 without a great increase in error.

The Feb'91 test set gets a maximum 12.8% reduction in error, achieved when using 34 features, and the Sep'92 has a maximum reduction of 14.2% error using also using 34

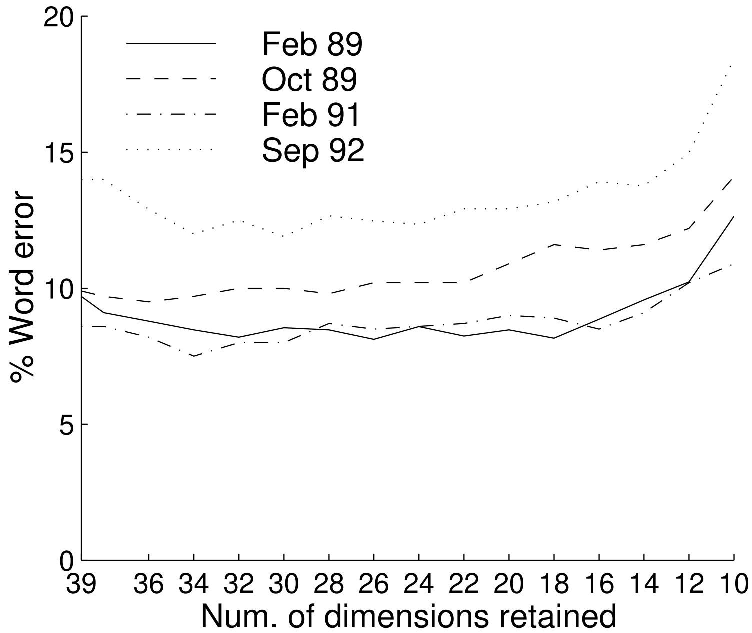

Figure 5.12: CDA confusion on different test sets. Confusions gathered using N-Best with a likelihood threshold of 400. Results are average % word error.

features.

5.8 Analysis of Discriminative Feature Selection

The evaluation of the CDA transform has shown that computation savings may be made over the original full covariance model set by using state-specific LDA or state-specific CDA using data-driven confusions. The N-Best confusion approach is superior to the other methods of generating transforms, and by incorporating near confusions as well as absolute confusions the estimates of the transforms can be made more robust.

An attempt at improving the state-specific LDA approach using PDF-based confusions which uses only those class distributions which are deemed confusable instead of all class distributions showed only limited success. A large dimensionality reduction could be made, but no substantial reduction in error. The method is clearly susceptible to the definition of confusable distributions, and the decision boundary chosen for the confusions has a large effect on the reductions achievable.

The biggest reduction in error on Feb'89 (15%) was achieved using the N-Best confusion gathering with a likelihood comparison threshold of 400. The largest reduction in dimensions and hence also computation was also achieved using the N-Best confusion approach.

5.8.1 Improving PDF-based Confusions

The PDF-based confusions are limited by the problems of determining which distributions are confusable. Only one distance metric, the directed divergence between two distributions, has been considered, although brief experiments with the full divergence measure showed that the directed divergence was superior. Other distance measures could be used, however, all measures are subject to the problems of defining a decision boundary.

A second possible improvement is to reduce the effect of a hard decision boundary. At present a frame is either confusable or not confusable. By incorporating a soft boundary, whereby the combination of distributions is weighted by the confusability of the distributions improved confusion distributions may be gained. This could be incorporated into the standard theory by replacing the binary valued $g(s_i)$ of equation 5.4 by a scaling value dependent on the confusability.

$$g(s_i) = x \tag{5.10}$$

where x is the degree of confusability, $x = 1$ indicates complete confusability, $x = 0$ indicates no confusability. Determining the degree of confusability can be performed using the distance measure with a gradient scale of confusability depending on the distance.

Given the poor performance of the PDF-based confusions in comparison to the N-Best approach, and noting that the hard-decision boundary produced poorer results than the state-specific LDA even when using low thresholds (i.e. very specific confusions), this extension of the method was not examined further.

5.8.2 Improving Data-Driven Confusions

The confusion distributions identified from the N-Best approach have been shown to be effective in the CDA framework. The use of fixed rank ordering and likelihood comparisons for the gathering of confusions both resulted in large numbers of confusions being gathered. The larger the number of identified confusions the more effective the CDA transformations. The major problem with this method is the computation involved in performing a recognition pass on the training data to generate the N-Best alternatives. It is also possible to weight the confusable frames based on a confusability measure. The weightings would be difficult to determine, however, and evaluation would be required to examine the best approach for such a scaling (e.g. weights based on sentence, word or phone confusability measures).

5.8.3 Application to Component Mixture Distributions

The discussion thus far has focused on the use of CDA with HMMs with a single full covariance Gaussian distribution. The application to the more general multiple component mixture density HMM case is obviously of great interest. In theory exactly the same approach can be used as the single Gaussian case, but treating each component Gaussian density as a separate class. This is certainly the case for LDA, as reported by Parris &

Carey [90], but for CDA this could affect the methods of gathering confusions. In this case, the component distributions within the same state should not be deemed confusable, and the assumption that frames allocated to different states within the same model are not confusable should be extended to cover this aspect.

For recognition confusions, the state alignments generated should be extended to mixture component alignments so that frames are assigned to individual components. The number of distributions has increased, thus the total number of confusions required for robust estimation of all the confusion distributions is increased. Indicating that larger thresholds should be used.

Due to the limited number of training utterances for RM, estimating model sets with multiple mixture component full covariance distributions to achieve reasonable performance was not feasible, so the approach to component mixture distributions could not be investigated properly.

5.9 Summary of Discriminative Feature Methods

The use of state-specific transforms has been investigated using the RM speaker independent database. Comparisons between the state-specific transforms and similar global approaches have shown the effectiveness of using state based information. A state-specific LDA approach, confusion discriminant analysis (CDA), allowed a much larger dimensionality reduction using global measures than a global LDA transform approach due to the ability to capture state-specific rotations to maximally separate the state from other distributions. The state-specific CDA used a global between-class distribution centered around the state under consideration. This approach is sub-optimal and based on expected confusable distributions. An attempt to improve the state-specific CDA approach by identifying only likely confusable distributions, the PDF-based approach, proved less successful. This was due to problems with distance measures and decision boundaries for identifying confusable distributions.

A more successful approach was that using statistics collected from a recognition of the training data to generate multiple sentence hypotheses. Accumulating data from the confusions and near confusions at the word level from the different hypotheses gave robust confusion distributions. This resulted in CDA dimensionality reduction being substantially more effective, in terms of reduction in error (15%), than the state-specific LDA transform. This could be achieved with no increase in computation during recognition. Extra computational savings could be made by reducing dimensionality further and performance equivalent to the baseline could still be achieved.

The CDA and state-based LDA approaches have an identical computation cost, and both can achieve a dimensionality reduction of over 50% allowing a computational saving over the original model set. The CDA approach with N-Best confusion gathering has a substantial computation overhead in computing the confusion distributions, but this is a once only cost.

The use of state-based criteria for generating rotations is clearly advantageous, and allows discrimination to be incorporated into the model parameters after maximum likelihood training has been completed.

Chapter 6

Speaker Adaptation Techniques

Transformation techniques may also be applied in the field of speaker adaptation. The model parameters or the input speech may be transformed so that the match between the acoustic model parameters and the acoustic signal is improved. In this chapter several techniques for speaker adaptation are reviewed. For comparison some non-transform methods are examined and the advantages and disadvantages of the different approaches are described.

6.1 Introduction to Speaker Adaptation

Current state of the art speech recognition technology is able to produce speaker independent recognisers which have extremely low error rates for small/medium vocabularies. For example average word error rates of around 3% have been achieved on the RM speaker independent test sets [85]. Although the average error rates are low, some speakers have recognition rates considerably worse than others. Due to the wide range of speakers used to train a speaker independent system the models have to cope with the inter-speaker variability as well as the intra-speaker variability. This leads to a large amount of variance within the model for each acoustic phenomena and may reduce the modelling accuracy for each individual speaker. This is demonstrated by a comparison of speaker independent (SI) and speaker dependent (SD) systems tested on the same speaker. Assuming that sufficient data is available to train both systems, speaker dependent systems typically perform 2-3 times better than equivalent speaker independent systems, as mentioned by Huang [50] and demonstrated by a comparison of results (Table 6.1). When the amount of speaker dependent data is limited, an improvement in performance over an SI system cannot be guaranteed due to poor estimates of the model parameters. In many applications it is not feasible for a speaker to provide enough data to reliably train a SD recognition system from scratch, and SI systems must be used.

This problem can be overcome, at least partially, by using **speaker adaptation** techniques the aim of which is to take an initial model system which is already trained, and use a sample of a new speakers data (termed *adaptation data*) to attempt to improve the modelling of the speaker with the current set of models. This can be achieved in two

Speaker	SI error rate	SD error rate
bef0_3	3.2	2.3
cmr0_2	7.4	1.6
das1_2	1.8	0.9
dms0_4	3.2	1.0
dtb0_3	3.3	1.2
dtd0_5	4.6	2.3
ers0_7	3.5	2.6
hxs0_6	6.5	1.5
jws0_4	4.5	1.8
pgh0_1	2.6	2.1
rkm0_5	8.3	2.6
tab0_7	2.2	1.8
Average	4.3	1.8

Table 6.1: Comparison of speaker dependent and speaker independent systems for the same speakers on RM1 speaker dependent test set. Identical systems are used, SI trained on SI-3990 sentences, SD system is SI system retrained on 600 SD sentences.

ways: *speaker normalisation* and *speaker adaptation*. Speaker normalisation is the process of normalising the input speech so that the speech of all speakers has similar characteristics. During speaker adaptation the parameters of the recognition system (input speech or model parameters) are adjusted to improve the modelling of the new speaker. Before these alternative approaches are examined the causes of speaker differences are discussed.

6.1.1 Speaker Variation

There are many causes of differences between speakers, mainly due to the personal characteristics of each person. A model of speaker characteristics is presented by Nolan [82] which suggests some of the factors which contribute to speaker variation. These can be split into two categories, acoustic (realisational, duration and physiological), and phonological (lexical and stress differences). The acoustic variation must be handled by the acoustic modelling component of a recognition system while the phonological differences can be handled by grammar and pronunciation models.

6.1.1.1 Linguistic Differences

Accents can account for much speaker variation between speakers, however there is still considerable variation within accents [102]. Pronunciations are different due to different phonetic realisations across accents, for example there is a vast difference between the pronunciation of *tomato* in RP /tə'mɑːtəʊ/ (using the IPA alphabet), and General American

/tɘ'meIto /. Vowels are particularly susceptible to variation even within a particular accent or dialect. The vowel in *coat, nose* and *snow* is monophthongal in some accents but dipthongal in others. In RP it is generally /eʊ/ but may be more rounded (/θʊ/) or less rounded (/əω/).

There are also phonotactic distribution and systemic differences between accents. These involve the different pronunciations of phonemes in different contexts, and the different phoneme sets used in the accents. Although all accents are based on the same IPA phone set, some accents ignore certain phones e.g. RP uses /ʊ/ and /uː/ for the vowels in *good* and *mood* whilst Scottish accents use /uː/ for both words, thereby using one less phoneme.

In some cases the pronunciation of one word can lead to predictions of the pronunciation of other words, e.g. the vowel in *bath* is usually consistent with the vowel in words such as grass, staff etc. for a given speaker. In other cases the pronunciation of words is speaker specific, e.g. the pronunciation of *either* generally gives no indication of how the speaker pronounces other words. Such examples are idiosyncratic of the speaker and lead to further speaker differences (usually termed lexical-distributional differences).

6.1.1.2 Physiological Differences

Physiological differences are the source of many speaker differences between speakers of the same accent. The values of formant frequencies are defined by physical attributes of the vocal tract, such as the size of the mouth and nasal cavity. Females tend to produce higher fundamental frequencies than males and more widely spaced formants, thought to be about 20% higher, due to the shorter vocal tract.

Other effects are more transitory in nature, being initiated by the physical and mental condition of the speaker. Tiredness can lead to slower more stilted speech, while anger may lead to louder, quicker speech. A cold can lead to the nasal passages becoming obstructed and the speech becomes fully non-nasal. For the purposes of current speech recognition techniques these transitory effects are ignored, and the normal behaviour of the speaker is considered. An ideal adaptation technique would consider transitory effects and change the modelling accordingly, but such a complete dynamic adaptation process is not currently feasible due to the poor understanding of the modelling involved and the short term nature of the effects.

6.1.2 Speaker Normalisation

In spite of the many differences in speaker characteristics outlined above, humans find it very easy to understand a wide variety of speakers with different accents and different physiological characteristics. This suggests that the human brain may be performing some kind of normalisation to the speech which filters out individual characteristics. If such a filtering system can be performed in a speech recogniser then the inter-speaker variation problem can be resolved.

The object of speaker normalisation is thus to construct a normalised speaker space into which speech from any speaker can be projected so that the inter-speaker differences are minimised (or even better, the acoustic features are invariant). Figure 6.1 shows a schematic diagram of speaker normalisation.

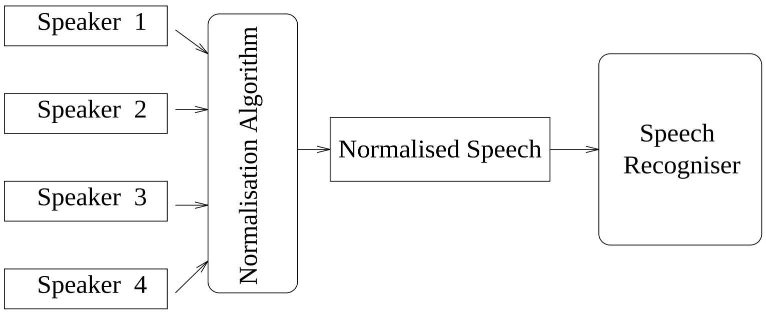

Figure 6.1: Schematic diagram of speaker normalisation process

The problem with speaker normalisation techniques is the amount of variation in the speech. Finding a general technique which performs a good mapping into the normalised space has proved difficult. Among the attempts that have been made are normalisations based on estimating the vocal tract length and then computing spectral shifts accordingly [86, 101]. Dynamic frequency warping has also been investigated [88] where the vowel spectra undergo a non-linear frequency alignment. This not only reduced the distances between equivalent vowel spectra, but also between phonetically distinct vowel spectra thereby reducing discrimination as a whole. Other normalisation techniques have investigated alternative frequency shift functions [99] and self-organising feature maps [62].

Overall, speaker normalisation techniques have been found to be ineffective due to the complex mappings required, and the fact that all acoustic information is treated in the same way, regardless of the acoustic event it represents. If the speaker normalisation was sound-specific it may be more effective.

6.1.3 Speaker Adaptation

Speaker adaptation techniques differ from normalisation techniques in that instead of using a general mapping for all speakers so that all speakers appear similar, a separate mapping is generated for each speaker. A sample of speech from the new speaker is used to generate an adaptation mapping and all further processing of the speaker uses the mapping, improving the recognition performance for the speaker. The aim is thus to make a speaker independent recogniser behave more like a speaker dependent recogniser.

During speaker adaptation either the input speech can be adapted (similar to speaker normalisation), or the models can be adapted, or both (Figure 6.2). The current adaptation techniques can be split into three categories: spectral transformation techniques which use mappings of the feature space, speaker classification in which an appropriate model set is selected, and techniques which re-estimate the model parameters. These techniques are discussed later in sections 6.3, 6.4 and 6.5.

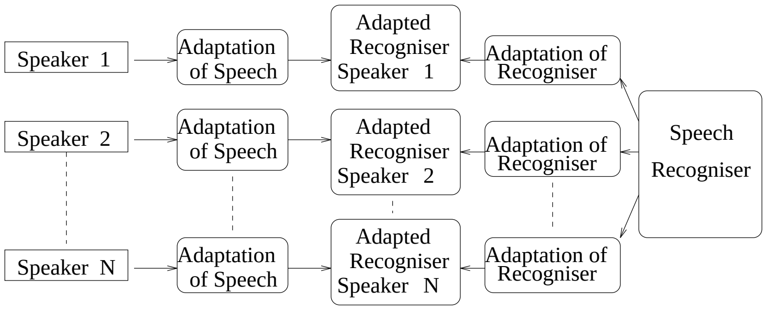

Figure 6.2: Schematic diagram of speaker adaptation process

6.2 Modes of Speaker Adaptation

The process of adapting the speech recognition system can be performed in several different modes, describing how and when the adaptation will take place, and how the adaptation data presented by the speaker is used. The adaptation is termed *supervised* if the identity of the adaptation data (e.g. a word transcription) is also given to the adaptation process, otherwise the adaptation is *unsupervised*. There are two categories for describing how the data is used:-

- Static Adaptation
  All of the adaptation data is presented to the adaptation process before the final adapted system is produced.
- Dynamic Adaptation (also termed *Incremental Adaptation*)
  An adapted system is produced after only part of the adaptation data is presented, and refined as more data is observed.

Thus if a speaker provides a sequence of known utterances and an adapted system is produced using all those utterances, the adaptation is static supervised.

The use of different adaptation modes has consequences for the adaptation techniques in terms of computational effort, accuracy of adaptation and the nature of the task in question. Clearly, if the adaptation data is correctly labelled, the appropriate adaptation mappings can be determined accurately whilst unsupervised data may lead to errors in the mappings. In some tasks a very quick enrollment time for a new speaker is desirable, hence it may be desirable to initially start with a speaker independent system and use unsupervised dynamic adaptation. In other tasks high recognition accuracy may be needed, hence more time may be available to gather adaptation data, and a static supervised mode may be used.

6.3 Speaker Clustering for Adaptation

One of the more simplistic ideas for implementing speaker adaptation is that of speaker clustering and classification. If it is assumed that there is a large number of model sets, each one corresponding to a different speaker, the adaptation method is simply to determine which model set belongs to the current speaker. In practice, generating a large number of well trained model sets is not feasible due to the training overheads and large amount of data required from each of many different speakers. Instead most implementations of this idea use a small number of model sets, and use each set to represent similar speakers. Speaker clustering algorithms are used to pool similar speakers into groups, and the speakers in each group are used to estimate a set of models. Thus the required training data from each speaker is reduced, and a smaller number of model sets can be trained. For recognition, the adaptation process is to select the speaker group that is most representative of the new speaker and use that model set.

Imamura [56] clusters the speakers by initially training a model for each speaker and then clustering the models based on a cross-entropy measure. Each model set is then restricted to have output distributions (in this case VQ codebook entries) which are dependent on the relevant group of speakers.

Kosaka [63] implements a scheme in which a hierarchical speaker model clustering algorithm is used. A model set is trained for each speaker and the sets are clustered to form a tree. The top node of the tree corresponds to the case where all speakers belong to the same cluster, and at successive levels of the tree, more classes are formed. A model set is associated with each node of the tree, and estimated from the individual speaker models. The adaptation process searches the tree for the best model set according to the best likelihood produced by recognition of the adaptation data.

A second method proposed by Kosaka [64] approaches the problem from a different angle. Given several sets of speaker dependent models for different speakers, the corresponding models in each set are merged by combining the different state distributions into a single state with equal weights. This gives a pseudo speaker independent system, termed a *speaker-mixture* model set. Adaptation of this system is performed by re-estimating the weights given to the distributions from each speaker according to the adaptation data.

All these methods suffer from several problems. The spectral variation of speakers, even within a cluster, may be large. The range of sample speakers is always limited, and the new speaker may not be represented well in the possible model set. It is assumed that the differences between speakers are uniform across a model set, i.e. the whole set of models is a good match for the new speaker, which is a poor assumption.

6.4 Spectral Transformation Approaches

The spectral mapping approach to speaker adaptation has been widely investigated in recent years. These methods attempt to map the input speech from a new speaker to match the speaker which the models are trained on (reference speaker). The reported spectral mapping implementations apply linear transformations to either the input vector, or to the whole model set. The main difference in the implementations is the manner in which the transform is estimated.

Spectral transformations were first considered for use with spectral template based recognisers [44, 45, 57]. These approaches considered the reference templates as representing speech from a reference speaker and generated spectral transformations to minimise the distance between the new speaker and the reference speaker.

Grenier [44, 45] proposed a canonical correlation approach, later extended by Choukri [24], whereby both the speech from the new speaker and the reference templates are projected into a common vector space which maximally correlates the two (Figure 6.3). This

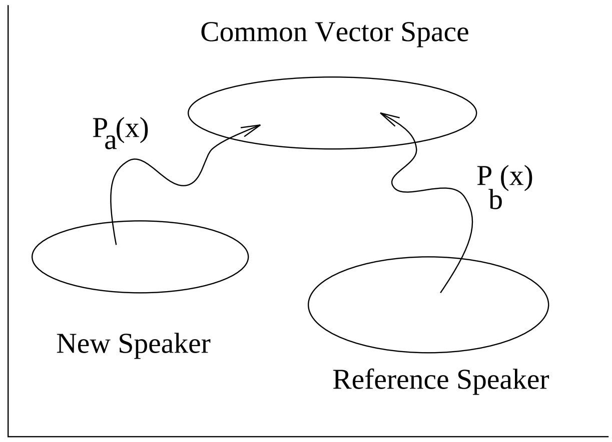

Figure 6.3: Example of the canonical correlation approach: The acoustic representations of each speaker are projected into a common feature space.

requires only two transformation functions to be estimated, one for the new speaker (P_a) and one for the templates (P_b).

$$\boldsymbol{y}_s = P_a(\boldsymbol{x}_s) \tag{6.1}$$

$$\boldsymbol{y}_r = P_b(\boldsymbol{x}_r) \tag{6.2}$$

where $\boldsymbol{x}_s$ and $\boldsymbol{y}_s$ are the original and transformed observation vectors of the speaker and $\boldsymbol{x}_r$ and $\boldsymbol{y}_r$ are the original and transformed templates of the reference speaker. The distance measure for recognition is then computed between $\boldsymbol{y}_s$ and $\boldsymbol{y}_r$. Jaschul [57] uses a simpler, but effective, transformation for spectral slice templates which can be viewed as a frequency shift transformation,

$$\boldsymbol{y} = \boldsymbol{T}\boldsymbol{x} + \boldsymbol{a} \tag{6.3}$$

where $\boldsymbol{T}$ is a transformation matrix and $\boldsymbol{a}$ is a vector allowing for mean effect. $\boldsymbol{x}$ is the input speech and $\boldsymbol{y}$ is the transformed speech (Figure 6.4). Jaschul implemented the transformation both on a global basis, where each frame of input speech is transformed only once and compared to each template, and on a more specific basis so that different transformations were used for each phone class (i.e. the speech had to be separately transformed for each template comparison). Although the separate transforms were computationally more expensive, and required more data for estimation, the resulting performance indicated that it should perform better than the global approach, though there are problems with data sufficiency for transform estimation. The parameters of the transform matrix were computed on a least squares error basis,

$$e = \frac{1}{N} \sum_{i=1}^{N} (W\hat{\boldsymbol{x}}_i - \boldsymbol{u}_i)^2 \tag{6.4}$$

where e is the error, $\boldsymbol{W}$ is the composite matrix $[\boldsymbol{T}, \boldsymbol{a}]$, $\boldsymbol{u}_i$ is the reference template corresponding to the i^{th} speech frame $\boldsymbol{x}_i$. $\hat{\boldsymbol{x}}_i$ is the extended vector $[\boldsymbol{x}_i, 1]$. To reduce the amount of data required for the estimation, a banded-diagonal matrix was also investigated. The diagonalised transform could still be implemented successfully, but required less data for estimation.

Jaschul's approach to transformations has been investigated further by Hewett [49]. Hewett uses dynamic time warping to align adaptation data to templates and estimates a single transformation matrix which combines the frequency shift and scaling by a least squares regression formulation. A comparison between the multivariate least squares regression transform and a canonical correlation method showed in favour of the regression transform.

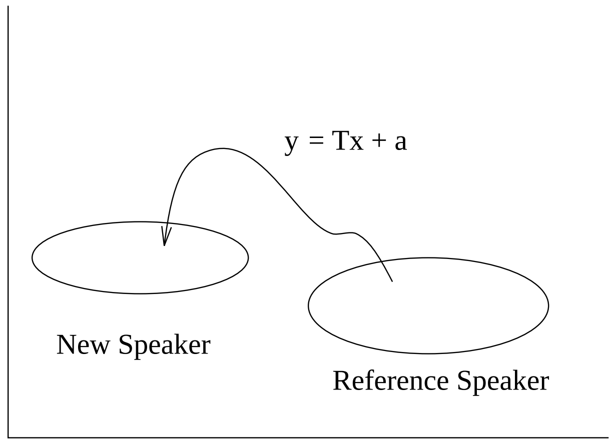

Figure 6.4: Example of spectral transformation using a single transform to project the reference speaker onto the new speaker

A similar transformation approach has been reported by Cox & Bridle [31] who implemented the approach on isolated vowel spectra, but in an unsupervised fashion. Schwartz *et al.* [97] use the transformation in the probabilistic domain to adjust the VQ codebook of discrete HMMs. Class *et al.* [26, 27] also apply a transformation to VQ codebooks, but use global transformations.

Kenny *et al.* [61] extended the transformation idea to continuous density HMMs, but only in a very restricted form. The mean vectors of the Gaussian distributions were updated by a translation,

$$\hat{\boldsymbol{\mu}} = \boldsymbol{\mu} + \boldsymbol{\nu}_{\text{phone}}. \tag{6.5}$$

The interesting aspect to the implementation is that the estimate of the translation vector ν_{phone} is achieved via a Baum-Welch type alignment of the data. The use of a simple translation is limited since the results reported by Kenny for adaptation were poorer than the results obtained using a standard reestimation of the means on the same data. This indicates that a translation even at a phone level is not sufficient to adequately capture the spectral variation from speaker to speaker in different states within an HMM.

The application of spectral transformations to HMMs has been considered further by Zhao [116] who views the inter-speaker variation as a two-source problem. The *acoustic* source is attributed to physical speaker differences, that cause spectral variations independent of phone units. The second source is *phone-specific*, attributed to a speaker's individual articulation (e.g. accent, idiosyncratic pronunciations). Each of these sources is modelled by a separate transform, so the complete spectral transformation is of the form,

$$\hat{\boldsymbol{x}} = \boldsymbol{H}\boldsymbol{L}_i\boldsymbol{x} \tag{6.6}$$

where $\boldsymbol{H}$ is the global acoustic transformation and $\boldsymbol{L}_i$ is the phone specific transformation for phone i. To estimate the transforms, $\boldsymbol{H}$ is first estimated from the average spectrums of the reference and the new speaker. The individual phone transforms are then estimated from the differenced spectra of the individual phone units after being adjusted by $\boldsymbol{H}$. The use of either the acoustic or the phone-specific transforms leads to an improvement in recognition accuracy, and the inclusion of both transforms leads to the best improvement.

A variation on spectral mapping techniques has been investigated by Bellegarda [9, 10]. The adaptation is performed on a piecewise basis, and is used to transform the existing data for the reference speaker. This data is added to that supplied by the new speaker and used to estimate a speaker dependent VQ codebook. A similar approach was also considered by Kubala [66], and later by Choi [21], where a spectral transform is used to map each speaker in the speaker independent training data onto a reference speaker before the system is trained. The resulting system is then assumed to be speaker dependent. During recognition, a further transformation is generated to map each speaker onto the reference speaker.

An investigation of a non-linear spectral transformation applied to the input speech has been performed by Gurgen [46]. A multi-layer perceptron was used to generate the transformation, and the resulting speech frames used in an HMM recogniser. The results showed that using a standard linear canonical correlation analysis actually performed better than the non-linear approach, indicating that linear transformations are capturing a large amount of the difference between speakers at a global level.

6.5 Model Parameter Adaptation Approaches

The model parameter adaptation approach to speaker adaptation has become a much researched area in recent years. The limitation of spectral input transformations and model selection techniques have become more apparent with the improving performance of speaker independent systems. Although some spectral transformation techniques do alter the model parameters, this is performed with the aim of improving the match between the new speaker and the reference speaker, not with the aim of improving the modelling accuracy for the new speaker.

Most model parameter adaptation techniques for continuous density HMMs use Bayesian MAP (*maximum a posteriori*) approaches to parameter reestimation. Brown [17] first suggested using Bayesian estimation for adaptation in a connected digit recogniser using CDHMMs, and the method has been developed further by Lee and Gauvain [43, 69].

In a MAP approach the parameter set is chosen to maximise

$$\mathcal{F}(\lambda|\boldsymbol{O}) = \frac{\mathcal{F}(\boldsymbol{O}|\lambda)\mathcal{F}(\lambda)}{\mathcal{F}(\boldsymbol{O})} \tag{6.7}$$

where $\boldsymbol{O}$ is the observation sequence of the adaptation data with a probability distribution function given by $\mathcal{F}(\boldsymbol{O})$, and λ is assumed to be the parameter set defining the distribution. If λ is assumed to be random with a prior distribution $\mathcal{F}_p(\lambda)$, the MAP estimate is obtained.

Different approaches to estimating the value of λ have been investigated. Lee proposed using a segmental K-means algorithm [69]. The MAP estimate was estimated by solving

$$\frac{\delta}{\delta\lambda}\mathcal{F}(\lambda, s|\boldsymbol{O}) = 0 \tag{6.8}$$

where s is a state sequence. An iterative procedure can be used to solve this, successively determining the optimal state sequence, and updating the parameters based on the sequence.

$$\hat{s} = \max_s \mathcal{F}(\boldsymbol{O}, s|\bar{\lambda})\mathcal{F}_p(\bar{\lambda}) \tag{6.9}$$

$$\bar{\lambda} = \max_\lambda \mathcal{F}(\boldsymbol{O}, \hat{s}|\lambda)\mathcal{F}_p(\lambda) \tag{6.10}$$

The MAP estimate can also be formulated in an EM algorithm [33]. If the complete set of the adaptation data is represented by $\boldsymbol{y}$, the auxiliary function used in the EM process is given by

$$Q(\lambda, \bar{\lambda}) = E[\log \mathcal{F}(\boldsymbol{y}|\bar{\lambda})|\boldsymbol{O}, \lambda]. \tag{6.11}$$

This function can be differentiated and separated into component parts representing each parameter in the model set. Hence, an iterative algorithm similar to that used in MLE can be used.

The choice of priors for λ can affect the outcome for the MAP estimate. If a non-informative prior is used (λ assumed fixed but unknown) the maximisation leads to the MLE formula (section 2.4).

SI	Adaptation Data		
	2 min	5 min	20 min
13.9	8.7	6.9	3.4

Table 6.2: Example MAP adaptation results on RM Feb'91 test set [43](%word error)

The effectiveness of the MAP estimate is shown by the results from Gauvain [43] (Table 6.2).

The major problems with the MAP adaptation approach are the estimation of prior densities, and the fact that the reestimation formulae only applies to individual model parameters. Thus, if a mixture component is not observed in the adaptation data it cannot be adapted.

The problem of unobserved mixture components has been addressed by Cox [29, 30] and Ahadi [1] by using prediction techniques for the unseen mixture components. These techniques stemmed from an initial study by Cox [28] on linear regression relationships between vowel sounds. Regression coefficients between different vowel sounds were computed from all the training speakers. Using five vowels from the test speaker all the vowel templates were updated based on regression. This improved classification rates for all the vowels. Furui [41] has also shown that regression analysis could be used to good effect to estimate unobserved models in isolated recognition.

This idea was incorporated into the basic MAP approach by augmenting the algorithm with a second stage which uses linear regression relationships between the distributions so that the unobserved distributions can still be updated. Speaker dependent model sets are used to derive regression coefficients between different mixture distributions. For the mean vectors of the distribution a multiple linear regression relationship of the form

$$y_j = a_0 + \sum_{i=1}^{P} a_i x_{ij} \tag{6.12}$$

is assumed where $\boldsymbol{y}$ is the target mean and $\boldsymbol{x}_i$ is the mean from the i^{th} source distribution. P is the regression order and the a_i are regression coefficients which are estimated by considering the source and target means from all the speaker dependent model sets. This method provides a significant improvement over the MAP estimate for limited adaptation data, and converges to the MAP estimate as more data becomes available, and less prediction is necessary [1].

The success of these techniques show that the relationships between the class distributions in different speakers may be represented in a regression formulation.

	SI	No. Adaptation Sentences		
		40	100	600
MAP	10.2	8.0	6.6	4.1
Prediction	10.2	7.1	6.2	4.1

Table 6.3: Improved adaptation by incorporating prediction methods into the MAP adaptation process [1] (% word error).

6.6 Approaches Combining Parameter and Spectral Adaptation

Although the spectral transformation and model adaptation approaches have been categorised separately, it can be seen that spectral transformations can equally be applied to the model parameters. The separation was deemed necessary since the spectral transformation approach optimised the match between paired speech frames (i.e. the reference frame and the speaker frames), but not the acoustic modelling of the speaker. The model-based approaches attempt to improve the modelling of the speaker. The adaptation method proposed in the next chapter combines the spectral transform estimation with a training algorithm which aims to improve the acoustic modelling [73]. A similar approach has since been proposed by Digalakis [34]. In both approaches the estimation of the transformation is formulated in a maximum likelihood framework. In the case proposed by Digalakis, only a diagonal transformation matrix has been considered, which, as will become apparent in the following chapters, is a very limited case.

The transform used by Digalakis takes the standard form of spectral transformations and is applied on a mixture component basis to both the mean and covariance,

$$\hat{\boldsymbol{\mu}}_i = \boldsymbol{T}\boldsymbol{\mu}_i + \boldsymbol{a} \tag{6.13}$$

$$\hat{\boldsymbol{\Sigma}}_i = \boldsymbol{T}\boldsymbol{\Sigma}_i\boldsymbol{T}' \tag{6.14}$$

where $\hat{\boldsymbol{\mu}}_i$ and $\hat{\boldsymbol{\Sigma}}_i$ are the adapted mean and covariance, $\boldsymbol{T}$ is the $n \times n$ transformation matrix, and $\boldsymbol{a}$ is an offset. To compensate for the lack of adaptation data the transformations are tied between many mixture components. The estimation of the transforms is performed using an EM algorithm. Since only diagonal covariances and diagonal scaling matrices are being considered the problem can be reduced to a single dimensional case.

6.7 Summary

The process of adapting HMMs to specific speakers with little training data has been described. This speaker adaptation can be performed in a variety of ways, either by selecting a model set, by spectral transformations of the speakers involved or by updating the parameters of the models. Examples of each of these techniques have been presented. Spectral transformations are the simplest to implement, since they can be performed on a global basis on the input speech. However, more success has been found by applying such transformations on a phone specific basis. This is less computationally expensive if the adaptation process transforms the HMM state distributions, leaving the input speech unchanged. This is slightly different from model reestimation since the aims are not the same. Spectral transforms attempt to minimise the difference between the speaker and the appropriate reference models. No account is taken of how this affects the other models.

The main problem with model reestimation techniques is that with limited adaptation data, not all the model parameters will be observed in the example data. A solution to this problem is to use a combined method of transform and parameter reestimation, by placing the estimation of the transforms into a standard reestimation framework. Such a method is detailed in the next chapter.

Chapter 7

A Transformation Approach to Speaker Adaptation

A new transformation based approach to speaker adaptation is presented in this chapter. The approach associates a transform matrix with the output distributions within each HMM. The transformation is estimated by aligning the adaptation data with the model states to capture the general characteristics of the speaker with respect to the current HMM parameters. The method is described in the following sections and formulae for estimating the transform are derived.

7.1 Maximum Likelihood Linear Regression

Many of the adaptation methods examined in chapter 6 use a global spectral transformation approach or only adapt parameters of those models which are observed in the adaptation data. A new method termed *Maximum Likelihood Linear Regression* (MLLR) is presented which overcomes these limitations.

MLLR uses a set of regression-based transforms to tune the HMM mean parameters to the new speaker. Each of the transformations is applied to a number of HMM mean parameters and estimated from the corresponding data. This sharing of transformations and data enables the method to estimate transforms on small amounts of data.

The aim of MLLR is to estimate an appropriate transformation for the mean vectors of each mixture component so that the original system is tuned to the new speaker. For mixture component s with mean $\boldsymbol{\mu}_s$, the adapted mean estimate $\hat{\boldsymbol{\mu}}_s$ is given by,

$$\hat{\boldsymbol{\mu}}_s = \boldsymbol{W_s} \boldsymbol{\xi}_s \tag{7.1}$$

where $\boldsymbol{W_s}$ is an $n \times (n+1)$ transformation matrix and $\boldsymbol{\xi}_s$ is the extended mean vector,

$$\boldsymbol{\xi}_s = [\omega, \mu_{s_1}, \ldots, \mu_{s_n}]'$$

where the value of ω indicates if an offset term is to be included: $\omega = 1$ for an offset, $\omega = 0$ for no offset.

Including the regression (transformation) matrices the mixture component density function becomes,

$$b_j(\boldsymbol{o}) = \frac{1}{(2\pi)^{\frac{n}{2}}|\boldsymbol{\Sigma}_s|^{\frac{1}{2}}} e^{-\frac{1}{2}(\boldsymbol{o}-\boldsymbol{W_s}\boldsymbol{\mu}_s)'\boldsymbol{\Sigma}_s^{-1}(\boldsymbol{o}-\boldsymbol{W_s}\boldsymbol{\mu}_s)}. \tag{7.2}$$

A maximum likelihood estimate of the $\boldsymbol{W_s}$ matrices is made which attempts to increase the likelihood of the model set generating the adaptation data.

The MLLR approach is similar to that used by Hewett [49], but is extended to work in an HMM framework with a more appropriate optimisation formula. Hewett's work was performed on DTW templates which allowed him to make the assumption that the distribution of each class around the template was common to all templates. In this case the maximum likelihood estimate for $\boldsymbol{W_s}$ is the least squares estimate. In HMMs the assumption of a common covariance for all states is not valid and the resulting least squares estimate of $\boldsymbol{W_s}$ is not guaranteed to be the best estimate. MLLR attempts to overcome this problem by using a maximum likelihood estimation which incorporates the effects of different class covariances.

Jaschul [57] has suggested that phone-dependent transforms may be more appropriate so the MLLR framework has been derived with a flexible scheme of transform sharing in mind. With a globally tied system a single $\boldsymbol{W_s}$ matrix is used and shared between all mixture components. At the opposite extreme, if a separate $\boldsymbol{W_s}$ is used for each mixture component then the adaptation process is an MLE reestimation of the component means using the adaptation data (since the same optimisation criteria is used). This method of tying transforms can be used to overcome problems of limited observation data, similar to methods such as tying of states which are used in parameter estimation [107, 110]. The estimation of the transformations then becomes a regression estimate of the average differences between the example speech data and the current system parameters. Since the method is being considered for the purposes of speaker adaptation with the aim of adapting the recognition system with as little data as possible, the emphasis is placed on the tying case.

Note that this method has only considered adapting the mean vectors and not the other model parameters. This is due to the assumption that the main differences between speakers lies in the average position of phones in the acoustic space rather than the distribution of the intra-speaker variation. Limiting adaptation to only updating the mean vectors restricts the number of parameters that need to be estimated during adaptation, and should allow adaptation on a small amount of adaptation data. The problem of updating other model parameters is considered in section 7.7.

7.2 Estimation of MLLR Regression Matrices

MLLR estimates the regression matrices $\boldsymbol{W_s}$ to maximise the likelihood of the adapted models generating the adaptation data. The derivation of the MLLR estimates [74] is given below, making the assumption that each HMM state has a single Gaussian output distribution.

7.2.1 Maximisation of Auxiliary Function

Assuming the adaptation data, $\boldsymbol{O}$, is a series of T observations

$$\boldsymbol{O} = \boldsymbol{o}_1 \ldots \boldsymbol{o}_T.$$

a re-estimated set of model parameters $(\bar{\lambda})$ can be generated from the current set of model parameters (λ) using MLE (section 2.4).

Denoting all possible state sequences of length T by the set $\boldsymbol{\Theta}$, a new set of parameters which increase the value of auxiliary function (equation 2.19) can be estimated. Since only the transformations $\boldsymbol{W_s}$ are to be re-estimated, only the output distributions b_s are affected, the auxiliary function, $Q(\lambda, \bar{\lambda})$, can be written as,

$$Q(\lambda, \bar{\lambda}) \quad = \quad constant + \sum_{\theta \in \Theta} \sum_{t=1}^{T} \mathcal{F}(\boldsymbol{O}, \boldsymbol{\theta}|\lambda) \log b_{\theta_t}(\boldsymbol{o}_t). \tag{7.3}$$

Defining S as the set of all states in the system, and $\gamma_s(t)$ as the *a posteriori* probability of occupying state s at time t given that the observation sequence $\boldsymbol{O}$ is generated,

$$\gamma_s(t) = \frac{1}{\mathcal{F}(\boldsymbol{O}|\lambda)} \sum_{\theta \in \Theta} \mathcal{F}(\boldsymbol{O}, \theta_t = s|\lambda). \tag{7.4}$$

Equation 7.3 can then be written as,

$$Q(\lambda, \bar{\lambda}) = a + \mathcal{F}(\boldsymbol{O}|\lambda) \sum_{j=1}^{S} \sum_{t=1}^{T} \gamma_j(t) \log b_j(\boldsymbol{o}_t). \tag{7.5}$$

where a is a constant. Expanding $\log b_j(\boldsymbol{o}_t)$ the auxiliary function is,

$$Q(\lambda, \bar{\lambda}) \quad = \quad a - \frac{1}{2}\mathcal{F}(\boldsymbol{O}|\lambda) \sum_{j=1}^{S} \sum_{t=1}^{T} \gamma_j(t)[n \log(2\pi) + \log|\boldsymbol{\Sigma}_j| + h(\boldsymbol{o}_t, j)] \tag{7.6}$$

where $\quad h(\boldsymbol{o}_t, j) = (\boldsymbol{o}_t - \hat{\boldsymbol{W}}_{\boldsymbol{j}}\boldsymbol{\xi}_j)'\boldsymbol{\Sigma}_j^{-1}(\boldsymbol{o}_t - \hat{\boldsymbol{W}}_{\boldsymbol{j}}\boldsymbol{\xi}_j)$.

The differential of $Q(\lambda, \bar{\lambda})$ with respect to $\hat{\boldsymbol{W}}_{\boldsymbol{s}}$ is,

$$\frac{d}{d\hat{\boldsymbol{W}}_{\boldsymbol{s}}} Q(\lambda, \bar{\lambda}) = -\frac{1}{2}\mathcal{F}(\boldsymbol{O}|\lambda) \frac{d}{d\hat{\boldsymbol{W}}_{\boldsymbol{s}}} \sum_{j=1}^{S} \sum_{t=1}^{T} \gamma_j(t)[n \log(2\pi) + \log|\boldsymbol{\Sigma}_j| + h(\boldsymbol{o}_t, j)] \tag{7.7}$$

Equating to zero to find the maximum of $Q(\lambda, \bar{\lambda})$,

$$\frac{d}{d\hat{\boldsymbol{W}}_{\boldsymbol{s}}} Q(\lambda, \bar{\lambda}) = \mathcal{F}(\boldsymbol{O}|\lambda) \sum_{t=1}^{T} \gamma_s(t) \boldsymbol{\Sigma}_s^{-1} \left[\boldsymbol{o}_t - \hat{\boldsymbol{W}}_{\boldsymbol{s}}\boldsymbol{\xi}_s\right] \boldsymbol{\xi}_s' = 0 \tag{7.8}$$

hence,

$$\sum_{t=1}^{T} \gamma_s(t) \boldsymbol{\Sigma}_s^{-1} \boldsymbol{o}_t \boldsymbol{\xi}_s' = \sum_{t=1}^{T} \gamma_s(t) \boldsymbol{\Sigma}_s^{-1} \hat{\boldsymbol{W}}_{\boldsymbol{s}} \boldsymbol{\xi}_s \boldsymbol{\xi}_s'. \tag{7.9}$$

Equation 7.9 gives the general form for computing $\hat{\boldsymbol{W}}_{\boldsymbol{s}}$. In the following sections a closed form solution for the case of diagonal covariance matrices is presented. The case of full covariance matrices has no closed form solution and is not considered further.

7.2.2 Re-estimation Formula for Tied Regression Matrices

When the regression matrices are tied across a number of distributions (e.g. a global regression matrix) the summations must be performed over all tied distributions. If $\boldsymbol{W}_{\boldsymbol{s}}$ is shared by R states $\{s_1, s_2 \ldots s_R\}$ equation 7.9 becomes,

$$\sum_{t=1}^{T} \sum_{r=1}^{R} \gamma_{s_r}(t) \boldsymbol{\Sigma}_{s_r}^{-1} \boldsymbol{o}_t \boldsymbol{\xi}_{s_r}' = \sum_{t=1}^{T} \sum_{r=1}^{R} \gamma_{s_r}(t) \boldsymbol{\Sigma}_{s_r}^{-1} \hat{\boldsymbol{W}}_{\boldsymbol{s}} \boldsymbol{\xi}_{s_r} \boldsymbol{\xi}_{s_r}'. \tag{7.10}$$

To derive a re-estimation formula for the tied case, equation 7.10 is first rewritten as

$$\sum_{t=1}^{T} \sum_{r=1}^{R} \gamma_{s_r}(t) \boldsymbol{\Sigma}_{s_r}^{-1} \boldsymbol{o}_t \boldsymbol{\xi}_{s_r}' = \sum_{r=1}^{R} \boldsymbol{V}^{(r)} \hat{\boldsymbol{W}}_{\boldsymbol{s}} \boldsymbol{D}^{(r)} \tag{7.11}$$

where $\boldsymbol{V}^{(r)}$ is the state distribution inverse covariance matrix scaled by the state occupation probability,

$$\boldsymbol{V}^{(r)} = \sum_{t=1}^{T} \gamma_{s_r}(t) \boldsymbol{\Sigma}_{s_r}^{-1} \tag{7.12}$$

and $D^{(r)}$ is the outer product of the extended mean vectors

$$\boldsymbol{D}^{(r)} = \boldsymbol{\xi}_{s_r} \boldsymbol{\xi}_{s_r}'. \tag{7.13}$$

If the right hand side of equation 7.11 is denoted by the $n \times (n+1)$ matrix $\boldsymbol{Y}$ with elements y_{ij}, the individual matrix elements of $\boldsymbol{V}^{(r)}$, $\boldsymbol{W}_{\boldsymbol{s}}$ and $\boldsymbol{D}^{(r)}$ are denoted by $v_{ij}^{(r)}$, w_{ij} and $d_{ij}^{(r)}$ respectively, then

$$y_{ij} = \sum_{p=1}^{n} \sum_{q=1}^{n+1} w_{pq} \left[\sum_{r=1}^{R} v_{ip}^{(r)} d_{qj}^{(r)} \right]. \tag{7.14}$$

If all covariances are diagonal, and since $\boldsymbol{D}$ is symmetric, then

$$\sum_{r=1}^{R} v_{ip}^{(r)} d_{qj}^{(r)} = \begin{cases} \sum_{r=1}^{R} v_{ii}^{(r)} d_{jq}^{(r)} & \text{when} \quad i = p \\ 0 & \text{when} \quad i \neq p \end{cases} \tag{7.15}$$

so

$$y_{ij} = \sum_{q=1}^{n+1} w_{iq} g_{jq}^{(i)}$$

where $g_{jk}^{(i)}$ are the elements of the $(n+1) \times (n+1)$ matrix $\boldsymbol{G}^{(i)}$, given by

$$g_{jq}^{(i)} = \sum_{r=1}^{R} v_{ii}^{(r)} d_{jq}^{(r)}. \tag{7.16}$$

If the left hand side of (7.11) is an $n \times (n+1)$ matrix denoted by $\boldsymbol{Z}$ with elements z_{ij}, then $\boldsymbol{Z} = \boldsymbol{Y}$ and

$$z_{ij} = y_{ij} = \sum_{q=1}^{n+1} w_{iq} g_{jq}^{(i)}. \tag{7.17}$$

It should be noted that z_{ij} and $g_{jq}^{(i)}$ are not dependent on $\hat{\boldsymbol{W}}_{\boldsymbol{s}}$ and both can be computed from the observation vectors and the model parameters. Hence, a system of simultaneous equations is generated to compute $\hat{\boldsymbol{W}}_{\boldsymbol{s}}$

$$\mathbf{w}_i' = \boldsymbol{G}^{(i)-1} \mathbf{z}_i' \tag{7.18}$$

where $\mathbf{w}_i$ and $\mathbf{z}_i$ are the i^{th} rows of $\hat{\boldsymbol{W}}_{\boldsymbol{s}}$ and $\boldsymbol{Z}$ respectively. These equations can be solved using Gaussian elimination or LU decomposition methods to calculate $\hat{\boldsymbol{W}}_{\boldsymbol{s}}$ on a row by row basis.

This estimate of $\hat{\boldsymbol{W}}_{\boldsymbol{s}}$ increases the likelihood of the adaptation data being generated by the models. By iterating the estimation, the likelihood can be maximised.

7.2.3 Extension to Mixture Distributions

For states with output distributions made up of M mixture components the probability density function for the state is given by

$$b_s(\boldsymbol{o}) = \sum_{k=1}^{M} c_{sk} b_{sk}(\boldsymbol{o}) \tag{7.19}$$

where $b_{sk}(\boldsymbol{o})$ is the k^{th} Gaussian mixture component of state s and c_{sk} the associated component weighting. Thus the likelihood function becomes

$$\begin{aligned}
\mathcal{F}(\boldsymbol{O}, \boldsymbol{\theta}|\lambda) &= a_{\theta_T,N} \prod_{t=1}^{T} a_{\theta_{t-1}\theta_t} \left[\sum_{k=1}^{M} c_{\theta_t k} b_{\theta_t k}(\boldsymbol{o}_t) \right] &(7.20)\\
&= \sum_{k_1=1}^{M} \sum_{k_2=1}^{M} \cdots \sum_{k_T=1}^{M} \left[a_{\theta_T,N} \prod_{t=1}^{T} a_{\theta_{t-1}\theta_t} c_{\theta_t k_t} b_{\theta_t k_t}(\boldsymbol{o}_t) \right]. &(7.21)
\end{aligned}$$

Defining

$$\mathcal{F}(\boldsymbol{O}, \boldsymbol{\theta}, K|\lambda) = a_{\theta_T N} \prod_{t=1}^{T} a_{\theta_{t-1}\theta_t} c_{\theta_t k_t} b_{\theta_t k_t}(\boldsymbol{o}_t) \tag{7.22}$$

and Ω_b as the set of all possible branch sequences (mixture component sequences) of length T, where such a sequence is $K = (k_1, k_2 \ldots, k_T)$, the joint density of the stochastic process is

$$\mathcal{F}(\boldsymbol{O}|\lambda) = \sum_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} \sum_{K \in \Omega_b} \mathcal{F}(\boldsymbol{O}, \boldsymbol{\theta}, K|\lambda). \tag{7.23}$$

This can be interpreted by considering each set of state sequences which generates $\boldsymbol{O}$ as being a superposition of M^T branch layers.

The auxiliary function is now redefined to take the branch layers into consideration:

$$\begin{aligned} Q(\lambda, \bar{\lambda}) &= \sum_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} \sum_{K \in \Omega_b} \mathcal{F}(\boldsymbol{O}, \boldsymbol{\theta}, K|\lambda) \log \mathcal{F}(\boldsymbol{O}, \boldsymbol{\theta}, K|\bar{\lambda}) \qquad (7.24) \\ &= \sum_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} \sum_{K \in \Omega_b} \mathcal{F}(\boldsymbol{O}, \boldsymbol{\theta}, K|\lambda) \left\{ \log \bar{a}_{\theta_T N} + \sum_{t=1}^{T} \log \bar{a}_{\theta_{t-1}\theta_t} + \right. \\ &\qquad \left. \sum_{t=1}^{T} \log \bar{b}_{\theta_t k_t}(\boldsymbol{o}_t) + \sum_{t=1}^{T} \log \bar{c}_{\theta_t k_t} \right\} \qquad (7.25) \end{aligned}$$

Again, only considering those elements relevant to the regression transform this reduces to

$$Q(\lambda, \bar{\lambda}) = constant + \sum_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} \sum_{K \in \Omega_b} \sum_{t=1}^{T} \mathcal{F}(\boldsymbol{O}, \theta_t = j, k_t = k|\lambda) \log \bar{b}_{jk}(\boldsymbol{o}_t) \tag{7.26}$$

Defining $\gamma_{jk}(t)$ as the total occupation probability of mixture component k of state j at time t given that the observation sequence $\boldsymbol{O}$ is generated:

$$\gamma_{jk}(t) = \frac{1}{\mathcal{F}(\boldsymbol{O}|\lambda)} \sum_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} \sum_{K \in \Omega_b} \mathcal{F}(\boldsymbol{O}, \theta_t = j, k_t = k|\lambda) \tag{7.27}$$

reduces equation 7.26 to

$$Q(\lambda, \bar{\lambda}) = constant + \mathcal{F}(\boldsymbol{O}|\lambda) \sum_{t=1}^{T} \gamma_{jk}(t) \log \bar{b}_{jk}(\boldsymbol{o}_t) \tag{7.28}$$

which is equivalent to equation 7.5 for the single Gaussian case.

Thus by substituting mixture component occupation probabilities for state occupation probabilities, transformations for individual mixture components can be derived.

7.2.4 Multiple Observation Sequences

The transform has been derived for the case of a single observation sequence. The extension to the more general case of multiple observation sequences is trivial.

Given a set of P observation sequences $\boldsymbol{O}^{(1)}\ldots\boldsymbol{O}^{(P)}$ with observation sequence $\boldsymbol{O}^{(p)}$ having T_p observation frames ($\boldsymbol{O}^{(p)} = \boldsymbol{o}_1^{(p)}\ldots\boldsymbol{o}_{T_p}^{(p)}$), and considering a single component mixture per state, equation 7.9 becomes:

$$\sum_{p=1}^{P}\sum_{t=1}^{T_p}\gamma_s^{(p)}(t)\boldsymbol{\Sigma}_s^{-1}\boldsymbol{o}_t^{(p)}\boldsymbol{\xi}_s' = \sum_{p=1}^{P}\sum_{t=1}^{T_p}\gamma_s^{(p)}(t)\boldsymbol{\Sigma}_s^{-1}\hat{\boldsymbol{W}}_s\boldsymbol{\xi}_s\boldsymbol{\xi}_s' \tag{7.29}$$

where

$$\gamma_s^{(p)}(t) = \frac{1}{\mathcal{F}(\boldsymbol{O}^{(p)}|\lambda)}\sum_{\boldsymbol{\theta}\in\boldsymbol{\Theta}_{T_p}}\mathcal{F}(\boldsymbol{O}^{(p)},\theta_t = s|\lambda) \tag{7.30}$$

and Θ_{T_p} is the set of all possible state sequences of length T_p.

The tied regression matrix formula (equation 7.11) then becomes

$$\sum_{p=1}^{P}\sum_{t=1}^{T_p}\sum_{r=1}^{R}\gamma_{s_r}(t)\boldsymbol{\Sigma}_{s_r}^{-1}\boldsymbol{o}_t^{(p)}\boldsymbol{\xi}_{s_r}' = \sum_{r=1}^{R}\boldsymbol{V}^{(r)}\hat{\boldsymbol{W}}_{\boldsymbol{s}}\boldsymbol{D}^{(r)} \tag{7.31}$$

where $\boldsymbol{D}^{(r)}$ is the extended mean outer product matrix as before and $\boldsymbol{V}^{(r)}$ is the state distribution inverse covariance matrix scaled by the state occupation probability:

$$\boldsymbol{V}^{(r)} = \sum_{p=1}^{P}\sum_{t=1}^{T}\gamma_{s_r}^{(p)}(t)\boldsymbol{\Sigma}_{s_r}^{-1} \tag{7.32}$$

Hence $\boldsymbol{G}^{(i)}$ and $\boldsymbol{Z}$ can be calculated and the transform estimated as before.

7.3 Incremental Adaptation

The MLLR equations presented in the solution of equation 7.11 assumed that all the adaptation data had been observed before the transform was estimated. By re-arranging the equations to separate the components which are dependent on time, a formulation for incremental adaptation is found.

Re-arranging equation 7.12 results in

$$\boldsymbol{V}^{(r)} = \left[\sum_{t=1}^{T}\gamma_{s_r}(t)\right]\boldsymbol{\Sigma}_{s_r}^{-1} \tag{7.33}$$

and the left hand side of equation 7.11, $\boldsymbol{Z}$, can be written as

$$\boldsymbol{Z} = \sum_{r=1}^{R}\boldsymbol{\Sigma}_{s_r}^{-1}\left[\sum_{t=1}^{T}\gamma_{s_r}(t)\boldsymbol{o}_t\right]\boldsymbol{\xi}_{s_r}'. \tag{7.34}$$

A full MLLR estimation of the transforms can then be made at any time, based on the observed data up to the current point in time. The adapted models may be used to get an improved alignment for future adaptation observations. This is equivalent to a single iteration of static adaptation, except that alignments are generated using repeatedly updated models.

7.4 Computational Considerations

Calculating the MLLR transform is computationally expensive. Consider a single transform. Computing each $g_{jq}^{(i)}$ requires $2R$ multiplications therefore each $\boldsymbol{G}^{(i)}$ requires $2R(n+1)^2$ multiplications. Assuming that the mixture component occupation probabilities and the observation vectors are efficiently accumulated $R(n+2)n$ multiplications are required to compute $\boldsymbol{Z}$. The computational overhead for one $(n \times n)$ regression matrix is $2Rn(n+1)^2 + Rn(n+2)$ multiplications plus n matrix inversions, i.e. computation of order n^3.

The matrix $\boldsymbol{G}^{(i)}$, which must be inverted, is the summed weighted outer product of two mean vectors. Each outer product is a singular matrix and the resulting matrix $\boldsymbol{G}^{(i)}$ may be very close to singular if only a few examples of regression class members are observed in the adaptation data. To overcome this problem it is suggested that the matrix inversion be performed using a robust method. The Moore-Penrose pseudo inverse is ideally suited to this problem, although computationally expensive,

$$\begin{array}{ccl} \text{If} & \mathbf{A} = \mathbf{XDY}' & \text{the singular value decomposition of } \mathbf{A} \\ \text{then} & \mathbf{A}^{\dagger} = \mathbf{Y}'\mathbf{D}^{-1}\mathbf{X} & \text{is the pseudo inverse} \end{array}$$

$\mathbf{X}$ and $\mathbf{Y}$ are orthogonal matrices and $\mathbf{D}$ is diagonal.

7.5 Diagonal Regression Matrices

A full $n \times (n+1)$ matrix implements n^{th} order multivariate linear regression. This can be computationally expensive and the complexity of the regression may be unnecessary. A considerable computational saving can be made if the regression matrices are restricted to a diagonal form. This effectively implements a single variable regression for each element within the feature vector. Consider the matrix

$$\hat{\boldsymbol{W}}_{\boldsymbol{s}} = \begin{pmatrix} w_{1,1} & w_{1,2} & 0 & \dots & 0 \\ w_{2,1} & 0 & w_{2,3} & \dots & 0 \\ \vdots & & & & \vdots \\ w_{n,1} & \dots & \dots & 0 & w_{n,n+1} \end{pmatrix} \tag{7.35}$$

then the transformation to update the mean value $\hat{\boldsymbol{\mu}} = \hat{\boldsymbol{W}}_{\boldsymbol{s}}\boldsymbol{\xi}$ results in each component of the mean undergoing a shift and a scaling

$$\hat{\mu}_i = \omega w_{1i} + w_{i+1,i}\mu_i. \tag{7.36}$$

Rewriting the matrix as a vector

$$\hat{\boldsymbol{w}}_s = \begin{pmatrix} w_{1,1} \\ \vdots \\ w_{n,1} \\ w_{1,2} \\ \vdots \\ w_{n,n+1} \end{pmatrix}$$

the quadratic form in the output density becomes

$$(\boldsymbol{o}_t - \hat{\boldsymbol{W}}_s \hat{\boldsymbol{\mu}}_s)' \boldsymbol{\Sigma}_s^{-1} (\boldsymbol{o}_t - \hat{\boldsymbol{W}}_s \hat{\boldsymbol{\mu}}_s) = (\boldsymbol{o}_t - \boldsymbol{D}_s \hat{\boldsymbol{w}}_s)' \boldsymbol{\Sigma}_s^{-1} (\boldsymbol{o}_t - \boldsymbol{D}_s \hat{\boldsymbol{w}}_s) \tag{7.37}$$

where $\boldsymbol{D}_s$ is a $n \times 2n$ matrix

$$\boldsymbol{D}_s = \begin{pmatrix} \omega & 0 & \dots & \dots & 0 & \mu_1 & 0 & \dots & & 0 \\ 0 & \omega & 0 & \dots & \dots & 0 & \mu_2 & 0 & \dots & 0 \\ \vdots & & & & & & & & & \vdots \\ 0 & \dots & 0 & \omega & 0 & \dots & \dots & 0 & \mu_{n-1} & 0 \\ 0 & \dots & \dots & 0 & \omega & 0 & \dots & \dots & 0 & \mu_n \end{pmatrix}. \tag{7.38}$$

Substituting the new form of the quadratic in the output density for each state and following the standard MLLR derivation leads to a formula for the values of the regression matrix elements in the tied case

$$\hat{\boldsymbol{w}}_s = \left[\sum_{r=1}^{R} \sum_{t=1}^{T} \sum_{t=1}^{T} \gamma_s(t) \boldsymbol{D}'_{s_r} \boldsymbol{\Sigma}_{s_r}^{-1} \boldsymbol{D}_{s_r} \right]^{-1} \left[\sum_{r=1}^{R} \sum_{t=1}^{T} \sum_{t=1}^{T} \gamma_{s_r}(t) \boldsymbol{D}'_{s_r} \boldsymbol{\Sigma}_{s_r}^{-1} \boldsymbol{o}_t \right] \tag{7.39}$$

When including the offset term one matrix inversion is required to calculate the regression matrix entries, and by ignoring the offset term ($\omega = 0$) the offset column can be ignored, and all matrices can be reduced to diagonal matrices making inversion trivial. Thus computation is significantly reduced over the full regression matrix case.

7.6 Regression Classes

The MLLR formulae have been derived with a flexible transformation tying structure. The effectiveness of the tying clearly depends on a good choice of mixture components to associate with each transformation matrix. For this purpose the idea of *regression classes* is introduced. The set of mixture components which are associated with the same transformation are placed in the same regression class. The aim is to select the members of each regression class so that a good transformation is generated for all members. The actual definition of a good transformation is an abstract term since the necessary transformations will be different for different speakers and the regression class definitions are needed to generate them in the first place. An interpretation using the acoustic space is considered instead. Assuming that if two acoustic classes are represented in a common region of acoustic space for a speaker (speaker A), then the same two acoustic classes will be represented in a common region (but possibly a different region to that of speaker A) of acoustic space for a different speaker (speaker B) (Figure 7.1).

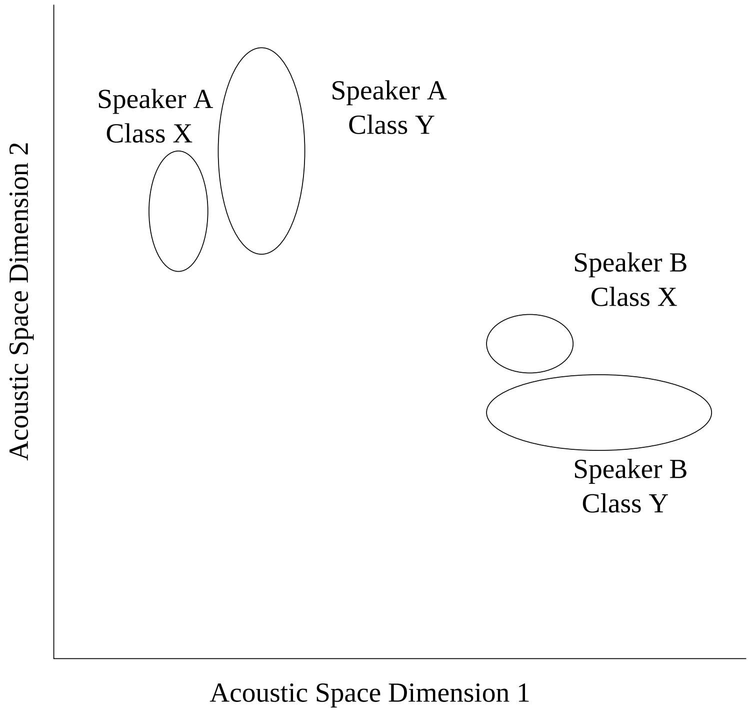

Figure 7.1: Example of assumption of similar relative acoustic distributions from speaker to speaker

Under this assumption the mixture components representing acoustic classes located in similar acoustic space should be placed in the same regression class (Figure 7.2). Although the relative positions of classes in acoustic space may vary with different speakers, if an average of many speakers is taken (e.g. from a speaker independent system) a good estimate of appropriate groupings may be made. A further consideration is that the mixture components in each regression class must contribute sufficient adaptation data to the class so that a robust estimate of the regression matrix can be made. Thus if the adaptation data is known in advance the regression classes can be selected accordingly.

7.7 Adapting Other Parameters with MLLR

The MLLR approach has only considered adaptation of the mean vectors of the mixture components. For complete adaptation of the system it would be necessary to update all the model parameters. However, this is similar to a complete retraining of the system which would require large amounts of speaker specific data. MLLR is intended to address the case where the amount of adaptation is limited, and hence makes a general update of the mean

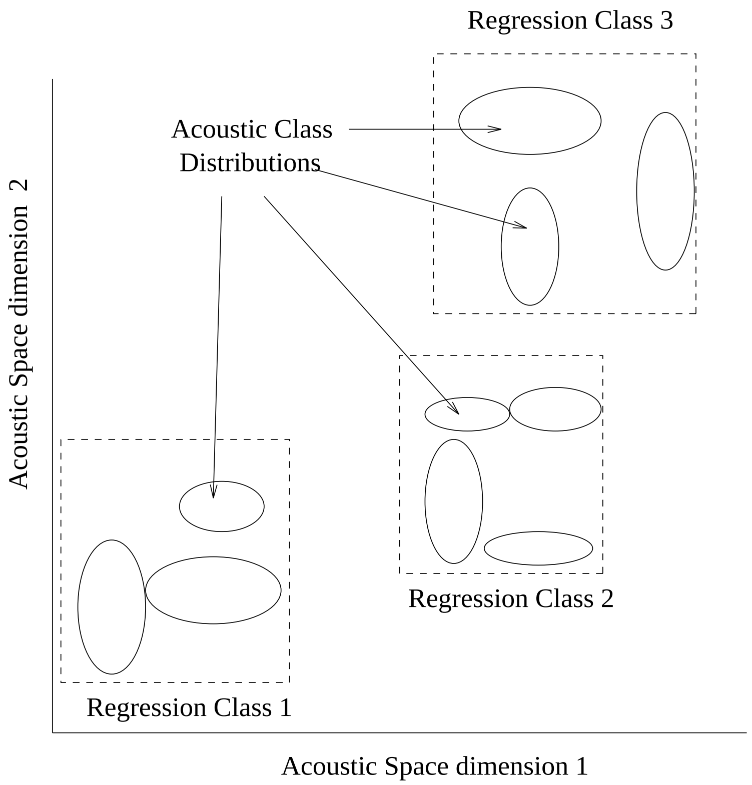

Figure 7.2: Example regression class members in a 2-dimensional acoustic space

parameters through the tying of transforms. The other parameters in the model set are not altered. The effect of not adapting the other parameters is briefly discussed below.

- The transition probabilities (a_{ij}).
  Very few methods adjust the transition probabilities on a speaker-by-speaker basis, with MAP adaptation being one example. However, the effect of transitions is small in a continuous density system, and not adapting the transition parameters will not adversely affect the adaptation.

- The mixture component weights (c_{ik}).
  Multiple mixture component states can be expanded to parallel individual states, with transition probabilities computed relative to the mixture weights. Hence, similar arguments to the case of transition probabilities can be made.

- The covariance matrices.
  Although tying of the covariance matrices has been performed within HMM systems (e.g. grand covariance systems), the nature of the covariance distribution makes it unsuitable for tying between transformations. The use of individual covariances in each distribution has been proved to be effective indicating the importance of having a correct shape for the distribution of examples of each acoustic class. Transformation of the covariance would result not in a change in position of the class in acoustic

space, but a change in the shape of the distribution. Although Digalakis [34] suggests adapting the covariance matrix with the same transformation of the mean, a similar idea is rejected in this case. The MLLR transform generates a relocation of the mean of each component in the current acoustic feature space, but does not transform the space. Applying the transform would result in the covariance of the distribution being rotated in the feature space, but the rotation would be generated from the change in the mean, not the appropriate covariance.

Although in theory these parameters should be adapted to the new speaker, it is clear from the success of spectral transformation techniques that successful adaptation can be performed without considering all of the parameters.

7.8 Alignment of Data

The success of the MLLR method depends on assigning frames of the adaptation data to the HMM states and mixture components. Assuming that the transcription of the adaptation data is given, standard alignment strategies such as the forward-backward algorithm or the Viterbi algorithm can be used. The forward-backward algorithm allows each frame of adaptation data to contribute to several states, which is possibly advantageous in the case of small amounts of data. A schematic diagram of the alignment process is shown in figure 7.3.

The adaptation data and transcriptions are first passed through a filter to resolve ambiguous pronunciations. This filter would normally consist of a Viterbi alignment using the SI models and a pronunciation network. The appropriate models representing the chosen pronunciations are then selected and ordered, and an alignment of the data produced. Each observation vector is then allocated to the regression classes with the appropriate weighting.

For the case of supervised adaptation the transcriptions are provided, for unsupervised adaptation the appropriate transcriptions must be generated within the system by passing the adaptation data through a recognition module.

7.9 MLLR Matrices as Input Transformations

In theory MLLR can be used to implement adaptation through speaker normalisation by applying the regression matrix to the input speech instead of the mixture component means.

Using a global transformation matrix to transform the observation vectors the probability density is

$$b_j(\boldsymbol{o}) = \frac{1}{(2\pi)^{\frac{n}{2}}|\boldsymbol{\Sigma}_s|^{\frac{1}{2}}} e^{-\frac{1}{2}(\boldsymbol{W}\hat{\boldsymbol{o}}-\boldsymbol{\mu}_s)'\boldsymbol{\Sigma}_s^{-1}(\boldsymbol{W}\hat{\boldsymbol{o}}-\boldsymbol{\mu}_s)}. \tag{7.40}$$

where $\hat{\boldsymbol{o}}$ is the extended observation vector for time t

$$\hat{\boldsymbol{o}}_t = \left[\ \omega, o_{t_1}, \ldots, o_{t_n}\ \right]'. \tag{7.41}$$

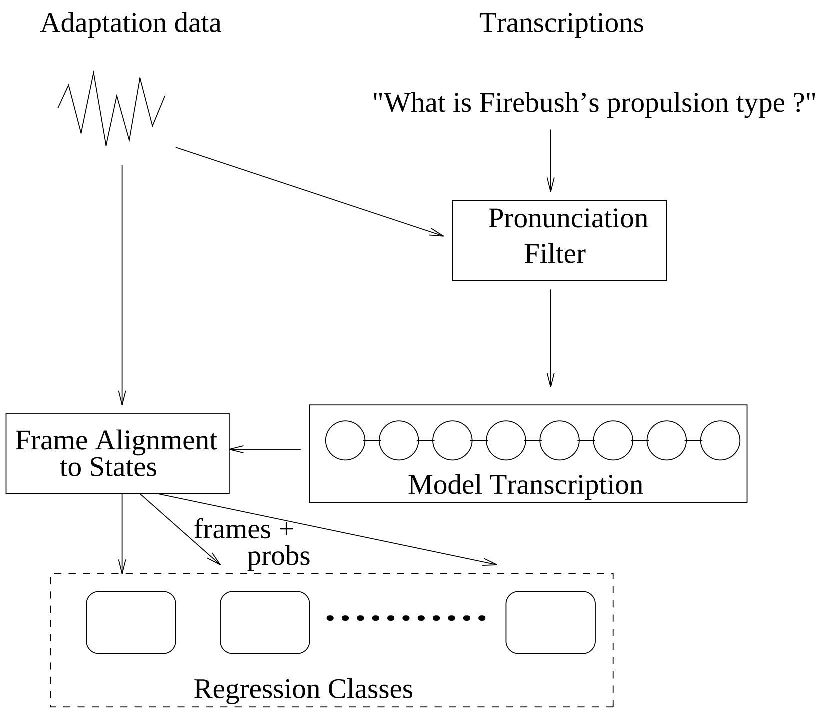

Figure 7.3: Alignment of the adaptation data, and allocation of data to tied regression classes

The estimate of the global transformation is performed in a similar manner to standard MLLR estimation. The auxiliary function is redefined as,

$$\begin{aligned} Q_s(\boldsymbol{W}, \hat{\boldsymbol{W}}) &= -\frac{1}{2}\mathcal{F}(\boldsymbol{O}|\lambda, \boldsymbol{W}) \sum_{t=1}^{T} \gamma_s(t)[n \log(2\pi) + \log|\boldsymbol{\Sigma}_s| + \\ &\qquad (\hat{\boldsymbol{W}}\hat{\boldsymbol{o}}_t - \boldsymbol{\mu}_s)'\boldsymbol{\Sigma}_s^{-1}(\hat{\boldsymbol{W}}\hat{\boldsymbol{o}}_t - \boldsymbol{\mu}_s)] \end{aligned} \tag{7.42}$$

with the differential

$$\frac{d}{d\hat{\boldsymbol{W}}} Q_s(\boldsymbol{W}, \hat{\boldsymbol{W}}) = \mathcal{F}(\boldsymbol{O}|\lambda, \boldsymbol{W}) \sum_{t=1}^{T} \gamma_s(t)\boldsymbol{\Sigma}_s^{-1} \left[\hat{\boldsymbol{W}}\hat{\boldsymbol{o}}_t - \boldsymbol{\mu}_s\right] \hat{\boldsymbol{o}}_t'. \tag{7.43}$$

Equating to zero and summing over all states,

$$\sum_{t=1}^{T} \sum_{s \in S} \gamma_s(t)\boldsymbol{\Sigma}_s^{-1}\boldsymbol{\mu}_s\hat{\boldsymbol{o}}_t' = \sum_{t=1}^{T} \sum_{s \in S} \boldsymbol{V}^{(s,t)}\hat{\boldsymbol{W}}\boldsymbol{D}^{(s,t)} \tag{7.44}$$

where $\boldsymbol{V}^{(s,t)}$ is the state distribution inverse covariance matrix scaled by the state occupation probability,

$$\boldsymbol{V}^{(s,t)} = \gamma_s(t)\boldsymbol{\Sigma}_s^{-1} \tag{7.45}$$

and $\boldsymbol{D}^{(s,t)}$ is the outer product of the extended observation vectors at time t

$$\boldsymbol{D}^{(s,t)} = \hat{\boldsymbol{o}}_t \hat{\boldsymbol{o}}_t' \tag{7.46}$$

If the individual matrix elements of $\boldsymbol{V}^{(s,t)}$,$\boldsymbol{W}$ and $\boldsymbol{D}^{(s,t)}$ are denoted by $v_{ij}^{(s,t)}$, w_{ij} and $d_{ij}^{(s,t)}$ respectively, the right hand side of equation (7.44) is an $n \times (n+1)$ matrix $\boldsymbol{Y}$ with individual elements y_{ij} given by

$$y_{ij} = \sum_{p=1}^{n} \sum_{q=1}^{n+1} w_{pq} \left[\sum_{s \in S} \sum_{t=1}^{T} v_{ip}^{(s,t)} d_{qj}^{(s,t)} \right] \tag{7.47}$$

Since $\boldsymbol{D}^{(s,t)}$ is symmetric and assuming diagonal covariances

$$y_{ij} = \sum_{q=1}^{n+1} w_{iq} g_{jq}^{(i)} \tag{7.48}$$

where $g_{jk}^{(i)}$ are the elements of the $(n+1) \times (n+1)$ matrix $\boldsymbol{G}^{(i)}$ which is the sum of the outer products of the extended observation vectors scaled by the i^{th} diagonal component of the weighted mixture component inverse variance of all mixture components ($\boldsymbol{D}^{(s,t)}$ scaled by $v_{ii}^{(s,t)}$)

$$g_{jq}^{(i)} = \sum_{s \in S} \sum_{t=1}^{T} v_{ii}^{(s,t)} d_{jq}^{(s,t)} \tag{7.49}$$

As before, the $n \times (n+1)$ matrix $\boldsymbol{Z}$ $(= [z_{ij}])$ is defined as the l.h.s. of equation 7.44,

$$\boldsymbol{Z} = \sum_{t=1}^{T} \sum_{s \in S} \gamma_s(t) \boldsymbol{\Sigma}_s^{-1} \boldsymbol{\mu}_s \hat{\boldsymbol{o}}_t' \tag{7.50}$$

and,

$$z_{ij} = y_{ij} = \sum_{q=1}^{n+1} w_{iq} g_{jq}^{(i)} \tag{7.51}$$

so a set of simultaneous equations can be generated and an estimate of $\boldsymbol{W}$ made.

Although this derivation follows similar arguments to the standard MLLR transform, the nature of the transformation generated is significantly different. Instead of generating a transform which relocates the position of each acoustic class in the acoustic space (transformation of the mean vectors), the observation vectors are transformed in an attempt to locate the appropriate vectors in the correct region of the acoustic space.

7.10 Comparison to Least Squares Regression

The MLLR approach is similar in essence to the least squares approach suggested by Hewett [49], although it differs significantly in the treatment of covariances. Least squares makes the assumption that shape of the distributions of each acoustic class modelled are identical.

The discussion in section 4.3 shows this is a poor assumption which MLLR overcomes through incorporating the covariances into the optimisation. Least squares regression can be shown to be a specific case of the more general MLLR.

Assuming all the covariance matrices of the distributions assigned to the same class are the same ($\boldsymbol{\Sigma}_{s_1} = \boldsymbol{\Sigma}_{s_2} = \ldots = \boldsymbol{\Sigma}_{s_R}$) equation 7.10 becomes,

$$\sum_{t=1}^{T}\sum_{r=1}^{R} \gamma_{s_r}(t)\boldsymbol{o}_t\hat{\boldsymbol{\mu}}'_{s_r} = \sum_{t=1}^{T}\sum_{r=1}^{R} \gamma_{s_r}(t)\hat{\boldsymbol{W}}_{\boldsymbol{s}}\hat{\boldsymbol{\mu}}_{s_r}\hat{\boldsymbol{\mu}}'_{s_r} \tag{7.52}$$

If each speech frame is assigned to exactly one distribution (e.g. by Viterbi alignment) so that

$$\gamma_{s_r}(t) = \begin{cases} 1 & \text{if } \boldsymbol{o}_t \text{ is assigned to state distribution } s_r \\ 0 & \text{otherwise} \end{cases} \tag{7.53}$$

then equation 7.52 becomes

$$\sum_{t=1}^{T} \boldsymbol{o}_t\hat{\boldsymbol{\mu}}'_{\theta_t}\delta_{s\theta_t} = \hat{\boldsymbol{W}}_{\boldsymbol{s}} \sum_{t=1}^{T} \hat{\boldsymbol{\mu}}_{\theta_t}\hat{\boldsymbol{\mu}}'_{\theta_t}\delta_{s\theta_t} \tag{7.54}$$

where

$$\delta_{s\theta_t} = \begin{cases} 1 & \text{if } \theta_t \in \{s_1 \ldots s_R\} \\ 0 & \text{otherwise} \end{cases} \tag{7.55}$$

Defining the matrices $\boldsymbol{X}$ and $\boldsymbol{Y}$ as

$$\begin{aligned} \boldsymbol{X} &= \begin{bmatrix} \hat{\boldsymbol{\mu}}_{\theta_1} & \hat{\boldsymbol{\mu}}_{\theta_2} & \ldots & \hat{\boldsymbol{\mu}}_{\theta_T} \end{bmatrix} \\ \boldsymbol{Y} &= \begin{bmatrix} \boldsymbol{o}_1\delta_{\theta_1 m} & \boldsymbol{o}_2\delta_{\theta_2 m} & \ldots & \boldsymbol{o}_T\delta_{\theta_T m} \end{bmatrix} \end{aligned}$$

equation 7.54 becomes,

$$\boldsymbol{Y}\boldsymbol{X}' = \hat{\boldsymbol{W}}_{\boldsymbol{s}}\boldsymbol{X}\boldsymbol{X}' \tag{7.56}$$

so the estimate of the regression matrix is

$$\hat{\boldsymbol{W}}_{\boldsymbol{s}} = \boldsymbol{Y}\boldsymbol{X}'(\boldsymbol{X}\boldsymbol{X}')^{-1} \tag{7.57}$$

which is the least squares estimate [59].

This suggests that MLLR should perform at least as well as the least squares approach.

7.11 Summary

The MLLR adaptation method has been described and appropriate optimisation formulae derived. This method transforms the mean vectors of all the mixture components in a continuous density HMM system. However, a closed form solution is only available for HMMs with diagonal covariances. The method can be implemented either using a global transformation, or, if there is sufficient data available to make robust estimates, more specific transformations. For this purpose the idea of regression classes has been introduced, with all the mixture components in each class being transformed by the same matrix.

The transformation matrices can either be full $n \times (n+1)$ matrices or more restricted diagonal matrices which implement single variable regression for each element of the feature vector. The diagonal matrices are computationally more efficient and require less data for robust estimation. However, they are less well suited to making general regression estimates when a large degree of tying is involved.

The MLLR method makes a maximum likelihood estimate of the regression matrices, and it can be shown that the least squares estimate is a special case. The MLLR method can also be used for implementing speaker normalisation by applying the transformation to the input speech instead of the mean vectors.

Chapter 8

Evaluation of MLLR Transformations for Speaker Adaptation

The efficacy of the MLLR transformation approach for adapting model parameters to a new speaker is now investigated. Many experiments have been performed to show that the MLLR estimate of the transformations is both effective and feasible in a general continuous density HMM framework. The basic experimental evaluation has been performed using the speaker dependent portion of the RM1 database. This has been selected since it provides a reasonable number of speaker dependent training utterances from 12 different speakers. Further evaluation used the much larger WSJ database to demonstrate the effectiveness of MLLR on different databases.

For the RM experiments, the adaptation data is drawn from the training utterances, and evaluation results generated using all of the speaker dependent test sentences. For the WSJ experiments the adaptation and evaluation correspond to the requirements of the 1994 ARPA sponsored continuous speech recognition evaluation [65].

8.1 Model Sets used for Adaptation Evaluation

The speaker adaptation techniques have initially been evaluated on the RM1 speaker dependent corpus, and extended to the Wall Street Journal corpus (Appendix A). The model sets used were originally created by others [1] for speaker independent tasks.

8.1.1 Models for RM experiments.

Two types of models have been used for the RM experiments - word-internal triphones, and cross-word triphones.

[1] Many thanks to Julian Odell and Phil Woodland for providing the model sets

8.1.1.1 Word Internal Triphones

A set of state clustered word-internal triphones were generated as described in [107]. The 130 function word dependent phone set described in section 4.1 was used to train a set of monophones with a single mixture component diagonal covariance per state. These models were cloned to generate 2427 non-tied single mixture component triphones and the models reestimated. The corresponding states of all allophones were then clustered using a hierarchical furthest neighbour clustering algorithm. Using the Euclidean distance between the means weighted by the geometric mean of the variances as the intra-class distances, those components which result in the minimum increase in intra-class distance are clustered.

After the initial clustering a data sufficiency check was made using the training data and any classes with insufficient data were clustered with the nearest class. The distributions associated with the clustered states were tied, as were the 130 transition matrices which were associated with each allophone. The parameters of the HMMs were then reestimated to form the $R1_{\text{diag}}$ model set. The resulting model set after eliminating models with identical distributions had 2029 models and 1811 states [107].

To form the other model sets the number of components in the mixture density for each state was successively increased, and the models re-estimated using all 3990 SI training utterances.

- $R1_{\text{diag}}$ - 1 Gaussian component mixture density per state.
- $R2_{\text{diag}}$ - 2 Gaussian component mixture density per state.
- $R4_{\text{diag}}$ - 4 Gaussian component mixture density per state.
- $R6_{\text{diag}}$ - 6 Gaussian component mixture density per state.
- $R8_{\text{diag}}$ - 8 Gaussian component mixture density per state.

8.1.1.2 Cross-Word Triphones - R_{cwt}

A set of cross-word triphones was defined using the tree-based state clustering method described in [110]. Initially a set of monophones were defined and trained. The monophones were cloned to context dependent models and retrained. For each monophone a state-based decision tree was built using the state alignment of the training data. The decision trees are then used to construct a complete set of triphones, including those not present in the training data. The number of mixture components in each state was then increased and the models re-estimated.

For the R_{cwt} set the model training results in a system with 1778 states each with 6 mixture components. This is the Rm3 system described in [85].

8.1.2 Wall Street Journal Models

Two WSJ model sets using cross-word triphones have been used. Both are trained on the combined WSJ0 and WSJ1 databases (Appendix A) using a modified version of the tree-based state clustering method described in section 8.1.1.2.

8.1.2.1 Gender Independent Models - W_{a}

Phone-level pronunciations were generated using a pronunciation dictionary (1993 LIMSI WSJ Lexicon and phone set) and the sentence orthography in a Viterbi alignment. The alignment was used to build monophones, then triphones and finally a phonetic decision tree. The tree was then used to cluster the triphone contexts and the resulting set of distributions reestimated. The number of mixture components per state was then increased, and the models reestimated [106].

The final W_{a} model set is a gender independent triphone model set with 6399 speech states each with 12 component Gaussian mixture densities.

8.1.2.2 Gender Dependent Models W_{b}

The W_{b} model set is a gender dependent quinphone model set with 9354 speech states each with 14 component Gaussian mixture densities. The quinphones are generated in exactly the same manner as the W_{a} triphones, but using a quinphone context decision tree. A gender independent model set is first generated, and the final gender dependent set is obtained by a final re-train to estimate the mixture component mean values, on gender dependent data whilst keeping the variances fixed.

8.1.2.3 WSJ Language Models

For the WSJ recognition tasks two language models were used. The WSJ spokes S0, S3 and S4 were restricted to 5000 words, and the MIT Lincoln Labs 5k trigram language model was used. For the hub task H1-P0 a 65k word dictionary was used and the language models (bigram, trigram and 4-gram) were generated using 227 million words of data from the WSJ CSRNAB1 corpus.

8.2 Implementation Issues

The two main issues regarding the implementation of MLLR are the generation of the observation vector/state alignment and associated probabilities, and the generation of appropriate regression classes.

8.2.1 Alignment of the Data

There are two approaches to aligning the data: the forward-backward algorithm and the Viterbi algorithm. The forward-backward algorithm assigns each observation vector to all states with a probability distribution whereas the Viterbi alignment assigns each observation vector to exactly one state (see chapter 2). To align the data, a model transcription must be generated. Since the dictionary used for the RM experiments contains only single pronunciations this is achieved by a simple dictionary lookup of the word transcription. For other cases where there may be multiple pronunciations of words (e.g. the WSJ database) a forced alignment of the possible model combinations should be generated, and the most likely model sequence determined. It must be noted that such a forced alignment will use an existing model set which is not tuned to the speaker. This has the result of mapping pronunciations of the new speaker to the speakers in the training which can be misleading if the new speaker has a different accent/dialect (e.g. pronouncing the same vowel in a different manner) to those in the training set.

8.2.2 Generation of Class Definitions

As explained in section 7.6 mixture components representing similar regions in the acoustic feature space should be placed in the same regression classes. Given an initial model set, some method of determining similar mixture components must be used. Possible approaches are :-

1. Phonetic Characteristics
   Similar acoustic classes can be grouped using phonetic knowledge. This assumes that utterances produced in a similar articulatory manner will be represented in a similar region of the feature space. This is clearly dependent on the speech coding used. A further assumption is that the acoustic classes modelled by the mixture components have some phonetic meaning. In general this is not the case since the model representing each phone is made up of several states each with multiple mixture components. Each state represents only part of a phone, and each mixture component represents a further division of the phone. On the positive side, small numbers of classes are trivial to define by using broad phoneme groupings (e.g. front vowels, back vowels, stop consonants etc.).

2. Distance Measures
   A logical method of determining similar mixture components is to use a distance measure between the distributions. There are two problems associated with this approach: the definition of the distance measure and the decision boundary between classes. A further consideration is that each class must contain a sufficient number of components to ensure that the estimate of the associated regression matrix is robust. Two possible distance measures which have been considered in this evaluation are the divergence measure and a likelihood measure (see Appendix B). The boundary

decision problem is implementation specific and depends on the distance measure. The distance criteria can be chosen so that each regression class has a sufficient number of members.

The use of multiple mixture components per state is intended to produce a complex modelling of the feature space for the sound class. Assigning the mixture components within the same state to different regression classes will result in the mixture components being transformed in different directions, and hence a change in the modality of the state distribution may result (Figure 8.1). Applying the same transform to all mixture components of the state will move the distributions in the same directions which preserves the shape of the state distribution, and is consistent with the mixture weights. However, since the transforms are estimated from the data assigned to each mixture component, a change in the state distribution shape may be desirable to improve the modelling of the speaker. For this reason no restrictions are placed on the regression class allocation of mixture components from the same state, although for phonetically based classes this naturally applies.

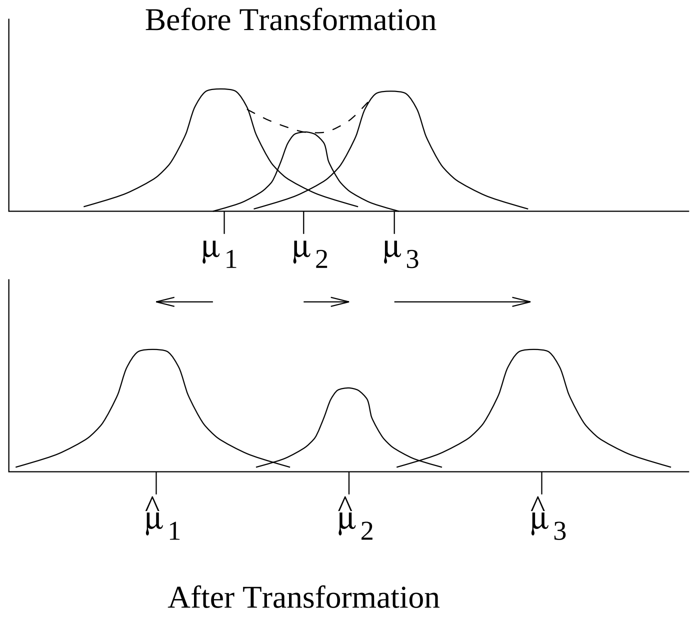

Figure 8.1: Example of change in modality when transforming mixture components within the same state using different transforms

In both types of class definition the mixture components representing silence should be considered separately. Given the same recording environment the non-speaker noise (background hum, microphone effects etc.) is common to all speakers. It should be noted that only clean speech recognition is considered here - background noise is minimal. Silence is therefore not included in any regression classes, and the mixture components within the silence models are ignored for the purpose of adaptation data collection, although they still form part of the alignment.

8.3 Class Definitions

The initial adaptation experiments used a set of class definitions determined before the adaptation process began. These class sets are termed *fixed class definitions*. An alternative method is to generate the classes after having seen all of the adaptation data. This can be achieved by using *regression class trees*.

8.3.1 Fixed Class Definitions

The two methods of generating class definitions were used for the experimental evaluation on the RM1 task. A set of phone-based class definitions based on broad phonetic classes was defined by associating all the component mixtures in a particular set of phone models with the same regression class. The division of phones into particular classes is described in Appendix C. A set of distance-based classes was defined using the likelihood measure, and hierarchical clustering based on pair-wise distances. Those components which were deemed close were pooled into a class. The classes were then repeatedly combined on a pair-wise basis (using a distance measure based on the average class membership) until the requisite number of classes was defined. The average distribution of a class with R mixture components is defined as,

$$\boldsymbol{\mu}_{\text{ave}} = \frac{1}{\sum_{i=1}^{R} N_i} \sum_{i=1}^{R} N_i \boldsymbol{\mu}_i \tag{8.1}$$

$$\boldsymbol{\Sigma}_{\text{ave}} = \frac{1}{\sum_{i=1}^{R} N_i} \sum_{i=1}^{R} N_i \left[\boldsymbol{\Sigma}_i + (\boldsymbol{\mu}_i - \boldsymbol{\mu}_{\text{ave}})(\boldsymbol{\mu}_i - \boldsymbol{\mu}_{\text{ave}})'\right] \tag{8.2}$$

where N_i represent the relative frequencies of the different distributions. For the purpose of adaptation using unknown data each mixture component is considered equally likely ($N_i = 1$ for all i). However, given *a priori* knowledge of the adaptation data, N_i can be chosen to generate task specific classes. For the class definitions used in the evaluation tests the adaptation data was assumed to be unknown.

8.3.2 Regression Class Trees

This fixed class definition approach may be replaced by a tree-based approach which allows class selection to be performed dynamically within the adaptation process. The tree structure (Figure 8.2) contains a set of nodes each of which represents a possible regression class. The regression class of each node contains all the mixture components in the regression classes in the sub-tree from the node. Hence, the root node represents the global regression class, and the children of each node represent more specific classes. For example, in the diagram node A represents the regression class made up of the mixture components in the

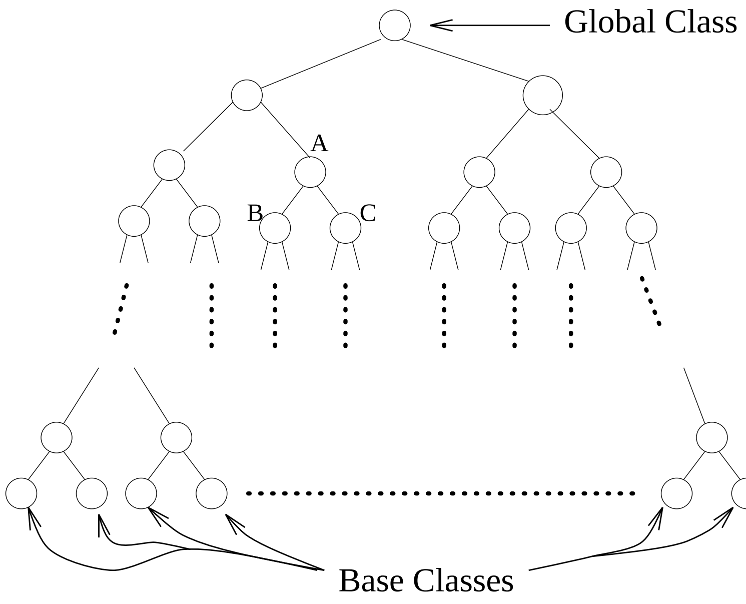

Figure 8.2: A regression class tree

classes represented by nodes B and C. The leaf nodes represent the most specific classes. For a complete tree the leaf nodes would represent individual mixture components, however, when considering large HMM sets the number of mixture components is large and with limited adaptation data it is unlikely that classes representing individual mixture components would be used. For practical purposes the leaf nodes represent *base classes* which contain a small number of similar mixture components. This reduces the complexity of the tree making tree searches less computationally expensive, and as long as the base class members are well chosen (i.e. they are similar and there is a small number of them) this should not limit the effect of the most specific transforms.

8.3.2.1 Traversing the Regression Class Tree

The regression class tree is used to determine the regression classes used during adaptation. Using the statistics from the adaptation data the amount of data allocated to each regression class can be determined. A top down search of the tree is performed to find the lowest node in each alternative branch with sufficient data to estimate a transform. This is performed in such a manner that sufficient transforms are estimated for all mixture components, and each mixture component is adapted using the most specific class that can be robustly estimated.

Starting the search at the root node:

For a node A

    if A is a leaf node

generate transform $\boldsymbol{W}$ *from data for A*
update all components in A using $\boldsymbol{W}$
if A has children B and C
if B has sufficient data
search through node B
else
generate transform $\boldsymbol{W}$ *from data for A*
update all components in B using $\boldsymbol{W}$
if C has sufficient data
search through node C
else
generate transform $\boldsymbol{W}$ *from data for A*
update all components in C using $\boldsymbol{W}$

The class tree can replace the use of fixed classes and remove the necessity of prior knowledge of the amount of adaptation data. The question of how much data is sufficient can be determined empirically, but should be similar to that determined using the fixed class approach.

8.3.2.2 Regression Class Tree Generation

The regression class tree may be generated using a clustering approach which is similar to that used to generate the fixed classes with distance measures. The base classes can be generated in an identical manner to the fixed classes, and the tree built from the base classes by pair-wise clustering. This process is repeated until the combination results in a global class.

For the RM cross-word triphone models a set of 200 base classes was defined using distance-based classes computed via the divergence measure. The average distribution of each class was computed from all the component class members and used to compute pair-wise class distances with the divergence measure. The regression class tree was then generated. Each base class had an average of 53 component members, and the resulting tree was 9 levels deep.

Trees were also generated for the WSJ model sets, but the number of base classes was increased to 750 due to the larger number of mixture components within the system. For the W_{a} model set each base class had an average of 102 mixture components and for the W_{b} model set there were 174 components in each base class. Both trees were 11 levels deep.

8.4 Initial MLLR Adaptation Experiments

Several experiments were performed using the RM database to assess the MLLR method. The most basic implementation used a global transformation. The use of different amounts of adaptation data, different alignment strategies, tied matrices and diagonal matrices were also evaluated. All the experiments were performed in a supervised manner, with the pronunciations for each word generated using the CMU RM dictionary [70]. Unless stated otherwise all alignments used the forward-backward algorithm and all regression transforms were full $n \times n+1$ matrices, generated using one iteration of MLLR. Although the method relies on an iterative optimisation criteria, it was found that only one iteration was necessary.

8.4.1 A Global MLLR Transform

The single mixture component diagonal model set $R1_{\text{diag}}$ was used for an initial assessment of the global MLLR transform. 40 utterances were used for adaptation and the resulting model set tested on all 100 test files for each speaker.

Speaker	Baseline	MLLR adaptation	% Improvement
bef0_3	10.5	8.7	11.5
cmr0_2	19.9	10.4	47.7
das1_2	6.7	3.3	50.7
dms0_4	11.2	6.0	46.4
dtb0_3	10.4	7.1	31.7
dtd0_5	11.7	9.3	20.5
ers0_7	13.6	11.7	14.0
hxs0_6	11.9	6.6	44.5
jws0_4	8.0	6.1	23.7
pgh0_1	9.1	7.0	23.1
rkm0_5	15.5	12.0	22.6
tab0_7	4.5	4.2	6.7
Average	11.1	7.7	30.6

Table 8.1: Results (% Word Error) using a global MLLR transform using a single mixture component per state ($R1_{\text{diag}}$). Adaptation using a full matrix estimated with 40 adaptation utterances.

The results (Table 8.1) show that a substantial improvement (averaging 30% reduction in error) over the speaker independent system can be achieved. The adapted models are better than the SI models for all speakers, even those which have a high initial error rate which may lead to poor model alignments.

Table 8.2 shows the changes in Viterbi likelihoods of the models on the test data. As would be expected a significant increase in the likelihood value per frame is achieved after adaptation. Thus the maximum likelihood estimation can be seen to be having the correct effect.

Speaker	Baseline Log Likelihood per frame	Adapted Log Likelihood per frame
bef0_3	-72.55	-70.52
cmr0_2	-74.06	-71.36
das1_2	-71.64	-69.40
dms0_4	-76.09	-72.77
dtb0_3	-74.32	-72.61
dtd0_5	-74.14	-72.23
ers0_7	-69.75	-68.71
hxs0_6	-72.21	-70.22
jws0_4	-76.27	-74.29
pgh0_1	-70.86	-69.17
rkm0_5	-75.99	-74.03
tab0_7	-71.33	-69.57

Table 8.2: Changes in log likelihood per frame for each speaker on the test data

8.4.2 Application to Multiple Component Mixture Distributions

The results for models with a single Gaussian distribution per state are encouraging, but these models have a high error rate. Equivalent SI model sets with increased numbers of Gaussian components mixture distributions per state were adapted using a global MLLR transform estimated on 40 adaptation utterances. These SI models give reduced error rates as the number of mixture components increases. Table 8.3 shows that the global MLLR transform gives a substantial reduction in error rate with every model set. For the 2 mixture component models the adapted models reduced the error rate for all speakers. For both the 4 mixture component and 8 mixture component model sets the adapted models increased the error rate for speaker ers0_7, the error rate for all other speakers was either better or equal to the baseline after adaptation.

The global MLLR transform is clearly effective even with model sets with large numbers of parameters. A 24% reduction in error rate using a global transform on a model set with a 6.6% initial error rate is considerable, and with more specific regression classes further reductions in error may be achieved.

8.4.3 Effect of Increasing the Number of Regression Classes

Table 8.4 shows the effect of using different methods to generate fixed regression classes for MLLR adaptation on $R2_{\mathrm{diag}}$ models using 40 utterances. The phone-based classes used broad phonetic definitions and the distance-based classes used the likelihood measure for clustering.

Num. Mixture Components	Baseline Models (SI)	Adapted Models	% Reduction in error
1	11.1	7.7	30.6
2	8.4	6.1	27.3
4	6.8	5.2	23.5
6	6.8	5.3	22.1
8	6.6	5.0	24.2

Table 8.3: Effect of MLLR on models with different numbers of component mixtures per state. Results are % word error averaged over all 12 speakers. 40 utterances were used to estimate a global adaptation transform.

As the number of regression classes is increased, the adapted models reduce the error further. Due to the smaller number of mixture components assigned to each class, the estimates of the regression matrices are more appropriate for the class members and better estimates of the adapted means are obtained. However, this is not a monotonic process, and there is an optimum number of classes to use. This optimum occurs due to the trade off between specificity of the regression classes and robustly estimating the transform. After this point the poorly estimated transforms can affect the performance quite dramatically, for example using a set of 47 regression classes based on the individual phone models (and ignoring silence) results in an average word error rate of 16.9%, which is considerably worse than the initial SI error rate.

There is little difference between the two types of class definitions in terms of the lowest error rates achievable. This is possibly due to the large number of mixture components within the model set, which results in a very broad allocation of mixture components to each class. Further tests with distance based classes based on the divergence measure also showed little difference.

8.4.4 Iterative Adaptation

The adaptation formulae are derived to maximise the auxiliary function. Although this is guaranteed to lead to an increase in the likelihood function an iterative procedure is required to obtain a local maximum Thus, once the models have been updated, a second alignment of data to models should be generated to collect new adaptation statistics and another update performed. This should be repeated until the likelihood function has converged (c.f. Baum-Welch parameter estimation).

Table 8.5 shows the effect of iterating the adaptation process for all speakers, using the 2 component mixture models. The likelihoods increase with each iteration until the point of convergence is reached, but the performance on the test data is not significantly changed. This indicates that very little change in the alignment of data is taking place and the estimates of the adapted means are not changing with each iteration. The objective function

No. Regression Classes	Class Definitions	
	Phone Based	Distance-Based
1	6.1	6.1
2	6.1	6.0
3	6.0	6.0
4	5.7	5.7
5	5.7	5.5
6	5.5	5.5
7	5.5	5.3
8	5.3	5.4
10	5.1	5.2

Table 8.4: Effect of MLLR with different numbers of regression classes defined using phonetic knowledge or distance measures between mixture component densities. Results are % word error averaged over all 12 speakers on the RM1 SD test sets. 40 utterances were used for adaptation of the $R2_{\text{diag}}$ models.

is close to the maximum after the first iteration and further iterations are unnecessary.

8.4.5 Adaptation of Specific Feature Elements

The feature vector contains MFCCs, an energy component and the first and second time derivatives. It may be the case that adapting all the elements of the feature vector is unnecessary. Some features (e.g. the dynamic coefficients) may be common among speakers and adapting them based on a small amount of adaptation data may have an adverse effect. By only adapting a portion of the mean vectors the number of parameters which need to be estimated in the regression matrices can be reduced. When implementing a selective feature vector adaptation scheme the regression matrices can be reduced to

$$\boldsymbol{W} = \begin{pmatrix} 0 & 0 & \dots & & & & & & \dots & 0 \\ 0 & 1 & 0 & \dots & & & & & \dots & 0 \\ \vdots & & \ddots & & & & & & & \vdots \\ 0 & \dots & 0 & 1 & 0 & \dots & & & & \vdots \\ w_{i,1} & 0 & \dots & 0 & w_{i,i+1} & \dots & w_{i,j+1} & 0 & \dots & 0 \\ \vdots & & & & \vdots & & \vdots & & & \vdots \\ w_{j,1} & 0 & \dots & 0 & w_{j,i+1} & \dots & w_{j,j+1} & 0 & \dots & 0 \\ \vdots & & & & & & & \ddots & & \vdots \\ 0 & 0 & \dots & & & & \dots & 0 & 1 & 0 \\ 0 & 0 & 0 & \dots & & & & \dots & 0 & 1 \end{pmatrix} \tag{8.3}$$

where features i to j are the elements within the vector to be adapted.

Iteration	% Word error	Training Data Likelihood
Baseline	8.4	-72.5
1	5.2	-68.3
2	5.2	-68.0
3	5.3	-68.0
4	5.3	-68.0

Table 8.5: Effect of iterative adaptation on SD tests. Adaptation on $R2_{\mathrm{diag}}$ models with 40 adaptation utterances and 10 regression classes. Table shows the average % word error on the test set and the average log likelihood per frame of the forward-backward alignment after each adaptation pass.

A comparison of adaptation using various combinations of the different coefficients is shown in Table 8.6. The largest gain after adaptation is certainly with the MFCC coefficients, however the other coefficients are important in reducing the error rate further. Including the energy and its derivatives gives the same amount of gain as including each of the first and second derivatives of the MFCCs.

Coefficients	Average % Word Accuracy		
Adapted	(a)	(b)	(c)
none	8.4	8.4	8.4
MFCCs	6.9	6.9	6.6
MFCCs + ΔMFCCs	6.7	6.7	6.3
MFCCs +ΔMFCCs +$\Delta\Delta$MFCCs	6.4	6.4	5.8
all	6.1	6.1	5.5

Table 8.6: The effect of adapting only selected coefficients for $R2_{\mathrm{diag}}$ models. Adaptation with (a) 5 utterances, global transform, (b) 40 utterances, global transform, (c) 40 utterances 5 classes.

Reducing the number of coefficients adapted does not appear to affect the amount of data required to estimate the transform, and for the adaptation to be fully effective all the elements of the feature vector should be adapted. No reduction in the number of regression parameters that need to be estimated can be achieved without a fall in adaptation performance.

8.5 Comparison with MAP adaptation

The MLLR adaptation method is successful at gaining an increase in recognition performance using a small number of adaptation utterances. Table 8.7 gives a comparison between

MLLR and the MAP adaptation algorithm described in section 6.5. Model sets with an identical structure to models $R2_{\text{diag}}$ and $R6_{\text{diag}}$ were used for the MAP adaptation, but were estimated in a slightly different manner [1]. The baseline results were marginally different, hence the comparison is made in terms of the reduction in error from the baseline. For the adaptation with 10 sentences a global transform was generated, and for 40 and 100 sentences 10 and 15 classes were used respectively.

Models	Number of Adaptation Sentences					
	10		40		100	
	MLLR	MAP	MLLR	MAP	MLLR	MAP
$R2_{\text{diag}}$	27.4	9.1	38.1	19.3	39.2	32.8
$R6_{\text{diag}}$	19.1	1.8	38.2	14.0	39.7	30.1

Table 8.7: Comparison of MLLR and MAP adaptation approaches using a small number of sentences for adaptation. Results are % error reductions obtained between baseline and adapted model sets.

The MLLR approach is substantially better when adapting with a small number of utterances. The MAP approach only updates those models for which there is data present in the adaptation data. Updating the parameters of all the models, even if in a general fashion, is clearly advantageous. MLLR can be seen to capture the general speaker characteristics quickly and use these to great effect in the speech modelling.

8.6 MLLR with High Accuracy Models

The previous section demonstrated the effectiveness of the basic MLLR approach. Attention is now focused on applying MLLR to an initial model set with very good performance. The model set (R_{cwt}) is based on cross word triphones and gives very good performance on the RM speaker independent test set, for example the word error rate on the Feb'91 SI test set was 2.5% [110].

On the speaker dependent test set this model set gives a word error of 4.3%, and after Baum-Welch re-training for each speaker with the 600 SD training files an average word error rate of 1.8% is achieved. The effect of re-training is particularly noticeable on those models which were initially poor, for example speaker rkm0_5 had the highest error rate in both the SI and SD systems, but performance improved from 8.3% to 2.6%.

In this section different aspects of implementing MLLR adaptation are investigated further. The relationship between the amount of adaptation data and the number of regression classes used is examined, and restricted cases of the general MLLR are tested.

8.6.1 Very Small Amounts of Adaptation Data

A global MLLR transform resulted in a $21 - 30\%$ error reduction for the word internal triphone models using 40 adaptation utterances. Figure 8.3 shows the effect of using a global MLLR transformation on the cross-word triphones using different amounts of data. The baseline SI and SD results are indicated on the figure and show that although adapta-

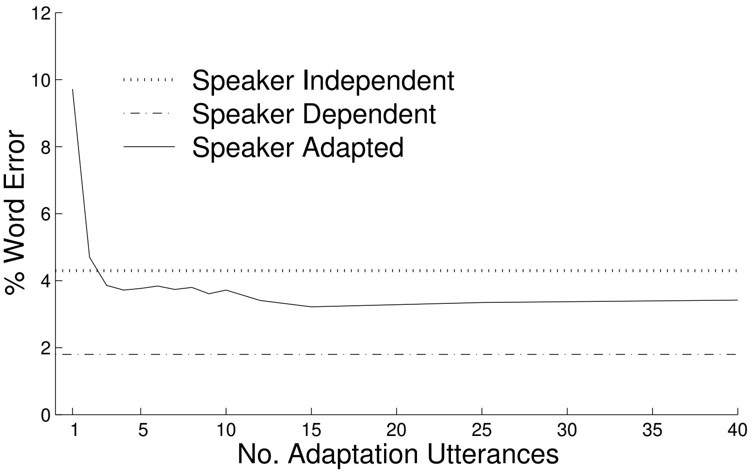

Figure 8.3: Supervised MLLR adaptation of cross-word triphone models using a global transformation

tion using 1 utterance is poor, using 3 or more utterances results in an improvement over the SI system. The poor performance of the adaptation using just 1 utterance is due to the lack of data for estimating the transform. The utterances are on average 3.4 seconds long, but this includes a significant amount of phrase initial/final silence. The number of models represented in one utterance is small and the transform is estimated with respect to a small number of mixture component examples. These mixture components dominate the estimate and lead to a poor transform for the majority of mixture components which has a severe effect on the recognition performance. Increasing the number of adaptation utterances increases the variety of mixture components observed in the adaptation data and the transform estimate is able to capture the general speaker characteristics.

An improvement in the adaptation performance is achieved by increasing the amount of adaptation data from 3 to 15 utterances, but further increasing the available data has little effect on the performance. With 15 utterances the transform has been estimated robustly and further data appears to simply reinforce the estimate.

Examining the effect on individual speakers it is encouraging to find that the adaptation performs well on the speakers which are poorly recognised with the SI models (Table 8.8). Some of the speakers with initially low error rates have robustly estimated transforms using just 5 utterances, while those speakers with higher SI error rates require more data.

Speaker		Num. Adapt. Utts			
	SI	1	5	15	SD
bef0_3	3.2	8.3	2.9	2.7	2.3
cmr0_2	7.4	19.7	6.2	4.8	1.6
das1_2	1.8	2.5	1.6	1.8	0.9
dms0_4	3.2	4.0	2.7	2.6	1.0
dtb0_3	3.3	7.4	2.8	2.4	1.2
dtd0_5	4.6	5.8	4.3	4.0	2.3
ers0_7	3.5	13.7	3.3	3.5	2.6
hxs0_6	6.5	12.8	4.3	3.2	1.5
jws0_4	4.5	7.9	4.1	3.3	1.8
pgh0_1	2.6	5.7	2.6	2.5	2.1
rkm0_5	8.3	24.8	7.6	5.5	2.6
tab0_7	2.2	3.1	2.6	2.2	1.8
Average	4.3	9.2	3.8	3.2	1.8

Table 8.8: Supervised MLLR adaptation performance for individual speakers using the cross-word triphones and a global transform. Results are average % word error rate over all speakers.

8.6.2 Varying the Number of Regression Classes

A global transformation cannot effectively exploit a lot of adaptation data. Once the matrix is well estimated, the addition of further data has little effect. By increasing the number of regression classes the data can be used to estimate several regression matrices. This allows more specific transforms to be generated for the component mixtures and increase the performance of the adaptation as indicated earlier in section 8.4.3. In that experiment only a limited number of classes were defined to compare phone-based and distance-based generation of regression classes which were found to give similar performance. It is however easier to automate the distance-based class generation and generate large numbers of classes. Figure 8.4 shows the effect of using different numbers of distance-based regression classes defined using the divergence measure. The trade off between well estimated transforms and specific matrices is clear. The error rate decreases when increasing the number of regression classes from 1 to 12, and is maintained up to around 20 classes. Increasing the number of classes further results in slightly less well estimated matrices and the adaptation error rate increases.

8.6.3 Diagonal Regression Matrices

All the MLLR experiments reported thus far have used the full $(n \times n+1)$ regression matrix. For the case of the 39 element observation vector, each matrix requires 1560 entries to be

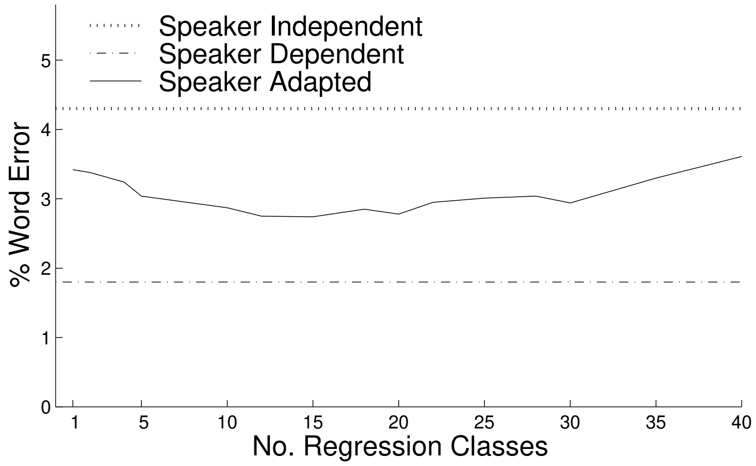

Figure 8.4: Effect of varying the number of regression classes when using cross-word triphone models. 40 utterances were used for supervised adaptation. Results are average word error rate averaged over all speakers.

estimated. This clearly limits the number of matrices that can be estimated from small amounts of data. Using the diagonal form of the MLLR matrices reduces the number of parameters to 79 per matrix, thus 20 diagonal matrices have the same number of parameters as one full matrix.

The effect of using diagonal regression matrices is shown in Figure 8.5. The diagonal matrices are not as effective at capturing the general regression estimates needed when using high degrees of tying. The assumption that single variable regression can model the complex mixture component adjustments is clearly poor. The performance of models adapted using 20 diagonal matrices is significantly worse than the SI model performance. One full matrix with the same data adapts the models effectively. Adaptation with the diagonal matrices only gives an improvement when using 30 classes or more, yet this is still marginally worse than the models adapted with one full matrix. Increasing the number of classes significantly reduces the error rate by only a small amount. Using 300 diagonal classes the performance is 3.2% word error, yet using 15 full matrices, which have an equivalent number of parameters, the adapted performance is 2.7% word error.

Using a large number of regression classes may lead to problems of data sufficiency in the matrix estimation. This is especially important if the adaptation data is not known when selecting the class definitions, especially when considering unsupervised adaptation. For this reason using a small number of full regression matrices is preferable, and the case of diagonal matrices is not considered further.

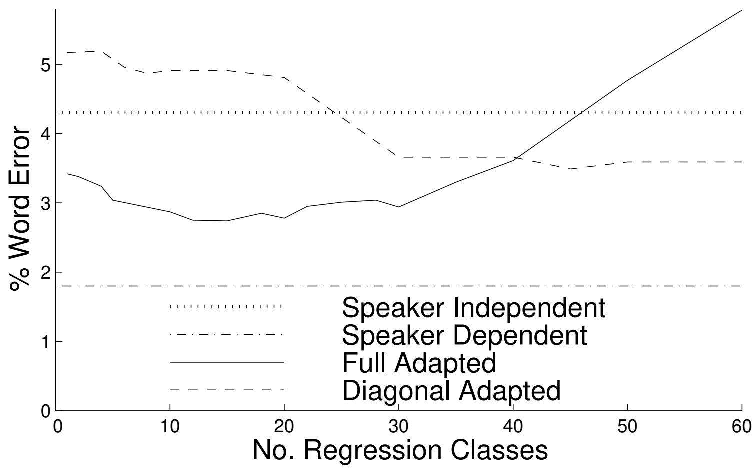

Figure 8.5: Effect using diagonal or full regression matrices for supervised adaptation of cross-word triphone models. 40 utterances were used for adaptation. Results are average word error rate.

8.6.4 Alignment Strategies

The MLLR estimation formulae have been shown to be equivalent to a least squares estimate if a) each observation frame is assigned to exactly one mixture component, and b) it is assumed that all covariances are equal. Table 8.9 shows a comparison between adaptation with the matrices generated by an MLLR estimate using a Viterbi alignment, an MLLR estimate using a forward-backward alignment and a least squares estimate. In all tests MLLR

No. Adapt. Utts	No. of Classes	Least Squares Adaptation	MLLR Adaptation	
			Viterbi	Forward-Backward
5	1	4.0	3.8	3.8
10	1	3.8	3.8	3.7
15	1	3.4	3.3	3.2
40	1	3.5	3.4	3.4
40	5	3.5	3.2	3.0
40	10	3.3	3.0	2.9
40	15	3.5	2.8	2.7
40	20	3.8	3.0	2.8

Table 8.9: Comparison of adaptation using least squares estimation and MLLR estimation using Viterbi and forward-backward alignment of the data (% word error).

adaptation is better than the least squares approach, and the difference gets significantly more noticeable as the transformations become more specific. The incorporation of variance terms into the transform estimation clearly has a positive effect. This effect will grow if the transformations are applied to fewer mixture components (i.e. using more regression classes).

The two alignment strategies for MLLR produce similar results, with the forward-backward approach being marginally better. This is not surprising for this model set since the alignments will be very similar due to the high accuracy of the SI models. However, in other cases (e.g. unsupervised adaptation), the difference may be more substantial due to the strict decisions of mixture component allocation made by the Viterbi alignment.

8.7 Supervised vs. Unsupervised Adaptation

Unsupervised adaptation using the MLLR transformation method can be easily implemented. As suggested in section 7.8 the labels for the unsupervised data can be generated using a recogniser. For this purpose the original speaker independent model set is used. The low initial error rate of the SI system will give a reasonable transcription for the adaptation data. Further, the use of the forward-backward model alignment and tied regression matrices should reduce the effect of mis-aligned data. This is demonstrated by unsupervised adaptation with a global matrix (Figure 8.6) which gives a similar performance to supervised adaptation. Adaptation with 15 utterances gives 3.2% word error in supervised mode, and 3.3% error in unsupervised mode.

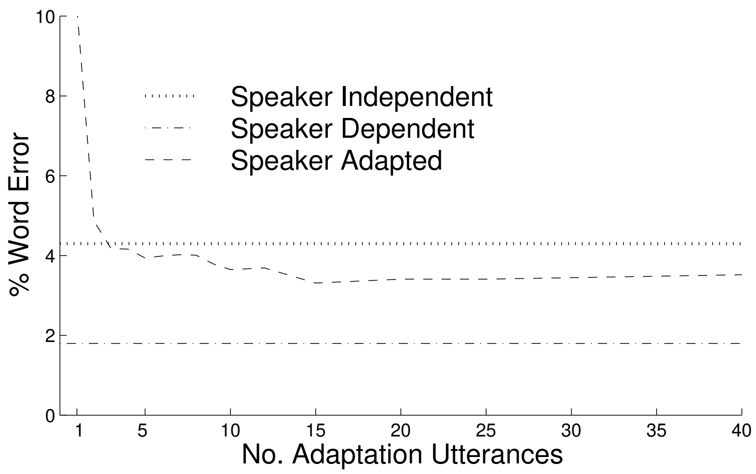

Figure 8.6: Unsupervised MLLR adaptation using a global transform. Results are average % word error rate over all speakers.

For a proper comparison of MLLR adaptation in supervised and unsupervised modes the set of regression class definitions was extended. The amount of adaptation data was varied and for each set of adaptation data the best number of class definitions (averaged over all speakers) was determined by experimentation. Table 8.10 shows that once the amount of data becomes sufficient to use multiple transforms there is a linear relationship between the amount of data and the appropriate number of classes to use, roughly 1 class for every 3 adaptation files. This is consistent with the earlier findings that 3 utterances were the minimum requirement for a global MLLR transform to give an improvement. On average 3 utterances corresponds to 10 seconds of speech per matrix, or 1000 frames of adaptation data (although a proportion of this is silence).

No. Utts	15	40	100	600
No. Classes	1	15	40	200
Supervised	3.2%	2.7%	2.3%	1.8%
Unsupervised	3.3%	2.9%	2.4%	1.9%

Table 8.10: Best performance for different amounts of adaptation data using cross-word triphone models.(% word error averaged over all speakers)

The performance of supervised and unsupervised adaptation modes are fairly similar, with supervised being marginally better (Figure 8.7). Adaptation using all the speaker dependent data gives equivalent performance to the SD trained models. This shows that the method gives a very good estimate of the adapted mean values. It also suggests that not adapting the covariances does not significantly affect performance. There is a slight caveat here in that there was not sufficient data to ensure that all the SD parameters were well estimated, so the true significance of the covariance parameters cannot be properly examined.

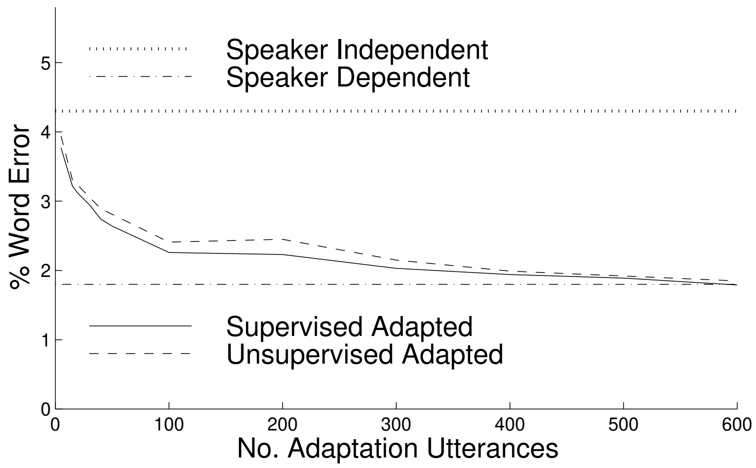

Figure 8.7: Effect of varying the number of regression classes with the amount of adaptation data using cross-word triphone models. Results are average word error rate achieved with the best set of regression classes.

8.8 Evaluation of Regression Trees

The RM database with cross-word triphones was used to compare the effect of using regression class selection via regression trees with fixed class definitions. The experimental setup described earlier was used, but the static supervised adaptation used a regression class tree instead of fixed class definitions.

The regression tree was generated using 200 base classes. The clustering criteria was the divergence between the average distribution of the mixture components in each node. Table 8.11 shows the results of the dynamic class allocation scheme using different thresholds for sufficient data to estimate the matrices.

Min. Occupation Threshold	% Word Accuracy
250	3.1
500	2.9
750	2.9
1000	2.8
1250	2.8
1500	3.1
2000	3.0
5000	3.3

Table 8.11: Adaptation results using a 200 leaf regression class tree for dynamic class allocation. Adaptation on cross-word triphone models R_{cwt} using 40 adaptation utterances.

The performance using a minimum node occupancy count of 1000 is very similar to that obtained when using the optimum number of fixed regression classes (10) determined by experimentation. An examination of the number of transforms generated at this threshold shows that an average of 9.7 classes were used for each speaker. Using a threshold of 250 uses on average 44.2 classes, while a 2000 count node threshold uses an average of only 5.3 classes.

The results confirm the earlier findings that an occupancy count of 1000 is needed for robust estimation of each transform matrix.

8.9 Incremental Adaptation

The dynamic approach to class generation allows MLLR to be extended to *incremental unsupervised* adaptation. As data is presented, the model set can be updated after every utterance (section 7.3). The regression class tree selects the appropriate class set for each adaptation phase.

Assuming that the frame/state alignment generated for each utterance is not changed after the models are updated, the accumulated adaptation data can be used in future updates. It was demonstrated earlier (section 8.4.4) that iterating the MLLR adaptation process provided little further improvement over the first adaptation pass when a good initial model set is used. This indicates that the initial alignment of the frames to states is very similar for both the initial models and the adapted models, although occupation probabilities may change. Thus the assumption is reasonable.

Using equations 7.33 and 7.34 the time dependent components can be accumulated by associating observation vectors and mixture component occupation counts with individual mixture components. The MLLR transforms can then be estimated at any time, with the estimate using all previously observed data, and not just the data presented since the last adaptation update.

8.9.1 Evaluation of Incremental Adaptation

The incremental MLLR adaptation approach has been evaluated on the S4 spoke of the 1994 WSJ NAB database [75]. The W_{a} cross word triphone models were used with a modified dynamic network decoder and a trigram language model with a 5k dictionary. The 750 base class regression tree was used for all adaptation.

The test sentences from each speaker were passed into the modified recognition system. A transcription for each sentence was generated by the recogniser which then used a forward-backward alignment of the data to collect adaptation statistics. After a specified number of sentences had been observed since the last update, the models were updated using the current adaptation statistics and the formulae in section 7.3.

Table 8.12 shows the results of the incremental adaptation on the S4 data using different intervals between adaptation updates. The error rates on both the development (Dev'94) and evaluation (Nov'94) data sets have been reduced. The effect is significantly more pronounced on the development data with a 26% reduction in error achieved. A smaller reduction in error (17%) was achieved on the evaluation data, but this started from a much lower baseline error rate.

Update	% Word Error	
Interval	S4 Dev'94	S4 Nov'94
baseline	9.08	7.76
1	6.66	6.43
5	6.69	6.58
10	6.76	6.62

Table 8.12: Unsupervised incremental adaptation on WSJ NAB 1994 spoke S4. A 750 base-class regression tree was used with a minimum occupancy level of 1000. Results are average % word error over all 4 speakers.

Each of the four speakers gives a reduction in error for the evaluation data (Table 8.13). Figure 8.8 gives an indication of what is happening during the adaptation when updating every sentence. This compares the error rates at various points in the recognition for all past sentences. After 10 sentences speakers 4tb and 4td are already being recognised substantially better by the adapted models, only 4te has a higher error rate. As the adaptation proceeds the adapted models are consistently better than the unadapted models. For speakers 4td and 4te, the difference in error rate between the adapted and unadapted models

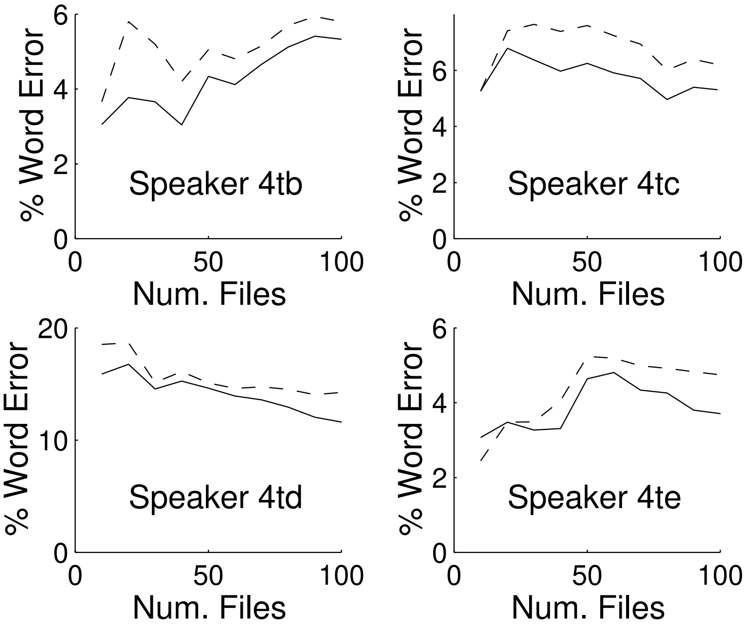

Figure 8.8: Comparison of error rates for S4 Nov'94 at various points during the recognition pass for each speaker. Dotted line indicates baseline error rate, solid line is the error rate for the recognition with incremental adaptation.

grows as more sentences are recognised, showing the incremental adaptation is having an increased effect as more data is seen. Speakers 4tb and 4tc also show an improvement, but the difference between adapted and unadapted models changes little with increased data. The error rate is the average of all past sentences so if further adaptation has only a small effect on the recognition rate over the interval being considered (every 10 sentences) the difference is not noticeable in the cumulative error rate. This can happen if either the sentences in the interval are either very easily recognised (e.g. one of the final intervals of 4tc has a 0.5% baseline error rate), or if the estimation of the transforms is dominated by earlier alignments in the incremental recognition.

Table 8.13 also shows the ratio of the time taken by the recognition run with adaptation every sentence and without adaptation. It is apparent that the adaptation process places a significant computational burden on the recogniser. The majority of this computation is in the computing of the transforms, and using the tree, the number of transforms computed increases with time (Figure 8.9). The collection of adaptation statistics is relatively efficient,

Speaker	Time ratio (adaptation/no adaptation)	% Word Error	
		Baseline	Adapted
4tb	1.37	5.57	4.98
4tc	1.16	6.32	5.37
4td	1.22	14.16	11.67
4te	1.12	4.80	3.71

Table 8.13: Error rates for individual speakers for Nov'94 S4, Unsupervised incremental adaptation with models updated every sentence.

and the updating of the means is a fixed computational overhead not dependent on the number of classes. The ratio can be reduced to closer to one if the models are updated less frequently.

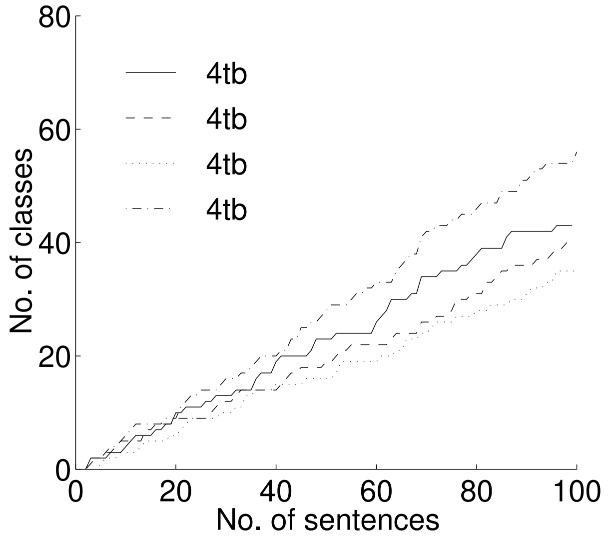

Figure 8.9: Number of classes used for incremental adaptation when adapting every sentence.

8.10 Static Adaptation of Poor Speakers

The MLLR adaptation approach relies on the statistics gathered from the state alignment of the adaptation data. So far only model sets with low error rates have been considered. Use of such model sets produces a good alignment of the data, especially in supervised adaptation. In a flexible framework a good match between speaker and model set cannot be taken for granted. To examine the robustness of MLLR when there is mismatch between the two, the case of non-native speakers is considered.

8.10.1 Non-Native Speakers

The use of speech from non-native speakers provides a very good test since there are many sources of modelling error. The main modelling problems relevant to MLLR adaptation are:-

1. Acoustic Modelling
   Clearly acoustic modelling is the most important aspect of the system. The non-native speakers have different phonetic characteristics, for example the articulation of vowels may be different. In extreme cases the phonetic alphabet used may be different (section 6.1.1).

2. Dictionary Lookup
   Non-native speakers have very different pronunciations to native speakers. This causes problems for the mapping of words to phones even though multiple pronunciations are available.

3. Language Model
   Non-native speakers are more prone to grammatical errors such as mixing tenses within a sentence. The effectiveness of the language model is also affected by mistakes which can result from poor acoustic modelling or bad pronunciations. The use of a strict language model may penalise such mistakes thus affecting the recognition.

When using a system trained for native speakers (in this case the native language is American English), the effects of these problems combine and result in a high error rate for non-native speakers. This is demonstrated in table 8.14 which compares the performance of a set of native speakers with a set of non-native speakers using the WSJ 1994 evaluation database (spokes S0 and S3 respectively). The results are generated using the W_{a} cross-word

System Setup	% Word Error	
	Native Speakers (S0)	Non-Native Speakers (S3)
for natives	5.67	20.72
for non-natives	6.47	16.67

Table 8.14: Native and non-native recognition results. The recognition set up is tuned to either native speakers or non-native speakers. Results are on the Nov'94 WSJ evaluation data, using the W_{a} system. S0 data for native speakers, S3 data for non-natives.

triphone system with a 5k vocabulary and a trigram language model. A dynamic network decoder [84] is used for the recogniser and the grammar scale factors and word insertion penalties are tuned to the data set using the Dev'94 development test set. The change in these factors is required to balance the insertion and deletion rate for the recognition errors. Without this adjustment the recognition of the non-native speakers is prone to large numbers of word insertions (6.9% before adjustment 2.1% after adjustment). The adjustment gives more weight to the language model, and encourages fewer words per sentence. This reduces the number of errors due to the inclusion of spurious function words.

8.10.2 Adaptation for Non-Native Speakers

The non-native speakers give approximately three times the word error rate of the native speakers. For adaptation this means that the alignments will be significantly poorer. However, the attraction of the MLLR method is that the probabilistic assignment of frames to mixture components and generation of general transformations compensate for poor alignments. Further, unlike the earlier cases examined which used good initial models, if the adaptation process changes the alignment significantly then the process can be iterated until the alignments (and hence likelihoods) converge.

The results on the WSJ S3 data confirm the need for iterative adaptation (Table 8.15). Although the first iteration provides a significant error reduction, a second iteration reduces

Iteration	Dev'94	Nov'94
Baseline	20.82	16.67
1	12.80	11.52
2	12.34	10.99
3	12.20	10.99

Table 8.15: Supervised MLLR adaptation on WSJ S3 non-native speakers. Adaptation uses 40 utterances and a 750 class regression tree. The recognition system is tuned to non-native speakers. Results are average % word error.

the error further. It then appears that further iterations do not affect the frame/state alignments significantly and hence have little additional effect. The use of the regression tree here is also effective since adaptation using a global transform (2 iterations) resulted in error rates of 16.10% for S3 Dev'94 and 13.81% for S3 Nov'94 [75].

8.11 Incorporating MLLR into SI Recognisers

The MLLR adaptation scheme has proved flexible with respect to specific adaptation tasks. As a final test of the true capability of this method, the incorporation of MLLR into a speaker-independent system has been considered. This is a specific case of the incremental adaptation algorithm placed in a standard recognition paradigm. However, two related problems need to be addressed: how to recognise a change in speaker, and what action to take when the speaker does change. Recognising speaker changes is not a trivial problem in speech recognition. Indeed much research is being performed in the area of speaker identification and separation [2, 76]. The problem is avoided in this case since the evaluation data is tagged with the speaker identity, however in a truly automatic system a speaker identification method would be necessary.

The adaptation data accumulated during incremental adaptation is speaker specific. When the speaker changes, this adaptation data becomes redundant. Incorporation of multiple speaker data in the collected adaptation statistics is an issue which has not been addressed in this work, but intuitively will lead to poorer estimates of the adapted model parameters for each speaker.

If the model set has been adapted to the previous speaker it will perform poorly on the new speaker. Thus the alignment of the data may be initially poor. To resolve this, each time a new speaker is identified, the model set is restored to the original parameters. This has the unfortunate effect of 'forgetting' the adaptation statistics and adapted model parameters for previous speakers so when they use the system again the adaptation restarts from scratch. By saving the speaker specific statistics this problem can be alleviated, but in true SI systems management of such stored data (i.e. identifying the speaker, loading/saving the correct data) becomes an intrinsically difficult problem.

8.11.1 The 1994 ARPA SI Evaluation

The incremental MLLR adaptation scheme was incorporated into the Cambridge University HTK HMM system [106] for the hub task (H1) of the 1994 ARPA continuous speech recognition evaluation. For these experiments a two pass dynamic network decoder was used. The first pass generated a set of word lattices using the W_{a} cross-word triphone models and a bigram language model with the 65k WSJ dictionary. The lattices were expanded to incorporate a 4-gram language model. The second pass rescored the lattices using the quinphone model set W_{b}. The first two utterances from each speaker are used to determine the gender of the speaker, and the appropriate gender dependent model set used for all further utterances of the speaker. Unsupervised incremental MLLR adaptation was used only in the second pass and the model parameters were updated every 2 utterances. Each speaker provided 15-20 sentences for the test set.

Adaptation	% Word Error	
	H1 Dev'94	H1 Nov'94
None	8.30	7.93
Incremental MLLR	7.28	7.18

Table 8.16: Results of 1994 ARPA CSR evaluation with unsupervised incremental MLLR adaptation. % Word error on the development data (H1 Dev'94) and evaluation test data (H1 Nov'94) using W_{b} models and 65k vocabulary with a 4-gram language model.

The results of this approach both with and without incremental adaptation are shown in Table 8.16. The incremental adaptation results in a 12.3% reduction in error for the development data and 9.5% reduction for the evaluation data. Considering the scale of the recognition task, the complexity of the models (the system has around 131000 mixture components), and the initial error rate, the adaptation has produced a reasonable reduction in error.

On the development data only one speaker had a higher error rate after adaptation. The evaluation data was more inconsistent with 7 out of the 20 speakers performing more poorly (Table 8.17).

Speaker	No. Files	Before	After	Speaker	No. Files	Before	After
4t0	15	12.88	8.73	4ta	15	6.86	6.33
4t1	21	6.99	5.15	4tb	15	11.21	11.84
4t2	15	5.45	5.69	4tc	15	4.78	5.26
4t3	16	2.04	2.30	4td	17	24.23	20.77
4t4	16	7.81	6.91	4te	16	4.66	5.59
4t5	15	4.36	4.12	4tg	15	10.68	7.42
4t6	15	19.42	18.90	4th	16	1.77	2.76
4t7	15	3.59	3.59	4ti	15	7.06	6.18
4t8	16	8.46	8.78	4tj	15	4.68	3.80
4t9	17	5.39	4.31	4tk	16	4.87	4.87

Table 8.17: Speaker breakdown of WSJ Nov'94 evaluation task H1 with and without adaptation.

The final system, including the adaptation, produced the lowest error rates for the H1-P0 test in the Nov'94 ARPA evaluation [89].

8.12 Discussion

This chapter has implemented and evaluated the MLLR adaptation method. Results on the Resource Management and Wall Street Journal databases have proved that MLLR adaptation scales to large continuous density HMM systems. This is very pleasing due to the broad nature of the transformations generated.

Each distribution in the system represents an acoustic class. The class is mapped into a feature space by the parameterisation of the observation vectors. The MFCC parameterisation captures general information about the speaker such as the formant frequencies and the distribution of the formants is characteristic of a speaker. Speaker independent systems have to model a wide variation of speakers, and therefore model the average distribution of the frequency of the speakers used to estimate the parameters. A specific speaker will have a different formant distribution from the average, but the differences are specific to each acoustic class. The global linear transformation is able model these various changes in the frequency distribution, at least in a general form. The changes are clearly class dependent as indicated by the improvement in adaptation using more specific transforms, and the diagonal regression transforms are unable to capture the variety of changes in distribution needed for a large number of classes. With larger systems, and especially the very large WSJ systems described, there is a large number of finely tuned mixture distributions, and the directions of transformation required are vastly more complex.

The experiments have shown that adapting time differential information is necessary for the adaptation to be fully effective. This indicates that as well as the distribution of the formant frequencies, the changes over time are indicative of the speaker.

The energy terms are also important, contributing as much to the adaptation as the inclusion of the MFCC time derivatives. This is not a surprise since the energy terms were found to be useful in aiding class discrimination in all feature selection methods (section 3.5.4). Energy is also a characteristic speaker trait, and the changes due to a speaker can obviously be captured by a linear transform.

8.13 Summary

This chapter has evaluated the basic MLLR technique and applied it to a variety of model sets using the RM speaker dependent database. A reduction in error on all RM model sets was achieved using a global transform and a small amount (15 utterances) of adaptation data. The best improvement in error rates tended to be for those models and speakers with initially high error rates. Adaptation using one component mixture models produced a larger reduction in error than models with more mixture components, although the error rate was still larger than that achieved by the higher mixture component models.

When more adaptation data is available the number of regression classes can be increased. There is a linear relationship between the optimum number of classes and the amount of adaptation data. This is due to the balance between making the regression classes specific and having sufficient data to estimate the matrix parameters. Reducing the number of parameters by using diagonal MLLR matrices is not as effective and large numbers of classes are required.

MLLR may be implemented in supervised or unsupervised modes with only a very marginal advantage in favour of supervised adaptation. This is due to the good use of data in the estimation of the transform. Generating mixture component occupancies using the forward-backward alignment allows each frame of data to contribute to several mixture components. This together with pooling the data into the regression classes reduces the effect of poorly labelled data, however, with good initial models the Viterbi alignment can give comparable results. A comparison with a least squares estimation of the transformation shows that the additional information obtained from the individual class covariances can be used to good effect in estimating the transforms. The effect increases as the transformations become more specific.

The MLLR adaptation method can thus be seen to be successful in improving the modelling of specific speakers. In fact, experiments on RM showed that MLLR adaptation of a SI system gives equivalent performance to reestimating the system using Baum-Welch.

Further experiments to assess the practical uses of MLLR were performed on the WSJ database. It was shown that MLLR could substantially reduce the error rate for non-native speakers, although the error was still much greater (about double) than that achieved for native speakers. Using iterative adaptation in such cases proved beneficial, but again the likelihood of the adaptation data converged rapidly so that more than two iterations had no further effect.

Unsupervised incremental adaptation was successfully applied to a good set of initial models. Even after a small number of sentences an improvement in error rate was gained, and maintained throughout the recognition pass.

Finally, as a demonstration of the use of adaptation in a speaker independent system, unsupervised incremental adaptation using MLLR was applied to the ARPA NAB 1994 H1-P0 task. Even though only a small number of sentences were available from each speaker, a reasonable reduction in error (average 10% over both Dev'94 and Nov'94) was achieved.

The results have shown that MLLR is both a robust and flexible method for speaker adaptation. Although effective at using very small amounts of adaptation data the method is easily extended to take advantage of larger amounts of data. MLLR has also been shown to compare favourably with other methods such as MAP adaptation and least squares transformation estimation.

Chapter 9

Conclusions

This thesis has investigated the use of transformations within HMMs as a post-training stage in improving the acoustic modelling. Particular emphasis has been placed on continuous speech with medium to large vocabularies so that the ideas could be seen to be feasible in a general HMM framework.

Two aspects of the acoustic modelling problem were identified in the introduction: improving discrimination between acoustically distinct classes; and, the problem of modelling individual speakers. These problems have been addressed through the use of transformations at a state-specific level of the HMM modelling, and it has been demonstrated that improving the modelling at the state level can have significant benefits at the word level, even for initially well trained models.

9.1 Review of Work

The two transformation approaches have been presented in two distinct sections. Chapters 3, 4 and 5 concentrated on transforms for improving the discrimination between individual acoustic classes, and chapters 6, 7 and 8 examined the use of transforms for improving the modelling of a new speaker.

9.1.1 Discriminative Feature Transforms

The existing methods for incorporating discrimination into HMMs were reviewed in chapter 3. These methods could be split into two categories, those which are based on reestimating the model parameters, and those which try to determine the elements of the feature vector containing the most discriminative information. Two types of optimisation criteria were used in examples for both categories, an information-theoretic measure (e.g. MMI, Bocchieri and Wilpon's feature selection), and effect on recognition rate (e.g. corrective training, Paliwal's feature selection). It was noted that the discriminative feature selection methods all used a global selection, except the confusion discriminant transform.

The standard global feature selection/extraction methods were evaluated in chapter 4 where it was demonstrated that simple feature selection is very limited and does not lead to an improvement in recognition rate when applied to word recognition in continuous speech. A global feature transformation using LDA gave only a very small improvement in recognition. A brief investigation of the LDA theory indicated that the reason for this was the assumption that all class distributions have similar shapes. This was shown to be a very poor assumption, and the examination switched to a state based transformation termed confusion discriminant analysis (CDA).

The theory of the CDA transform was derived and it was shown that if the number of dimensions in the chosen discriminative subspace could be reduced by a factor of two, a computational saving over the original model system could be made during recognition. Using a distribution based on the between-class covariance the number of dimensions in the discriminative feature space could be reduced to as low as 16 from the original 39, without a loss of performance. The examination of other distributions with the same transform indicated that the nature of the between-class covariance was important. Using the global covariance of the data as a whole proved not as successful, and this was put down to the lack of class-specific information. Previous work had shown that the CDA transform could be more effective if the distributions used in the computation were based more on the actual confusions that could occur. The problems of implementing such an approach in large vocabulary continuous speech were identified as: the definition of confusable frames; the difficulty in identifying confusable words; and, the location of confusable boundaries. There was also the additional problem of ensuring robust estimates for the estimated confusion distributions.

Methods for overcoming these problems were proposed in chapter 5. Two types of method were investigated, one based on confusions identified during the recognition of the training data (data-driven confusions), and the other based on using the confusability of the class distributions (PDF-based confusions). It was found that the PDF-based confusions performed poorly in comparison to using a global between-class distribution. This was due to the problems in defining confusable classes with a fixed decision boundary, and the lack of a readily available weighting to combine the confusable distributions into a single distributions. The data-driven confusions were more successful, allowing an even greater reduction in dimensionality than the global between-class covariance, without loss of performance. Using half the number of dimensions of the original feature space a 15% reduction in error can be achieved. This performance can only be achieved by using an N-Best framework to generate large numbers of confusable frames. The confusion gathering had to be extended to include large numbers of near confusions to ensure robust estimates of the transforms, and the general conclusion appeared to be that the more near confusions that were identified, the better the transforms that were generated.

The discriminative feature transform can be seen to be an effective way of improving the discrimination between acoustic classes, and results in an improvement in recognition at the word level. A trade off between recognition improvement and reduction in computation can be made using CDA. The N-Best confusion distributions are computationally expensive to compute, but result in a worthwhile error reduction. Using a global between-class distribution allows a similar reduction in computation during recognition, but the reduction in computation does not lead to improved performance.

9.1.2 Transforms for Speaker Adaptation

Previous work on speaker adaptation used either model selection, spectral transformations, or model parameter reestimation, as discussed in chapter 6. The theory presented in chapter 7, termed maximum likelihood linear regression (MLLR), is aimed at combining the parameter reestimation approach with the transformation approach by using standard HMM training techniques to estimate the transform parameters. A transformation is associated with each acoustic class (each Gaussian mixture component), and to compensate for the limited amount of adaptation data, many classes share the same transformation matrices. The transform sharing is formulated in such a way that the parameters can be reestimated within a standard MLE framework. A closed form reestimation formula is only available for HMMs using Gaussians with diagonal covariances. The derived formula can be shown to be equivalent to both a global spectral transformation method (using least squares estimation) and a standard Baum-Welch MLE training algorithm, depending on the degree of transform tying.

For the purpose of tying transforms between mixture components the notion of regression classes has been introduced. These are groups of acoustic classes which are deemed to be similar enough so that they require similar transformations. When there is insufficient data to estimate a transform for a particular component, the transform is computed from the data associated with all the components within the same regression class.

The MLLR transformation approach is evaluated in chapter 8. The initial experiments demonstrated that the MLLR approach was practical for use with HMMs systems with states made up of multiple mixture distributions. The investigation then examined the effect of increasing the number of transformations used, and found that there was a trade off between the number of regression classes and the ability to robustly estimate the classes with the available adaptation data. By arranging the regression classes in a tree the selection of appropriate regression classes can be made. An unsupervised MLLR adaptation approach was compared to the supervised method, and the supervised method was only marginally better. This can be explained by the general use of transform tying and probabilistic assignment of adaptation data. Misalignments of data have only a small effect on the estimation the transforms, and since the models were initially very good, the number of misaligned speech frames is small.

The application of the MLLR method to a state of the art system was reported in the later part of chapter 8. Adaptation was applied to a non-native speaker task, using initial native models. This showed a large reduction in error could be achieved with adaptation, but the error rate was still approximately double that of native speakers. Unsupervised incremental adaptation using MLLR was also examined using the same model set using native speakers, and a reduction in error was achieved after only a few utterances and was maintained throughout the test.

Finally, to assess the use of adaptation within speaker independent recognisers, the unsupervised incremental adaptation approach was applied to a speaker independent system with very short term speaker dependent tests (about 15 sentences per speaker). An average 10% reduction in error was achieved using an open vocabulary task with a 65000 word dictionary. This system including the MLLR adaptation had the lowest error rate in the 1994 ARPA continuous speech recognition evaluation.

9.1.3 Summary

The two transformation approaches have been shown to be successful in their respective tasks. Both methods take models with parameters estimated using standard MLE, and improve the modelling at the acoustic class level. In both cases this results in a reduction in error, and in the case of discriminative feature transforms, can also lead to a reduction in computation.

It can be concluded that transforms applied in a post-training phase are very useful in fine-tuning the acoustic modelling. The transforms are most effective when applied directly to the acoustic classes being modelled, but sufficient data must be available to estimate them.

9.2 Ideas for Future Work

Although both of the transformation methods presented indicate that the acoustic modelling in HMMs can be improved, there are still many aspects of these transformations which need investigation.

9.2.1 Discriminative Feature Transforms

The discriminative feature transformation has only been examined for HMMs with single Gaussian full covariance state distributions. These models are limited in their ability to fully capture the variability of speech that needs to be modelled, as indicated by the fact that using multiple component mixture state distributions perform substantially better. The application of CDA to model sets with better initial recognition performance, such as context dependent models or models with multiple component mixture distributions, is the next step. Due to the problems of data sufficiency using RM to generate high accuracy full covariance models this has not been examined. The application should be a straightforward extension of the method proposed, treating each individual mixture component as a separate class. The main question is whether a similar reduction in error can be achieved.

Although the N-Best data-driven approach proved successful, one of the problems encountered during the evaluation was that of which utterances to use to identify confusions. This was performed in a rather ad-hoc manner, increasing the number of utterances considered until sufficient confusions were identified. Further research is needed to determine optimal thresholds and devise methods for identifying optimal sets of confusions for each state.

One interesting aspect to consider is that of reestimating the models after they have been transformed. This is not a trivial task since the current set of state transformations must be applied to the frames assigned to each state, before the reestimation statistics can be accumulated. This will allow the dimensionality reduction to be estimated in a step fashion. Generate an initial transform, retrain the models with the transform, collect more statistics, and reestimate a new transform which to reduce the dimensionality further. Such an approach, however, would require an extremely large amount of compute power.

Finally, application of the CDA transforms to a larger task, such as WSJ, can be considered. This will allow the training of a much larger set of models, with an increased number of mixture components.

9.2.2 Speaker Adaptation Transforms

Although many aspects of the MLLR transform approach have been investigated there are still many area open for future research. Each of the ideas presented can be examined further to determine the factors which contribute to the success of the algorithm. Some of the areas which are of the greatest interest are :-

- Reestimation procedure
  The basic method has used the maximum likelihood criteria for estimating the transform parameters. It is possible to formulate the objective function for a different criteria such as maximum mutual information. A different set of estimation formula would be generated and it is possible that no closed form solution would exist. However, as with other MMI training implementations, it may be possible to use a gradient descent approach for the estimation.

- Regression class definitions
  The experiments reported here have generally used only a very limited amount of adaptation data, and as a result the number of regression classes has been small. This makes the regression classes extremely broad, and the effects of poor allocation of mixture components to classes is difficult to assess.

- Adaptation of variances
  The transformation derived is only applied to the mean vectors of the mixture components. The adaptation of variances was not initially considered due to the consideration of limited adaptation data. A transformation matrix for the covariance (since only diagonal covariances are used, it should be a scaling matrix) may be examined within the same framework.

- Full covariance models
  For MLLR to be considered a general adaptation method, the application to full

covariance models must be addressed. It is possible to assume that a full covariance matrix is diagonal for the calculation of the transform, but this is not optimal.

Given the success of the MLLR method these extensions merit further consideration.

9.3 Summary

Transformation approaches to improving the basic HMM modelling capabilities at the state level have been considered with respect to large vocabulary speech recognition. The two approaches, one aimed at improved acoustic class separation, and the other to improve the modelling of individual speakers, have been shown to be successful in reducing the word error rate on tasks with vocabularies of 1000 or more words.

Appendix A

Databases

The different databases used during the experimental evaluations are described. The speech waveforms have been parameterised in exactly the same manner for all databases.

A.1 Parameterisation of Databases

All the databases used in this thesis have been parameterised in the same way. Each utterance was sampled at 16KHz, and split into frames of 25 msecs, with a new frame starting every 10msecs. The speech wave is pre-emphasised by a factor of $k = 0.97$,

$$s'_n = s_n - ks_{n-1} \tag{A.1}$$

Each frame is windowed by a Hamming function defined by,

$$s'_n = 0.54 - 0.46\cos(\frac{2\pi n}{N-1})s_n \tag{A.2}$$

where N is the number of samples in a frame and s_n is sample n.

Each frame is coded into 12 Mel-Frequency Cepstral Coefficients (MFCCs), plus a log energy term, which then undergo cepstral liftering by a factor of $L = 22$,

$$c'_i = (1 + \frac{L}{2}\sin(\frac{\pi n}{L}))c_i \tag{A.3}$$

The 12 MFCCs and energy term are combined with first order, and second order time differentials to form a 39 element vector which is used for all experiments.

A.2 The TIMIT Database

The TIMIT database is a phoneme based continuous speech database for speaker independent tests. It contains a total of 6300 sentences, 10 sentences spoken by each of 630 speakers from 8 major dialect regions of the United States. The text material in the prompts consists of dialect sentences (sa), phonetically-compact sentences (sx) and phonetically-diverse sentences (si). For the purposes of this work the dialect sentences (sa) were ignored in both training and testing as they were considered unsuitable for speaker independent phone recognisers.

The database is split into a training set of 3696 utterances and a test set of 1344 utterances. Dialect region 1 (dr1) is used for all tests and consists of 8 utterances from each of 11 speakers (ignoring the sa sentences).

A.3 The Resource Management Database

The Resource Management (RM) database is a continuous speech database based on a naval task. The speech is from a variety of speakers of North American English recorded in a quiet environment using close talking head mounted microphones. There are two portions to the database, one speaker independent and the other speaker dependent. The sentences for both tasks use a vocabulary of 991 words, and are generated from a list of 900 different template queries. A single pronunciation was defined for each word based on the CMU dictionary and a set of 48 phones [71].

A.3.1 Speaker Independent RM Database

The training data for the speaker independent RM database consists of 3990 sentences corresponding to about 3.3 hours of data. 109 different speakers were used to collect the data, with 2830 sentences from 78 different male speakers and 1160 sentences from 31 females.

The SI database has four different test sets, corresponding to the DARPA speech recognition evaluations in February 1989, October 1989, February 1991 and September 1992. Each test set (referred to as Feb'89, Oct'89, Feb'91 and Sep'92) respectively consists of 300 sentences made up of 30 utterances from each of 10 speakers.

A.3.2 Speaker Dependent RM Database

The speaker dependent portion of the RM database consists of data for each of 12 speakers. There are 600 training sentences per speaker, corresponding to approximately 30 minutes of data. A further 100 sentences per speaker are provided for test data.

A.3.3 RM Word-Pair Language Model

For the purposes of recognition a word-pair grammar is provided. This is constructed from the finite state network which was used to construct the training and test sentences. This grammar simply constrains the words which may immediately follow a given word. All allowable successor words are assigned equal probability.

A.3.4 Function Words

The 32 most common words in the RM (Table A.1) database are defined as function words for the definition of some model sets.

A	FOR	ME	THERE
ALL	FROM	MORE	TO
AND	GET	OF	WAS
ANY	HAVE	ON	WHAT
ARE	HOW	ONE	WERE
AT	IN	THAN	WHICH
BE	IS	THAT	WILL
BY	MANY	THE	WITH

Table A.1: Words defined as function words in RM

A.4 The Wall Street Journal Database

The Wall Street Journal (WSJ) database is a very large vocabulary continuous speech task consisting of passages read from the Wall Street Journal (a daily American financial newspaper). The database was collected using a variety of desktop and close talking head microphones.

A.4.1 WSJ Training Database Description

The database contains a large amount of speech data, some with verbalised punctuation where the punctuation is explicitly included in the sentence, and the remainder non-verbalised where the punctuation symbols are not included.

The initial WSJ data is split into two sections:

- WSJ0

  The WSJ0 corpus is designed for training purposes. The speakers were asked to speak the given prompts exactly as given. Non-verbalised pronunciation was used for half the prompts and the pronunciation was verbalised for the remainder. The SI84 subset of the database was used, and those sentences which contained mispronounced words or word fragments were discarded, leaving a total of 7185 sentences. The phrase initial and phrase final silence in each sentence was stripped to leave a maximum of 200ms of initial and final silence. This resulted in around 14 hours of training data.

- WSJ1

  The WSJ1 portion of the data is larger but less varied than WSJ0. Speakers were allowed to use their normal speaking style and no fixed specification for numbers or abbreviations was enforced. The data was processed in the same manner as WSJ0, resulting in 29308 sentences representing 52 hours of speech.

The WSJ0 and WSJ1 portions of the database combine to form the training corpus for the WSJ experiments.

A.4.2 The 1994 NAB Database

For the purpose of the 1994 ARPA continuous speech recognition evaluation the corpus was extended to include texts from other North American business publications. These included Reuters North American Business Report, the Dow Jones Information Service, the New York Times, the Los Angeles Times and the Washington Post. No new acoustic training data was made available, but a development test set (Dev'94) was supplied to tune the recognition setup. Evaluation tests were performed on the evaluation test set (Nov'94) under the guidelines of the ARPA evaluation [65].

The test sets were split into different categories and the ones that are used in this work are:-

- Hub H1 - the main ARPA evaluation for SI systems.

  Dev'94 H1 - 15-18 sentences from each of 20 speakers (total 310 sentences).

  Nov'94 H1 - 15-21 sentences from each of 20 speakers (total 316 sentences).

- Spoke 0 (S0) - Native speaker SI tests.

  Dev'94 S0 - 20-23 sentences from each of 20 speakers (total 424 sentences).

  Nov'94 S0 - 19-23 sentences from each of 20 speakers (total 425 sentences).

- Spoke 3 (S3) - Non-native speaker SI tests.

  Dev'94 S3 - 20-23 sentences from each of 11 speakers (total 238 sentences).

  Nov'94 S3 - 20-23 sentences each from 10 speakers (total 213 sentences).

- Spoke 4 (S4) - Incremental adaptation tests.

  Dev'94 S4 - 100-102 sentences from each of 4 speakers (total 403 sentences)

  Nov'94 S4 - 99-101 sentences from each of 4 speakers (total 401 sentences).

Appendix B

Distance Measures

B.1 Derivation of Closed Form of Divergence Measures

The divergence is a measure of the separability of two distributions based on the difference of their means.

Given two Gaussian distributions $p_1(\boldsymbol{y})$ for class ω_1, with mean $\boldsymbol{\mu}_1$ and covariance $\boldsymbol{\Sigma}_1$, and $p_2(\boldsymbol{y})$ for class ω_2, with mean $\boldsymbol{\mu}_2$ and covariance $\boldsymbol{\Sigma}_2$, the log likelihood ratio is,

$$\Lambda(\boldsymbol{y}) = \ln\left(\frac{p_1(\boldsymbol{y})}{p_2(\boldsymbol{y})}\right) \tag{B.1}$$

and for $i = 1$ or $i = 2$,

$$E[\Lambda(\boldsymbol{y})|\omega_i] = \int_{-\infty}^{\infty} \ln\left(\frac{p_1(\boldsymbol{y})}{p_2(\boldsymbol{y})}\right) p_i(\boldsymbol{y})d\boldsymbol{y} \tag{B.2}$$

Define H as:

$$H(i,j) = \int_{-\infty}^{\infty} (\ln \frac{p_i(\boldsymbol{y})}{p_j(\boldsymbol{y})}) p_i(\boldsymbol{y})d\boldsymbol{y} = E[\ln \frac{p_i(\boldsymbol{y})}{p_j(\boldsymbol{y})}|\omega_i] \tag{B.3}$$

The directed divergence between ω_1 and ω_2 is ,

$$D_{dir} = H(\omega_1, \omega_2) \tag{B.4}$$

and the symmetric divergence is,

$$D_{sym} = H(\omega_1, \omega_2) + H(\omega_2, \omega_1) \tag{B.5}$$

The log likelihood ratio may be expressed as,

$$\begin{aligned} \ln \frac{p_1(\boldsymbol{y})}{p_2(\boldsymbol{y})} &= \frac{1}{2}(\boldsymbol{y} - \boldsymbol{\mu}_2)'\boldsymbol{\Sigma}_2^{-1}(\boldsymbol{y} - \boldsymbol{\mu}_2) \\ &\quad -\frac{1}{2}(\boldsymbol{y} - \boldsymbol{\mu}_1)'\boldsymbol{\Sigma}_1^{-1}(\boldsymbol{y} - \boldsymbol{\mu}_1) + \frac{1}{2}\ln\frac{|\boldsymbol{\Sigma}_2|}{|\boldsymbol{\Sigma}_1|} \end{aligned} \tag{B.6}$$

$$\begin{aligned} H(\omega_1,\omega_2) &= \frac{1}{2}E[(\boldsymbol{y}-\boldsymbol{\mu}_2)'\boldsymbol{\Sigma}_2^{-1}(\boldsymbol{y}-\boldsymbol{\mu}_2)|\omega_1] \\ &\quad +\frac{1}{2}E[(\boldsymbol{y}-\boldsymbol{\mu}_1)'\boldsymbol{\Sigma}_1^{-1}(\boldsymbol{y}-\boldsymbol{\mu}_1)|\omega_1] + \frac{1}{2}\ln\frac{|\boldsymbol{\Sigma}_2|}{|\boldsymbol{\Sigma}_1|} \end{aligned} \tag{B.7}$$

and since,

$$(\boldsymbol{y}-\boldsymbol{\mu}_i)'\boldsymbol{\Sigma}_i^{-1}(\boldsymbol{y}-\boldsymbol{\mu}_i) = tr\left((\boldsymbol{y}-\boldsymbol{\mu}_i)'\boldsymbol{\Sigma}_i^{-1}(\boldsymbol{y}-\boldsymbol{\mu}_i)\right) \tag{B.8}$$

$$\begin{aligned} E[(\boldsymbol{y}-\boldsymbol{\mu}_1)'\boldsymbol{\Sigma}_1^{-1}(\boldsymbol{y}-\boldsymbol{\mu}_1)|\omega_1] &= E\left[tr\left(\boldsymbol{\Sigma}_1^{-1}(\boldsymbol{y}-\boldsymbol{\mu}_1)(\boldsymbol{y}-\boldsymbol{\mu}_1)'\right)|\omega_1\right] \\ &= tr\left(\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\Sigma}_1\right) = tr(\boldsymbol{I}) \\ E[(\boldsymbol{y}-\boldsymbol{\mu}_2)'\boldsymbol{\Sigma}_2^{-1}(\boldsymbol{y}-\boldsymbol{\mu}_2)|\omega_1] &= E\left[tr\left(\boldsymbol{\Sigma}_2^{-1}(\boldsymbol{y}-\boldsymbol{\mu}_2)(\boldsymbol{y}-\boldsymbol{\mu}_2)'\right)|\omega_1\right] \\ &= tr\left(\boldsymbol{\Sigma}_2^{-1}(\boldsymbol{\Sigma}_1+(\boldsymbol{\mu}_1-\boldsymbol{\mu}_2)(\boldsymbol{\mu}_1-\boldsymbol{\mu}_2)')\right) \\ &= tr\left(\boldsymbol{\Sigma}_2^{-1}\boldsymbol{\Sigma}_1\right)+(\boldsymbol{\mu}_1-\boldsymbol{\mu}_2)\boldsymbol{\Sigma}_2^{-1}(\boldsymbol{\mu}_1-\boldsymbol{\mu}_2)') \end{aligned} \tag{B.9}$$

Thus from above, the directed divergence D_{dir} is,

$$\begin{aligned} D_{dir} &= H(\omega_1,\omega_2) \\ &= \frac{1}{2}tr(\boldsymbol{\Sigma}_2^{-1}\boldsymbol{\Sigma}_1-\boldsymbol{I})+\frac{1}{2}(\boldsymbol{\mu}_1-\boldsymbol{\mu}_2)'\boldsymbol{\Sigma}_2^{-1}(\boldsymbol{\mu}_1-\boldsymbol{\mu}_2)+\frac{1}{2}\ln\frac{|\boldsymbol{\Sigma}_2|}{|\boldsymbol{\Sigma}_1|} \end{aligned} \tag{B.10}$$

and the symmetric divergence D_{sym} is,

$$\begin{aligned} D_{sym} &= H(\omega_1,\omega_2)+H(\omega_2,\omega_1) \\ &= \frac{1}{2}tr(\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\Sigma}_2+\boldsymbol{\Sigma}_2^{-1}\boldsymbol{\Sigma}_1-2\boldsymbol{I})+\frac{1}{2}(\boldsymbol{\mu}_1-\boldsymbol{\mu}_2)'(\boldsymbol{\Sigma}_1^{-1}+\boldsymbol{\Sigma}_2^{-1})(\boldsymbol{\mu}_1-\boldsymbol{\mu}_2) \end{aligned} \tag{B.11}$$

B.2 Likelihood Measure

The likelihood measure is a measure of similarity between distributions, and is used in model decision tree building for clustering of distributions [85].

For single Gaussian distributions the log likelihood of the distribution generating an observation vector $\boldsymbol{o}$ is,

$$\begin{aligned} \log(b_s(\boldsymbol{o})) &= \log\left(\frac{1}{(2\pi)^{n/2}|\boldsymbol{\Sigma}_s|^{1/2}}e^{-\frac{1}{2}(\boldsymbol{o}-\boldsymbol{\mu}_s)'\boldsymbol{\Sigma}_s^{-1}(\boldsymbol{o}-\boldsymbol{\mu}_s)}\right) \\ &= -\frac{1}{2}\left(n\log(2\pi)+\log(|\boldsymbol{\Sigma}_s|)+(\boldsymbol{o}-\boldsymbol{\mu}_s)'\boldsymbol{\Sigma}_s^{-1}(o-\boldsymbol{\mu}_s)\right). \end{aligned} \tag{B.12}$$

So, assuming that the data is the training data and consists of T observations, and $\gamma_s(t)$ is the probability of occupying state s at time t, the likelihood measure is,

$$\mathcal{L} = \sum_{t=1}^{T}-\frac{\gamma_s(t)}{2}\left(n\log(2\pi)+\log(|\boldsymbol{\Sigma}_s|)+(\boldsymbol{o}_t-\boldsymbol{\mu}_s)'\Sigma_s^{-1}(\boldsymbol{o}_t-\boldsymbol{\mu}_s)\right) \tag{B.13}$$

The parameter reestimation formulae for the covariance using a single observation sequence is [113],

$$\boldsymbol{\Sigma}_s = \frac{\sum_{t=1}^{T} \gamma_s(t)(\boldsymbol{o}_t - \boldsymbol{\mu}_s)(\boldsymbol{o}_t - \boldsymbol{\mu}_s)'}{\sum_{t=1}^{T} \gamma_s(t)}$$

So,

$$\sum_{t=1}^{T} (\boldsymbol{o}_t - \boldsymbol{\mu}_s)' \boldsymbol{\Sigma}_s^{-1} (\boldsymbol{o}_t - \boldsymbol{\mu}_s) \gamma_s(t) = n \sum_{t=1}^{T} \gamma_s(t) \tag{B.14}$$

This gives

$$\mathcal{L} = \sum_{s \in S} -\frac{1}{2} \left(n(1 + \log(2\pi)) + \log(|\boldsymbol{\Sigma}_s|) \right) \sum_{t=1}^{T} \gamma_s(t). \tag{B.15}$$

The change in likelihood when a set of distributions D are merged to produce a single distribution m, is then given by,

$$\delta\mathcal{L} = \left(\sum_{d \in D} \frac{1}{2} \log(|\boldsymbol{\Sigma}_d|) \sum_{t=1}^{T} \gamma_d(t) \right) - \left(\frac{1}{2} \log(|\boldsymbol{\Sigma}_m|) \sum_{t=1}^{T} \gamma_m(t) \right). \tag{B.16}$$

Replacing each $(\sum_{t=1}^{T} \gamma_i(t))$s with a specific weight (w_i), and considering the case of two distributions (d_1 and d_2) merging into a single distribution s the change in likelihood measure is,

$$\delta\mathcal{L} = \frac{1}{2} w_{d_1} \log(|\boldsymbol{\Sigma}_{d_1}|) + \frac{1}{2} w_{d_2} \log(|\boldsymbol{\Sigma}_{d_2}|) - \frac{1}{2} w_m \log(|\boldsymbol{\Sigma}_m|). \tag{B.17}$$

The weights w_1,w_2 and w_3 can be determined either from the data or can be set empirically. The merged distribution is produced by averaging the two distributions using the weights.

Appendix C

Broad Phonetic Classes

For the allocation of phones to classes in the phonetic-based regression class definition the following groupings were used.

Class	Members
Very Front Vowels	ih iy ix
Near Front Vowels	ae eh
Front Dipthongs	ey ay
Back Dipthongs	aw ow oy
Near Back Vowels	aa uh ah er
Very Back Vowels	ao ax uw
Liquids	l r w y
Nasals	en m n ng
Strong Fricative	dh jh ts v z
Weak Fricative	ch f hh s sh th
Closures	dd kd pd td
Unvoiced Stops	k p t
Voiced Stops	b d dx g

Table C.1: Broad Phonetic Classes

Bibliography

[1] S.M. Ahadi and P.C. Woodland. Rapid Speaker Adaptation using Model Prediction. *Proc. ICASSP*, 1995. To appear.

[2] Y. Ariki and K. Dori. Speaker Recognition Based on Subspace Methods. *Proc. International Conference on Spoken Language Processing*, Vol. 4, pp. 1859–1862, 1994.

[3] X. Aubert, R. Haeb-Umbach, and H. Ney. Continuous Mixture Densities and Linear Discriminant Analysis for Improved Context-Dependent Acoustic Models. *Proc. ICASSP*, Vol. 2, pp. 648–651, 1993.

[4] C.M. Ayer. *A Discriminatively Derived Linear Transform Capable of Improving Speech Recognition Accuracy.* PhD thesis, Imperial College, London, 1992.

[5] C.M. Ayer, M.J. Hunt, and D.M. Brookes. A Discriminatively Derived Linear Transform for Improved Speech Recognition. *Proc. EuroSpeech*, Vol. 1, pp. 583–586, 1993.

[6] L.R. Bahl, P.F. Brown, P.V. de Souza, and R.L. Mercer. A New Algorithm for the Estimation of Hidden Markov Model Parameters. *Proc. ICASSP*, Vol. 1, pp. 493–496, 1988.

[7] L.R. Bahl, P.F. Brown, P.V. de Souza, and R.L. Mercer. Speech Recognition with Continuous Parameter Hidden Markov Models. *Proc. ICASSP*, Vol. 1, pp. 40–43, 1988.

[8] L.E. Baum. An Inequality and Associated Maximization Technique in Statistical Estimation for Probabilistic Functions of Markov Processes. *Inequalities*, 3:1–8, 1972.

[9] J.R. Bellegarda, P.V. de Souza, et al. Robust Speaker Adaptation using a Piecewise Linear Acoustic Mapping. *Proc. ICASSP*, Vol. 1, pp. 445–448, 1992.

[10] J.R. Bellegarda, P.V. de Souza, et al. The Metamorphic Algorithm: A Speaker Mapping Approach to Data Augmentation. *IEEE Trans. Speech and Audio Processing*, 2(3):413–419, July 1994.

[11] J.R. Bellegarda and D. Nahamoo. Tied Mixture Continuous Parameter Modeling for Speech Recognition. *IEEE Trans. ASSP*, 38(12):2033–2045, December 1990.

[12] A. Bhattacharyya. On a Measure of Divergence between two Statistical Populations Defined by their Probability Distributions. *Bull. Calcutta Math. Soc.*, Vol. 35(No.3):99–110, 1943.

[13] E.L. Bocchieri and G.R. Doddington. Frame Specific Statistical Features for Speaker Independent Speech Recognition. *IEEE Trans. ASSP*, ASSP-34(4):755–764, August 1986.

[14] E.L. Bocchieri and J.G. Wilpon. Discriminative Analysis for Feature Reduction in Automatic Speech Recognition. *Proc. ICASSP*, Vol. 1, pp. 501–504, 1992.

[15] E.L. Bocchieri and J.G. Wilpon. Discriminative Feature Selection for Speech Recognition. *Computer Speech and Language*, 7(3):229–246, July 1993.

[16] P. Brown. *The Acoustic-Modelling Problem in Automatic Speech Recognition.* PhD thesis, IBM T.J. Watson Research Center, 1987.

[17] P.F. Brown, C.-H. Lee, and J.C. Spohrer. Bayesian Adaptation in Speech Recognition. *Proc. ICASSP*, Vol. 2, pp. 761–764, 1983.

[18] R. Cardin, Y. Normandin, and E. Millien. Inter-word Coarticulation Modelling and MMIE Training for Improved Connected Digit Recognition. *Proc. ICASSP*, Vol. 2, pp. 243–246, 1993.

[19] R. Cardin, Y. Normandin, and R. De Mori. High Performance Connected Digit Recognition using Maximum Mutual Information Estimation. *Proc. ICASSP*, Vol. 1, pp. 533–536, 1991.

[20] J-K. Chen and F.K. Soong. An N-Best Candidates-Based Discriminative Training for Speech Recognition Applications. *IEEE Trans. Speech and Audio Processing*, 2(1):206–216, January 1991.

[21] H.C. Choi and R.W. King. A Two-Stage Spectral Transformation Approach to Fast Speaker Adaptation. *Proc. Speech Science and Technology*, Vol. 2, pp. 540–545, Perth, Australia, December 1994.

[22] W. Chou, B.-H. Juang, and C.-H. Lee. Segmental GPD Training of HMM Based Speech Recognizer. *Proc. ICASSP*, Vol. 1, pp. 473–476, 1992.

[23] W. Chou, C.-H. Lee, and B.-H. Juang. Minimum Error Rate Training based on N-Best String Models. *Proc. ICASSP*, Vol. 2, pp. 652–655, 1993.

[24] K. Choukri, G. Chollet, and Y. Grenier. Spectral Transformations through Canonical Correlation Analysis for Speaker Adaptation in ASR. *Proc. ICASSP*, Vol. 4, pp. 2659–2662, 1986.

[25] Y.-L. Chow. Maximum Mutual Information Estimation of HMM Parameters for Continuous Speech Recognition Using the N-Best Algorithm. *Proc. ICASSP*, Vol. 2, pp. 701–704, 1990.

[26] F. Class, A. Kaltenmeier, et al. Fast Speaker Adaptation for Speech Recognition Systems. *Proc. ICASSP*, Vol. 1, pp. 133–136, 1990.

[27] F. Class, A. Kaltenmeier, and P. Regel-Brietzmann. Fast Speaker Adaptation Combined with Soft Vector Quantization in an HMM Speech Recognition System. *Proc. ICASSP*, Vol. 1, pp. 461–464, 1992.

[28] S.J. Cox. Speaker Adaptation in Speech Recognition using Linear Regression Techniques. *Electronics Letters*, 28(22):2093–2094, October 1992.

[29] S.J. Cox. Speaker Adaptation using a Predictive Model. *Proc. EuroSpeech*, Vol. 3, pp. 2283–2286, 1993.

[30] S.J. Cox. Predictive Speaker Adaptation in Speech Recognition. *Computer Speech and Language*, 9(1), January 1995.

[31] S.J. Cox and J.S. Bridle. Unsupervised Speaker Adaptation by Probabilistic Spectrum Fitting. *Proc. ICASSP*, Vol. 1, pp. 294–297, 1989.

[32] S.B. Davis and P. Mermelstein. Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences. *IEEE Trans. ASSP*, 28(4):357–366, August 1980.

[33] A. Dempster, N. Laird, and D. Rubin. Maximum Likelihood from Incomplete Data via the EM algorithm. *Journal of the Royal Statistical Society*, 39:1–38, 1977. Ser. B.

[34] V. Digalakis, D. Rtischev, and L. Neumeyer. Fast Speaker Adaptation using Constrained Estimation of Gaussian Mixtures. *IEEE Trans. Speech and Audio Processing*, 1995. To appear.

[35] G.R. Doddington. Frame Specific Variance Weighting for Improved Speech Recognition. Presented at *ICASSP 1990*. Not included in proceedings.

[36] G.R. Doddington. Phonetically Sensitive Discriminants for Improved Speech Recognition. *Proc. ICASSP*, Vol. 1, pp. 556–559, 1989.

[37] Y. Ephraim, A. Dembo, and L.R. Rabiner. A Minimum Discriminative Information Modelling Approach for Hidden Markov Modelling. *Proc. ICASSP*, Vol. 1, pp. 25–27, 1987.

[38] Y. Ephraim and L.R. Rabiner. On the Relations Between Modeling Approaches for Information Sources. *Proc. ICASSP*, Vol. 1, pp. 24–27, 1988.

[39] H. Franco and A. Serralheiro. Training HMMs using a Minimum Recognition Error Approach. *Proc. ICASSP*, Vol. 1, pp. 357–360, 1991.

[40] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, New York, 2nd edition, 1990.

[41] S. Furui. A Training Procedure for Isolated Word Recognition Systems. *IEEE Trans. ASSP*, 28(2):129–136, April 1980.

[42] J-L. Gauvain and C.-H. Lee. Improved Acoustic Modelling with Bayesian Learning. *Proc. ICASSP*, Vol. 1, pp. 481–484, 1992.

[43] J-L. Gauvain and C.-H. Lee. Maximum a Posteriori Estimation for Multivariate Gaussian Mixture Observations of Markov Chains. *IEEE Trans. Speech and Audio Processing*, 2(2):291–298, 1994.

[44] Y. Grenier. Speaker Adaptation through Canonical Correlation Analysis. *Proc. ICASSP*, Vol. 3, pp. 888–891, 1980.

[45] Y. Grenier, L. Miclet, J.C. Maurin, and H. Michel. Speaker Adaptation for Phoneme Recognition. *Proc. ICASSP*, Vol. 3, pp. 1273–1275, 1981.

[46] F.S. Gurgen and H.C. Choi. On the Frame-Based and Segment-Based Non-linear Spectral Transformation for Speaker Adaptation. *Proc. Speech Science and Technology*, Vol. 2, pp. 534–539, Perth, Australia, December 1994.

[47] R. Haeb-Umbach, D. Geller, and H. Ney. Improvements in Connected Digit Recognition using Linear Discriminant Analysis and Mixture Densities. *Proc. ICASSP*, Vol. 2, pp. 239–242, 1993.

[48] R. Haeb-Umbach and H. Ney. Linear Discriminant Analysis for Improved Large Vocabulary Continuous Speech Recognition. *Proc. ICASSP*, Vol. 1, pp. 13–16, 1992.

[49] A.J. Hewett. *Training and Speaker Adaptation in Template-based Speech Recognition*. PhD thesis, Cambridge University, 1989.

[50] X.D. Huang and K.F. Lee. On Speaker-Independent, Speaker-Dependent and Speaker-Adaptive Speech Recognition. *Proc. ICASSP*, Vol. 2, pp. 877–880, 1991.

[51] M.J. Hunt. A Statistical Approach to Metrics for Word and Syllable Recognition. *Journal of Acoustical Society America*, 66:S535–536, 1979. (Abstract only).

[52] M.J. Hunt and C. Lefebvre. Speaker Dependent and Independent Speech Recognition Experiments with an Auditory Model. *Proc. ICASSP*, Vol. 1, pp. 215–218, 1988.

[53] M.J. Hunt and C. Lefebvre. A Comparison of Several Acoustic Representations for Speech Recognition with Degraded and Undegraded Speech. *Proc. ICASSP*, Vol. 1, pp. 262–265, 1989.

[54] M.J. Hunt and S.M. Richardson. Use of Linear Discriminant Analysis in a Speech Recogniser. *Official Proceedings of Voice Systems Worldwide*, Vol. 1, pp. 87–93, 1990.

[55] M.J. Hunt, S.M. Richardson, et al. An Investigation of PLP and IMELDA Acoustic Representations and of their Potential for Combination. *Proc. ICASSP*, Vol. 2, pp. 881–884, 1991.

[56] A. Imamura. Speaker Adaptive HMM-Based Speech Recognition with a Stochastic Speaker Classifier. *Proc. ICASSP*, Vol. 2, pp. 841–844, 1991.

[57] J. Jaschul. Speaker Adaptation by a Linear Transformation with Optimised Parameters. *Proc. ICASSP*, Vol. 3, pp. 1657–1670, 1982.

[58] B.-H. Juang. Maximum Likelihood Estimation for Mixture Multivariate Stochastic Observations of Markov Chains. *A.T. & T. Technical Journal*, 64(6):1235–1249, July-August 1985.

[59] M.G. Kendall and A. Stuart. *The Advanced Theory of Statistics.* Charles Griffin & Company, 1971.

[60] P. Kenny, M. Lennig, and P. Mermelstein. A Linear Predictive HMM for Vector Valued Observations with Applications to Speech Recognition. *IEEE Trans. ASSP*, 38(2):220–225, February 1990.

[61] P. Kenny, M. Lennig, and P. Mermelstein. Speaker Adaptation in a Large-Vocabulary Gaussian HMM Recogniser. *IEEE Trans. PAMI*, 12(9):917–920, September 1990.

[62] L. Knohl and A. Rinscheid. Speaker Normalization and Adaptation Based on Feature Map Selection. *Proc. EuroSpeech*, Vol. 1, pp. 367–370, 1993.

[63] T. Kosaka and S. Sagayama. Tree Structured Speaker Clustering for Fast Speaker Adaptation. *Proc. ICASSP*, Vol. 1, pp. 245–248, 1994.

[64] T. Kosaka, J. Takami, and S. Sagayama. Rapid Speaker Adaptation using Speaker Mixture Allophone Models Applied to Speaker Independent Speech Recognition. *Proc. ICASSP*, Vol. 2, pp. 570–573, 1993.

[65] F. Kubala. Design of the 1994 Benchmark Test. *Proc. ARPA Spoken Language Technology Workshop*, Barton Creek, 1995. Morgan Kaufmann.

[66] F. Kubala, R. Schwartz, and C. Barry. Speaker Adaptation from a Speaker Independent Training Corpus. *Proc. ICASSP*, Vol. 1, pp. 137–140, 1990.

[67] S. Kullback and R.A. Liebler. On Information and Sufficiency. *Ann.Math.Stat.*, 22:79–86, 1951.

[68] C.-H. Lee, E. Giachin, et al. Improved Acoustic Modelling for Speaker Independent Large Vocabulary Continuous Speech Recognition. *Proc. ICASSP*, Vol. 1, pp. 161–164, 1991.

[69] C.-H. Lee, C.-H. Lin, and B.-H. Juang. A Study on Speaker Adaptation of Continuous Density HMM Parameters. *Proc. ICASSP*, Vol. 1, pp. 145–148, 1990.

[70] K.-F. Lee. *Large Vocabulary Speaker Independent Continuous Speech Recognition: The SPHINX system.* PhD thesis, Carnegie Mellon University, 1988.

[71] K-F. Lee. *Automatic Speech Recognition: The Development of the SPHINX System.* Kluwer Academic, 1989.

[72] K.-F. Lee and S. Mahajan. Corrective and Reinforcement Learning for Speaker-Independent Continuous Speech Recognition. *Computer Speech and Language*, 4(4):231–245, 1990.

[73] C.J. Leggetter and P.C. Woodland. Speaker Adaptation of Continuous Density HMMs using Multi-variate Linear Regression. *Proc. International Conference on Spoken Language Processing*, Vol. 2, pp. 451–454, 1994.

[74] C.J. Leggetter and P.C. Woodland. Speaker Adaptation Using Linear Regression. Technical Report CUED/F-INFENG/TR.181, Cambridge University Engineering Department, June 1994.

[75] C.J. Leggetter and P.C. Woodland. Flexible Speaker Adaptation using Maximum Likelihood Linear Regression. *Proc. ARPA Spoken Language Technology Workshop*, Barton Creek, 1995.

[76] A. Lipeika and J. Lipeikiene. The use of Pseudo-Stationary Segments for Speaker Identification. *Proc. EuroSpeech*, Vol. 3, pp. 2303–2306, 1993.

[77] L.A. Liporace. Maximum Likelihood Estimation for Multivariate Observations of Markov Sources. *IEEE Trans. Information Theory*, IT-28(5):729–734, September 1982.

[78] G.J. McLachlan. *Discriminative Analysis and Statistical Pattern Recognition.* John Wiley and Sons, 1992.

[79] S. Mizuta and K. Nakajima. An Optimal Discriminative Training Method for Continuous Density HMMs. *Proc. International Conference on Spoken Language Processing*, Vol. 1, pp. 245–248, 1990.

[80] A. Nadas. A Decision Theoretic Formulation of a Training Problem in Speech Recognition and a Comparison of Training by Unconditional Versus Conditional Maximum Likelihood. *IEEE Trans. ASSP*, 31(4):814–817, August 1983.

[81] A. Nadas, D. Nahamoo, and M.A. Picheny. On a Model-Robust Training Method for Speech Recognition. *IEEE Trans. ASSP*, 36(9):1432–1435, September 1988.

[82] F. Nolan. *The Phonetic Bases of Speech Recognition.* Cambridge University Press, 1983.

[83] Y. Normandin, R. Cardin, and R. de Mori. High-Performance Connected Digit Recognition using Maximum Mutual Information Estimation. *IEEE Trans. Speech and Audio Processing*, 2(2):299–311, April 1994.

[84] J.J. Odell, V. Valtchev, P.C. Woodland, and S.J. Young. A One Pass Decoder Design for Large Vocabulary Recognition. *Proc. ARPA Human Language Technology Workshop*, Vol. 1, pp. 405–410. Morgan Kaufmann, 1994.

[85] J.J. Odell, P.C. Woodland, and S.J. Young. Tree-Based State Clustering for Large Vocabulary Speech Recognition. *Proc. International Symposium on Speech, Image Processing and Neural Networks*, Vol. 2, pp. 690–693, Hong Kong, April 1994.

[86] Y. Ono, H. Wakita, and Y. Zhao. Speaker Normalization using Constrained Spectra Shifts in Auditory Filter Domain. *Proc. EuroSpeech*, Vol. 1, pp. 355–358, 1993.

[87] K.K. Paliwal. Dimensionality Reduction of the Enhanced Feature Set for the HMM-Based Speech Recogniser. *Digital Signal Processing*, (2):157–173, 1992.

[88] K.K. Paliwal and W.A. Ainsworth. Dynamic Frequency Warping for Speaker Adaptation in Automatic Speech Recognition. *Journal of Phonetics*, (13):123–134, 1985.

[89] D.S. Pallett, J.G. Fiscus, W.M. Fisher, et al. 1994 Benchmark Tests for ARPA Spoken Language Program. *Proc. ARPA Spoken Language Technology Workshop*, Barton Creek, 1995. Morgan Kaufmann.

[90] E.S. Parris and M.J. Carey. Estimating Linear Discriminant Parameters for Continuous Density Hidden Markov Models. *Proc. International Conference on Spoken Language Processing*, Vol. 1, pp. 215–218, 1994.

[91] Louis Pols. Real Time Recognition of Spoken Words. *IEEE Trans. ASSP*, C-20(9):972–978, Sept. 1971.

[92] 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. *A.T. & T. Technical Journal*, 64(6):1211–1234, 1985.

[93] D. Rainton and S. Sagayama. Minimum Error Classification Training of HMMs - Implementation Details and Experimental Results. *Journal Acoustic Soc. Japan.*, 13(6):379–386, 1992.

[94] A.J. Robinson. An Application of Recurrent Nets to Phone Probability Estimation. *IEEE Trans. Neural Networks*, 5(2):298–305, March 1994.

[95] R. Schwartz and S. Austin. A Comparison of Several Approximate Algorithms for Finding Multiple (N-Best) Sentence Hypotheses. *Proc. ICASSP*, Vol. 2, pp. 701–704, 1991.

[96] R. Schwartz and Y.-L. Chow. The N-Best Algorithm: An Efficient and Exact Procedure for Finding the N Most Likely Sentence Hypotheses. *Proc. ICASSP*, Vol. 1, pp. 81–84, 1990.

[97] R. Schwartz, Y-L. Chow, and F. Kubala. Rapid Speaker Adaptation using a Probabilistic Spectral Mapping. *Proc. ICASSP*, Vol. 1, pp. 633–636, 1987.

[98] F.K. Soong and E.-F. Huang. A Tree-Trellis Based Fast Search for Finding the N-Best Sentence Hypotheses in Continuous Speech Recognition. *Proc. ICASSP*, Vol. 1, pp. 705–708, 1991.

[99] C. Tuerk and T. Robinson. A New Frequency Shift Function for Reducing Inter-Speaker Variance. *Proc. EuroSpeech*, Vol. 1, pp. 351–354, 1993.

[100] A.J. Viterbi. Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm. *IEEE Trans. Information Theory*, 13:260–269, 1967.

[101] H. Wakita. Normalisation of Vowels by Vocal Tract Length and its Application to Vowel Identification. *IEEE Trans. ASSP*, 25(2):183–192, April 1977.

[102] J.C. Wells. *Accents of English*. Cambridge University Press, 1982.

[103] J.G. Wilpon, C.-H. Lee, and L.R. Rabiner. Improvements in Connected Digit Recognition using Higher Order Spectral and Energy Features. *Proc. ICASSP*, Vol. 1, pp. 349–352, 1991.

[104] P.C. Woodland. Hidden Markov Models using Vector Linear Predictions and Discriminative Output Distributions. *Proc. ICASSP*, Vol. 1, pp. 509–512, 1992.

[105] P.C. Woodland and D.R. Cole. Optimising Hidden Markov Models using Discriminative Output Distributions. *Proc. ICASSP*, Vol. 1, pp. 545–548, 1991.

[106] 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. *Proc. ARPA Spoken Language Technology Workshop*, Barton Creek, 1995. Morgan Kaufmann.

[107] P.C. Woodland and S.J. Young. The HTK Tied-State Continuous Speech Recogniser. *Proc. EuroSpeech*, Vol. 3, pp. 2207–2210, 1993.

[108] S.J. Young. Competitive Training in Hidden Markov Models. *Proc. ICASSP*, Vol. 1, pp. 681–684, 1989.

[109] S.J. Young. The General Use of Tying in Phoneme-Based HMM Speech Recognisers. *Proc. ICASSP*, Vol. 1, pp. 569–572, 1992.

[110] S.J. Young, J.J. Odell, and P.C. Woodland. Tree-Based State Tying for High Accuracy Acoustic Modelling. *Proc. ARPA Human Language Technology Workshop*, Vol. 1, pp. 286–291, Princeton, March 1994.

[111] S.J. Young, N.H. Russell, and J.H.S. Thornton. Token Passing: A Conceptual Model for Connected Speech Recognition Systems. Technical Report F_INFENG/TR.38, Cambridge University Engineering Department, 1989.

[112] S.J. Young and P.C. Woodland. The Use of State Tying in Continuous Speech Recognition. *Proc. EuroSpeech*, Vol. 3, pp. 2203–2206, 1993.

[113] S.J. Young, P.C. Woodland, and W.J. Byrne. *HTK - Hidden Markov Model Toolkit, Version 1.5.* Cambridge University Engineering Department and Entropic Research Laboratories Inc.

[114] G. Yu, W. Russull, R. Schwartz, and J. Makhoul. Discriminant Analysis and Supervised Vector Quantization for Continuous Speech Recognition. *Proc. ICASSP*, Vol. 2, pp. 685–688, 1989.

[115] S.A. Zahorian, D. Qian, and A.J. Jagharghi. Acoustic-Phonetic Transformations for Improved Speaker-Independent Isolated Word Recognition. *Proc. ICASSP*, Vol. 1, pp. 561–564, 1991.

[116] Y. Zhao. An Acoustic-Phonetic Based Speaker Adaptation Technique for Improving Speaker Independent Continuous Speech Recognition. *IEEE Trans. Speech and Audio Processing*, 2(3):380–394, July 1994.