CROSS-LANGUAGE ACOUSTIC ADAPTATION FOR AUTOMATIC SPEECH RECOGNITION

by

Christoph Nieuwoudt

Submitted in partial fulfilment of the requirements for the degree

Philosophiae Doctor

in the

Faculty of Engineering

UNIVERSITY OF PRETORIA

April 2000

Summary

Speech recognition systems have been developed for the major languages of the world, yet for the majority of languages there are currently no large vocabulary continuous speech recognition (LVCSR) systems. The development of an LVCSR system for a new language is very costly, mainly because a large speech database has to be compiled to robustly capture the acoustic characteristics of the new language.

This thesis investigates techniques that enable the re-use of acoustic information from a source language, in which a large amount of data is available, in implementing a system for a new target language. The assumption is that too little data is available in the target language to train a robust speech recognition system on that data alone, and that use of acoustic information from a source language can improve the performance of a target language recognition system.

Strategies for cross-language use of acoustic information are proposed, including training on pooled source and target language data, adaptation of source language models using target language data, adapting multilingual models using target language data and transforming source language data to augment target language data for model training. These strategies are allied with Bayesian and transformation-based techniques, usually used for speaker adaptation, as well as with discriminative learning techniques, to present a framework for cross-language re-use of acoustic information. Extensions to current adaptation techniques are proposed to improve the performance of these techniques specifically for cross-language adaptation. A new technique for transformation-based adaptation of variance parameters and a cost-based extension of the minimum classification error (MCE) approach are proposed.

Experiments are performed for a large number of approaches from the proposed framework for cross-language re-use of acoustic information. Relatively large amounts of English speech data are used in conjunction with smaller amounts of Afrikaans speech data to improve the performance of an Afrikaans speech recogniser. Results indicate that a significant

reduction in word error rate (between 26% and 50%, depending on the amount of Afrikaans data available) is possible when English acoustic data is used in addition to Afrikaans speech data from the same database (i.e. both sets of data were recorded under the same conditions and the same labelling process was used). For same-database experiments, best results are achieved for approaches that train models on pooled source and target language data and then perform further adaptation of the models using Bayesian or discriminative techniques on target language data only. Experiments are also performed to evaluate the use of English data from a different database than the Afrikaans data. Peak reductions in word error rate of between 16% and 35% are delivered, depending on the amount of Afrikaans data available. Best results are achieved for an approach that performs a simple transformation of source model parameters using target language data, and then performs Bayesian adaptation of the transformed model on target language data.

Keywords: multilingual speech recognition, cross-language acoustic adaptation, Bayesian adaptation, transformation-based adaptation, minimum classification error adaptation

Opsomming

Spraakherkenningstelsels is reeds ontwikkel vir die groot tale van die wêreld, maar vir die meerderheid van tale bestaan daar tans geen groot-woordeskat kontinuespraakherkenningstelsels nie. Die ontwikkeling van 'n groot-woordeskat kontinuespraakherkenningstelsel vir 'n nuwe taal is baie duur, hoofsaaklik omdat 'n groot databasis opgestel moet word om die akoestiek van 'n nuwe taal op robuuste wyse te vervat.

Die tesis ondersoek tegnieke wat die hergebruik van akoestiese inligting van 'n brontaal, waarvoor 'n groot hoeveelheid data beskikbaar is, toe te laat in die implementering van 'n stelsel vir 'n nuwe teikentaal. Die aanname word gemaak dat te min data beskikbaar is vir die teikentaal om 'n robuuste spraakherkenningstelsel mee af te rig, en dat akoestiese inligting in 'n brontaal gebruik kan word om die herkenning van 'n teikentaalstelsel te verbeter.

Strategieë vir die gebruik van akoestiese inligting oor taalgrense heen word voorgestel en sluit in: afrigting op gepoelde brontaal- en teikentaaldata, aanpassing van brontaalmodelle met teikentaaldata, aanpassing van multitaalmodelle met teikentaaldata en transformasie van brontaaldata om teikentaaldata aan te vul vir afrigting van modelle. Hierdie strategieë word met Bayes en transformasie tegnieke, wat gewoonlik vir sprekeraanpassing gebruik word, en diskriminerende afrigtingstegnieke gebruik om 'n raamwerk vir die gebruik van akoestiese inligting oor taalgrense daar te stel. Uitbreidings van bestaande tegnieke word voorgestel om die herkenning van die tegnieke te verbeter vir kruis-taal aanpassing. 'n Nuwe tegniek vir transformasie van variansieparameters en 'n kostegebaseerde uitbreiding van die minimum klassifikasiefout tegniek word voorgestel.

Eksperimente word uitgevoer vir 'n groot aantal benaderings uit die voorgestelde raamwerk vir kruis-taal hergebruik van akoestiese inligting. Relatief groot hoeveelhede Engelse spraakdata word gebruik tesame met kleiner hoeveelhede Afrikaanse spraakdata om die werkverrigting van 'n Afrikaanse herkenningstelsel te verbeter. Die resultate dui aan dat 'n beduidende vermindering in woordfouttempo (tussen 26% en 50%, afhangende van die hoe-

veelheid Afrikaanse data wat beskikbaar is) moontlik is wanneer Engelse data tesame met Afrikaanse data van dieselfde databasis gebruik word (dit wil sê beide datastelle is onder dieselfde toestande opgeneem en dieselfde etiketteringsproses is gebruik). Vir dieselfde-databasis eksperimente word die beste resultate bereik vir benaderings wat modelle afrig op gepoelde brontaal- en teikentaaldata, en wat dan verdere afrigting van modelle volgens Bayes of diskriminasiegebaseerde tegnieke uitvoer met slegs teikentaaldata. Eksperimente word ook uitgevoer om die gebruik van Engelse spraakdata van 'n verskillende databasis as die Afrikaanse data te evalueer. Piek verminderings in fouttempo tussen 16% en 35% word gelewer, afhangende van die hoeveelheid Afrikaanse data wat beskikbaar is. Beste resultate word bereik vir 'n benadering wat 'n eenvoudige transformasie van bronmodelparameters uitvoer met gebruik van teikentaaldata, en dan Bayes aanpassing van die getransformeerde model uitvoer met teikentaaldata.

Sleutelwoorde: multitaalspraakherkenning, kruis-taal akoestiese aanpassing, Bayes aanpassing, parameter transformasie, minimum klassifikasiefout aanpassing

Contents

List of Abbreviations

ANN	Artificial neural network
CBLF	Cost-based loss function
CBMM	Cost-based misclassification measure
CDHMM	Continuous density hidden Markov model
CDR	Connected digit recognition
CMS	Cepstral mean subtraction
CRBMM	Cost and reward-based misclassification measure
DCT	Discrete cosine transform
DTW	Dynamic time warping
EM	Expectation maximisation
ESHMM	Expanded state hidden Markov model
FFT	Fast Fourier transform
GPD	Generalised probabilistic descent
HMM	Hidden Markov model
LDA	Linear discriminant analysis
LVCSR	Large vocabulary continuous speech recognition
MAP	Maximum *a posteriori*
MAPLR	Maximum *a posteriori* linear regression
MCE	Minimum classification error
MFCC	Mel-scaled cepstral coefficient
ML	Maximum likelihood
MLLR	Maximum likelihood linear regression
MLP	Multi-layer perceptron
MMI	Maximum mutual information
MSE	Minimum square error
PCA	Principal component analysis
SA	Speaker adaptive
SD	Speaker dependent
SI	Speaker independent
VTLN	Vocal tract length normalisation
WER	Word error rate

List of Symbols

$\mathbf{A}$	a state transition probability matrix
D	the feature dimension
K	the number of mixtures in a state
M	the number of classes/HMMs
N	the number of states in an HMM
$\mathbf{R}$	a Gaussian precision matrix
$\mathbf{S}$	a sample variance
T	the number of time frames in an observation sequence
$\mathbf{W}$	a transformation matrix
$\mathbf{X}$	an observation sequence
a	an HMM transition probability
c	a mixture weight
$\mathbf{q}$	a state sequence
$\mathbf{m}$	the mean vector of a Gaussian prior distribution
r	a Gaussian precision value
v	a parameter of the mixture weight Dirichlet prior distribution
$\mathbf{v}$	the target variance parameter
w	the relative variance of the mean in the prior
$\mathbf{x}$	a feature vector
$\mathbf{\Lambda}$	the parameters of a set of HMMs
$\mathbf{\Sigma}$	a Gaussian covariance matrix
$\mathbf{\Upsilon}$	the precision of the covariance Wishart prior distribution
α	a parameter of a gamma distribution
β	a parameter of a gamma distribution
γ	the state/mixture occupancy variable
	the slope of the sigmoid in the MCE loss function
ϵ	the MCE update parameter
ζ	the cost associated with a phoneme misclassification
η	a parameter of the transition probability Dirichlet prior distribution
	the scaling factor in the MCE misclassification measure
θ	the offset in the MCE loss function
κ	the reward associated with a phoneme classification decision
$\boldsymbol{\lambda}$	the parameters of an HMM
$\boldsymbol{\mu}$	a Gaussian mean vector
ξ	the transition count variable
$\boldsymbol{\sigma}$	a Gaussian variance vector (for diagonal covariance)
$\boldsymbol{v}$	the extended mean vector
τ	the variance of the mean in the prior (univariate)
ϖ	a learning rate parameter

Chapter 1

Introduction

Speech is a natural and efficient way for humans to communicate. Automatic speech recognition for computers introduces a fundamental shift in the human-machine interface, leading to myriads of new applications and greatly improving the usability of many existing applications. Human to human communication will be significantly enhanced in the future through the co-development of speech recognition in multiple languages combined with automatic translation between languages.

For most languages of the world, however, no speech recognition systems exist. The standard methods used for constructing speech recognition systems have been shown to work well for a large number of languages. The methods, however, necessitate a large amount of training data to deliver acceptable performance. The collection of speech data and the subsequent labelling of that data is currently an expensive and labour-intensive process. For the majority of languages of the world the lack of sufficient databases is the barrier that limits the development of speech recognition technology.

An interesting field of research in speech recognition technology is the development of systems that can explicitly recognise speech in multiple languages. Multilingual systems generally necessitate the use of acoustic information from multiple languages in a single

modelling environment to avoid the cost of keeping full models sets for each language, to facilitate improved systems integration and to enable recognition of words from multiple languages in the same utterance. Although the aim of multilingual systems as such is not the re-use of acoustic information across language boundaries, it does present an approach for the use of acoustic information from existing databases in the development of a speech recognition system for a new language for which a limited amount of data is available.

Another field of research that is of interest is the field of speaker adaptation. Speaker adaptation techniques change model parameters to improve recognition performance for a new target speaker based on a limited amount of data from the new speaker. We, however, propose using speaker adaptation techniques for the purpose of changing models that were trained on a source language or languages, to improve performance for a target language. We propose that in this way, a system for a new target language can be developed that uses acoustic information from existing source language databases, but the performance of which is optimised for the target language using whatever target language data is available. This thesis details how multilingual data should be used in conjunction with adaptation techniques to deliver optimal performance for a new target language in the absence of sufficient amounts of target language data for the development of a stand alone speech recognition system.

1.1 Speech recognition fundamentals

Current leading edge speech recognition systems are based firmly on statistical pattern recognition principles [1, 2]. As such, these systems are data driven, i.e. are the result of training models of suitable complexity on large amounts of data. To increase recognition performance, models of increasing complexity are used - which in turn need increased amounts of training data to train accurately. It is expected that this trend will continue for some time. Large projects have been launched to collect and label spoken data for many of the major language groupings of the world such as American English [3, 4, 5], Japanese

[6], French [7, 8], German [9], as well as for the collection of multilingual databases[10, 11]. The existence of a comprehensive collection of data is a prerequisite for the development of a successful speech recognition system using current algorithms and technology.

Large-vocabulary continuous-speech recognition (LVCSR) systems comprise of two main parts, firstly *acoustic modelling* of the basic sounds or phones of speech, and secondly *language modelling* which captures the statistics of sequences of words. Pre-processing or feature extraction forms an important part of acoustic modelling by transforming the raw speech signal into an acoustic vector sequence $\mathbf{X} = \mathbf{x}_1, \mathbf{x}_2, ..\mathbf{x}_T$ that is more amenable to modelling. Bayes' rule expresses the probability $P(W|\mathbf{X})$ of a word sequence $W = w_1, w_2, ..w_n$, given an observed acoustic vector sequence $\mathbf{X}$ by

$$P(W|\mathbf{X}) = \frac{P(\mathbf{X}|W)P(W)}{P(\mathbf{X})}, \tag{1.1}$$

where $P(W)$ represents the *a priori* probability of observing the sequence of words W, independent of the observed signal, and $P(\mathbf{X}|W)$ represents the conditional probability of observing the vector sequence $\mathbf{X}$, given the word sequence W. The probability $P(W)$ is language specific and is determined by a language model, often in the form of a conceptually simple bigram or trigram that may contain millions of discrete probabilities. The estimation of the parameters of these language models are facilitated by the large amounts of text available electronically. The training of the acoustic model $P(\mathbf{X}|W)$, however, depends on the availability of speech databases that are phonetically labelled, or at least transcribed. To achieve good performance, the training data should also fit the expected use of the system as closely as possible and should include data from many speakers for speaker independent (SI) recognition.

A basic premise of acoustic modelling is that a speech signal consists of short periods exhibiting stationary behaviour. This leads to the simplification of subdividing a speech signal into frames of relatively short length (typically 10-25 ms) with respect to the periods over which speech is stationary. A further assumption is that words can be modelled as the concatenation of a sequence of basic sounds or phones. Hidden Markov models (HMMs)

[1] are used to model phones via a sequence of states with quasi-stationary behaviour in each state. If every phone is represented by an HMM, words and sentences can be modelled by a concatenation of HMMs. The distribution of acoustic parameters in each state of an HMM is typically modelled with parametric continuous-density output distributions such as multivariate Gaussian mixtures.

Context has a large influence on the way that phones are produced and thus also influences the acoustic properties of the phones. To obtain good phonetic discrimination it is desirable to train different HMMs for phones in different contexts if enough speech data is available. A solution is to use triphones, where there is a distinct model for each phone combined with a unique pair of left and right neighbours. In practice this leads to an extremely large number of model parameters, which is reduced by making use of state tieing. The idea is to tie together states that are acoustically indistinguishable or at least very similar. Data associated with each individual state are pooled, giving more robust estimates for the parameters of a tied state. Even if enough data is not available to train accurate context dependent models, context independent models should at least allow for relatively complex distributions (such as Gaussian mixture distributions) to be able to model different contexts of each phoneme model. In any event, the amount of parameters to be estimated is large and predicates the use of large databases for training.

For many languages, including 10 of the 11 official languages of South Africa (all but English), very little or no speech data is available for training acoustic models. Even for South African English, speech data from various local population groupings would be needed to develop a system with robust performance on the South African accents. As far as language modelling is concerned, the situation is somewhat better since moderate amounts of electronic text are available in at least some of the languages. The speech databases that are available for South African languages include a database for South African English and Afrikaans [12] and a Xhosa database [13]. It is foreseeable that some data may be collected for more of the local languages, but it is unlikely that the quantity of data collected will approach the amount of data routinely used in developing LVCSR systems for the major languages of the world.

It is apparent that techniques must be found to enable the training of robust acoustic models in the absence of large quantities of speech data. One possible way would be to attempt to use expert knowledge from phoneticians in the target language. Approaches based largely on phonetic knowledge have been superseded by the statistical modelling approach and do not present a feasible solution. The only other option available then is to find methods that can use available data from other languages. It is hoped that these methods can improve the performance of acoustic models for a target language in which little training data is available. The field of *multilingual* speech recognition, which investigates the sharing of phoneme sets across languages, is a starting point for this research. It should be noted that the main focus of multilingual research is the creation of systems that can explicitly recognise speech in multiple languages, which may be in conflict with our goal of optimising performance for any specific language.

Another set of techniques that may be of use in developing robust acoustic models for a new language are techniques used for speaker adaptation. Generally, speaker adaptation techniques have been applied and optimised to improve speaker dependent modelling performance given a certain limited amount of speaker specific data. Although they are called *speaker* adaptation techniques, they also adapt models to recording and transmission channel conditions they are exposed to. These techniques do not have to be directed at a specific speaker and can be performed in multispeaker or even speaker independent mode and in our case are investigated for their use in cross-language adaptation of acoustic models.

1.2 Multilingual speech recognition systems

Multilingual speech recognition has generally been researched for the development of systems that can handle speech input in multiple languages [14, 15], or for the bootstrapping of seed models for forced alignment of speech data in a new language [16, 17, 18]. Some studies have researched the explicit sharing of acoustic information between languages by constructing multilingual phone sets [19, 20], but have in most cases reported some recogni-

tion performance degradation in return for simplified modelling of acoustic parameters and easily integrated multilingual recognition. Few studies have considered using cross-language acoustic information for the explicit goal of improving the performance of a speech recogniser in a new target language. One study [20] pooled cross-language and target language data to improve recognition for a target language application. Another two studies [21, 22] performed mean-only Bayesian adaptation of source language models using target language data and showed improvements in recognition rate under certain conditions.

The re-use of acoustic information across language boundaries for improving recognition in a new target language is only partially addressed by current research. Especially the application of adaptation algorithms for this purpose needs further investigation. We next discuss the main categories of speaker adaptation algorithms that are relevant for this thesis before we continue with the discussion on their use for cross-language adaptation.

1.3 Speaker adaptation techniques

The field of speaker adaptation is usually of interest when considering the adaptation of acoustic model parameters to new speakers or new conditions. Speaker adaptation techniques generally attempt to adapt acoustic parameters from the speaker independent (SI) scenario to improve performance on the data from a specific speaker. Research in speaker adaptation, to a large degree, focuses on achieving good adaptation performance using as little data as possible from a new speaker, enabling faster enrolment for dictation systems and also enabling the use of speaker adaptation techniques for a wider range of applications. Our interest in speaker adaptation algorithms lies with their application for the adaptation of acoustic models from a source language using speech from a limited number of speakers in a target language. In this way we aim to train target language models that retain some of their original acoustic properties, rendering them more robust and leading to improved recognition performance in the target language.

Bayesian methods were amongst the first methods used for speaker adaptation [23, 24]. Bayesian methods are especially applicable if a sufficient amount of adaptation data is available and suitable prior distributions can be estimated for system parameters. Bayesian methods assume a prior distribution $P_o(\boldsymbol{\lambda})$ for the model parameters, usually determined from training with a large set of SI data and use observations from a new speaker to determine the *a posteriori* distribution of the model parameters. Using Bayes' theorem we may write the posterior distribution $P(\boldsymbol{\lambda}|\mathbf{X})$ as

$$P(\boldsymbol{\lambda}|\mathbf{X}) = \frac{P(\mathbf{X}|\boldsymbol{\lambda})P_o(\boldsymbol{\lambda})}{P(\mathbf{X})}. \tag{1.2}$$

The prior distribution, $P_o(\boldsymbol{\lambda})$, effectively biases the parameter distribution with the statistics for the speaker independent (SI) scenario. Bayesian estimation in known to work well for the SI to speaker dependent (SD) mapping since the SI case is a generalisation of the SD case. This is not true for a cross-language mapping, i.e. observations from a new language are not expected to be distributed according to a subset of the distribution of a source language, and may thus limit the performance achievable with Bayesian adaptation. An advantage of using Bayesian estimation, though, is that it has the property that the parameters converge to the target dependent parameters if enough adaptation data is available. Since we expect at least reasonably large amounts of data to be available for adaptation to the target language, the asymptotic performance property of a Bayesian estimator is desirable.

In speech recognition literature the method most commonly used for Bayesian adaptation is that of maximum *a posteriori* (MAP) parameter estimation. MAP estimation [25] chooses the mode of the posterior parameter distribution (the mode of $P(\boldsymbol{\lambda}|\mathbf{X})$) to represent the estimate of the parameter and is thus related to *maximum likelihood* (ML) estimation, which chooses the mode of the likelihood function (the mode of $P(\mathbf{X}|\boldsymbol{\lambda})$). Bayesian estimation can also be based on the use of a loss function to ensure that in some sense the minimum risk is associated with the estimate. The use of loss function-based Bayes estimators is investigated in this thesis in addition to MAP estimation.

A second class of methods for speaker adaptation is based on the transformation of the acoustic model parameters. Speaker adaptation via transformation does not attempt to directly estimate the new SD parameters, but rather estimates a transformation of the acoustic parameters from the SI models to the SD models. As such, it is suitable even when the SI models do not represent a prior for the SD models. A transformation may have few parameters - far fewer than the models being transformed, allowing the method to work reasonably well even when very little data is available. Transformation-based approaches were originally used to perform spectral transformation for template adaptation, accounting for microphone and channel effects and also changing the spectrum to better match the spectral characteristics of a new speaker [26, p. 286]. More recently, linear transformations of model parameters such as implemented by the *maximum likelihood linear regression* (MLLR) technique [27], rather than feature space transformations, have been commonly used. By grouping phones into classes for transformation, multiple transformations can be estimated, increasing the ability of the approach to perform complex adaptation tasks. Transformation-based adaptation generally performs well when significant bias exists between source and destination parameters, but its performance for cross-language adaptation of acoustic parameters, which may entail managing a complex set of uncorrelated differences between source and target acoustics, has yet to be fully investigated.

A third class of methods, based on discriminative training, has only recently been applied to the problem of speaker adaptation. A particularly promising implementation of discriminative training, called the minimum classification error (MCE) [28] approach, has been shown for some applications to improve performance beyond that obtained with the traditional MAP approach [29]. The MCE approach differs from Bayesian and ML approaches in that it attempts to directly minimise the number of misclassification errors, rather than maximising the *a posteriori* model likelihood or the data likelihood. Because MCE is really an error-function optimisation approach, it has considerable flexibility, leading us to consider its use for the complex task of cross-language adaptation. Unfortunately, MCE also suffers from problems such as being prone to converge to local minima. Using MCE for cross-language adaptation has the advantages over Bayesian and transformation-based methods

that MCE does not make the assumption that the parameters of a suitable prior distribution can be found, nor does it assume that a linear transformation of parameter space is applicable. MCE is not suited for the removal of consistent bias (such as transformation-based methods are well suited for), but can effect very complex 'tuning' of parameters. Similar to Bayesian adaptation, only observed parameters are adapted, predicating the availability of reasonably large amounts of adaptation data for good performance.

In the next section we proceed to discuss how we applied the methods from the research fields covered so far, namely multilingual speech recognition and speaker adaptation, to our principal problem of cross-language data re-use.

1.4 Cross-language re-use of acoustic information

Previous research has shown the feasibility of using acoustic information from languages for which large databases exist in aiding the development of speech recognition systems for new languages. Source language models have been shown to be useful for bootstrapping models in a new language. Most studies indicate, however, that the sharing of acoustic models in a multilingual context leads to some performance degradation in return for simplified modelling [19, 20, 30], because model accuracy is reduced when the same model is used across multiple languages. Research [20] shows that sharing phones can work well if the languages have large acoustic similarities e.g. Italian and Spanish. For some new target languages it may be possible to find an acoustically similar language in which large amounts of speech data are available, but there may still be some sounds that occur only in the target language and have no near counterpart in the source language. Even for phones that occur in both source and target languages, there are bound to be some systematic differences in pronunciation, as well as differences with respect to the context of the phones. Simple sharing of acoustic information across language boundaries thus does not present an optimal solution to the problem in general.

An alternative to the pooling of data is to train models on large amounts of source language data and to then adapt the acoustic parameters from the source language to the target language in the same way that acoustic parameters are adapted from speaker independent (SI) models to the speaker dependent (SD) models. Some issues have to be addressed, however, since cross-language adaptation entails an SI to SI mapping and not an SI to SD mapping. Our aim with cross-language adaptation is to retain the SI properties of the acoustic models from the source language while changing them to better reflect the overall distributions of feature parameters in the target language. Typically, more data is available for cross-language adaptation than is usually used for speaker dependent adaptation, since a more complex mapping is expected to be necessary and also since the process can be performed off-line. This implies that techniques which can efficiently use larger amounts of data, rather than techniques specialised for rapid adaptation, are expected to deliver better performance.

A problem with the application of speaker adaptation techniques for cross-language adaptation is the assumption that the same set of phonemes can be used, which is not true in general for different databases in different languages. To address this problem it is necessary to make use of phonetic experts, or to use distance metrics to determine which phone classes should be used in conjunction with which other phone classes in the different databases and languages. For models in the target language that have poor correspondence in the source language, cross-language use of data does not guarantee acceptable performance and adaptation has to be able to significantly alter the model parameters to achieve good performance.

Two main classes of methods have been employed for cross-language adaptation in previous research namely Bayesian methods such as MAP and transformation-based techniques such as MLLR. We also apply these two methods, albeit more comprehensively than previous studies, to cross-language adaptation. Our implementation of cross-language Bayesian estimation uses the first language models to provide *a priori* information on the expected distribution of the second language model parameters and we adapt Gaussian mean and variance parameters as well as the mixture weight and transition probability parameters.

We show that adapting all model parameters in a Bayesian framework leads to superior performance when compared to the mean-only adaptation approach reported in previous research [22, 21]. A further improvement is obtained when prior distributions for MAP adaptation are estimated from models trained on pooled source and target language data, especially when source and target language data present a close match. This strategy of first training on pooled multilingual data and then performing further target language specific adaptation is well suited to the Bayesian adaptation paradigm, because use of multilingual data is more likely to produce suitable prior distributions than use of source data alone.

We find that use of the MLLR technique does not achieve the same level of performance as that achieved with the MAP technique. We propose a method to also transform the Gaussian variance parameters, greatly improving performance, but still not achieving as good performance as with MAP. We find, however, that use of MLLR adaptation is especially applicable when the source and target databases differ in terms of the recording conditions so that there are spectral differences between the source and target signals. In such cases, MLLR is used to produce transformed models, which in turn are used to seed prior distributions for MAP adaptation, achieving the best performance on the independent test set for cross-database adaptation.

We find that models trained on pooled multilingual data present good initial models for discriminative adaptation, especially if the pooled data sets were closely matched. Adaptation of the multilingual models is done with MCE, using target language data only, thereby improving the performance of the models for the target language. Discriminative training at this stage allows the models to retain the multilingual acoustic distributions as far as possible, changing them only with respect to errors incurred on the target language data. We propose an extension to the MCE framework that modifies the MCE misclassification measure to associate a cost with each phoneme misclassification error. The cost is based on the probability of a phoneme error leading to a word error and is shown to deliver improved performance for cross-language MCE adaptation. We also apply discriminative adaptation to models that have already been optimised for target language performance using other approaches and find that the MCE approach can improve on the performance achieved with

the MAP and combined MLLR and MAP approaches, but that improved performance is not guaranteed.

Finally, we propose a data augmentation strategy for cross-language use of acoustic information. Data augmentation comprises computing a relatively simple transformation of source language data to better match target language data and then a pooling of the transformed data and the target language data. This pooled data set is termed the *augmented* data set and is used for model training. Trained models can be subjected to further target language dependent training to improve performance, especially since the data transformation may not accurately capture all the differences between the acoustics of the respective languages.

Overall, we find that cross-language use of acoustic information can lead to greatly improved target language performance. We present a framework of strategies and techniques for cross-language adaptation and perform experiments to evaluate the performance of a variety of the approaches.

1.5 Organisation of thesis

The outline of the thesis is now given. Chapter 2 gives background on the hidden Markov modelling approach followed. A relatively comprehensive coverage of basic material is given for reference purposes from later chapters as well as to at least partially document the algorithms used in the development of the Hidden Markov Toolkit for Speech Recognition (HMTSR) software by Darryl Purnell and the author during their Ph.D. studies. Also as part of the background, Chapter 2 contains a discussion of previous research in the field of multilingual speech recognition, which sets the stage for the research undertaken in this thesis. Chapter 3 treats techniques commonly used for speaker adaptation as their use for cross-language adaptation is extensively evaluated in a later chapter. Improvements to current techniques are also proposed. Discriminative learning methods, especially the minimum classification error (MCE) technique, are discussed in depth in Chapter 4, as well

as a cost-based extension of the MCE framework. Chapters 3 and 4 form the basis for the presentation in Chapter 5, which describes strategies for cross-language use of acoustic information, as well as factors to be considered in applying both speaker adaptation methods and discriminative training methods to cross-language adaptation of acoustic models.

Cross-language English-Afrikaans experiments on the SUN Speech database [12] are presented and discussed in Chapter 6, showing large improvements in recognition performance through cross-language re-use of acoustic information. Chapter 7 extends the results from Chapter 6 to include cross-language use of acoustic information between the TIMIT [31] and SUN Speech databases. Finally, the conclusion is presented in Chapter 8.

1.6 Contributions of thesis

The original contributions presented in this thesis include the following points.

- We present a framework of strategies and techniques for cross-language use of acoustic information [32]. New strategies are proposed, such as first training models on pooled source and target language data, followed by adaptation, as well as a cross-language data augmentation approach which transforms source language data for a better match with target language data. We use the strategies to apply specific techniques from the field of speaker adaptation and discriminative learning and show that our newly proposed approach of pooling-adaptation leads to superior performance for same-database experiments than source model adaptation.
- Our complete implementation, evaluation and comparison of Bayesian and transformation-based adaptation techniques (initial results published in [33, 34]), as applied to the task of cross-language adaptation, provides insights as to the conditions under which the algorithms perform well. Previous studies only adapted Gaussian mean parameters and we show that adaptation of Gaussian variance and other HMM parameters lead to large performance improvements. As part of describing Bayesian estimation,

we note that although MAP is almost exclusively and sometimes interchangeably used for Bayesian adaptation in the speech recognition community, that alternative implementations defined by loss functions exist. Along with the well documented MAP estimators, we also provide Bayes estimators for a mean square error loss function and experimentally compare the approaches.

- We propose a technique that, in conjunction with MLLR transformation of the Gaussian means, performs a full matrix transformation of the (diagonal) Gaussian variance values based on the least squares estimation. The transformation is computed in log-space, maintaining constraints on the variance values and minimising relative error in the transformation. Our experimental results show that the proposed approach outperforms standard MLLR, linear variance transformation and variance re-estimation in all experiments.

- We implement the recently proposed MAPLR approach, which combines Bayesian and transformation-based adaptation of Gaussian mean parameters. We use the same concept to extend our log-space variance transformation technique to incorporate a MAP-like term, improving generalisation and especially improving sensitivity of the transformation with respect to the number of regression classes by reducing overfitting.

- We derive and implement a comprehensive version of the MCE algorithm, adapting all HMM parameters, including duration modelling parameters in a unified framework utilising both "true" class derivatives and the "false" class derivatives. We extend the MCE framework to include a cost associated with each misclassification into the misclassification measure. We derive equations to base the estimation of the cost of phoneme misclassification on word error rate. We show that the cost-based extension to MCE achieves superior performance for multilingual model adaptation than the standard approach in our experiments.

- We evaluate cross-language performance for a continuous speech recognition task and show that cross-language use of acoustic information from the same or a different

database can greatly improve the performance of a continuous speech recogniser beyond that achievable using only target language data. We present a feasible and useful approach for the development of a speech recognition system in a new language when only a limited amount of data is available in the new language.

Chapter 2

Background

This chapter discusses in detail the background to the research that was performed for this thesis. Firstly an overview is given of the theory of hidden Markov modelling that was applied. The basic notation is given and algorithms and equations that are required for understanding the proposed ideas are discussed for reference from later chapters. This section also serves to at least partially document the algorithms used in the development of the Hidden Markov Toolkit for Speech Recognition (HMTSR) C++ software by Darryl Purnell and the author during their Ph.D. studies. The software is included on the compact disc inserted inside the back cover of this thesis.

Previous research in multilingual speech recognition is discussed next, focusing on how these systems re-use acoustic information between multiple languages and specifically how cross-language use of acoustic information has benefited the development of speech recognisers in a new target language. Limitations of previous research is pointed out, in particular the partial implementation of speaker adaptation techniques, leading us to consider improved use of these techniques in following chapters.

2.1 Hidden Markov modelling framework

The main components used in the training and testing of the speech recognition system that was developed are:

- **feature extraction** in which speech signals are converted into sequences of mel-scaled cepstral coefficient vectors along with their time derivatives,
- **training** of HMMs, which includes fixed segment initialisation, Viterbi alignment re-estimation and the expectation maximisation or Baum-Welch procedure,
- **continuous speech recognition** in which the feature vectors are matched using dynamic programming to a set of trained HMMs constrained by a finite state grammar.

We now proceed to discuss each of these items in detail, including various choices with respect to parameters of especially the feature extraction process. The selection of parameters of the general system is included in this background section on HMMs because the values of these parameters are fairly standard and are not considered to significantly impact the experiments discussed in a later section.

2.1.1 Feature extraction

The speech signal is blocked into frames of 16 ms spaced 10 ms apart - delivering 6 ms of overlap between successive frames. This choice has been empirically determined to deliver good performance. At a 16 kHz sampling rate, which is used in all experiments, each 16 ms frame consists of 256 samples. Hamming windowing and a fast Fourier transform (FFT) is performed on each frame and the result is multiplied by its complex conjugate to deliver a real valued power spectrum. The next step is applying a mel-spaced filter bank to produce 24 mel-spaced filtered coefficients. The logarithm of each coefficient is taken and a discrete cosine transform (DCT) is performed on the coefficients to deliver what is referred to as

mel-scaled cepstral coefficients or MFCCs. In all experiments 13 coefficients are used as we have previously found this to deliver good performance for a connected digit recognition task [35] performed with the HMM software and is also commonly reported in literature. Temporal information about the speech signal is incorporated by estimating first and second time derivatives for each of the 13 coefficients. A second order linear regression is applied to each set of five consecutive coefficients in order to obtain a smoothed estimate of the first and second time derivatives. The observation vector $\mathbf{x}$ thus consists of the 13 mel-scaled cepstral coefficients plus first and second order time derivatives, totalling 39 elements at each frame time. A detailed discussion of issues concerning the feature extraction process can be found in [26].

2.1.2 Continuous density hidden Markov models

A continuous density hidden Markov model (CDHMM), hereafter referred to simply as an HMM, signified by $\boldsymbol{\lambda}$, is described by two sets of parameters:

- a state transition matrix $\mathbf{A} = \{a_{ij}\}$ reflecting the probabilities of making transitions from each state i to each other state j and
- a continuous state observation density function $b_j(\mathbf{x})$ reflecting the likelihood of observing observation vector $\mathbf{x}$ in state j.

To simplify the equations we consider the initial state probabilities to be given by a_{0i} for each state i, without any loss of generality.

The models are first order HMMs since each transition probability to a next state depends only on the current state, and not on which states were previously traversed. Left-to-right HMMs without skipping transitions are commonly used for speech recognition and is the connectionist strategy we use for the purpose of this thesis. The state transition probabilities a_{ij} satisfy the constraints $\sum_{j=1}^{N} a_{ij} = 1$ and for left-to-right models without

skipping transitions the additional constraints are that $a_{ij} \neq 0$ only for $j = i$ or $j = i + 1$. The assumption with left-to-right modelling is that observation sequences corresponding to the same HMM traverse the same discrete sequence of statistical properties. This agrees with our phonetic understanding of speech as exhibiting piecewise continuous behaviour to a large degree. This unfortunately does not explicitly allow for the modelling of too much variation in the way that the same word may be pronounced other than for time warping of the speech signal, but at least leads to very efficient implementation.

The Markov models are termed "hidden" due to the fact that the states are not observed directly in the observation sequence, but rather indirectly through modelling of observation distributions in each state. Gaussian mixtures are used to model the observation probability density functions. The p.d.f. of observation $\mathbf{x}_t$ at time t in state j takes the form

$$\begin{aligned} b_j(\mathbf{x}_t) &= \sum_{k=1}^{K} c_{jk} \mathcal{N}[\mathbf{x}_t, \boldsymbol{\mu}_{jk}, \boldsymbol{\Sigma}_{jk}] \\ &= \sum_{k=1}^{K} c_{jk} (2\pi)^{-D/2} |\boldsymbol{\Sigma}_{jk}|^{-1/2} e^{-(1/2)(\boldsymbol{\mu}_{jk} - \mathbf{x}_t)^T \boldsymbol{\Sigma}_{jk}^{-1} (\boldsymbol{\mu}_{jk} - \mathbf{x}_t)} \end{aligned} \tag{2.1}$$

where K is the number of mixture components, D is the number of feature vector elements, c_{jk} is the weight associated with the kth mixture in the jth state, $\mathcal{N}$ is the multivariate normal density, $\boldsymbol{\mu}_{jk}$ is the mean vector of the kth mixture in the jth state and $\boldsymbol{\Sigma}_{jk}$ is the covariance matrix of the kth mixture in the jth state. To greatly reduce the number of parameters and since the elements of $\mathbf{x}_t$ are largely uncorrelated, we make the assumption that $\boldsymbol{\Sigma}_{jk}$ is diagonal. The observation density function becomes

$$b_j(\mathbf{x}_t) = \sum_{k=1}^{K} c_{jk} \prod_{l=1}^{D} (2\pi)^{-1/2} \sigma_{jkl}^{-1} \, e^{-(x_{tl} - \mu_{jkl})^2 / 2\sigma_{jkl}^2} \tag{2.2}$$

where x_{tl} is the lth element of the observation vector at time t, μ_{jkl} is the lth element of the mean vector in mixture k of state j and σ_{jkl}^2 is the lth variance value on the diagonal of $\boldsymbol{\Sigma}_{jk}$.

2.1.3 Duration modelling

It is commonly accepted that the duration modelling aspect of the HMM approach to speech recognition is a major weakness. Conventional HMMs implicitly model state duration by a geometric distribution, i.e.

$$\rho_j(\tau) = a_{jj}^{\tau-1}(1 - a_{jj}), \tag{2.3}$$

where a_{jj} is the auto-transition probability in state j and τ is the duration in number of frames. The geometric distribution is not able to model individual state duration probabilities well since it can only represent an exponentially decreasing probability density function. Explicit duration densities for states may be specified and in such a case the models are called semi-Markov models [26]. State duration density may be modelled with estimated discrete duration probabilities $d_j(\tau), \tau = 1, 2, .., \tau_{max}^j$ for each duration up to a maximum duration τ_{max}^j. This approach has the disadvantage that a large number of parameters have to be estimated. Modelling duration with parametric functions greatly reduces the number of parameters. A popular function for modelling state duration probability is the Gamma distribution

$$\rho(\tau) = \frac{\beta^\alpha}{\Gamma(\alpha)}\tau^{\alpha-1}e^{-\beta\tau}, \tag{2.4}$$

which has only the parameters α and β that have to be estimated for each state duration model. Initial algorithms for duration modelling were very computationally expensive [26], and a post-processing approach [36] was often used. The post-processing method uses duration metrics to re-score a number of the best paths obtained from a search process. This approach fails where the best re-scored path is not amongst the obtained best paths, and is thus not re-scored. Another approach that was more recently investigated was the use of the so-called expanded-state HMM (ESHMM) [37, 38] that provided moderate performance improvement, but at the cost of between 2-times and 4-times speed degradation.

An efficient approach towards incorporating duration modelling into the search process has

been proposed by Du Preez [39] and a similar approach was later independently proposed by Burshtein [40]. Both approaches add a duration metric at each time frame to automatically obtain a state duration probability weighted path in a computationally efficient manner, causing only marginal speed degradation. The method proposed by Burshtein was used in the experiments where applicable. Implementation of the method is discussed in the next section along with the algorithms used for the training of HMMs.

2.1.4 Hidden Markov model training

Current large vocabulary continuous speech recognition (LVCSR) systems make use of phone models to efficiently capture the necessary acoustic information for modelling large vocabulary speech through use of pronunciation dictionaries. Separate HMMs are used to model each phone and if a sufficient amount of data is available for training, head-body-tail or explicit trigram-type phone models are used. An HMM is also used to model silence at the beginning and end of utterances and between words. A clustering method, often based on data likelihood e.g. the Bayesian information criterion method, is used to decide on which trigrams to group together to constitute the set of phones. In this study only monophones, which do not take context into account, are used since the experiments mostly do not use enough data to warrant the training of more complex models and also because of the greater computational cost associated with the adaptation of large numbers of context dependent model parameters.

The parameters that have to be estimated in training an N state HMM are:

- $N+1$ independent transition probabilities (since $a_{j,j+1} = 1 - a_{jj}$ for the left-to-right model and other off-diagonal values are zero),
- NK mixture weights,
- NKD mean and covariances values and
- $2N$ duration parameters if duration modelling with the Gamma distribution is used.

Methods used for training HMMs are usually based on the *maximum likelihood* principle. The maximum likelihood estimate $\boldsymbol{\lambda}_{\mathrm{ML}}$ of the parameters of an HMM given an observation sequence $\mathbf{X} = (\mathbf{x}_1, .., \mathbf{x}_T)$ is given by the mode (maximum) of the likelihood function

$$\boldsymbol{\lambda}_{\mathrm{ML}} = \max_{\boldsymbol{\lambda}} f(\mathbf{X}|\boldsymbol{\lambda}). \tag{2.5}$$

The ML estimate is usually found by setting the derivate of either the likelihood function or the log-likelihood function with respect to the parameters of the model to zero and solving for the ML estimate of the parameters.

The likelihood of a sequence of observations given an HMM denoted by $\boldsymbol{\lambda}$ has the form

$$f(\mathbf{X}|\boldsymbol{\lambda}) \propto \sum_{\mathbf{q}} \prod_{t=1}^{T} \left[a_{q_{t-1}q_t} \sum_{k=1}^{K} c_{q_t k} \mathcal{N}(\mathbf{x}_t, \boldsymbol{\mu}_{q_t k}, \boldsymbol{\Sigma}_{q_t k}) \right], \tag{2.6}$$

where summation takes place over all possible observation sequences $\mathbf{q}$. ML estimation of the parameters of an HMM is not trivial because a *sufficient statistic* of fixed dimension does not exist for observations of an HMM. The likelihood function is not expressible in terms of a fixed number of parameters and thus cannot be maximised easily. The lack of a sufficient statistic of fixed dimension is due to the hidden process of an HMM, namely the fact that state and mixture occupancy is not observable. This lack of observability causes HMM estimation to be termed an *incomplete data* estimation problem. The solution is to use an iterative procedure such as expectation maximisation [41] procedure. The EM procedure estimates state and mixture occupancy sufficient statistics in a first part. With the availability of the calculated state and mixture occupancy statistics together with the observation sequence, the problem becomes a *complete data* estimation problem. This enables the computation of the ML parameter estimate in the second part for the complete data problem. The EM procedure consisting of the calculation of occupancy statistics followed by ML parameter estimation is repeated until convergence or a fixed number of iterations have occurred.

In our system we use iterative estimation in each of three different training stages, namely

fixed segmentation initialisation in the first stage, then segmental training (encompassing Viterbi-alignment and ML estimation) and finally training with the standard EM or Baum-Welch algorithm that computes statistics with the forward-backward algorithm and uses them for ML estimation. Our three stage training process progresses from simple, computationally inexpensive initialisation to the more complex and slower EM training. The three training stages are discussed next.

Initialisation

The parameters are estimated by examining the distribution of features in training data. The state transition matrix A is initialised according to left-to-right constraints. To bootstrap the parameters, each observation feature vector sequence corresponding to a single HMM is subdivided into as many segments of equal length as there are states in the HMM. The mean and variance of the first Gaussian mixture component in each state is initialised to the sample mean and (diagonal) covariance of the corresponding speech feature segments. After initialisation the training process commences.

Segmental training

In segmental training the Viterbi [42] algorithm is used to compute the single most likely state alignment of each observation sequence. Given the estimated alignment, the *complete data* modelling problem is solved using maximum likelihood estimates of the mixture weights, means and covariances of the Gaussian mixture models at each state and of the transition probabilities. Alignment and parameter estimation is repeated iteratively. We give details of the Viterbi dynamic programming algorithm since it is used in the implementation of the adaptation algorithms and is also used for both the training and testing of HMMs. We present the Viterbi algorithm mostly following the syntax from [26].

For an HMM, the Viterbi algorithm finds the most likely state sequence $\bar{\mathbf{q}} = (\bar{q}_1, \bar{q}_2, .., \bar{q}_T)$

for a given observation sequence $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, .., \mathbf{x}_T)$ as well as the likelihood associated with this sequence

$$\begin{aligned} P(\mathbf{X}|\boldsymbol{\lambda}) &= \max_{\mathbf{q}} P(\mathbf{X}, \mathbf{q}|\boldsymbol{\lambda}) \\ &= P(\mathbf{X}, \bar{\mathbf{q}}|\boldsymbol{\lambda}) \\ &= \prod_{t=1}^{T} [a_{\bar{q}_{t-1}\bar{q}_t} b_{\bar{q}_t}(\mathbf{x}_t)]. \end{aligned} \tag{2.7}$$

As part of the definition of the Viterbi algorithm we define

$$\Phi_j(t) = \max_{q_1, q_2, .., q_{t-1}} P[q_1 q_2, .., q_{t-1}, q_t = j, \mathbf{x}_1 \mathbf{x}_2, .., \mathbf{x}_t|\boldsymbol{\lambda}], \tag{2.8}$$

the highest probability along a single path, at time t, that accounts for the first t observations and ends in state j. By induction, the Viterbi recursion is defined as

$$\Phi_j(t+1) = \max_{0 \le i \le N} [\Phi_i(t) a_{ij}]\, b_j(\mathbf{x}_{t+1}), \quad 1 \le j \le N. \tag{2.9}$$

The probability in the final state at the final time frame, $\Phi_N(T)$, indicates the score for the match between model and observation sequence. When the Viterbi algorithm is used to align speech with model states, such as is used in training, it is necessary to keep track of the path followed via

$$\psi_j(t+1) = \arg\max_{0 \le i \le N} [\Phi_i(t) a_{ij}]. \tag{2.10}$$

This path can be backtracked from $\psi_N(T)$ to deliver the highest scoring path $\bar{\mathbf{q}}$.

Note that in Equation 2.9 we have included transitions from state 0 to the current state to incorporate the initial state probabilities. To initialise the Viterbi search for left to right operation we define $\Phi_0(0) = 1$, $\Phi_j(0) = 0, j \neq 0$ and $\Phi_0(t) = 0, t > 0$. When we discuss the implementation of successive Viterbi searches in a later section, the value associated with state zero ($\Phi_0(t)$) for $t > 0$ might not have the value 0, but may represent the final value of

a previous level in the search, e.g. $\Phi_N^{(r)}(t)$ for highest scoring model r in the previous level.

The training of parameters takes place after statistics from an entire batch of training utterances are collected. The result of each application of the Viterbi algorithm is a state-aligned set of observation features. After statistics have been collected for the batch of training samples, new mean, variance and transition probability values are computed for the Gaussian mixture models. Training using Viterbi-alignment is also called *segmental* training since the observation sequence is segmented, with each segment being used to update the parameters of a particular state. For the update, we first need to define the posterior state probability variable

$$\gamma_j(t) = P(q_t = j|\mathbf{X}, \boldsymbol{\lambda}), \tag{2.11}$$

which expresses the probability of being in state j at time t, given the observation sequence $\mathbf{X}$ and the model $\boldsymbol{\lambda}$. When *segmental* training is used, $\gamma_j(t)$ is simply equal to 1 when $\bar{q}_t = j$ and zero otherwise, i.e. $\gamma_j(t) = \delta(\bar{q}_t - j)$ where δ denotes the Kronecker delta function. Since we use Gaussian mixture distributions, we proceed to define the posterior mixture observation probability variable

$$\gamma_{jk}(t) = \gamma_j(t)\frac{c_{jk}\mathcal{N}[\mathbf{x}_t, \boldsymbol{\mu}_{jk}, \boldsymbol{\Sigma}_{jk}]}{b_j(\mathbf{x}_t)} \tag{2.12}$$

which expresses the joint probability of being in state j at time t and observing mixture k, given the observation sequence $\mathbf{X}$ and the model $\boldsymbol{\lambda}$. Finally, we define $\xi_{ij}(t)$, the probability of being in state i at time t and in state j at time $t+1$ by

$$\xi_{ij}(t) = P(q_t = i, q_{t+1} = j|\mathbf{X}, \boldsymbol{\lambda}), \tag{2.13}$$

which simply becomes $\xi_{ij}(t) = \delta(\bar{q}_t - i)\delta(\bar{q}_{t+1} - j)$ for segmental training.

Now we can define the update equations for the coefficients of the mixture density in

iteration n in terms of the sufficient statistics:

$$a_{ij}(n) = \frac{\sum_{t=1}^{T-1} \xi_{ij}(t)}{\sum_{t=1}^{T-1} \gamma_i(t)} \tag{2.14}$$

$$c_{jk}(n) = \frac{\sum_{t=1}^{T} \gamma_{jk}(t)}{\sum_{t=1}^{T} \gamma_j(t)} \tag{2.15}$$

$$\boldsymbol{\mu}_{jk}(n) = \frac{\sum_{t=1}^{T} \gamma_{jk}(t)\mathbf{x}_t}{\sum_{t=1}^{T} \gamma_{jk}(t)} \tag{2.16}$$

$$\boldsymbol{\Sigma}_{jk}(n) = \frac{\sum_{t=1}^{T} \gamma_{jk}(t)(\boldsymbol{\mu}_{jk}(n) - \mathbf{x}_t)(\boldsymbol{\mu}_{jk}(n) - \mathbf{x}_t)^T}{\sum_{t=1}^{T} \gamma_{jk}(t)}. \tag{2.17}$$

Our implementation of the segmental training process is now elaborated in more detail. As a result of the initialisation process described at the start of this section, only the first mixture component has non-zero values after the first iteration. If after an iteration, there are mixtures with zero mixture weights and sufficient data is available to warrant splitting, the component with the largest mixture weight is split to produce two components with means only slightly offset from each other in the direction of maximum variance. This process of alignment, re-estimation and mixture splitting is repeated iteratively until either convergence occurs, or a predetermined number of iterations have been completed.

Expectation maximisation

We use the Baum-Welch method, which is an implementation of the expectation maximisation or EM method for HMMs, to perform final training of the HMMs. Each HMM is trained using the set of speech segments accorded to it in the labelling process. The Baum-Welch method iteratively updates the means, covariances, mixture weights and state

transition probabilities at each state in much the same way as done with *segmental* training, but uses the forward-backward algorithm instead of the Viterbi algorithm to obtain statistics from the training utterances. Once the sufficient statistics have been computed, Equations 2.14-2.17 are used for the update. The implementation of the forward-backward algorithm is not discussed here as we do not use it for implementing the adaptation approaches (i.e. we only use it for initial model training). We use a *segmental* implementation for the adaptation algorithms since it is faster and is the method most commonly used in research on adaptation methods. Detail regarding use of the forward-backward algorithm and parameter update using the EM method can be found in [26].

Duration model training

Lastly, training of only the duration parameters is done through the Viterbi state alignment of HMMs to the utterances they represent. For each alignment, the sum of the first and second moments of the number of frames corresponding to each state in each HMM is collected. The empirical expectation values of the mean($\hat{E}\{\tau\}$) and variance($\hat{E}^2\{\tau\}$) of each duration can be calculated and used to obtain the Gamma distribution parameters ($\hat{\alpha}$ and $\hat{\beta}$) through

$$\hat{\alpha} = \frac{\hat{E}^2\{\tau\}}{\widehat{VAR}\{\tau\}}, \quad \hat{\beta} = \frac{\hat{E}\{\tau\}}{\widehat{VAR}\{\tau\}}. \tag{2.18}$$

This concludes the training process. In the next section we discuss the implementation of the recogniser.

2.1.5 Pattern matching

We first discuss the incorporation of duration modelling into the Viterbi algorithm. Duration modelling is implemented according to the synchronous frame by frame method suggested by Burshtein [40]. The method modifies the Viterbi recursion (Equation 2.9) by

incorporating a duration penalty C_{ij}^t of making a transition from state i to state j at time $t+1$ within the term that is maximised by the recursion. When written in log format for implementation efficiency, the Viterbi recursion becomes

$$\Phi_j(t+1) = \max_{0 \le i \le N} [\Phi_i(t) + \log(a_{ij}) + \log(C_{i,j}^t)] + \log b_j(\mathbf{x}_{t+1}), \quad 1 \le j \le N. \tag{2.19}$$

To compute the duration penalty, the method keeps track of the number of successive self-transitions in each state. The duration $D_i(t)$ of a state i at time t is equal to one plus the number of successive self-transitions in that state. Let M_i denote the duration at which the Gamma distribution $p(\tau)$ at state i reaches a maximum value. The duration penalty $C_{i,j}^t$ is then given by

$$C_{i,j}^t = \begin{cases} 0 & i = j,\ D_i(t) < M_i \\ \log(D_i(t+1)) - \log(D_i(t)) & i = j,\ D_i(t) \ge M_i \\ \log(D_i(t)) & i \ne j,\ D_i(t) < M_i \\ \log(M_i) & i \ne j,\ D_i(t) \ge M_i. \end{cases} \tag{2.20}$$

The working of the method can be understood in the following way. The duration probability density function is used to modify the probability of a transition occurring, based on the duration spent in the state from which the transition occurs. When a transition to a different state is taken, the exact duration is known and can be used to modify the probability. In considering self-transitions, however, the penalty can not be incorporated on a frame by frame basis since the eventual duration in a state is yet unknown. Incorporating the duration probability at each frame as if it were the last time step in a state would penalise initial self-transitions in a state – causing an incorrect bias towards transitions from the previous state. Therefore the method should not penalise self-transitions until the peak duration probability is reached in a state. After the point of peak duration probability, duration penalty is applied in accordance with the duration probability density.

With duration modelling now incorporated into the Viterbi search, we turn our attention to the implementation of the level building algorithm. When recognition of a sequence of spoken words is attempted, it is desired to find the best match across all possible sequences of word and pause models. An exhaustive search of depth V, containing R possibilities in each level leads to R^V Viterbi alignments. Even for a simple task like connected digit recognition it leads to 10^{10} Viterbi alignments (if pause models are ignored) for a string of at most 10 digits - which is not computationally feasible. The level building algorithm [26] dramatically reduces the computational cost by performing only R searches at each of the V levels, thus effectively $V \times R$ Viterbi alignments.

The level building algorithm works by computing at each successive level l the most likely final state probability (P_t^l) at each frame t over all R models in the search path

$$P_t^l = \max_{1 \leq r \leq R} [\Phi_N^r(t)]. \tag{2.21}$$

After a level has been completed, the final state probabilities are used as initial state probabilities for all Viterbi searches at the next level, i.e. we now set $\Phi_0^r(t) = P_t^l, 1 \leq r \leq R$. This process continues until the desired number of levels have been searched. The most likely sequence ends at the level given by

$$\arg \max_{1 \leq l \leq V} (P_T^l). \tag{2.22}$$

From the most likely final state at the final frame it is easy to backtrack the complete path followed through all levels provided that the backtracking information from each individual Viterbi alignment has been retained. Note that the most likely solution does not necessarily present itself at the last level. The level building technique can thus be used to find unknown length word strings up to the maximum depth for which was searched. In continuous speech recognition experiments we set the maximum depth large enough so as to not influence the results.

The level building algorithm was initially used for experiments, but was later superseded

by the use of a frame synchronous trellis search [43] using the Viterbi algorithm, yielding identical results with less computational cost. With the general HMM system background now covered, we turn to previous research in the field of multilingual speech recognition.

2.2 Multilingual speech recognition

To study the similarities between the phones of different languages, one has to examine the relatively new field of multilingual speech recognition. Multilinguality refers to the property of a system to be capable of understanding speech input in more than one language, i.e. it includes both the acoustic and so-called *language* modelling of the relevant languages. As far as acoustic modelling in the multilingual field is concerned, research ranges from systems that have a unified architecture, yet have separate models for each language to systems that share increased numbers of acoustic parameters. Language identification systems are also multilingual systems in a certain sense, but focus mainly on language models to perform discrimination between languages with multiple acoustic models used primarily to extract phone sequences, but also to provide some discriminative information [44, 45].

Speech translation systems from the Verbmobil [10] project, specifically the JANUS [46] system, are amongst the first applications of multilingual speech recognition. The JANUS system is architecturally language independent and each speech recognition module is loaded with models for the specific language it has to recognise. It therefore has a common modelling structure for speech from different languages, but does not share acoustic information between languages.

The field of multilingual information systems has also been actively researched. In the development of the MIT VOYAGER [14] multilingual system, separate context-independent acoustic models were trained for English, Japanese and Italian. The English version used 58 models based on the labels used in the TIMIT [31] database and was trained with data from the same database. For the Japanese and Italian versions, the models were initialised by

seeding them from their most phonetically similar English counterparts. This was reported to enable the further training of the Japanese and Italian models on language specific speech data that was only transcribed but not aligned, thus saving the great amount of work needed to manually align the speech. The Mandarin Chinese version of the GALAXY system, called YINHE [18], followed on the VOYAGER system and also used English models to seed near-neighbour Mandarin acoustic models. Another large multilingual speech recognition system is the BYBLOS Callhome system [15]. The system performs task specific speech recognition in multiple languages using the same architecture, but using separate acoustic models for each language in question.

2.2.1 Bootstrapping of new target language recognisers

Studies have been performed to quantify the effect on system performance of the cross-language bootstrapping of acoustic models. Wheatley *et al.* [16] compared the performance when Japanese acoustic models are bootstrapped with English acoustic models trained on TIMIT to flat-start training of the Japanese models, as well as initialisation of the Japanese models with a limited number of hand-picked representative examples. The application was a connected digit recognition system with some control words and was modelled with whole word models. In the case of bootstrapping with English models the Japanese word models were initialised with sequences of English phone models. Compared to the flat-start approach, the cross-language bootstrapped models and the hand-picked representative example approaches achieved better performance after 2 training iterations. After 10 iterations, the bootstrapped models exhibited a small improvement in overall performance over the other two approaches. The authors also performed a cross-language smoothing experiment. When only a small amount of Japanese data was used, smoothing of the final model by interpolation between English and Japanese models achieved slightly better performance than using the Japanese models directly.

Schultz *et al.* performed bootstrapping of a Japanese recogniser with models from a German recogniser [47] and showed bootstrapping to be an efficient method of initialising the target

language models. In a subsequent study [17] a multilingual phoneme set comprising of the collection of the language dependent phonemes of German, English, Japanese and Spanish was created and used to bootstrap recognisers in Chinese, Croatian and Turkish. Bootstrapping was done through a five step process namely

1. the determining of a mapping of language specific phones to the multilingual set by phonetic experts,
2. initialisation of the acoustic models according to the mapping,
3. maximum likelihood linear regression (MLLR) transformation of the models using language specific data along with language specific linear discriminant analysis (LDA) calculation and K-means codebook clustering,
4. four training iterations and
5. repetition of steps 3 and 4.

Bootstrapping was shown to result in better performance than is achievable with flat start training on target language data when only a few iterations of training is done. Schultz & Waibel later also investigated a simpler form of bootstrapping by performing cross-language training [30] of acoustic models. Various monolingual model sets as well as a multilingual set of HMMs were used as starting models for 2 iterations of Viterbi training on German data. It was shown that using the multilingual phone set as initial model was slightly superior to using 3 of the 5 languages (Turkish, Croatian and Spanish) and was far superior to using Japanese and Korean initial models.

The bootstrapping results discussed in this subsection indicate that cross-lingual models provide good initial models for training in a new language. None of the bootstrapping results, unfortunately, indicate a real advantage in terms of recognition rate of using cross-lingual information. The methods do, however, show the advantage of requiring fewer training iterations for convergence when cross-language seed models are used.

2.2.2 Explicitly multilingual systems

When acoustic information in a source language is used to bootstrap models for a new language, the source language data is only used to construct seed models for initial alignment of unlabelled data and is not used in subsequent re-alignment and re-estimation of models for the new language. In this case separate recognisers are realised for each target language. Other studies have explicitly used multilingual phoneme sets, in which case the eventual models exhibit characteristics of multiple languages. Köhler [48] studied isolated phoneme recognition on continuous American English, German and Spanish telephone speech. He found that the sharing of acoustic information across languages leads to some performance degradation, but that a representation with fewer mixtures than that of the combined models from the three languages still delivered reasonable performance. Weng *et al.* [19] used shared Gaussian codebooks across Swedish and English phones and reported that allowing the sharing of data across phones from the two languages also did not improve performance, but lead to a system capable of performing language identification as part of the decoding process. Bonaventura *et al.* [20] performed experiments to quantify the performance of a system with a language-independent phonetic inventory on Italian, Spanish, English and German words. Dissimilarity measures were proposed to enable automatic determination of which phones from the different language to merge into multilingual phones. Significant reduction of the total number of phones needed was achieved at the cost of some degradation of performance with respect to language dependent phones.

In a detailed study covering five languages (Croatian, Japanese, Korean, Spanish and Turkish), Schultz & Waibel [30] found that monolingual systems outperformed a system with shared multilingual acoustic models and the same number of parameters as the five monolingual systems combined, by approximately 1% (27% versus 28%) in terms of word error rate (WER). The reason given for the decrease in performance is that language independent modelling decreases the precision of the acoustic models. In a study also covering five languages (French, German, Italian, Portuguese and Spanish) Köhler [21] found that a multilingual approach to acoustic modelling yielded a 3.2% (14.2% versus 11.0%) increase

in average WER for an isolated word recognition task and a 4.9% (43.7% versus 48.6%) decrease in correct phone recognition rate compared to a monolingual approach.

A study performed by Uebler *et al.* [49] targeted performance improvement in a bilingual environment where L1 German and Italian speakers spoke both languages, producing L1 and L2 German and Italian speech. The study found that a bilingual German/Italian system outperformed two separate monolingual systems on the test database of L1 and L2 German and Italian speech. The improvement of 1.2% (11.3% versus 12.5%) in WER of the bilingual system is attributed to the large variation in accents and dialects of the L2 speakers in both languages being better captured by the bilingual system than by the monolingual systems.

The research discussed so far in this section has focussed either on bootstrapping to avoid the manual labelling effort in a new language, or on creating shared multilingual phone sets to facilitate integrated multilingual recognition. The latter approach has mostly lead to a degradation in performance over monolingual systems, except where L1 and L2 speech were mixed in an application [49]. Little research has been performed with the goal of improving performance in a specific target language through explicit use of cross-language acoustic information. Research conducted with this specific goal in mind is discussed next.

2.2.3 Cross-language use of acoustic data for new target languages

Bonaventura *et al.* [20] performed experiments where it was assumed that little data was available for training Spanish models. The application was an isolated word recognition system with a vocabulary of 70 words. It was found that the use of phone models trained on both Italian and Spanish data, i.e. on the pooled multilingual data, lead to between 0.6% (12.9% versus 13.5%) and 3% (20% versus 23%) reduction in WER over a system trained only on the Spanish data, depending on the amount of adaptation data used.

The use of on-line Bayesian learning for cross-language adaptation was investigated by Bub *et al.* [22] and applied to Slovene isolated digit recognition. The method used the on-line maximum *a posteriori* (MAP) algorithm, updating only the Gaussian means via linear interpolation between the original and sample means - i.e. the Gaussian variance, mixture weight and transitions probabilities from the original models were not changed. The adaptation of monolingual and a multilingual (German, American English and Spanish) HMM systems to Slovene was considered. Results show that MAP adaptation on 646 utterances of 12 isolated Slovene digits improved the performance of the baseline multilingual HMM system from 76.5% to 85.0%. Unfortunately no comparable results for direct training on the Slovene digit data are given. The WER of 15% is also high for an isolated digit recognition system.

Köhler [21] investigated the cross-language use of multilingual acoustic models (trained on American English, Italian, French, Portuguese and Spanish) in developing a German speech recognition system. A bootstrapping method from Schultz & Waibel [17] was compared to cross-language mean-only MAP adaptation on German adaptation data and it was found that for little adaptation data, the cross-language adaptation approach achieved better performance than a bootstrapping or a flat-start approach. When most of the adaptation data was used, a flat-start German system was found to achieve the best performance. The relatively poor performance of the adapted system when more data is available is probably due to the fact that only the mean parameters were adapted- since it is known that the performance of the MAP algorithm is asymptotic with the task dependent performance as the amount of data increases (Lee *et al.* [24]).

In this section we discussed the various approaches that were followed in previous research on multilingual recognition. Some adaptation algorithms, notably MAP and MLLR were used, albeit in a limited fashion. Proper use of adaptation algorithms presents the logical extension to the research covered in this chapter. In the next chapter we therefore proceed to discuss adaptation algorithms in depth to apply these methods for cross-language adaptation.

Chapter 3

Speaker adaptation theory

This chapter discusses previous research in speaker adaptation, but places it within the context of our topic of cross-language adaptation. Reasonably detailed derivations of algorithms are given, especially when understanding of the algorithms are necessary for their proper use for cross-language adaptation versus for speaker adaptation as such.

3.1 Background on speaker adaptation

It has been established that if sufficient data is available, a speaker-dependent system outperforms a speaker independent system [50]. Research [51] has shown that differences between speakers is of much smaller magnitude than differences between phonemes, which is why speaker independent systems function reasonably well in spite of not modelling the exact characteristics of any given speaker. Speaker independent systems, however, perform poorly for speakers with different accents than those with which the system was trained. For most applications there is not enough speaker dependent data for the training of robust models and therefore speaker independent or speaker adaptive training is used. Speaker adaptive training uses large amounts of existing information from many speakers to improve the estimation of model parameters when faced with little data from a new speaker.

3.1.1 Speaker variation

The reason for performing adaptation is that there exists variation between the speech of different speakers. This variation can be classified into two main categories [52]:

- acoustic level differences, including
  - realisational,
  - physiological and
  - durational differences, and
- phonological level differences, including
  - lexical and
  - stress differences.

In this thesis we are mainly interested in the former category of speaker differences, or more accurately, in the correspondence between variation at this level across different languages. To the degree that the acoustic level speaker differences are not language specific, we expect direct cross-language re-use of acoustic information to be useful. In terms of a speech recognition system, the latter category of phonological differences between speakers is dealt with at the language (grammar) and pronunciation modelling level and is thus very language specific. However, since we deal with acoustic modelling, phonological speaker differences are not of direct importance.

Realisational factors comprise different methods of using the articulatory organs to produce wanted sounds. Physiological factors influence the generation of sounds by constraining the possible range of sounds that can be generated by an individual. For example the physical dimensions of the articulatory organs and notably the length of the vocal tract is known to influence the formant frequencies of voiced speech. Durational differences between speakers relate to the timing of the different aspects of generating specific sounds.

3.1.2 Speaker normalisation

Speaker normalisation groups together techniques that attempt to remove, or at least reduce, the differences between the speech of different speakers, while retaining the characteristics that distinguish the different phonetic categories. Vocal tract length normalisation (VTLN) is one such technique that estimates vocal tract length and computes a spectral shift accordingly [53]. An important aspect of normalisation is taking into account not only the characteristics of the particular speaker, but also being able to compensate for recording channel mismatch between training and testing conditions. Subtraction of the estimated mismatch between the training and testing conditions is often done in the cepstral domain, also called cepstral mean subtraction (CMS). CMS performs a cancellation of the effect of any linear operator in the frequency domain, such as is caused by using a microphone with a different frequency transfer function or by frequency filtering due to a transmission channel.

Normalisation is usually applied to the speech signal, or at least to the observation vector sequence as part of the pre-processing stage of the classifier. Normalisation can also be applied during the training phase to compensate for channel variations if applicable, or to reduce spectral differences between training speakers, resulting in more accurate models [54]. When considering the use of multiple databases for cross-language use of data it may be important to apply a normalisation technique such as CMS to take care of recording channel mismatch between the databases. Normalisation will, however, also remove overall spectral differences between the languages, influencing the distribution of feature vectors for all phones. The languages and databases concerned may differ significantly with respect to the phones and the relative quantities of these phones they contain, causing application of CMS to entire databases to be biased. A solution may be to weight the contribution of the data associated with each individual phone in computing the cepstral mean for a database. We discuss this topic in more detail in Section 5.1 where aspects regarding cross-database use of acoustic information are discussed.

Normalisation overlaps to a large degree with speaker adaptation, with normalisation usu-

ally seen as the application of adaptation techniques at the feature level, rather than at the model level. Some types of model adaptation, such as transformation-based adaptation, may implicitly perform normalisation, such as done by CMS, with an offset term and can approximate the spectral shift performed with VTLN in the cepstral mean transformation matrix [55], thus further blurring the distinction between adaptation and normalisation. Other adaptation techniques, such as Bayesian or discriminative training-based adaptation can not efficiently remove bias and thus the use of normalisation such as CMS in conjunction with adaptation may still be important to achieve good recognition performance. Zhao [56] performed experiments showing that acoustic normalisation (via CMS) followed by Bayesian adaptation achieved improved performance compared to performing only Bayesian adaptation when training on speech from one database and testing on speech from another database not exactly matched to the first.

3.1.3 Modes of applying speaker adaptation

Speaker adaptation can be applied in an on-line or an off-line mode. For dictation systems speaker adaptation can generally be performed in off-line or static mode, with adaptation occurring after initial enrolment and at intervals after collection of more data. For telephone-based systems, adaptation, if any, has to be applied on-line or dynamically on a per-call basis. The main difference between static and dynamic adaptation is in terms of the need for real-time implementation. Real-time constraints force dynamic adaptation to be performed on very little data, typically a single utterance, while static adaptation such as used for dictation systems, may use perhaps 30 minutes of speaker specific data. On-line methods use incremental techniques that typically only slightly change model parameters with each additional utterance used, while off-line methods perform batch-mode parameter updates that may completely re-estimate parameters. Cross-language adaptation is performed off-line since real-time constraints are not applicable. On-line adaptation may of course still be used after this to further increase performance when the system is applied to specific speakers.

Another important aspect to take into account is whether adaptation will be supervised or unsupervised. In supervised adaptation the adaptation speech has been labelled, or at least a transcription of the adaptation speech is available. In unsupervised adaptation, the speech to be used for adaptation is unknown and has to be recognised first before it can be used for adaptation. Chapter 2 discussed cross-language use of bootstrapping methods where transcriptions of the data in the target language were available, but the data was not labelled at phone level. Completely unsupervised cross-language adaptation is probably not feasible since the mismatch between the models and data would probably be too great for recognition in the target language to give acceptable results for further training or adaptation.

3.1.4 Categories of speaker adaptation

Speaker adaptation techniques have previously been classified into three categories [54] namely: (i) speaker classification, (ii) spectral transformation and (iii) speaker adaptive re-estimation of model parameters. We use a similar structure for our discussion of speaker adaptation techniques, but consider the transformation category to encompass newer techniques using transformations of model parameters and not only spectral or feature space transformations. Furthermore the third category of speaker adaptive re-estimation is quite wide and we limit ourselves in this chapter to the discussion of Bayesian adaptation techniques. A further field only recently applied to speaker adaptation, namely discriminative learning, is discussed in the next chapter. An overview of the three categories of

- speaker classification,
- transformation-based adaptation and
- Bayesian adaptation

is given next.

Speaker classification attempts to identify a specific set of models that best exhibit the characteristics of a new speaker and uses those models to perform recognition. The speaker classification category is of little interest to our research as it cannot change the characteristics of the acoustic space except to cluster it into segments. It is unlikely that significant overlap will occur between the clusters of speakers in different languages and even if there were significant overlap, the method would still only be useful in terms of handling speaker specific characteristics and not performing any adaptation to the new target language. The other two categories are more interesting to our research as they both can change source language model parameters in a structured way to better reflect the characteristics of the target language.

Transformation-based adaptation entails computing a transformation of pre-trained model parameters to better fit the speech of a new target speaker. This type of adaptation has ties with the technique of normalisation, discussed in Section 3.1.2, which operates on either frequency or cepstral domain features of the observation sequence to reduce spectral differences between speakers. Transformation-based adaptation, at its simplest, may entail only the subtraction of a global cepstral offset term, thereby improving the spectral match between the pre-trained model and the target speakers' speech in the same way that would be achieved with normalisation. However, when we refer to transformation-based adaptation, we usually imply that a matrix transformation of the model parameters is estimated - a more complex and powerful approach than frequency equalisation since (i) correlated noise can be removed and (ii) different transformations can be estimated for different phone groupings.

Transformation-based techniques assume a large degree of correlation between the feature distribution expressed by the current model and the feature distribution of the target speaker. This paradigm is well suited for the removal of correlated noise between source and target parameters. In contrast, Bayesian learning does not assume correlation with respect to changes from a current model, but assumes that prior knowledge exists about the distribution of the model parameters. Observations from a new speaker are treated as adding to the prior knowledge of the parameter distributions, thereby improving the estimate of the

parameters. We expect Bayesian methods to work well in an environment where we have reasonably robust models in general, but which may need complex fine-tuning to achieve improved performance for a specific speaker or environment. The next two sections discuss in detail the implementation of Bayesian and transformation methods used in this thesis.

3.2 Bayesian adaptation

Bayesian estimation presents an alternative to maximum likelihood estimation and is preferred in particular when information about the distribution of unknown parameters is available. Bayesian learning also provides a framework for parameter smoothing and speaker adaptation when faced with a limited amount of data. In this section we are specifically interested in the use of Bayesian methods for adaptation. Bayesian estimation is well suited to the speaker adaptation paradigm, because information about (prior) distributions of model parameters can be estimated beforehand from a large set of speakers and then fine-tuned using measured observations from a new speaker.

Bayesian methods consider model parameters to be random variables with known *a priori* distributions. Observation of sample data from a new speaker converts the *a priori* density of a parameter into an *a posteriori* density, improving the estimate of the true value of the parameter and converging to the true value as the amount of observations increases. In Bayesian estimation, the unknown, but desired p.d.f. $p(\mathbf{x})$ is estimated by using the observed data $\mathbf{X} = \{\mathbf{x}_1, ..., \mathbf{x}_n\}$ and integrating over the parameter vector $\boldsymbol{\theta}$, which is considered a random variable taking values in the space Θ. The integral is expressed by [25, p. 51]

$$\begin{aligned} p(\mathbf{x}|\mathbf{X}) &= \int_{\Theta} p(\mathbf{x}, \boldsymbol{\theta}|\mathbf{X}) d\boldsymbol{\theta} \\ &= \int_{\Theta} p(\mathbf{x}|\boldsymbol{\theta}) p(\boldsymbol{\theta}|\mathbf{X}) d\boldsymbol{\theta}. \end{aligned} \tag{3.1}$$

Using Bayes rule, Equation 3.1 can be written, using the notation $g(\boldsymbol{\theta})$ for the prior distri-

bution and $f(\mathbf{X}|\boldsymbol{\theta})$ for the likelihood function, by

$$p(\mathbf{x}|\mathbf{X}) = \int_{\Theta} p(\mathbf{x}|\boldsymbol{\theta}) \frac{f(\mathbf{X}|\boldsymbol{\theta})g(\boldsymbol{\theta})}{p(\mathbf{X})} d\boldsymbol{\theta}, \tag{3.2}$$

where the observation probability, $p(\mathbf{X}) = \int_{\Theta} f(\mathbf{X}|\boldsymbol{\theta})g(\boldsymbol{\theta})d\boldsymbol{\theta}$, is a constant that normalises the posterior density function. In practice Equation 3.2 does not offer a computationally feasible solution with current speech modelling techniques and computer technology due to the integration term. However, if $p(\boldsymbol{\theta}|\mathbf{X})$ peaks very sharply about some value $\hat{\boldsymbol{\theta}}$, Equation 3.2 may be approximated by

$$p(\mathbf{x}|\mathbf{X}) \approx p(\mathbf{x}|\hat{\boldsymbol{\theta}}). \tag{3.3}$$

This is especially applicable according to the Bayesian learning paradigm described by Duda & Hart [25, p. 54], which states in general that as the number of observations from a given distribution increases, the posterior distributions of the parameters peak more sharply around the true values of the parameters, ultimately approaching Dirac delta functions at the true values of the parameters as the number of observations approaches infinity. In this case the approximation is therefore entirely applicable.

However, even if the posterior parameter distribution is not sufficiently peaked, to reach a computationally feasible solution, it may still be necessary to estimate a single parameter value $\hat{\boldsymbol{\theta}}$ for use in place of the integration over the parameter space of Equation 3.2. The next section discusses a procedure to estimate such a parameter.

3.2.1 Bayes estimators

Because Bayesian methods consider parameters to be random variables, distributions of parameters are used, rather than fixed values. For efficiency, a single suitable value for the parameter may need to be estimated and for this purpose an estimator is used. The form of the estimator is not prescribed in Bayesian learning and remains to be decided by

the statistician. The most important requirement of an estimator δ is that it delivers an estimate $\delta(\mathbf{X})$ (based on the observed data $\mathbf{X}$) that is close to the actual value $\mathbf{a}$ of the parameter $\boldsymbol{\theta}$ in an experiment. A sensible way of determining an estimator is by specifying a *loss function* $L(\mathbf{a}, \hat{\boldsymbol{\theta}})$ which measures the loss or cost when the true value of the parameter is $\boldsymbol{\theta} = \mathbf{a}$ and the estimate is $\hat{\boldsymbol{\theta}}$. The *Bayes estimator* [57, p. 275] is then given by the function $\delta^*(\mathbf{X})$ that, for every possible value $\mathbf{x}$ of $\mathbf{X}$, delivers the minimum expected loss, i.e.

$$\mathrm{E}[L(\boldsymbol{\theta}, \delta^*(\mathbf{X}))|\mathbf{X}] = \min_{\hat{\boldsymbol{\theta}} \in \Theta} \mathrm{E}[L(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}})|\mathbf{X}], \tag{3.4}$$

where the unknown value $\hat{\boldsymbol{\theta}}$ of $\boldsymbol{\theta}$ takes values in the space Θ.

Minimum square error Bayes estimation

The loss function that is most commonly used is the *squared error loss function*, $L(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}) = (\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})^T(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})$. When the squared error loss function is used, the Bayes estimate is the value of $\hat{\boldsymbol{\theta}}$ for which $\mathrm{E}[(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})^T(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})|\mathbf{X}]$ reaches a minimum value. The Bayes estimator for the squared error loss function is found by finding the root of the quadratic, i.e.

$$\begin{aligned}
\frac{\partial}{\partial \hat{\boldsymbol{\theta}}} \mathrm{E}[(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})^T(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})|\mathbf{X}] &= 0 \\
\frac{\partial}{\partial \hat{\boldsymbol{\theta}}} \left[\mathrm{E}[\boldsymbol{\theta}^T\boldsymbol{\theta}|\mathbf{X}] - 2\hat{\boldsymbol{\theta}}^T \mathrm{E}[\boldsymbol{\theta}|\mathbf{X}] + \hat{\boldsymbol{\theta}}^T\hat{\boldsymbol{\theta}} \right] &= 0 \\
-2\mathrm{E}[\boldsymbol{\theta}|\mathbf{X}] + 2\hat{\boldsymbol{\theta}} &= 0
\end{aligned}$$

and thus the Bayes estimator is simply equal to the expectation value of the parameter $\boldsymbol{\theta}$,

$$\begin{aligned}
\delta^*(\mathbf{X}) = \hat{\boldsymbol{\theta}} &= \mathrm{E}[\boldsymbol{\theta}|\mathbf{X}] \\
&= \int_{\Theta} \boldsymbol{\theta} \cdot p(\boldsymbol{\theta}|\mathbf{X}) d\boldsymbol{\theta} \\
&= \int_{\Theta} \boldsymbol{\theta} \cdot \frac{f(\mathbf{X}|\boldsymbol{\theta}) g(\boldsymbol{\theta})}{p(\mathbf{X})} d\boldsymbol{\theta},
\end{aligned} \tag{3.5}$$

which in turn equals the first moment of the posterior density function $f(\mathbf{X}|\boldsymbol{\theta})g(\boldsymbol{\theta})/p(\mathbf{X})$. We refer to the Bayes estimator of Equation 3.5 as the MSE estimator in subsequent discussions since it produces the minimum squared error (MSE) solution to the Bayes loss function.

Other loss functions exist and may lead to different Bayes estimators, such as the absolute error loss function which leads to the Bayes estimate being equal to the median of the posterior distribution [57, p. 277]. An alternative to using a loss function in the Bayesian framework is to simply use the maximum value of the posterior distribution as the estimate, which in general will differ from the mean for asymmetric functions. This method is discussed next.

MAP Bayes estimation

Maximum *a posteriori* (MAP) estimation uses the parameter associated with the maximum *a posteriori* probability as the Bayes estimate. The MAP estimate for a parameter $\boldsymbol{\theta}$, given a prior distribution $g(\boldsymbol{\theta})$ and observation sequence $\mathbf{X} = \{\mathbf{x}_1, .., \mathbf{x}_n\}$ is given by the mode of the posterior density function, i.e.

$$\boldsymbol{\theta}_{MAP} = \arg\max_{\boldsymbol{\theta}} p(\boldsymbol{\theta}|\mathbf{X}) = \arg\max_{\boldsymbol{\theta}} f(\mathbf{X}|\boldsymbol{\theta})g(\boldsymbol{\theta}). \tag{3.6}$$

If $g(\boldsymbol{\theta})$ is considered fixed, but unknown, also known as a non-informative prior, then there is no knowledge about $\boldsymbol{\theta}$ and the MAP estimate is equal to the maximum likelihood (ML) estimate. We thus consider the selection of a suitable informative prior. The choice of a prior distribution is predicated as much by its suitability for expressing the prior distribution as by the possibility of deriving a solution for the Bayesian/MAP estimation problem. Similar to ML estimation, the computation of the MAP estimate is relatively easy when the family of p.d.f.'s $\{f(\cdot|\boldsymbol{\theta}), \boldsymbol{\theta} \in \Theta\}$ possesses a *sufficient statistic* of fixed dimension. For HMMs in the *incomplete data* modelling problem this is not true, but is addressed by iterative methods that solve the *complete data* modelling problem for which a sufficient statistic exists. Given

that the family $\{f(\cdot|\boldsymbol{\theta}), \boldsymbol{\theta} \in \Theta\}$ possesses a sufficient statistic $t(\mathbf{X})$ of fixed dimension for the parameter $\boldsymbol{\theta}$, $f(\mathbf{X}|\boldsymbol{\theta})$ can be factored into two terms $f(\mathbf{X}|\boldsymbol{\theta}) = h(\mathbf{X})k(\boldsymbol{\theta}, t(\mathbf{X}))$ such that $h(\mathbf{X})$ is independent of $\boldsymbol{\theta}$ and $k(\boldsymbol{\theta}, t(\mathbf{X}))$ is the *kernel density*, which is a function of $\boldsymbol{\theta}$ and depends on $\mathbf{X}$ only through the sufficient statistic $t(\mathbf{X})$. If the prior density is thus chosen in a *conjugate family* $\{k(\cdot|\boldsymbol{\psi}), \boldsymbol{\psi} \in \Psi\}$ which includes the kernel density of the likelihood function $f(\cdot|\boldsymbol{\theta})$, the MAP estimate is greatly simplified since the posterior density is then of the same form as the prior, i.e. $k(\boldsymbol{\theta}|\boldsymbol{\psi}') \propto k(\boldsymbol{\theta}|\boldsymbol{\psi})k(\boldsymbol{\theta}, t(\mathbf{X}))$. With such a choice of prior, the procedure for finding the MAP estimate is similar to solving for the ML estimate - i.e. both find the mode of the kernel density.

Having a simple posterior density also eases implementation of other Bayesian estimators such as the MSE estimator which finds the mean of the posterior distribution. For symmetric distributions, such as the normal distribution, the mode and mean are equal and thus the MSE and MAP estimates are the same, while for asymmetric distributions, such as the Gamma distribution, the estimates will generally differ. We note at this point that it is only because a limited amount of adaptation data is used that the difference between the MAP and MSE estimators is considered. We do not expect the difference between the estimates produced by the methods to be large, but still wish to quantify the difference.

With some basic theory behind Bayesian estimation now covered, we proceed to discuss the implementation of Bayesian adaptation, and more specifically MSE and MAP adaptation for both the (single) Gaussian observation density case as well as for the more general Gaussian mixture distribution case. We assume that we are solving the *complete data* modelling problem as we shall discuss the implementation of the iterative estimation algorithm [24, 58] for the *incomplete data* modelling problem for HMMs in Section 3.2.4.

3.2.2 Gaussian density parameter distributions

In this section it is assumed that a sample from a Gaussian distribution is available and it is desired to derive the posterior distributions of the parameters of the Gaussian, i.e. the

mean and variance of the Gaussian. The derivations closely follow DeGroot [59].

Mean-only adaptation

The simplest and also most used approach for Bayesian adaptation is to assume a normal distribution with mean m and precision τ (inverse of the variance) as the prior for the mean parameter μ (to be estimated) of the Gaussian observation distribution and a fixed, known value for the Gaussian precision parameter r. The prior distribution of μ

$$g(\mu) \propto \tau^{1/2} e^{-(\tau/2)(\mu-m)^2} \tag{3.7}$$

and the likelihood function $f(X|\mu)$ for observations $X = \{x_1, ..., x_n\}$

$$\begin{aligned} f(X|\mu) &\propto r^{n/2} e^{-(r/2)\sum_{i=1}^{n}(\mu-x_i)^2} \\ &\propto r^{n/2} e^{-(r/2)\,[nS+n(\mu-\bar{x})^2]} \end{aligned} \tag{3.8}$$

where $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ is the sample mean and

$$S = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2 \tag{3.9}$$

is the sample variance of the observations, can be combined to form the posterior p.d.f. $g(\mu|X)$ given by

$$g(\mu|X) \propto f(X|\mu)g(\mu) \propto \tau^{1/2} r^{n/2} e^{-(1/2)[\tau(\mu-m)^2+nr(\mu-\bar{x})^2+nrS]}. \tag{3.10}$$

By using the equality

$$\tau(\mu-m)^2 + nr(\mu-\bar{x})^2 = (\tau+nr)\Big(\mu - \frac{\tau m + nr\bar{x}}{\tau+nr}\Big)^2 + \frac{\tau nr}{\tau+nr} r(m-\bar{x})^2, \tag{3.11}$$

it is noted that the posterior p.d.f. $g(\mu|X)$ of μ is also a normal distribution (similar to the prior of μ in Equation 3.7), with mean $\frac{\tau m+nr\bar{x}}{\tau+nr}$ and precision $\tau+nr$ [59, p. 167] and is given

by

$$g(\mu|X) \propto e^{-\frac{\tau+nr}{2}(\mu-\frac{\tau m+nr\bar{x}}{\tau+nr})^2}. \tag{3.12}$$

Since the mode and the mean value of the normal distribution are equal, both the MAP estimate as well as the minimum squared error Bayesian estimate for μ are given by the mean of Equation 3.12, namely

$$\mu_{\text{MAP}} = \mu_{\text{MSE}} = \frac{\tau m + nr\bar{x}}{\tau + nr}. \tag{3.13}$$

Note that we refer to the mean value of a distribution as the expectation value of the parameter on which the distribution is conditioned. The estimate of μ is a linear combination of the prior mean m and the speaker dependent sample mean $\bar{x}$. When $n = 0$, no observations are available and the MAP estimate is simply equal to the prior mean value m. When a large number of observations are available, the MAP estimate converges to the ML estimate $\bar{x}$ asymptotically. When the prior precision τ is large, high confidence is associated with the prior mean m and a larger sample size will be necessary to significantly change the MAP estimate from the prior mean value than when the prior precision value is small. The difficulty with implementing Bayesian adaptation is in choosing suitable values for the prior distribution parameters. In the MAP approach suggested by Lee *et al.* [24], prior distribution parameters are estimated from speaker independent training data. Speaker independent Gaussian mixture distribution models are used to estimate weighted mean and precision values for the prior as well as the expected value of the precision through

$$m = \sum_{k=1}^{K} \tilde{c}_k \tilde{m}_k, \tag{3.14}$$

$$1/\tau = \sum_{k=1}^{K} \tilde{c}_k (\tilde{m}_k - m)^2 \tag{3.15}$$

and

$$1/r = \sum_{k=1}^{K} \tilde{c}_k \tilde{\sigma}_k^2 \tag{3.16}$$

where $\tilde{c}_k$ is the weight, $\tilde{m}_k$ is the mean and $\tilde{\sigma}_k^2$ is the variance of the kth mixture component of the speaker independent model. The weighted value of m is simply the sample mean of the speaker independent data when the weights are ML estimates while $1/\tau$ equals the variance of the mixture means around the global mean value and $1/r$ is the weighted average variance within a mixture. This choice of estimating the prior distribution and fixed variance makes especially good sense when we expect each mixture distribution to be representative of an individual speaker or type of speaker since Equation 3.15 then represents the expected between-speaker variance and Equation 3.16 the expected within-speaker variance.

Variance-only adaptation

Variance adaptation is proposed by Lee *et al.* [24] by assuming the value of the mean m to be fixed, but unknown and the variance a random variable with a prior distribution $g(\sigma^2)$ of the form

$$g(\sigma^2) = \begin{cases} \text{constant} & \text{if} \quad \sigma^2 \leq \sigma^2_{\min} \\ 0 & \text{otherwise,} \end{cases} \tag{3.17}$$

where $\sigma^2_{\min}$ is estimated from a large amount of speech data and should be a reasonable lower bound on the variance. We have arbitrarily chosen $\sigma^2_{\min} = 10^{-4}$. The MAP estimate for the variance is then given by

$$\sigma^2_{\text{MAP}} = \begin{cases} S & \text{if} \quad S \geq \sigma^2_{\min} \\ \sigma^2_{\min} & \text{otherwise,} \end{cases} \tag{3.18}$$

where S is the sample variance as in Equation 3.9. While Equation 3.18 is not really useful by itself for speaker adaptation since the Gaussian variance plays a much less important role than the Gaussian mean value in speaker adaptation, it is of much use in any training situation when little data is available. The training procedure detailed in Chapter 2.1.4 and all adaptation methods detailed in this thesis also implement Equation 3.18 during parameter re-estimation in the form of a variance floor. This prevents variance values from reaching unrealistically low values when little data is used for estimation or adaptation purposes, thereby improving generalisation.

Mean and variance adaptation

Lee *et al.* [24] proposes a third approach where mean and precision parameters are adapted according to a joint mean and precision prior distribution derived from the set of speaker independent Gaussian mixtures. It has been shown [59, p. 169] that the choice of a normal-Gamma joint prior distribution forms a conjugate family for the mean and precision of a sample from a normal distribution. The joint prior distribution of the mean μ and precision r parameters is as follows: the conditional distribution of μ given r is a normal distribution with mean m and precision wr where $w > 0$, and the marginal distribution of r is a Gamma distribution with parameters $\alpha > 0$ and $\beta > 0$, i.e.,

$$g(\mu, r) \propto r^{1/2} e^{-(wr/2)(\mu-m)^2} r^{\alpha-1} e^{-\beta r}. \tag{3.19}$$

The Gaussian likelihood function given by (similar to Equation 3.8)

$$f(X|\mu, r) \propto r^{n/2} e^{-(r/2)[nS+n(\mu-\bar{x})^2]} \tag{3.20}$$

can be combined with the prior $g(\mu, r)$ of Equation 3.19 to form the posterior p.d.f. $g(\mu, r|X)$

$$g(\mu, r|X) \propto f(X|\mu, r) g(\mu, r) \propto \{r^{1/2} e^{-(1/2)[wr(\mu-m)^2+nr(\mu-\bar{x})^2]}\} r^{\alpha+n/2-1} e^{-\beta r-(nr/2)S}. \tag{3.21}$$

By using the equality

$$wr(\mu - m)^2 + nr(\mu - \bar{x})^2 = (w+n)r\left(\mu - \frac{wm + n\bar{x}}{w+n}\right)^2 + \frac{wn}{w+n}r(m - \bar{x})^2, \tag{3.22}$$

it is noted that the posterior p.d.f. $g(\mu, r|X)$ (from Equation 3.21) of μ and r is also a joint normal-Gamma distribution (similar to the joint prior of μ and r in Equation 3.19) with the following form [59, p. 169]

$$g(\mu, r|X) \propto \{r^{1/2}e^{-(r/2)(w+n)(\mu-\hat{m})^2}\}r^{\hat{\alpha}-1}e^{-\hat{\beta}r}, \tag{3.23}$$

which is discussed in detail next. The part between braces on the right hand side of Equation 3.23 expresses the conditional distribution of μ for a given r and given the observations, which is a normal distribution with mean $\hat{m}$ given by

$$\hat{m} = \frac{wm + n\bar{x}}{w+n} \tag{3.24}$$

and precision $(w+n)r$. The second part on the right hand side of Equation 3.23 expresses the marginal distribution of r given the observations, which is a Gamma distribution with parameters $\hat{\alpha}$ and $\hat{\beta}$ given by

$$\hat{\alpha} = \alpha + n/2 \tag{3.25}$$

and

$$\hat{\beta} = \beta + \frac{n}{2}S + \frac{wn(m - \bar{x})^2}{2(w+n)}. \tag{3.26}$$

It is perhaps not immediately apparent from Equation 3.23 that the marginal distribution of r is simply the second part on the right hand side of the equation, until one considers that the integral over μ of the normalised first part on the right hand side of the equation (the normal distribution) is independent of r, rendering the remaining part the marginal

distribution of r, i.e.

$$\begin{aligned} g(r|X) = \int g(\mu, r|X)d\mu &\propto \int \{[(w+n)r]^{1/2} e^{-(r/2)(w+n)(\mu-\hat{m})^2}\} r^{\hat{\alpha}-1} e^{-\hat{\beta}r} d\mu \\ &\propto r^{\hat{\alpha}-1} e^{-\hat{\beta}r} \int [(w+n)r]^{1/2} e^{-(r/2)(w+n)(\mu-\hat{m})^2} \, d\mu \\ &\propto r^{\hat{\alpha}-1} e^{-\hat{\beta}r}. \end{aligned} \tag{3.27}$$

The posterior distribution of μ and r shows that they are dependent. The joint MAP estimate of μ and r is given by the mode of the 2-dimensional posterior distribution, while the MSE estimate is given by the mean of the distribution. Inspection reveals that the joint posterior distribution (Equation 3.23) has an axis of symmetry along $\mu = \hat{m}$ and thus the expectation value of μ, as well as the value of the mode of μ are independent of r and are equal to $\hat{m}$. Both the MAP and the MSE estimates for μ are given by the mean of the normal distribution

$$\mu_{\text{MAP}} = \mu_{\text{MSE}} = \hat{m} = \frac{wm + n\bar{x}}{w+n}. \tag{3.28}$$

The MAP estimate of the Gaussian precision is calculated by differentiating the joint posterior distribution (Equation 3.23) with respect to r and finding the root of the equation. The calculation is greatly simplified since we know that the mode is located along $\mu = \hat{m}$ and thus we calculate

$$\frac{\partial}{\partial r} g(\hat{m}, r|X) = \frac{\partial}{\partial r} r^{\hat{\alpha}-1/2} e^{-\hat{\beta}r} = 0$$

$$(\hat{\alpha} - 1/2) r^{\hat{\alpha}-3/2} e^{-\hat{\beta}r} - r^{\hat{\alpha}-1/2}(-\hat{\beta}) e^{-\hat{\beta}r} = 0$$

and therefore the MAP estimate of the Gaussian precision is given by

$$r_{\text{MAP}} = \frac{\hat{\alpha} - 1/2}{\hat{\beta}} = \frac{2\alpha - 1 + n}{2\beta + nS + \frac{wn}{w+n}(m - \bar{x})^2}. \tag{3.29}$$

The MSE estimate of the Gaussian precision is simply the mean of the marginal posterior Gamma distribution and is given by (from Equations 3.25 and 3.26)

$$r_{\text{MSE}} = \frac{\hat{\alpha}}{\hat{\beta}} = \frac{2\alpha + n}{2\beta + nS + \frac{wn}{w+n}(m - \bar{x})^2}. \tag{3.30}$$

We note at this point that Lee *et al.* [24] used the *mean* (not the mode as in Equation 3.29) of the marginal distribution of r, i.e. the mean of the Gamma p.d.f. $\hat{\alpha}/\hat{\beta}$, as the MAP estimate. This choice is inconsistent with the definition of MAP estimation, requiring use of the mode of the posterior distribution. In a later paper, Gauvain & Lee [58] refer to the correct MAP estimate $\hat{r} = (\hat{\alpha} - 1/2)/\hat{\beta}$. There is, however, a problem with using the mode of the posterior, since as Equation 3.29 shows, the precision is only valid (larger than zero) for $\hat{\alpha} = \alpha + n/2 > 1/2$. This may pose a problem when no observations are made ($n = 0$), depending on the value of α, for which case it is probably sensible to select to use the mean of the posterior, i.e. $\hat{\alpha}/\hat{\beta}$.

With the MAP estimates now derived, the selection of parameters for the normal-Gamma prior distributions remains to be addressed. Making exactly the same choices as in Equations 3.14-3.16 with respect to the prior normal distribution values (m and $\frac{1}{wr}$), as well as the prior expectation value of the variance (β/α), we get

$$m = \sum_{k=1}^{K} \tilde{c}_k \tilde{m}_k, \tag{3.31}$$

$$\frac{1}{wr} = \sum_{k=1}^{K} \tilde{c}_k (\tilde{m}_k - m)^2 \tag{3.32}$$

and

$$\beta/\alpha = \sum_{k=1}^{K} \tilde{c}_k \tilde{\sigma}_k^2. \tag{3.33}$$

By choosing somewhat arbitrarily the value of $\beta = 1$ we can solve for (Equation 3.33)

$$\alpha = \frac{1}{\sum_{k=1}^{K} \tilde{c}_k \tilde{\sigma}_k^2} \tag{3.34}$$

and using the prior mean value of the Gamma distribution $\alpha/\beta = \alpha$ in place of r in Equation 3.32 we solve for

$$w = \frac{\sum_{k=1}^{K} \tilde{c}_k \tilde{\sigma}_k^2}{\sum_{k=1}^{K} \tilde{c}_k (\tilde{m}_k - m)^2}. \tag{3.35}$$

Since β was chosen arbitrarily, the prior variance of the precision was not considered. We know, however, that for a sample from a Gamma distribution the expectation value of the variance is given by α/β^2, which in our case simply equals α. It is intuitively pleasing that the variance of the precision in the prior is equal to the chosen expectation value of the precision, meaning that large prior values of the precision are associated with larger variance and thus less certainty than for lower values of the precision.

The MAP equations we derived here are the same as those derived by Lee *et al.* [24], except for the offset in the variance estimate, but our derivation shows perhaps more clearly the meaning of the choices with respect to the prior parameters. The procedure outlined above is only for parameter estimation of univariate Gaussian distributions. This is not a problem if diagonal covariance matrices are used with multivariate Gaussian distributions, as they then simplify to independent univariate estimation problems. The next section discusses the implementation of Bayesian adaptation for the general multivariate case.

Multivariate normal distribution adaptation

The derivation of posterior distributions for a multivariate Gaussian distribution is a generalisation of the discussion in the previous section. We proceed to give the derivation of the Bayesian estimates for a joint mean and variance prior distribution. It has been shown [59, p. 177] that the choice of a normal-Wishart joint prior distribution forms a jointly conju-

gate family for the mean and precision of a sample from a multivariate normal distribution. The joint prior distribution of the mean $\boldsymbol{\mu}$ and precision $\mathbf{R}$ parameters is as follows: the conditional distribution of $\boldsymbol{\mu}$ given $\mathbf{R}$ is a normal distribution with mean vector $\mathbf{m}$ and precision matrix $w\mathbf{R}, w > 0$, and the marginal distribution of $\mathbf{R}$ is a Wishart distribution with $\alpha > D - 1$ degrees of freedom and a symmetric positive definite precision matrix $\boldsymbol{\Upsilon}$. The joint prior normal-Wishart distribution is given by [59, p. 178]

$$g(\boldsymbol{\mu}, \mathbf{R}) \propto |\mathbf{R}|^{1/2} e^{-(w/2)(\boldsymbol{\mu}-\mathbf{m})^T \mathbf{R}(\boldsymbol{\mu}-\mathbf{m})} \, |\mathbf{R}|^{(\alpha-D-1)/2} e^{-(1/2)\,\mathrm{tr}[\boldsymbol{\Upsilon}\mathbf{R}]}. \tag{3.36}$$

With the multivariate Gaussian likelihood function for observations $\mathbf{X} = \{\mathbf{x}_1, ..., \mathbf{x}_n\}$ given by

$$\begin{aligned} f(\mathbf{X}|\boldsymbol{\mu}, \mathbf{R}) &\propto |\mathbf{R}|^{n/2} e^{-(1/2)\sum_{i=1}^{n}(\mathbf{x}_i-\boldsymbol{\mu})^T \mathbf{R}(\mathbf{x}_i-\boldsymbol{\mu})} \\ &\propto |\mathbf{R}|^{n/2} e^{-(1/2)\sum_{i=1}^{n}(\mathbf{x}_i-\bar{\mathbf{x}})^T \mathbf{R}(\mathbf{x}_i-\bar{\mathbf{x}}) + n(\boldsymbol{\mu}-\bar{\mathbf{x}})^T \mathbf{R}(\boldsymbol{\mu}-\bar{\mathbf{x}})} \\ &\propto |\mathbf{R}|^{n/2} e^{-(n/2)\,[\mathrm{tr}(\mathbf{S}\mathbf{R}) + (\boldsymbol{\mu}-\bar{\mathbf{x}})^T \mathbf{R}(\boldsymbol{\mu}-\bar{\mathbf{x}})]} \end{aligned} \tag{3.37}$$

and using the equality

$$\begin{aligned} &w(\boldsymbol{\mu}-\mathbf{m})^T \mathbf{R}(\boldsymbol{\mu}-\mathbf{m}) + n(\boldsymbol{\mu}-\bar{\mathbf{x}})^T \mathbf{R}(\boldsymbol{\mu}-\bar{\mathbf{x}}) = \\ &\quad (w+n)\Big(\boldsymbol{\mu} - \frac{w\mathbf{m}+n\bar{\mathbf{x}}}{w+n}\Big)^T \mathbf{R}\Big(\boldsymbol{\mu} - \frac{w\mathbf{m}+n\bar{\mathbf{x}}}{w+n}\Big) + \frac{wn}{w+n}(\mathbf{m}-\bar{\mathbf{x}})^T \mathbf{R}(\mathbf{m}-\bar{\mathbf{x}}) \end{aligned} \tag{3.38}$$

the posterior p.d.f. $g(\boldsymbol{\mu}, \mathbf{R}|\mathbf{X}) \propto f(\mathbf{X}|\boldsymbol{\mu}, \mathbf{R}) g(\boldsymbol{\mu}, \mathbf{R})$ is also a normal-Wishart distribution with the following form [59, p. 178]

$$g(\boldsymbol{\mu}, \mathbf{R}|\mathbf{X}) \propto \{|\mathbf{R}|^{1/2} e^{-(1/2)(w+n)(\boldsymbol{\mu}-\hat{\mathbf{m}})^T \mathbf{R}(\boldsymbol{\mu}-\hat{\mathbf{m}})}\} \, \{|\mathbf{R}|^{(\alpha+n-D-1)/2} e^{-(1/2)\,\mathrm{tr}[\hat{\boldsymbol{\Upsilon}}\mathbf{R}]}\}, \tag{3.39}$$

which is discussed next. The posterior conditional distribution of $\boldsymbol{\mu}$ for a given $\mathbf{R}$ and given the observations is a normal distribution with mean $\hat{\mathbf{m}}$ given by

$$\hat{\mathbf{m}} = \frac{w\mathbf{m}+n\bar{\mathbf{x}}}{w+n} \tag{3.40}$$

and precision $(w+n)\mathbf{R}$. The marginal posterior distribution of $\mathbf{R}$ given the observations is a Wishart distribution with $\alpha+n$ degrees of freedom and precision matrix $\hat{\Upsilon}$ given by

$$\hat{\Upsilon} = \Upsilon + n\mathbf{S} + \frac{wn}{w+n}(\mathbf{m}-\bar{\mathbf{x}})(\mathbf{m}-\bar{\mathbf{x}})^T. \tag{3.41}$$

Since the posterior conditional normal distribution has an axis of symmetry along $\boldsymbol{\mu} = \hat{\mathbf{m}}$, the MAP and MSE estimates of $\boldsymbol{\mu}$ are independent of $\mathbf{R}$ and are given by

$$\boldsymbol{\mu}_{\text{MAP}} = \boldsymbol{\mu}_{\text{MSE}} = \hat{\mathbf{m}} = \frac{w\mathbf{m}+n\bar{\mathbf{x}}}{w+n}. \tag{3.42}$$

The MSE estimate of the Gaussian covariance can be written in terms of the mean value of the posterior marginal Wishart p.d.f.

$$\mathbf{R}^{-1}_{\text{MSE}} = \frac{\hat{\Upsilon}}{\alpha+n} = \frac{\Upsilon + n\mathbf{S} + \frac{wn}{w+n}(\mathbf{m}-\bar{\mathbf{x}})(\mathbf{m}-\bar{\mathbf{x}})^T}{\alpha+n} \tag{3.43}$$

while the MAP estimate can be derived by calculating the derivative of $g(\hat{\mathbf{m}}, \mathbf{R}|\mathbf{X})$ (from Equation 3.39) with respect to $\mathbf{R}$ and setting it equal to zero, which delivers

$$\mathbf{R}^{-1}_{\text{MAP}} = \frac{\Upsilon + n\mathbf{S} + \frac{wn}{w+n}(\mathbf{m}-\bar{\mathbf{x}})(\mathbf{m}-\bar{\mathbf{x}})^T}{\alpha+n-D}. \tag{3.44}$$

It can be attempted to estimate values for the parameters of the prior distributions from speaker independent mixture models in the same way as for the univariate case, using the criteria of Equations 3.31-3.33:

$$\mathbf{m} = \sum_{k=1}^{K} \tilde{c}_k \tilde{\mathbf{m}}_k, \tag{3.45}$$

$$(w\mathbf{R})^{-1} = \sum_{k=1}^{K} \tilde{c}_k (\tilde{\mathbf{m}}_k - \mathbf{m})(\tilde{\mathbf{m}}_k - \mathbf{m})^T \tag{3.46}$$

and

$$\Upsilon/\alpha = \sum_{k=1}^{K} \tilde{c}_k \tilde{\boldsymbol{\Sigma}}_k, \tag{3.47}$$

where $\tilde{\Sigma}_k$ is the covariance matrix for mixture k of the speaker independent model. Setting $\mathbf{R}^{-1}$ equal to Υ/α in Equation 3.47 uses the expectation value of the prior covariance, but causes the equations to have no solution since $\mathbf{R}$ is over-determined. However, if diagonal dominance of the precision is assumed, use of the trace on both sides of Equation 3.46 allows a reasonable solution to be found for w. A choice with respect to either Υ or α still needs to be made. Without further information, it may be necessary to make an arbitrary assignment. A choice that will satisfy the constraints is e.g. selecting $\alpha = D + 1$. We do not discuss prior estimation for the multivariate case in more detail here, but return to the topic in Section 3.2.5 where a method for estimation of prior parameters for a multivariate mixture distribution is discussed.

The preceding procedures are applicable for the estimation of (single) Gaussian distributions, which we have found to be useful for speaker adaptation, even when cross-language prior models are used [33]. However, to estimate complex models commonly used for speaker independent recognition, we have to consider the problem of adaptation of mixture density models, which is addressed in the next section.

3.2.3 Mixture density HMM parameter distributions

This section expands on the previous sections that dealt with mean and variance adaptation in a Gaussian framework and places those derivations in the context of Gaussian mixture densities used as output distributions in an HMM with state transition probabilities. Gauvain & Lee [58, 60, 61] suggested applying Bayesian learning of Gaussian mixture components to speaker adaptation of CDHMMs. The method uses parameters of individual Gaussian components in a speaker independent HMM to compute prior distribution parameters for the adaptation of the Gaussian mean, variance and component weight, as well as for the adaptation of state transition parameters within a single framework. We proceed to discuss the prior distribution for a mixture density.

Mixture weight distributions

The Gaussian mixture density for a given state j can be considered a density associated with a statistical population consisting of a mixture of K component populations with mixing proportions $c_{j1}, ..., c_{jK}$. The sizes of the component populations can then be considered to be distributed according to a multinomial distribution, given by

$$f(n_{j1}, .., n_{jK}|c_{j1}, .., c_{jK}) \propto \prod_{k=1}^{K} c_{jk}^{n_{jk}} \tag{3.48}$$

where n_{jk} occurrences of each of the $1 \leq k \leq K$ mixture densities in state j are observed. It is known that the Dirichlet density [59, p. 174]

$$g(c_{j1}, .., c_{jK}) \propto \prod_{k=1}^{K} c_{jk}^{v_{jk}-1} \tag{3.49}$$

with prior parameters $v_{j1}, .., v_{jK}$ in this case, is a conjugate density for a sample from the multinomial distribution and is thus suitable for expressing prior information about the mixing proportions. The posterior Dirichlet p.d.f. of the mixing proportions, or mixture weights as we refer to them, is simply given by

$$\begin{aligned} g(c_{j1}, .., c_{jK}|n_{j1}, .., n_{jK}) &\propto f(n_{j1}, .., n_{jK}|c_{j1}, .., c_{jK}) g(c_{j1}, .., c_{jK}) \\ &\propto \prod_{k=1}^{K} c_{jk}^{n_{jk}} \prod_{k=1}^{K} c_{jk}^{v_{jk}-1} \\ &\propto \prod_{k=1}^{K} c_{jk}^{v_{jk}+n_{jk}-1}. \end{aligned} \tag{3.50}$$

The MAP estimate for the mixture weight is given by the mode of Equation 3.50 [61]

$$c_{jk\,\mathrm{MAP}} = \frac{\hat{v}_{jk} - 1}{\sum_{l=1}^{K} (\hat{v}_{jl} - 1)} \tag{3.51}$$

where $\hat{v}_{jk} = v_{jk} + n_{jk}$ is the parameter of the posterior Dirichlet distribution. The MSE

estimate for the mixture weight is given by the mean of Equation 3.50 [59, p. 51]:

$$c_{jk\,\mathrm{MSE}} = \frac{\hat{v}_{jk}}{\sum_{l=1}^{K} \hat{v}_{jl}}. \tag{3.52}$$

Transition probability distributions

The HMM state transition probability parameters can be dealt with in much the same way as the mixture weight parameters. If the assumption is made that the transition probability parameters are independent of the other HMM parameters and that each row of the transition probability matrix $\mathbf{A}$ is independent, which is true for a first order Markov process, each row of the transition probability matrix can be considered to be the parameter of a multinomial distribution, characterising the number of transitions from state i to each state in the HMM, with likelihood function

$$f(n_{i1}, .., n_{iN} | a_{i1}, .., a_{iN}) \propto \prod_{j=1}^{N} a_{ij}^{n_{ij}} \tag{3.53}$$

where n_{ij} transitions from state i to each of the $1 \leq j \leq N$ states are observed. The prior Dirichlet density is expressed by

$$g(a_{i1}, .., a_{iN}) \propto \prod_{j=1}^{N} a_{ij}^{\eta_{ij}-1} \tag{3.54}$$

with prior parameters $\eta_{i1}, .., \eta_{iN}$ for the transition probabilities from state i. Similar to Equation 3.50, but calculating the joint p.d.f. of the transition probabilities from each state including dummy state 0, we derive the joint posterior distribution

$$\begin{aligned} g(\mathbf{A} | \{n_{ij}\}_{\,i=0,..,N;\,j=1,..,N}) &\propto \prod_{i=0}^{N} f(n_{i1}, .., n_{iN} | a_{i1}, .., a_{iN}) g(a_{i1}, .., a_{iN}) \\ &\propto \prod_{i=0}^{N} \Big[\prod_{j=1}^{N} a_{ij}^{\eta_{ij}+n_{ij}-1} \Big]. \end{aligned} \tag{3.55}$$

The MAP estimate for the transition probability parameters is given by the mode of Equation 3.55:

$$a_{ij\,\mathrm{MAP}} = \frac{\hat{\eta}_{ij} - 1}{\sum_{l=1}^{K}(\hat{\eta}_{il} - 1)} \tag{3.56}$$

where $\hat{\eta}_{ij} = \eta_{ij} + n_{ij}$ is the parameter of the posterior Dirichlet distribution. The MSE estimate for the mixture weight is given by the mean of Equation 3.55:

$$a_{ij\,\mathrm{MSE}} = \frac{\hat{\eta}_{ij}}{\sum_{l=1}^{K}\hat{\eta}_{il}}. \tag{3.57}$$

Now that we have prior distribution families for every parameter of the Gaussian mixture HMM in isolation, we combine these prior distributions to form a joint prior distribution.

Joint prior distribution for HMM parameters

Assuming independence between the transition probability parameters, the mixture weight parameters and the parameters of the mixture distribution, the prior distributions of the parameters of the Gaussian mixture HMM $\boldsymbol{\lambda}$ can be combined in a joint prior distribution

$$g(\boldsymbol{\lambda}) \propto \prod_{i=1}^{N}\left[a_{0i}^{\eta_{0i}-1}\left[\prod_{j=1}^{N}a_{ij}^{\eta_{ij}-1}\right]\left[\prod_{k=1}^{K}c_{ik}^{v_{ik}-1}g(\boldsymbol{\mu}_{ik},\mathbf{R}_{ik})\right]\right] \tag{3.58}$$

with the prior normal-Wishart mixture parameter distribution given by (see Equation 3.36)

$$g(\boldsymbol{\mu}_{ik},\mathbf{R}_{ik}) \propto |\mathbf{R}_{ik}|^{1/2}e^{-(w_{ik}/2)(\boldsymbol{\mu}_{ik}-\mathbf{m}_{ik})^T\mathbf{R}_{ik}(\boldsymbol{\mu}_{ik}-\mathbf{m}_{ik})}\,|\mathbf{R}_{ik}|^{(\alpha_{ik}-D-1)/2}e^{-(1/2)\,\mathrm{tr}[\boldsymbol{\Upsilon}_{ik}\mathbf{R}_{ik}]}. \tag{3.59}$$

Under the complete data density assumption, which explicitly uses state and mixture alignment, posterior distributions for the parameters of an HMM can be derived. This is done next.

Complete data HMM likelihood function

The *complete data* likelihood for a mixture density HMM $\boldsymbol{\lambda}$ is the joint likelihood of the observations $\mathbf{X} = \{\mathbf{x}_1, .., \mathbf{x}_T\}$, the state alignment given by $\mathbf{q} = \{q_1, ..., q_T\}$ and the mixture alignment given by $\mathbf{l} = \{l_1, ..., l_T\}$ (see Equation 2.6):

$$f(\mathbf{X}, \mathbf{q}, \mathbf{l}|\boldsymbol{\lambda}) \propto \prod_{t=1}^{T} \left[a_{q_{t-1}q_t} c_{q_t l_t} |\mathbf{R}_{q_t l_t}|^{1/2} e^{-(1/2)(\boldsymbol{\mu}_{q_t l_t} - \mathbf{x}_t)^T \mathbf{R}_{q_t l_t} (\boldsymbol{\mu}_{q_t l_t} - \mathbf{x}_t)} \right], \tag{3.60}$$

From the state and mixture alignments, mixture occupancy $\gamma_{jk}(t)$ and transition occupancy $\xi_{ij}(t)$ (described by Equations 2.12 and 2.13 respectively) can be computed. From a decoding point of view, this correspond to Viterbi state alignment and choosing the most likely mixture at each state aligned observation frame. We note that the forward-backward algorithm can also be used to calculate values for the statistics $\gamma_{jk}(t)$ and $\xi_{ij}(t)$, but for the complete data likelihood we assume exact state and mixture alignment. In the following section (Section 3.2.4) this constraint will be eased when the estimation strategy is discussed. Further statistics can be defined:

$$\gamma_{ik} = \sum_{t=1}^{T} \gamma_{ik}(t), \tag{3.61}$$

$$\xi_{ij} = \sum_{t=1}^{T} \xi_{ij}(t), \tag{3.62}$$

$$\bar{\mathbf{x}}_{ik} = (1/\gamma_{ik}) \sum_{t=1}^{T} \gamma_{ik}(t) \mathbf{x}_t \tag{3.63}$$

and

$$\mathbf{S}_{ik} = (1/\gamma_{ik}) \sum_{t=1}^{T} \gamma_{ik}(t) (\mathbf{x}_t - \bar{\mathbf{x}}_{ik})(\mathbf{x}_t - \bar{\mathbf{x}}_{ik})^T \tag{3.64}$$

where γ_{jk} is the total occupancy of mixture k in state j, ξ_{ij} is number of transitions in the aligned data from state i to state j, and $\bar{\mathbf{x}}_{ik}$ is the sample mean and $\mathbf{S}_{ik}$ the sample variance of observations in mixture k of state i. Using the statistics of Equations 3.61-3.64 and the

compact form of the Gaussian likelihood function (see Equation 3.37), the complete data likelihood function of Equation 3.60 can then be written as

$$f(\mathbf{X},\mathbf{q},\mathbf{l}|\boldsymbol{\lambda}) \propto \prod_{i=1}^{N}\left[a_{0i}^{\xi_{0i}}\left[\prod_{j=1}^{N}a_{ij}^{\xi_{ij}}\right]\left[\prod_{k=1}^{K}c_{ik}^{\gamma_{ik}}|\mathbf{R}_{ik}|^{\gamma_{ik}/2}e^{-(\gamma_{ik}/2)[\mathrm{tr}(\mathbf{S}_{ik}\mathbf{R}_{ik})+(\boldsymbol{\mu}_{ik}-\bar{\mathbf{x}}_{ik})^T\mathbf{R}_{ik}(\boldsymbol{\mu}_{ik}-\bar{\mathbf{x}}_{ik})]}\right]\right]. \tag{3.65}$$

Complete data posterior distribution

The prior $g(\boldsymbol{\lambda})$ (Equation 3.58) includes the kernel density of the complete data likelihood function $f(\mathbf{X},\mathbf{q},\mathbf{l}|\boldsymbol{\lambda})$ (Equation 3.65) and is thus a conjugate prior distribution for the complete data density. From Equations 3.58 and 3.65, the joint posterior distribution $g(\mathbf{q},\mathbf{l},\hat{\boldsymbol{\lambda}}|\mathbf{X})$ for the complete data density is therefore given by

$$\begin{aligned} g(\mathbf{q},\mathbf{l},\hat{\boldsymbol{\lambda}}|\mathbf{X}) \propto f(\mathbf{X},\mathbf{q},\mathbf{l}|\boldsymbol{\lambda})g(\boldsymbol{\lambda}) \propto \prod_{i=1}^{N}\Bigg[a_{0i}^{\eta_{0i}-1}\Big[\prod_{j=1}^{N}a_{ij}^{\eta_{ij}-1}\Big] \\ \Big[\prod_{k=1}^{K}c_{ik}^{v_{ik}-1}|\mathbf{R}_{ik}|^{1/2}e^{-(w_{ik}/2)(\boldsymbol{\mu}_{ik}-\mathbf{m}_{ik})^T\mathbf{R}_{ik}(\boldsymbol{\mu}_{ik}-\mathbf{m}_{ik})}\,|\mathbf{R}_{ik}|^{(\alpha_{ik}-D-1)/2}e^{-(1/2)\,\mathrm{tr}[\boldsymbol{\Upsilon}_{ik}\mathbf{R}_{ik}]}\Big]\Bigg] \\ \prod_{i=1}^{N}\left[a_{0i}^{\xi_{0i}}\left[\prod_{j=1}^{N}a_{ij}^{\xi_{ij}}\right]\left[\prod_{k=1}^{K}c_{ik}^{\gamma_{ik}}|\mathbf{R}_{ik}|^{\gamma_{ik}/2}e^{-(\gamma_{ik}/2)[\mathrm{tr}(\mathbf{S}_{ik}\mathbf{R}_{ik})+(\boldsymbol{\mu}_{ik}-\bar{\mathbf{x}}_{ik})^T\mathbf{R}_{ik}(\boldsymbol{\mu}_{ik}-\bar{\mathbf{x}}_{ik})]}\right]\right]. \end{aligned} \tag{3.66}$$

By re-arranging terms, $g(\mathbf{q},\mathbf{l},\hat{\boldsymbol{\lambda}}|\mathbf{X})$ can be written in the same form as the joint prior distribution $g(\boldsymbol{\lambda})$ (Equation 3.58) by:

$$\begin{aligned} g(\mathbf{q},\mathbf{l},\hat{\boldsymbol{\lambda}}|\mathbf{X}) \propto \prod_{i=1}^{N}\Bigg[a_{0i}^{\eta_{0i}+\xi_{0i}-1}\Big[\prod_{j=1}^{N}a_{ij}^{\eta_{ij}+\xi_{ij}-1}\Big] \\ \Big[\prod_{k=1}^{K}c_{ik}^{v_{ik}+\gamma_{ik}-1}|\mathbf{R}_{ik}|^{1/2}e^{-(1/2)(w_{ik}+\gamma_{ik})(\boldsymbol{\mu}_{ik}-\hat{\mathbf{m}}_{ik})^T\mathbf{R}_{ik}(\boldsymbol{\mu}_{ik}-\hat{\mathbf{m}}_{ik})}\,|\mathbf{R}_{ik}|^{(\alpha_{ik}+\gamma_{ik}-D-1)/2}e^{-(1/2)\,\mathrm{tr}[\hat{\boldsymbol{\Upsilon}}_{ik}\mathbf{R}_{ik}]}\Big]\Bigg], \end{aligned} \tag{3.67}$$

where the mean of the posterior Gaussian mean $\hat{\mathbf{m}}_{ik}$ is given by (see Equation 3.40)

$$\hat{\mathbf{m}}_{ik} = \frac{w_{ik}\mathbf{m}_{ik} + \gamma_{ik}\bar{\mathbf{x}}_{ik}}{w_{ik} + \gamma_{ik}}, \tag{3.68}$$

and the precision of the posterior Wishart precision $\hat{\boldsymbol{\Upsilon}}_{ik}$ is given by (see Equations 3.41 and 3.38)

$$\begin{aligned}\hat{\boldsymbol{\Upsilon}}_{ik} &= \boldsymbol{\Upsilon}_{ik} + \gamma_{ik}\mathbf{S}_{ik} + \frac{w_{ik}\gamma_{ik}}{w_{ik} + \gamma_{ik}}(\mathbf{m}_{ik} - \bar{\mathbf{x}}_{ik})(\mathbf{m}_{ik} - \bar{\mathbf{x}}_{ik})^T \\ &= \boldsymbol{\Upsilon}_{ik} + \sum_{t=1}^{T} \gamma_{ik}(t)(\hat{\mathbf{m}}_{ik} - \mathbf{x}_t)(\hat{\mathbf{m}}_{ik} - \mathbf{x}_t)^T + w_{ik}(\hat{\mathbf{m}}_{ik} - \mathbf{m}_{ik})(\hat{\mathbf{m}}_{ik} - \mathbf{m}_{ik})^T. \end{aligned} \tag{3.69}$$

The solutions to the other posterior distribution parameters are also similar to those presented in Section 3.2.2 and this section, except that they are given in terms of the sufficient statistics of Equations 3.61-3.64. The parameters of the posterior Dirichlet transition probability and mixture weight densities ($\hat{\eta}_{ij}$ and $\hat{v}_{ik}$ respectively), the relative precision of the conditional posterior mean density, w_{ik} and the number of degrees of freedom of the posterior Wishart precision density are given by:

$$\hat{\eta}_{ij} = \eta_{ij} + \xi_{ij} \tag{3.70}$$

$$\hat{v}_{ik} = v_{ik} + \gamma_{ik} \tag{3.71}$$

$$\hat{w}_{ik} = w_{ik} + \gamma_{ik} \tag{3.72}$$

and

$$\hat{\alpha}_{ik} = \alpha_{ik} + \gamma_{ik}. \tag{3.73}$$

MAP and MSE parameter estimates

From the posterior distributions, MAP and MSE parameter estimates can be made. For the Gaussian mean distribution, the mean and the mode of the posterior distributions are

the same and the MAP and MSE parameters are given by $\hat{\mathbf{m}}_{ik}$ (Equation 3.68)

$$\boldsymbol{\mu}_{ik\,\mathrm{MAP}} = \boldsymbol{\mu}_{ik\,\mathrm{MSE}} = \frac{w_{ik}\mathbf{m}_{ik} + \gamma_{ik}\bar{\mathbf{x}}_{ik}}{w_{ik} + \gamma_{ik}}. \tag{3.74}$$

For the Dirichlet and Wishart distributions the mean and mode differs. The MAP parameters are given by (see Equations 3.56, 3.51 and 3.44)

$$a_{ij\,\mathrm{MAP}} = \frac{\eta_{ij} + \xi_{ij} - 1}{\sum_{l=1}^{K}(\eta_{il} + \xi_{il} - 1)} \tag{3.75}$$

$$c_{ik\,\mathrm{MAP}} = \frac{v_{ik} + \gamma_{ik} - 1}{\sum_{l=1}^{K}(v_{il} + \gamma_{il} - 1)} \tag{3.76}$$

$$\mathbf{R}^{-1}_{ik\,\mathrm{MAP}} = \frac{\boldsymbol{\Upsilon}_{ik} + \sum_{t=1}^{T}\gamma_{ik}(t)(\hat{\mathbf{m}}_{ik} - \mathbf{x}_t)(\hat{\mathbf{m}}_{ik} - \mathbf{x}_t)^T + w_{ik}(\hat{\mathbf{m}}_{ik} - \mathbf{m}_{ik})(\hat{\mathbf{m}}_{ik} - \mathbf{m}_{ik})^T}{\alpha_{ik} + \gamma_{ik} - D}, \tag{3.77}$$

and the MSE parameters are given by (see Equations 3.57, 3.52 and 3.43)

$$a_{ij\,\mathrm{MSE}} = \frac{\eta_{ij} + \xi_{ij}}{\sum_{l=1}^{K}(\eta_{il} + \xi_{il})} \tag{3.78}$$

$$c_{ik\,\mathrm{MSE}} = \frac{v_{ik} + \gamma_{ik}}{\sum_{l=1}^{K}(v_{il} + \gamma_{il})} \tag{3.79}$$

$$\mathbf{R}^{-1}_{ik\,\mathrm{MSE}} = \frac{\boldsymbol{\Upsilon}_{ik} + \sum_{t=1}^{T}\gamma_{ik}(t)(\hat{\mathbf{m}}_{ik} - \mathbf{x}_t)(\hat{\mathbf{m}}_{ik} - \mathbf{x}_t)^T + w_{ik}(\hat{\mathbf{m}}_{ik} - \mathbf{m}_{ik})(\hat{\mathbf{m}}_{ik} - \mathbf{m}_{ik})^T}{\alpha_{ik} + \gamma_{ik}}. \tag{3.80}$$

It is apparent that the MAP estimates (Equations 3.75-3.77) are invalid under certain conditions ($\eta_{ij} + \xi_{ij} < 1$, $v_{ik} + \gamma_{ik} < 1$ and $\alpha_{ik} + \gamma_{ik} \leq D$). This is because the mode of the posterior distribution is undefined under these conditions. The MSE estimates do not suffer from this problem though.

The MAP and MSE estimates of Equations 3.74-3.80 have been derived based on the complete data assumption, i.e. that state and mixture alignment information is available. In practice, this information has to be computed from the adaptation data. The next section discusses an iterative estimation technique for the incomplete data scenario where

state and mixture occupancy is not observed and also generalises the results of this section to include all possible state and mixture sequences.

3.2.4 Estimation algorithm

Gauvain & Lee [61] propose using an expectation maximisation (EM) [41] estimation strategy for MAP parameter estimation. The proposed strategy is based on the maximisation of the auxiliary function $R(\boldsymbol{\lambda}, \hat{\boldsymbol{\lambda}})$, representing the *expectation* of the complete data posterior model log-likelihood ($\log[f(\mathbf{X}, \mathbf{q}, \mathbf{l}|\hat{\boldsymbol{\lambda}})g(\hat{\boldsymbol{\lambda}})]$)

$$\begin{aligned} R(\boldsymbol{\lambda}, \hat{\boldsymbol{\lambda}}) &= E\big[\log[f(\mathbf{X}, \mathbf{q}, \mathbf{l}|\hat{\boldsymbol{\lambda}})g(\hat{\boldsymbol{\lambda}})|\mathbf{X}, \boldsymbol{\lambda}]\big] \\ &= E\big[\log[f(\mathbf{X}, \mathbf{q}, \mathbf{l}|\hat{\boldsymbol{\lambda}})|\mathbf{X}, \boldsymbol{\lambda}]\big] + \log g(\hat{\boldsymbol{\lambda}}) \\ &= Q(\boldsymbol{\lambda}, \hat{\boldsymbol{\lambda}}) + \log g(\hat{\boldsymbol{\lambda}}), \end{aligned} \tag{3.81}$$

given the observations $\mathbf{X}$, a current model $\boldsymbol{\lambda}$ and where $Q(\boldsymbol{\lambda}, \hat{\boldsymbol{\lambda}})$ is the auxiliary equation for conventional Gaussian mixture ML procedures and is given by

$$Q(\boldsymbol{\lambda}, \hat{\boldsymbol{\lambda}}) = \frac{1}{f(\mathbf{X}|\boldsymbol{\lambda})} \sum_{\mathbf{q}} \sum_{\mathbf{l}} f(\mathbf{X}, \mathbf{q}, \mathbf{l}|\boldsymbol{\lambda}) \log f(\mathbf{X}, \mathbf{q}, \mathbf{l}|\hat{\boldsymbol{\lambda}}). \tag{3.82}$$

Similar to maximising $Q(\boldsymbol{\lambda}, \hat{\boldsymbol{\lambda}})$ (see [62]), maximising $R(\boldsymbol{\lambda}, \hat{\boldsymbol{\lambda}})$ in each iteration, $R(\boldsymbol{\lambda}, \hat{\boldsymbol{\lambda}}) > R(\boldsymbol{\lambda}, \boldsymbol{\lambda})$ implies a monotonic increase in posterior likelihood $f(\mathbf{X}|\hat{\boldsymbol{\lambda}})g(\hat{\boldsymbol{\lambda}}) > f(\mathbf{X}|\boldsymbol{\lambda})g(\boldsymbol{\lambda})$ until $\hat{\boldsymbol{\lambda}}$ reaches a critical point where $f(\mathbf{X}|\hat{\boldsymbol{\lambda}})$ attains a local maximum. Maximisation of $R(\boldsymbol{\lambda}, \hat{\boldsymbol{\lambda}})$ according to the procedure defined by [61] leads to exactly the re-estimation equations derived in the previous section (Equations 3.74-3.77), as we shall show shortly. The auxiliary function $Q(\boldsymbol{\lambda}, \hat{\boldsymbol{\lambda}})$ can be expanded (following [63, p. 9]):

$$\begin{aligned} Q(\boldsymbol{\lambda}, \hat{\boldsymbol{\lambda}}) &= \frac{1}{f(\mathbf{X}|\boldsymbol{\lambda})} \sum_{\mathbf{q}} \sum_{\mathbf{l}} f(\mathbf{X}, \mathbf{q}, \mathbf{l}|\boldsymbol{\lambda}) \Big[\sum_{t=1}^{T} \log a_{q_{t-1}q_t} + \sum_{t=1}^{T} \log c_{q_t l_t} + \sum_{t=1}^{T} \log \mathcal{N}[\mathbf{x}_t, \boldsymbol{\mu}_{q_t l_t}, \mathbf{R}_{q_t l_t}] \Big] \\ &= \sum_{i=0}^{N} Q_{a_i}[\boldsymbol{\lambda}, \{a_{ij}\}_{j=1}^{N}] + \sum_{i=1}^{N} Q_{c_i}[\boldsymbol{\lambda}, \{c_{ik}\}_{k=1}^{K}] + \sum_{i=1}^{N} \sum_{k=1}^{K} Q_{\mathcal{N}}[\boldsymbol{\lambda}, \boldsymbol{\mu}_{ik}, \mathbf{R}_{ik}], \end{aligned} \tag{3.83}$$

where

$$
\begin{aligned}
Q_{a_i}[\boldsymbol{\lambda}, \{a_{ij}\}_{j=1}^{N}] &= \frac{1}{f(\mathbf{X}|\boldsymbol{\lambda})} \sum_{\mathbf{q}} \sum_{\mathbf{l}} \sum_{t=1}^{T} \sum_{j=1}^{N} f(\mathbf{X}, q_{t-1} = i, q_t = j, \mathbf{l}|\boldsymbol{\lambda}) \log a_{ij} \\
&= \sum_{t=1}^{T} \sum_{j=1}^{N} \xi_{ij}(t) \log a_{ij} \quad \text{(reversing order of summation and using Equation 2.13)} \\
&= \sum_{j=1}^{N} \xi_{ij} \log a_{ij} \quad \text{(using Equation 3.62),}
\end{aligned}
\tag{3.84}
$$

$$
\begin{aligned}
Q_{c_i}[\boldsymbol{\lambda}, \{c_{ik}\}_{k=1}^{K}] &= \frac{1}{f(\mathbf{X}|\boldsymbol{\lambda})} \sum_{\mathbf{q}} \sum_{\mathbf{l}} \sum_{k=1}^{K} \sum_{t=1}^{T} f(\mathbf{X}, q_t = i, l_t = k|\boldsymbol{\lambda}) \log c_{ik} \\
&= \sum_{k=1}^{K} \sum_{t=1}^{T} \gamma_{ik}(t) \log c_{ik} \quad \text{(reversing order of summation and using Equation 2.12)} \\
&= \sum_{k=1}^{K} \gamma_{ik} \log c_{ik} \quad \text{(using Equation 3.61),}
\end{aligned}
\tag{3.85}
$$

and

$$
\begin{aligned}
Q_{\mathcal{N}}[\boldsymbol{\lambda}, \boldsymbol{\mu}_{ik}, \mathbf{R}_{ik}] &= \frac{1}{f(\mathbf{X}|\boldsymbol{\lambda})} \sum_{\mathbf{q}} \sum_{\mathbf{l}} \sum_{t=1}^{T} f(\mathbf{X}, q_t = i, l_t = k|\boldsymbol{\lambda}) \log \mathcal{N}[\mathbf{x}_t, \boldsymbol{\mu}_{ik}, \mathbf{R}_{ik}] \\
&\propto \sum_{t=1}^{T} \gamma_{ik}(t) \log \left[|\mathbf{R}_{ik}|^{1/2} e^{-(1/2)(\boldsymbol{\mu}_{ik} - \mathbf{x}_t)^T \mathbf{R}_{ik} (\boldsymbol{\mu}_{ik} - \mathbf{x}_t)} \right] \\
&\quad \text{(reversing order of summation and using Equation 2.12)} \\
&\propto \gamma_{ik} \log \left[|\mathbf{R}_{ik}|^{1/2} e^{-(1/2)[\mathrm{tr}(\mathbf{S}_{ik}\mathbf{R}_{ik}) + (\boldsymbol{\mu}_{ik} - \bar{\mathbf{x}}_{ik})^T \mathbf{R}_{ik} (\boldsymbol{\mu}_{ik} - \bar{\mathbf{x}}_{ik})]} \right] \\
&\quad \text{(following Equation 3.37).}
\end{aligned}
\tag{3.86}
$$

If we consider maximising $\Psi(\boldsymbol{\lambda}, \hat{\boldsymbol{\lambda}}) = e^{R(\boldsymbol{\lambda}, \hat{\boldsymbol{\lambda}})}$, we get

$$\begin{aligned}
\Psi(\boldsymbol{\lambda}, \hat{\boldsymbol{\lambda}}) &= e^{Q(\boldsymbol{\lambda}, \hat{\boldsymbol{\lambda}}) + \log g(\hat{\boldsymbol{\lambda}})} \\
&= g(\hat{\boldsymbol{\lambda}}) e^{\sum_{i=0}^{N} Q_a[\boldsymbol{\lambda}, \{a_{ij}\}_{j=1}^{N}] + \sum_{i=1}^{N} Q_{c_i}[\boldsymbol{\lambda}, \{c_{ik}\}_{k=1}^{K}] + \sum_{i=1}^{N} \sum_{k=1}^{K} Q_{\mathcal{N}}[\boldsymbol{\lambda}, \boldsymbol{\mu}_{ik}, \mathbf{R}_{ik}]} \\
&\propto g(\hat{\boldsymbol{\lambda}}) \left[\prod_{i=0}^{N} e^{Q_a[\boldsymbol{\lambda}, \{a_{ij}\}_{j=1}^{N}]} \right] \left[\prod_{i=1}^{N} e^{Q_{c_i}[\boldsymbol{\lambda}, \{c_{ik}\}_{k=1}^{K}]} \right] \left[\prod_{i=1}^{N} \prod_{k=1}^{K} e^{Q_{\mathcal{N}}[\boldsymbol{\lambda}, \boldsymbol{\mu}_{ik}, \mathbf{R}_{ik}]} \right] \\
&\propto g(\hat{\boldsymbol{\lambda}}) \prod_{i=1}^{N} \left[a_{0i}^{\xi_{0i}} \left[\prod_{j=1}^{N} a_{ij}^{\xi_{ij}} \right] \left[\prod_{k=1}^{K} c_{ik}^{\gamma_{ik}} |\mathbf{R}_{ik}|^{\gamma_{ik}/2} e^{-(\gamma_{ik}/2)[\mathrm{tr}(\mathbf{S}_{ik}\mathbf{R}_{ik}) + (\boldsymbol{\mu}_{ik} - \bar{\mathbf{x}}_{ik})^T \mathbf{R}_{ik} (\boldsymbol{\mu}_{ik} - \bar{\mathbf{x}}_{ik})]} \right] \right],
\end{aligned} \tag{3.87}$$

which is of exactly the same form as the joint posterior distribution for the complete data density given in Equation 3.66 and therefore maximisation of Equation 3.87 leads to the MAP estimation equations derived in the previous section (Equations 3.74-3.77). Use of the auxiliary function in the derivation ensures that the likelihood of the MAP estimates monotonically increase in every iteration. Unfortunately this theoretical result does not apply for the MSE estimates. Since the maximisation of the auxiliary function is done for arbitrary unknown state and mixture alignment, either of the two main methods for iterative estimation of HMM parameters, namely the segmental and forward-backward methods of Chapter 2 can be used to calculate the sufficient statistics for the approximation of the posterior parameters. For computational efficiency we select to use the *segmental adaptation* method to locally maximise $f(\mathbf{X}, \mathbf{q}|\boldsymbol{\lambda})g(\boldsymbol{\lambda})$, but we could also have used the more general solution offered by the *forward-backward* adaptation algorithm to locally maximise $f(\mathbf{X}|\boldsymbol{\lambda})g(\boldsymbol{\lambda})$, as was assumed in the derivation of the maximisation of the auxiliary function $R(\boldsymbol{\lambda}, \hat{\boldsymbol{\lambda}})$.

In the implementation of the segmental Bayesian adaptation algorithm, the Viterbi algorithm is used to compute the state alignment $(\bar{\mathbf{q}}(n))$ in iteration n of the observations with the current model estimate:

$$\bar{\mathbf{q}}(n) = \arg\max_{\mathbf{q}} f(\mathbf{X}, \mathbf{q}|\boldsymbol{\lambda}(n)). \tag{3.88}$$

The state alignment in iteration n is used, in turn, to estimate the statistics of Equations 3.61-3.64 and the MAP parameters of iteration $n+1$, as described by

$$\boldsymbol{\lambda}(n+1) = \arg\max_{\boldsymbol{\lambda}} f(\mathbf{X}, \bar{\mathbf{q}}(n)|\boldsymbol{\lambda})g(\boldsymbol{\lambda}), \tag{3.89}$$

where $\boldsymbol{\lambda}(0)$ is initialised to the model estimate when no data is observed, which is usually just the model that was used to seed the prior distribution. When applying the segmental Bayesian algorithm for speaker adaptation, use of only a single iteration may suffice, but we expect that for cross-language adaptation a relatively large number of iterations may be necessary, especially if there is a large mismatch between source and target data distributions. When a large number of iterations take place, unobserved model mixtures (mixtures with very low output probabilities) may converge to feature space regions where they contribute to the *a posteriori* probability function and are therefore adapted. We now turn our attention to the determination of the parameters of the prior distribution.

3.2.5 Prior density estimation

Section 3.2.2 discussed a method (from [24]) for prior density estimation for the mean and variance (or precision) parameters of a univariate Gaussian (Equations 3.31-3.33) and a multivariate Gaussian (Equations 3.45-3.47). The discussion centred around a way of using speaker independent Gaussian mixture models to estimate a normal-Gamma (univariate) and normal-Wishart (multivariate) prior distribution for the mean and variance of observations from the Gaussian observation distribution. One may apply this approach directly for Gaussian mixture observation distributions, but it would imply use of an identical prior distribution for every mixture. Another way of estimating parameters for the prior distribution is to set the prior mode equal to the parameters of a given HMM [61], typically an HMM trained on speaker independent data. The prior distribution, however, contains five parameters (v_{ik}, $\mathbf{m}_{ik}$, w_{ik}, α_{ik} and $\boldsymbol{\Upsilon}_{ik}$) for each mixture, while only three parameters ($\tilde{c}_{ik}$, $\tilde{\mathbf{m}}_{ik}$ and $\tilde{\mathbf{r}}_{ik}$) are associated with each mixture of the speaker independent HMM, essentially implying that we are unable to estimate the variance of the prior mean and, similar to the

other approaches, that our estimate of the mean and the variance of the prior precision are dependent.

An elegant solution [61] can be found by limiting the family of the prior distribution to that of the kernel density of the complete-data likelihood. The prior family is expressed as a joint Dirichlet-normal-Wishart distribution (Equation 3.58) while the complete data likelihood function (Equation 3.65) is a *dependent* Dirichlet-normal-Wishart function. Element-wise comparison of the two equations delivers the following correspondence

$$\eta_{ij} - 1 \leftrightarrow \xi_{ij} \tag{3.90}$$

$$v_{ik} - 1 \leftrightarrow \gamma_{ik} \tag{3.91}$$

$$\alpha_{ik} - D \leftrightarrow \gamma_{ik} \tag{3.92}$$

$$w_{ik} \leftrightarrow \gamma_{ik}. \tag{3.93}$$

By selecting to retain η_{ij} and w_{ik}, the other two parameters of the prior distribution, namely v_{ik} and α_{ik}, can be written in terms of w_{ik} by

$$v_{ik} = w_{ik} + 1 \tag{3.94}$$

$$\alpha_{ik} = w_{ik} + D. \tag{3.95}$$

This reduction of the prior renders it of the same distribution family as the complete data likelihood function and the remaining parameters can then be estimated directly from the seed model parameters by using the prior transition probability

$$\eta_{ij} = \tilde{a}_{ij}, \tag{3.96}$$

the prior mixture weight value

$$w_{ik} = \tilde{c}_{ik}, \tag{3.97}$$

and similar to Equations 3.45 and 3.47:

$$\mathbf{m}_{ik} = \tilde{\mathbf{m}}_{ik}, \tag{3.98}$$

$$\frac{\boldsymbol{\Upsilon}_{ik}}{w_{ik} + D} = \tilde{\boldsymbol{\Sigma}}_{ik}. \tag{3.99}$$

To evaluate the meaningfulness of these choices we rewrite the affected posterior parameter estimates. The parameter reductions of Equations 3.94 and 3.95 are applied, as well as the choice of prior seed values (Equations 3.96-3.99) for the Gaussian mean estimates (from Equation 3.74)

$$\boldsymbol{\mu}_{ik\,\mathrm{MAP}} = \boldsymbol{\mu}_{ik\,\mathrm{MSE}} = \frac{\tilde{c}_{ik}\tilde{\mathbf{m}}_{ik} + \gamma_{ik}\bar{\mathbf{x}}_{ik}}{\tilde{c}_{ik} + \gamma_{ik}}, \tag{3.100}$$

for the MAP parameters (from Equations 3.75-3.77)

$$a_{ij\,\mathrm{MAP}} = \frac{\tilde{a}_{ij} + \xi_{ij} - 1}{\sum_{l=1}^{K}(\tilde{a}_{il} + \xi_{il} - 1)} \tag{3.101}$$

$$c_{ik\,\mathrm{MAP}} = \frac{\tilde{c}_{ik} + \gamma_{ik}}{\sum_{l=1}^{K}(\tilde{c}_{il} + \gamma_{il})} \tag{3.102}$$

$$\mathrm{R}^{-1}_{ik\,\mathrm{MAP}} = \frac{(\tilde{c}_{ik} + D)\tilde{\boldsymbol{\Sigma}}_{ik} + \sum_{t=1}^{T}\gamma_{ik}(t)(\hat{\mathbf{m}}_{ik} - \mathbf{x}_t)(\hat{\mathbf{m}}_{ik} - \mathbf{x}_t)^T + \tilde{c}_{ik}(\hat{\mathbf{m}}_{ik} - \tilde{\mathbf{m}}_{ik})(\hat{\mathbf{m}}_{ik} - \tilde{\mathbf{m}}_{ik})^T}{\tilde{c}_{ik} + \gamma_{ik}}, \tag{3.103}$$

and for the MSE parameters (from Equations 3.78-3.80)

$$a_{ij\,\mathrm{MSE}} = \frac{\tilde{a}_{ij} + \xi_{ij}}{\sum_{l=1}^{K}(\tilde{a}_{il} + \xi_{il})} \tag{3.104}$$

$$c_{ik\,\mathrm{MSE}} = \frac{\tilde{c}_{ik} + \gamma_{ik} + 1}{\sum_{l=1}^{K}(\tilde{c}_{il} + \gamma_{il} + 1)} \tag{3.105}$$

$$\mathrm{R}^{-1}_{ik\,\mathrm{MSE}} = \frac{(\tilde{c}_{ik} + D)\tilde{\boldsymbol{\Sigma}}_{ik} + \sum_{t=1}^{T}\gamma_{ik}(t)(\hat{\mathbf{m}}_{ik} - \mathbf{x}_t)(\hat{\mathbf{m}}_{ik} - \mathbf{x}_t)^T + \tilde{c}_{ik}(\hat{\mathbf{m}}_{ik} - \tilde{\mathbf{m}}_{ik})(\hat{\mathbf{m}}_{ik} - \tilde{\mathbf{m}}_{ik})^T}{\tilde{c}_{ik} + \gamma_{ik} + D}. \tag{3.106}$$

Although the parameter reduction has produced a MAP estimate that is defined for all valid prior parameter values, an artifact of the seeding is that the MAP variance estimate

(Equation 3.103) is not equal to the prior variance when no observations are available. We propose to remedy this by seeding the mode Υ_{ik}/w_{ik} (in place of the mean as in Equation 3.99) of the variance prior. This results in an elegant formula for the MAP variance estimate which is independent of the feature dimension D and is given by

$$\mathrm{R}^{-1}_{ik\,\mathrm{MAP}} = \frac{\tilde{c}_{ik}\tilde{\boldsymbol{\Sigma}}_{ik} + \sum_{t=1}^{T}\gamma_{ik}(t)(\hat{\mathbf{m}}_{ik} - \mathbf{x}_t)(\hat{\mathbf{m}}_{ik} - \mathbf{x}_t)^T + \tilde{c}_{ik}(\hat{\mathbf{m}}_{ik} - \tilde{\mathbf{m}}_{ik})(\hat{\mathbf{m}}_{ik} - \tilde{\mathbf{m}}_{ik})^T}{\tilde{c}_{ik} + \gamma_{ik}}. \tag{3.107}$$

Examination of the posterior mean estimate (Equation 3.100) and the posterior variance estimates (Equations 3.103, 3.106 and 3.107) shows that $\tilde{c}_{ik}$ can be interpreted as a prior weighting factor associated with the kth mixture of state i. When $\tilde{c}_{ik}$ is large the mean and variance prior densities are sharply peaked around the values used for seeding the prior and less adaptation occurs than when $\tilde{c}_{ik}$ is small. This choice implies that we expect the weight associated with a mixture to express the confidence associated with the mixture, which makes intuitive sense. While the choice of seed value (Equation 3.97) makes sense, it leads to prior weight values in the range $[0, 1]$, which in Equations 3.100-3.107 implies that the weight associated with the prior distribution is less than that associated with a single observation frame. The prior weight $\tilde{c}_{ik}$ assigned to the prior distribution for each mixture is therefore multiplied by a global prior weight scaling factor ϖ. Unfortunately, the optimal value of ϖ cannot be determined easily from a small amount of training data, since it needs to be evaluated on independent data (target data not used for adaptation). We do not follow a cross-validation approach, but in experiments (Chapters 6 and 7) rather explicitly show the effect of the prior weight scaling factor on recognition performance. More detailed aspects of the application of Bayesian techniques for cross-language adaptation are covered in Chapter 5.

The Bayesian framework for estimation that we discussed in this section focussed heavily on the use of existing knowledge when facing the design of a new system, or when changing a system based on new observations. In the next section we discuss methods that attempt to exploit correlation between parameters when changing a current model to better reflect

the characteristics of a new sample.

3.3 Transformation-based adaptation

Transformation-based techniques estimate a transformation of model parameters using a limited amount of observation data. A linear transformation of model parameters is usually computed and applied to an existing model for the model to better reflect the characteristics of the observations. Non-linear transformations, such as those implemented with multi-layer perceptrons (MLPs), have also been applied for the transformation of model parameters.

The motivation for using transformation-based adaptation, versus say Bayesian adaptation, is that if the changes in the observation characteristics can be approximated well enough by a simple parameter transformation, then only the parameters of the transformation have to be estimated which will typically be far fewer than those of the model being transformed. Parameters of unobserved distributions are adapted by implementing the same transformation for all the parameters or for groups of parameters and rapid adaptation can thus be achieved on little target data. When a reasonably large amount of adaptation data is available, such as for our application of cross-language adaptation, transformation-based adaptation does not automatically guarantee asymptotic behaviour with respect to a language dependent system.

The transformation approach can be applied at the feature or at the model level. When applied at the feature level, it is referred to as feature space adaptation or *spectral* transformation [54]. Feature space transformation can be implemented as part of the pre-processing stage of a system, transforming incoming speech from a new speaker to better match that of a reference speaker or speakers - thus normalising the speech of the new speaker with respect to the reference. Feature space transformation can also be used to perform compensation for spectral mismatch of recording conditions and channel effects between training and testing environments. When the transformation is implemented on cepstral features, as

is usually done, a linear process in the frequency domain can be implemented (or counteracted) with a simple offset in the cepstral domain. Frequency warping or other non-linear frequency domain processes can be approximately implemented or counteracted with full transformations of the cepstral features. Feature space transformations have been used to perform phone-specific transformations to some degree by estimating several transformations across the entire feature space and implementing transformation of specific features using fuzzy class membership rules [64].

Model space transformations are generally accepted [65] to deliver better performance than feature space transformations since different transformations can be estimated for different phonetic groupings and also other parameters, such as Gaussian variance, can be transformed separately from the Gaussian mean parameters. Model space transformations can make better use of available data than feature space transformations by estimating few transformations when little adaptation data is available and estimating many transformations when a large amount of adaptation data is available.

An application of feature space transformation that is promising is the use of transformation to normalise speech from the training speakers with respect to some reference and then to retrain the models [66]. This approach is related to data augmentation, which transforms speech data from speakers close to the target speaker and subsequently performs retraining of models [67]. We discuss these methods in the context of using them for cross-language data augmentation, i.e. performing cross-language transformation and subsequent retraining. We now proceed to discuss the method most commonly used for transformation-based adaptation namely the linear transform.

3.3.1 Linear transformation of the Gaussian mean

Linear transformation of the Gaussian mean model parameters using target data attempts to improve the match between the model and target data through correlation between the distribution the model represents and the distribution of the target data. The Gaussian

mean parameters are usually transformed since they specify positions in feature space that represent nuclei of the model distribution and can thus be directly compared with target data distributions. Transformation-based adaptation is usually performed with a linear transformation because it is well understood and leads to simple implementation. When a linear transformation $\mathbf{y} = \mathbf{W}\mathbf{x}$ from parameters or observations $\mathbf{X} = \{\mathbf{x}_1, ..., \mathbf{x}_T\}$ to parameters or observations $\mathbf{Y} = \{\mathbf{y}_1, ..., \mathbf{y}_T\}$ is estimated, the squared error is given by

$$E = \sum_{t=1}^{T} (\mathbf{y}_t - \mathbf{W}\mathbf{x}_t)^T (\mathbf{y}_t - \mathbf{W}\mathbf{x}_t) = \mathrm{tr}\big[(\mathbf{Y} - \mathbf{W}\mathbf{X})(\mathbf{Y} - \mathbf{W}\mathbf{X})^T\big] \tag{3.108}$$

and the minimum squared error (MSE) solution is found using the pseudo inverse form for the transformation matrix

$$\mathbf{W} = \mathbf{Y}\mathbf{X}^T(\mathbf{X}\mathbf{X}^T)^{-1} \tag{3.109}$$

which is given in transpose form by

$$\mathbf{W}^T = (\mathbf{X}\mathbf{X}^T)^{-1}\mathbf{X}\mathbf{Y}^T \tag{3.110}$$

and for the transpose of row l of $\mathbf{W}$ by

$$\mathbf{w}_l^T = (\mathbf{X}\mathbf{X}^T)^{-1}\mathbf{X}\mathbf{y}_l^T \tag{3.111}$$

for comparison with later equations, where $\mathbf{y}_l$ is the lth row of $\mathbf{Y}$ (not to be confused with a column $\mathbf{y}_t$ of $\mathbf{Y}$). Least squares linear regression has been used for estimation of feature space transformations [54], as well as for model adaptation by estimating transformations of parameters of CDHMM [68]. Cox [69] also used regression to estimate linear transformation of individual sound classes, exploiting correlation between classes. The most popular approach for estimating linear transformations is related to the least squares estimate and is discussed next.

Maximum likelihood linear regression

A maximum likelihood-based approach for linear transformation, termed maximum likelihood linear regression (MLLR), was proposed by Leggetter and Woodland [63, 27]. In the Gaussian mixture density HMM framework, MLLR estimates the linear transformation of the Gaussian means

$$\hat{\boldsymbol{\mu}}_{jk} = \mathbf{W}\boldsymbol{\mu}_{jk} \tag{3.112}$$

that maximises the likelihood $f(\mathbf{X}|\hat{\boldsymbol{\lambda}})$ of the observations given the transformed model

$$\hat{\boldsymbol{\lambda}} = \{A, (c_{jk}, \mathbf{W}\boldsymbol{\mu}_{jk}, \mathbf{R}_{jk})_{j=1,k=1}^{N,K}\}. \tag{3.113}$$

The transformation matrices can be found by maximising the auxiliary function

$$Q(\boldsymbol{\lambda}, \hat{\boldsymbol{\lambda}}) = \sum_{\mathbf{q}} f(\mathbf{X}, \mathbf{q}|\boldsymbol{\lambda}) \log(f(\mathbf{X}, \mathbf{q}|\hat{\boldsymbol{\lambda}})) \tag{3.114}$$

with respect to $\mathbf{W}$ where $\hat{\boldsymbol{\lambda}}$ is the transformed model of Equation 3.113. Using the transformed model in the expansion of the auxiliary equation (Equation 3.86) delivers

$$\begin{aligned} Q_{\mathcal{N}}[\boldsymbol{\lambda}, \mathbf{W}\boldsymbol{\mu}_{ik}, \mathbf{R}_{ik}] &\propto \gamma_{ik} \log\left[|\mathbf{R}_{ik}|^{1/2} e^{-(1/2)[\mathrm{tr}(\mathbf{S}_{ik}\mathbf{R}_{ik}) + (\mathbf{W}\boldsymbol{\mu}_{ik} - \bar{\mathbf{x}}_{ik})^T \mathbf{R}_{ik}(\mathbf{W}\boldsymbol{\mu}_{ik} - \bar{\mathbf{x}}_{ik})]}\right] \\ &\propto \gamma_{ik}\left[\frac{1}{2}\log|\mathbf{R}_{ik}| - \frac{1}{2}\mathrm{tr}(\mathbf{S}_{ik}\mathbf{R}_{ik}) - \frac{1}{2}(\mathbf{W}\boldsymbol{\mu}_{ik} - \bar{\mathbf{x}}_{ik})^T \mathbf{R}_{ik}(\mathbf{W}\boldsymbol{\mu}_{ik} - \bar{\mathbf{x}}_{ik})\right] \end{aligned} \tag{3.115}$$

To maximise $Q(\boldsymbol{\lambda}, \hat{\boldsymbol{\lambda}})$, its derivative w.r.t. $\mathbf{W}$ is computed and equated to zero, i.e.

$$\begin{aligned} \frac{dQ(\boldsymbol{\lambda}, \hat{\boldsymbol{\lambda}})}{d\mathbf{W}} &= \frac{d}{d\mathbf{W}} \sum_{i=1}^{N}\sum_{k=1}^{K} Q_{\mathcal{N}}[\boldsymbol{\lambda}, \mathbf{W}\boldsymbol{\mu}_{ik}, \mathbf{R}_{ik}] \\ &= \sum_{i=1}^{N}\sum_{k=1}^{K} \gamma_{ik} \frac{d}{d\mathbf{W}}\left[\frac{1}{2}\log|\mathbf{R}_{ik}| - \frac{1}{2}\mathrm{tr}(\mathbf{S}_{ik}\mathbf{R}_{ik}) - \frac{1}{2}(\mathbf{W}\boldsymbol{\mu}_{ik} - \bar{\mathbf{x}}_{ik})^T \mathbf{R}_{ik}(\mathbf{W}\boldsymbol{\mu}_{ik} - \bar{\mathbf{x}}_{ik})\right] \\ &= \sum_{i=1}^{N}\sum_{k=1}^{K} \gamma_{ik}\mathbf{R}_{ik}(\mathbf{W}\boldsymbol{\mu}_{ik} - \bar{\mathbf{x}}_{ik})\boldsymbol{\mu}_{ik}^T = 0, \end{aligned} \tag{3.116}$$

which delivers

$$\sum_{i=1}^{N}\sum_{k=1}^{K}\gamma_{ik}\mathbf{R}_{ik}\mathbf{W}\boldsymbol{\mu}_{ik}\boldsymbol{\mu}_{ik}^{T}=\sum_{i=1}^{N}\sum_{k=1}^{K}\gamma_{ik}\mathbf{R}_{ik}\bar{\mathbf{x}}_{ik}\boldsymbol{\mu}_{ik}^{T}. \tag{3.117}$$

For a diagonal covariance matrix (and thus diagonal precision also), the lth row on both sides of Equation 3.117 is given by

$$\mathbf{w}_l\sum_{i=1}^{N}\sum_{k=1}^{K}\gamma_{ik}r_{ikl}\boldsymbol{\mu}_{ik}\boldsymbol{\mu}_{ik}^{T}=\sum_{i=1}^{N}\sum_{k=1}^{K}\gamma_{ik}r_{ikl}\bar{x}_{ikl}\boldsymbol{\mu}_{ik}^{T} \tag{3.118}$$

and we therefore find that the maximum likelihood estimate of the mean transformation matrix $\mathbf{W}$ can be expressed in a much simpler format than in the original publications [63, 27] by the expression

$$\mathbf{w}_l^{T}=\Big[\sum_{i=1}^{N}\sum_{k=1}^{K}\gamma_{ik}r_{ikl}\boldsymbol{\mu}_{ik}\boldsymbol{\mu}_{ik}^{T}\Big]^{-1}\Big[\sum_{i=1}^{N}\sum_{k=1}^{K}\gamma_{ik}r_{ikl}\bar{x}_{ikl}\boldsymbol{\mu}_{ik}\Big] \tag{3.119}$$

for the lth row of $\mathbf{W}$. Equation 3.119 also clearly shows the relationship between the MLLR estimate and the MSE transformation estimate of Equation 3.111. The MLLR estimate is simply an MSE estimate that weights the contribution of each mixture component to the pseudo inverse with the amount of data associated with the mixture (γ_{ik}) multiplied by the precision of the mixture component separately for each feature dimension (r_{ikl}). The MLLR estimate can be written in the exact form of an MSE estimate (Equation 3.109) with $D\times KN$ dimensional matrices $\mathbf{X}$ and $\mathbf{Y}$, with the $(i\times k)$th column of $\mathbf{X}$ given by $(\gamma_{ik}r_{ikl})^{1/2}\boldsymbol{\mu}_{ik}$ and the $(i\times k)$th column of $\mathbf{Y}$ given by $(\gamma_{ik}r_{ikl})^{1/2}\bar{\mathbf{x}}_{ik}$.

If an offset term is added to the multiplicative term to make the transformation more general, the transformation of the kth Gaussian mean in the ith state can be expressed as the transformation

$$\hat{\boldsymbol{\mu}}_{ik}=\mathbf{W}\boldsymbol{v}_{ik} \tag{3.120}$$

of the extended mean vector $\boldsymbol{v}_{ik}=[\mu_{ik1},...,\mu_{ikD},1]^{T}$ by a $D\times(D+1)$ matrix $\mathbf{W}$. Closely

following Equation 3.119, but sharing the transformation $\mathbf{W}_s$ across the K components of an arbitrary set of R states $\{s_1, ..., s_R\}$, the lth row of the transformation matrix $\mathbf{W}$ is given by

$$\mathbf{w}_{s_l}^T = \Big[\sum_{r=1}^{R}\sum_{k=1}^{K}\gamma_{s_rk}\tau_{s_rkl}\boldsymbol{\upsilon}_{s_rk}\boldsymbol{\upsilon}_{s_rk}^T\Big]^{-1}\Big[\sum_{r=1}^{R}\sum_{k=1}^{K}\gamma_{s_rk}\tau_{s_rkl}\bar{x}_{s_rkl}\boldsymbol{\upsilon}_{s_rk}\Big]. \tag{3.121}$$

Usually transformation-based adaptation is performed because there is too little data for re-estimation of parameters, thus necessitating the tieing of transformations across multiple states of multiple HMMs to obtain a robust estimates of the transformation. Equation 3.121 does not explicitly show tieing between states of different HMMs, but the group of states (called a regression class) tied in the transformation may be associated arbitrarily with different HMMs. The implementation of tieing used in this thesis groups together HMMs according to phonetic categories. A clustering algorithm may also be used to group together mixtures that are close to each other in feature space according to some metric [70]. When using phonetic groupings, the assumption is that sounds from the same categories undergo similar transforms, while the clustering approach assumes that mixtures that are closely located in feature space undergo similar transforms.

Inspection of Equation 3.121 reveals that the rank of the matrix that is inverted is less than or equal to the number of observed independent Gaussian mean vectors (at most RK). Since the matrix contains $D+1$ rows and columns, it follows that it is necessary that $RK \geq D+1$ for a non-singular matrix and thus for a unique solution to be found for $\mathbf{w}_{s_l}$. Writing Equation 3.121 in the familiar $\mathbf{Ax} = \mathbf{b}$ notation, it is apparent that $\mathbf{b}$ is within the column-space of $\mathbf{A}$ when the coefficients of the summation are not degenerate and therefore a solution exists, irrespective of the degree of mixture tieing. However, when $RK < D+1$, the solution is not unique and a range of values for $\mathbf{W}$ exist that exactly reproduce the maximum likelihood values for all the tied Gaussian means, i.e. the values obtained if mean-only training (re-estimation) is done on the adaptation data. The use of singular value decomposition is preferred in general for the solution of the least squares problem and may be used to determine a suitable transformation matrix $\mathbf{W}$, irrespective

of whether the matrix $\mathbf{A}$ is singular or not. The transformation may, however, not be very useful if it merely implements re-estimation of the means.

When little target data is available, re-estimation is particularly troublesome as the re-estimated means are likely to be inaccurate. In this case a value of $RK >> D+1$ is desired to tie the transformation across a large number of mixtures for accuracy. However, if the amount of available target data increases, less tieing, i.e. more regression classes and thus a smaller RK is desired so that transformations may group together more closely related mixtures. In the event of a very large amount of data being available, $RK \leq D+1$ (effectively re-estimation) does not present a problem and may even be desirable because accurate estimates can be made on the target data alone. This is a very important point to make since it indicates that MLLR can exhibit asymptotic behaviour (in terms of mean estimation) with respect to a system trained on target data only, if the number of regression classes is allowed to increase in relation to the amount of target data available.

Discussion of this aspect in the original MLLR paper [27] attributes poor performance in the extreme case of calculating a transformation of few tied mixtures using little data, to the accumulated matrices being close to singular and (matrix inversion) therefore causing computational errors. We feel that this is not the true reason for poor performance in the case of little data and few tied mixtures (many regression classes). Rather, as mentioned, the reason is that re-estimation on small amounts of target data is undesirable and therefore the inter-dependencies between more parameters should be shared in the transformation. Inspection of the condition of the matrices (from Equation 3.121) calculated in experiments in Chapters 6 and 7 also reveals that numerical accuracy is not of concern - also substantiated by the fact that the same results are obtained with Gauss-Jordan elimination (with full pivoting) than with a singular value decomposition-based approach.

Implementation of adaptation procedure

An iterative procedure is typically used to estimate the transformations, consisting of the following steps:

1. initialising current model estimates to trained source models,
2. computing sufficient statistics (Equations 2.12, 2.13 and 3.61-3.64) from target data using current model estimates and either Viterbi-alignment or forward-backward approaches,
3. computing the transformation for each regression class (Equation 3.121),
4. updating current model estimates (Equation 3.112) and
5. repeating the process from step 2 for a limited number of iterations or until convergence occurs.

The procedure usually converges within only a few iterations, but more iterations may be needed if the original source models match very poorly with the target data, which may be the case in particular for cross-language model transformation.

3.3.2 Variance transformation

A method for the transformation of both the Gaussian mean and variance parameters that is closely related to MLLR was suggested by Digalakis *et al.* [71]. The method computes the linear transformation of both Gaussian mean and variance parameters through the estimation of a transformation matrix $\mathbf{W}$ and an offset vector $\mathbf{b}$, yielding transformed Gaussian mean

$$\hat{\mu}_{jk} = \mathbf{W}\mu_{jk} + \mathbf{b} \tag{3.122}$$

and variance values

$$\hat{\Sigma}_{jk} = \mathbf{W}\Sigma_{jk}\mathbf{W}^T. \tag{3.123}$$

Unfortunately a closed form solution exists only for diagonal transformation matrices and therefore the transformation for each feature dimension is computed separately. The method has been found [65] not to perform as well as the standard MLLR approach, even though it also adapts the variance parameters, since it does not make use of dependencies between different feature dimensions. For this reason we did not pursue it further.

Maximum likelihood variance transformation

Another method for transforming both Gaussian mean and variance parameters, based on the extension of the MLLR adaptation framework, was proposed by Gales and Woodland [72]. Unlike the approach suggested by Digalakis *et al.* [71], the method optimises the mean and variance parameters in separate iterations, termed unconstrained transformation, thereby allowing a closed form solution for the ML variance transform estimate to be found. The Gaussian mean parameters are transformed in a first step using the standard MLLR approach discussed in the previous section (Equation 3.121). The Gaussian variance parameters are updated in a second step through

$$\hat{\Sigma}_{jk} = \mathbf{B}_{jk}^T\mathbf{H}\mathbf{B}_{jk}, \tag{3.124}$$

where $\mathbf{H}$ is the transformation to be estimated and $\mathbf{B}_{jk}$ is the inverse of the Choleski factor ($\mathbf{C}_{jk}$) of Σ_{jk}^{-1}, i.e.

$$\mathbf{B}_{jk} = \mathbf{C}_{jk}^{-1} \tag{3.125}$$

where

$$\boldsymbol{\Sigma}_{jk}^{-1} = \mathbf{C}_{jk}\mathbf{C}_{jk}^{T}. \tag{3.126}$$

The updated variance model $\bar{\boldsymbol{\lambda}}$ is given by

$$\bar{\boldsymbol{\lambda}} = \{\mathbf{A}, (c_{jk}, \hat{\boldsymbol{\mu}}_{jk}, \mathbf{B}_{jk}^{T}\mathbf{H}\mathbf{B}_{jk})_{j=1,k=1}^{N,K}\} \tag{3.127}$$

where $\hat{\boldsymbol{\mu}}_{jk}$ is the MLLR updated Gaussian mean estimate.

Similar to the MLLR derivation, the transformation matrix $\mathbf{H}$ can be found by performing the derivative of the auxiliary function $Q(\hat{\boldsymbol{\lambda}}, \bar{\boldsymbol{\lambda}})$ (where $\hat{\boldsymbol{\lambda}}$ represents the MLLR updated mean model obtained using Equation 3.121 and $\bar{\boldsymbol{\lambda}}$ the MLLR updated mean and variance model) with respect to $\mathbf{H}$ and finding the root of the equation. For a transformation H_s, shared by the K components of a set of R states $\{s_1, ..., s_R\}$, each associated with observation sequences of length T_{s_r}, the estimation of the tied variance transformation can be represented by [72]

$$\mathbf{H}_s = \frac{\sum_{r=1}^{R}\sum_{k=1}^{K}\left\{\mathbf{C}_{s_rk}^{T}\left[\sum_{t=1}^{T_{s_r}}\gamma_{s_rk}(t)(\mathbf{x}_{s_rt} - \hat{\boldsymbol{\mu}}_{s_rk})(\mathbf{x}_{s_rt} - \hat{\boldsymbol{\mu}}_{s_rk})^{T}\right]\mathbf{C}_{s_rk}^{T}\right\}}{\sum_{r=1}^{R}\sum_{k=1}^{K}\sum_{t=1}^{T_{s_r}}\gamma_{s_rk}(t)} \tag{3.128}$$

where $\hat{\boldsymbol{\mu}}_{s_rk}$ is the MLLR updated Gaussian mean estimate and $\mathbf{C}_{s_rk}$ is given by Equation 3.126. The estimate of $\mathbf{H}_s$ in Equation 3.128 results in a full transformed covariance matrix. Full covariance matrices, however, are rarely used in speech recognition systems due to their greatly increased computational requirements. For diagonal covariance, which we also use, the diagonal entries of $\hat{\boldsymbol{\Sigma}}_{jk}$ are only affected by the diagonal entries of $\mathbf{H}$. The results is thus a diagonal transformation of variance - which does not take dependencies between the feature dimensions into account. In experiments, Gales and Woodland [72] reported an additional decrease in word error rate (WER) of 2% for speaker adaptation by using this mean and variance adaptation approach versus only MLLR mean adaptation, which by itself achieved 13% decrease in WER. Results [72] for noise and channel compensation produced greater increases due to variance adaptation (7% reduction in WER).

For cross-language adaptation, adaptation of the variance components may result in larger performance gains than for speaker adaptation, but may require a more complex approach than diagonal transformation. Recently, Gales [73] proposed a method for unconstrained full variance transformation which uses an iterative estimation algorithm to solve for the transformation. We, however, propose and evaluate an alternative approach.

Minimum squared error variance transformation

We propose a method for unconstrained full variance transformation that uses weighted least squares estimation to compute the variance transformation in a single iteration. The standard MLLR algorithm (Equation 3.121) is used to estimate transformed Gaussian mean parameters in a first stage, similar to the approach suggested by Gales & Woodland [72], followed by Gaussian variance transformation in the next stage. Since almost exclusive use is made of diagonal covariance matrices in speech recognition systems, we only consider the transformation of the variance parameter vector $\boldsymbol{\sigma}^2_{s_r k}$ on the diagonal of the covariance matrix $\boldsymbol{\Sigma}_{s_r k}$. A full transformation of the variance parameters associated with the K component mixtures of a set of R states $\{s_1, ..., s_R\}$ can be expressed by

$$\hat{\boldsymbol{\sigma}}^2_{s_r k} = \mathbf{W}^*_s \boldsymbol{\sigma}^2_{s_r k} \tag{3.129}$$

where $\mathbf{W}^*_s$ is the (full) shared variance transformation matrix. We consider calculating the maximum likelihood estimate of the variance transformation of Equation 3.129, but find that the estimate can not be written in a closed-form, which reduces the attractiveness of the approach. We therefore consider using least squares estimation for the computation of $\mathbf{W}^*_s$. The squared error for the variance transformation of Equation 3.129 can be computed directly from the observation data and is then expressed by

$$E_1 = \sum_{r=1}^{R} \sum_{t=1}^{T_{s_r}} \sum_{k=1}^{K} \gamma_{s_r k}(t) \left[(\mathbf{x}_{s_r t} - \hat{\boldsymbol{\mu}}_{s_r k})^2 - \mathbf{W}^*_s \boldsymbol{\sigma}^2_{s_r k} \right]^T \left[(\mathbf{x}_{s_r t} - \hat{\boldsymbol{\mu}}_{s_r k})^2 - \mathbf{W}^*_s \boldsymbol{\sigma}^2_{s_r k} \right] \tag{3.130}$$

where $\hat{\boldsymbol{\mu}}_{s_r k}$ is an MLLR updated mean value (Equation 3.121) and assuming that the square of a vector implies computing the component-wise square of the vector (in the first term in brackets). Alternatively the squared error can also be expressed in terms of a statistic measuring the expected variance of the observation data by

$$E_2 = \sum_{r=1}^{R} \sum_{k=1}^{K} \gamma_{s_r k} \left[\mathbf{v}_{s_r k} - \mathbf{W}_s^* \boldsymbol{\sigma}_{s_r k}^2 \right]^T \left[\mathbf{v}_{s_r k} - \mathbf{W}_s^* \boldsymbol{\sigma}_{s_r k}^2 \right], \tag{3.131}$$

where $\mathbf{v}_{s_r k}$ is the target variance (vector) for mixture k of state s_r and is given by

$$\mathbf{v}_{s_r k} = \frac{\sum_{t=1}^{T_{s_r}} \gamma_{s_r k}(t)(\mathbf{x}_{s_r t} - \hat{\boldsymbol{\mu}}_{s_r k})^2}{\gamma_{s_r k}}. \tag{3.132}$$

We prefer to use Equation 3.131 because Equation 3.130 computes the fourth power of the distance between each observation and the transformed mean value, leading to very large estimates of the variance, while Equation 3.131 uses the average variance as computed in Equation 3.132. There are still, however, fundamental problems with the use of the variance transformation of Equation 3.129 as optimised using Equation 3.131 since:

- the constraint $\hat{\sigma}_{jkl}^2 > 0$ is not guaranteed and
- the least squares error function measures an additive error and not a relative error, thereby biasing the transformation to decrease the error produced by large variance values and causing large relative errors for small variance values.

The transformed variance values can be forced to be valid by applying a variance floor, such as described in Section 3.2.2, but this does not really present a desirable solution. Also, if the magnitude of the variance values grouped together in a transformation have a large range, the relative error may be very large for small variance values, even if the relative error is small for large variance values. A better method for the MSE variance transform that overcomes both these problems is given next.

Minimum squared error log-variance transformation

We propose transforming variance parameters in log-space, thereby maintaining the constraint $\hat{\sigma}^2_{jkl} > 0$ and also minimising the relative error (in place of the absolute error) in the estimation of $\hat{\sigma}^2_{jkl}$. The transformation of the log-variance parameters by transformation matrix $\mathbf{W}_s^\dagger$ is given by

$$\log \hat{\boldsymbol{\sigma}}^2_{s_r k} = \mathbf{W}_s^\dagger \log \boldsymbol{\sigma}^2_{s_r k} \tag{3.133}$$

where $\log \boldsymbol{\sigma}^2_{s_r k}$ is the element-wise logarithm of $\boldsymbol{\sigma}^2_{s_r k}$. The squared error to be minimised can be written as

$$E = \sum_{r=1}^{R} \sum_{k=1}^{K} \gamma_{s_r k} \left[\log \mathbf{v}_{s_r k} - \mathbf{W}_s^\dagger \log \boldsymbol{\sigma}^2_{s_r k} \right]^T \left[\log \mathbf{v}_{s_r k} - \mathbf{W}_s^\dagger \log \boldsymbol{\sigma}^2_{s_r k} \right], \tag{3.134}$$

where the target variance $\mathbf{v}_{s_r k}$ is given by Equation 3.132. By writing the squared error in the following format

$$E = \sum_{r=1}^{R} \sum_{k=1}^{K} \gamma_{s_r k} \left[\log \frac{\mathbf{v}_{s_r k}}{\hat{\boldsymbol{\sigma}}^2_{s_r k}} \right]^T \left[\log \frac{\mathbf{v}_{s_r k}}{\hat{\boldsymbol{\sigma}}^2_{s_r k}} \right], \tag{3.135}$$

it is evident that the log-variance transform minimises the *relative* error between the transformed variance $\hat{\boldsymbol{\sigma}}^2_{s_r k}$ and the target variance $\mathbf{v}_{s_r k}$ and is therefore not as sensitive to the relative magnitudes of the variance components as the direct variance transformation.

Finally, the least squares estimate for the log-variance transformation matrix is given in pseudo inverse form solution (as in Equation 3.109):

$$\mathbf{W}_s^\dagger = \left[\sum_{r=1}^{R} \sum_{k=1}^{K} \gamma_{s_r k} \log \mathbf{v}_{s_r k} \log \boldsymbol{\sigma}^2_{s_r k}{}^T \right] \left[\sum_{r=1}^{R} \sum_{k=1}^{K} \gamma_{s_r k} \log \boldsymbol{\sigma}^2_{s_r k} \log \boldsymbol{\sigma}^2_{s_r k}{}^T \right]^{-1}. \tag{3.136}$$

We note that the same discussion that applied to the MLLR estimation equation (Equation 3.121 in Section 3.3.1) applies here with respect to the number of transformed mixtures and the dimension of the transformation. When equal or fewer mixtures than the dimension

of the transformation are used, exact re-estimation of the variance values is the result. This, however, implies that inversion of the right-hand-side of Equation 3.136 is not attempted, but that the solution is found through e.g. a singular value decomposition-based approach. The more mixtures are grouped together in a transformation, the more robust, yet less accurate, the transformation becomes. When little data is available, few transformations should be calculated since direct estimation of the variance is problematic on little data.

This concludes our discussion of linear transformation-based adaptation. For speaker adaptation mean-only transformations are usually used, but we have covered variance adaptation in depth since it is important for cross-language adaptation. We have omitted discussion of the adaptation of mixture weight and transition probability parameters because it is inappropriate to apply transformation-based adaptation to them. For cross-language purposes, adaptation of mixture weight and transition probability parameters may be warranted. Other forms of adaptation as in Section 3.2 or even re-estimation may then be used on these parameters as they require far smaller amounts of data to estimate reliably than the Gaussian mean and variance parameters. We now proceed to discuss the application of *non-linear* transformation methods for adaptation.

3.3.3 Non-linear transformation adaptation

Non-linear transformation presents a more powerful paradigm than linear transformation, but present serious challenges in finding a suitable functional form for the transformation and also in optimising the parameters of the transform. As was noted in the previous section on linear transformation, only limited amounts of data are usually available. Relatively simple and well understood estimation techniques such as linear regression are able to use data relatively efficiently, while for the non-linear transformation approach gradient-based techniques must generally be used, which may not use limited data as efficiently.

Non-linear transformation of acoustic parameters has been performed for speaker adaptation using multi-layer perceptrons (MLPs) by Abrash *et al.* [74]. Gaussian mean compo-

nents of a speaker independent model were adapted on speech from non-native American English speakers. A single non-linear (sigmoidal output function) hidden layer was used for the MLP. A linear transformation was used in parallel with the MLP, effectively adding direct connections from the inputs to the linear output neurons. The weights of the MLP were initialised to small random values, and the linear transformation was set to an identity matrix. Training both the linear transform and the MLP with gradient descent to maximise the observation data likelihood did not achieve the peak performance achieved with an MLLR-estimated linear transform when many transformation classes were allowed. However, when the linear transform was initialised with the MLLR estimate, a modest improvement on standard MLLR was achieved by applying gradient descent to both the linear transform and the MLP.

Choi and King [54] compared the performance of using an MLP with using linear transformations for speaker adaptation and found that the linear transformation delivered significantly better performance. The two studies thus indicate that, using current techniques, it may be difficult for non-linear transformations to improve on the performance of multiple linear transformations. For these above reasons, we restrict our further experimental investigations to linear transforms.

3.3.4 Transformation for normalisation before training

The use of transformations as a pre-processing stage for the normalisation of speech from different speakers before commencing with HMM training has shown promising results. A procedure for *data augmentation* was suggested by Bellegarda *et al.* [75] that performs a least squares linear mapping from the acoustic space of a reference speaker to that of a new speaker. A large amount of data from a reference speaker is transformed to augment the little data from a new speaker to serve for the training of speaker dependent models for the new speaker. Separate linear transformations are estimated for the data associated with groups of elementary speech models. A problem that was reported with the procedure was that too much transformed data from a single reference speaker overwhelmed the small

amount of speaker specific data. This situation was improved in subsequent research [76] by

- implementing transformations from multiple reference speakers - thereby reducing the amount of data per reference speaker to approximately the amount of data available for the new speaker,
- implementing a selection procedure to choose reference speakers that are "close" in some sense to the new speaker and
- tieing all the models for a reference speaker in estimating the transformation.

Further improvements to the approach are detailed in [67] and include using MLLR to estimate the transformations and using gender dependent models to estimate alignments instead of using reference speaker specific models.

Procedures related to the previous approach have been used for speaker normalisation before training. Ishii and Tonomura [77] implemented a procedure for speaker normalisation through transformation. The method estimates MLLR mappings from each speaker to the SI model trained on speech from all the speakers, subtracts the MLLR offsets from the speech data and retrains the SI models. This procedure is repeated iteratively and delivers speaker independent models that do not model speaker variation offset and may thus have narrower distributions. For recognition purposes MLLR is used to estimate the transformation (including offset) from the normalised SI models for a new speaker. A closely related approach was also proposed by Nagesha and Gillick [66]. MLLR mappings are also estimated from SI models to each of a set of speakers, but speaker specific data is then transformed using the inverse of the MLLR estimated transformation for each speaker. The reverse transformed data is then used to retrain SI models and the procedure is repeated. Speaker independent models are thus produced that are invariant to linear transformations of the speech from speakers used to train them. Obviously, to accurately recognise speech from a new speaker, a transformation from the normalised models to the new speaker must first be estimated.

The procedures discussed in this section are of interest for cross-language adaptation, because they may be applied to the normalisation of data from multiple databases containing multiple languages to a single target language. Further detail regarding application of the methods is given in Chapter 5. The next section discusses how Bayesian and transformation-based techniques can be combined to improve adaptation performance.

3.4 Combined Bayesian and transformation-based adaptation

Both the Bayesian adaptation approach detailed in Section 3.2 and the transformation-based adaptation approach detailed in Section 3.3 have their respective strengths and weaknesses. Bayesian methods have in particular two perceived advantages over the transformation-based approach namely that with Bayesian methods

- expected performance is asymptotic with respect to a target system - i.e. the performance converges to that of a target dependent system when a large amount of data is available, and
- the degree to which adaptation takes place is automatically controlled by the amount of adaptation data available - i.e. when little data is available little adaptation takes place and as more data is available, more adaptation takes place.

We note that the asymptotic performance property of Bayesian techniques is not true for a transformation-based adaptation approach in general, but as we discussed in Section 3.3.1, may be achieved for transformed values if the number of transformation classes is allowed to increase with the amount of adaptation data. The Gaussian mean values then eventually agree with the target dependent mean values when there are fewer independent Gaussian mixtures per transformation than the dimension of the transformation itself. This argument may be extended to the Gaussian variance values if they are transformed separately

from the means. Transformation-based adaptation, on the other hand, has the advantage over Bayesian adaptation that by sharing transformations across groups of phonemes, unobserved parameters can be adapted, leading to more rapid adaptation than is possible with Bayesian adaptation.

Methods to combine Bayesian and transformation-based adaptation are researched in an attempt to retain desired properties from both strategies. We discuss two main techniques that combine Bayesian and transformation-based methods, the first technique focusing on combining rapid transformation-based adaptation with the asymptotic performance property of Bayesian adaptation and the second technique focusing on using Bayesian techniques to control transformation-based adaptation when little data is available.

3.4.1 Linear transformation-MAP

Digalakis and Neumeyer [78] proposed combining Bayesian and transformation-based adaptation in two stages. Constrained transformation-based adaptation [71] is performed in the first stage, using a diagonal transformation to adapt both mean and variance (Equations 3.122 and 3.123) parameters with the adaptation data for a new speaker. This has the advantage of rapidly and accurately compensating for significant bias between source models and target data, such as is exhibited by channel effects. The resulting (speaker adapted) models are used as the starting point for the second adaptation stage, implementing an approximate MAP (AMAP) adaptation algorithm for the Gaussian mean and variance parameters. The Gaussian mean parameters are estimated using an interesting variation to the MAP mean estimate of Equation 3.74 given for mixture k of state i by [78]

$$\boldsymbol{\mu}_{ik\,\mathrm{AMAP}} = \frac{\varpi\gamma_{ik}^{\mathrm{SI}}\boldsymbol{\mu}_{ik}^{\mathrm{SA}} + (1-\varpi)\gamma_{ik}^{\mathrm{SD}}\bar{\mathbf{x}}_{ik}^{\mathrm{SD}}}{\varpi\gamma_{ik}^{\mathrm{SI}} + (1-\varpi)\gamma_{ik}^{\mathrm{SD}}}, \tag{3.137}$$

where $\gamma_{ik}^{\mathrm{SI}}$ and $\gamma_{ik}^{\mathrm{SD}}$ are the mixture occupancy statistics of the speaker independent and speaker dependent data respectively, $\boldsymbol{\mu}_{ik}^{\mathrm{SA}}$ is the (speaker adaptive) transformed mean value, $\bar{\mathbf{x}}_{ik}^{\mathrm{SD}}$ is the sample mean of the speaker dependent data and ϖ is a global adaptation

rate factor, in this case taking on values between zero and one. Gaussian variance parameter estimation is computed in a similar fashion to the mean estimation, calculating a linear combination of transformed variance statistics and speaker dependent variance statistics. Digalakis and Neumeyer report [79] that their technique approximately halves the recognition error rate for non-native speakers of American English with only a small amount of adaptation data, approaching the speaker independent accuracy achieved for native speakers.

Comparing the method to MAP mean estimation as derived in Section 3.2.5 (Equation 3.100 in particular), the mixture weight prior seed $\tilde{c}_{ik}$ associated with a mixture in the prior has been replaced by the occupancy statistics for that mixture and the learning factor ϖ is incorporated in a different way. Using occupancy statistics for weighting causes mixtures with high occupancy in the prior to be adapted more slowly than mixtures for which little data was observed when the prior was trained. This may be useful for speaker adaptation, but not necessarily for cross-language adaptation, as the frequency of occurrence of a phoneme in a source language may not give an accurate indication as to its suitability for seeding a prior distribution for target language model estimation. In fact, we found that use of source language occupancy statistics (as in Equation 3.137) delivered poorer performance than use of the mixture weight prior seed $\tilde{c}_{ik}$ (as in Equation 3.100) and therefore in experiments in Chapters 6 and 7 we used the MAP estimates of Section 3.2.5 in implementing MLLR-MAP adaptation.

Thelen *et al.* [80] also implemented a combination of linear transformation and MAP adaptation, similar to that of Digalakis and Neumeyer [78], but using least squares to estimate a full transformation matrix for the Gaussian mean parameters and then used the standard MAP algorithm to derive the final mean values. Better results were obtained with phonetically derived regression classes than with clustering procedures. Interestingly, they reported that their linear regression-MAP algorithm did not achieve asymptotic performance with a speaker dependent system as was planned. They give as a reason the fact that, even with a large amount (several hours) of adaptation data from a single speaker, less than 70% of the transformed densities are observed during MAP adaptation and are

thus unadapted. Most parameters of the adapted system are therefore not optimised for the target speaker beyond the initial transformation. The percentage of unobserved densities may have been even higher if the initial transformation had not been performed. This points to a deficiency in the Bayesian estimation framework, namely that when the distribution of the adaptation data differs significantly from the distribution of the data that was trained on, only a fraction of the total parameter set that corresponds to the adaptation data is adapted. For cross-language and cross-database acoustic adaptation we expect that the overlap between source and target feature distributions may be relatively poor, which may negatively influence recognition performance. We therefore evaluate the performance of using MLLR-MAP, showing in Chapter 7 that it leads to improved performance for cross-language, cross-database adaptation.

3.4.2 MAP-MLLR

Chou [81] recently proposed an alternative combination of Bayesian and transformation-based adaptation termed maximum *a posteriori* linear regression (MAPLR). The goal of the method is not to ensure asymptotic performance, but to control the amount of adaptation when little data is available by using prior distributions. It incorporates prior knowledge by biasing the MLLR transformation to more closely match a unity transformation when little adaptation data is available and to more closely match the MLLR estimate when a large amount of adaptation data is available.

MAPLR

MAPLR assumes an elliptic symmetric *a priori* distribution for the transformation matrix. The solution to MAPLR entails diagonalising the matrix inversion of the MLLR estimate

(Equation 3.121) through the addition of a diagonal matrix, i.e.

$$\begin{aligned}\hat{\mathbf{w}}_{sl}^T &= (\hat{\mathbf{G}}_{sl})^{-1}\hat{\mathbf{z}}_{sl} \\ &= (\mathbf{G}_{sl} + \mathbf{D}_{sl})^{-1}(\mathbf{z}_{sl} + \mathbf{D}_{sl}\tilde{\mathbf{w}}_l^T)\end{aligned} \tag{3.138}$$

where $\mathbf{G}_{sl}$ is equal to the first term in brackets and $\mathbf{z}_{sl}$ the second term in brackets on the right hand side of Equation 3.121, $\mathbf{D}_{sl}$ is the scale factor (acting as a diagonalising term) and $\tilde{\mathbf{w}}_l^T$ is the lth row of the location parameter of the transformation. Choosing the location parameter ($\tilde{\mathbf{W}}$) to be the identity matrix backs off the transformation to an identity transform when there are no observations and is the implementation approach that we use. When a large number of observations are available, the occupancy statistics of $\mathbf{G}_{sl}$ and $\mathbf{z}_{sl}$ dominate the equation, ensuring convergence to the MLLR estimate.

Chou [81] uses a global MLLR transformation to estimate the prior location parameters ($\tilde{\mathbf{w}}_l$), but does not describe how to estimate the scale parameters ($\mathbf{D}_{sl}$). We propose using as diagonalising term the diagonal of $\mathbf{G}_{sl}$ (from Equation 3.121), normalised with respect to the amount of data and multiplied by an overall prior weight scaling factor ϖ, i.e.

$$d_{sli} = \varpi \frac{\sum_{r=1}^{R}\sum_{k=1}^{K}\gamma_{s_rk}r_{s_rkl}v_{s_rki}^2}{\sum_{r=1}^{R}\sum_{k=1}^{K}\gamma_{s_rk}}, \tag{3.139}$$

where d_{sli} is the ith term on the diagonal of $\mathbf{D}_{sl}$ and v_{s_rki} is the ith term of the extended mean vector $\boldsymbol{v}_{s_rk}$. The value of ϖ depends on the suitability of the prior distribution and should be determined empirically.

MAP-like log variance transformation

We propose using a similar approach to MAPLR for the diagonalisation of the MSE log-variance transformation. Since our attempts at obtaining an ML estimate for the variance and log-variance transformations did not produce a closed-form solution, MAP estimation is not attempted. We propose simply adding a diagonalising term (scaling parameter)

$\mathbf{D}_s^\dagger$, similar to the scaling parameter in Equation 3.138, to the pseudo inverse solution of Equation 3.136, producing the MAP-like estimate

$$\hat{\mathbf{W}}_s^\dagger = \left[\sum_{r=1}^{R}\sum_{k=1}^{K}\gamma_{s_rk}\log\mathbf{v}_{s_rk}\log\boldsymbol{\sigma}_{s_rk}^{2\,T} + \mathbf{D}_s^\dagger\right]\left[\sum_{r=1}^{R}\sum_{k=1}^{K}\gamma_{s_rk}\log\boldsymbol{\sigma}_{s_rk}^{2}\log\boldsymbol{\sigma}_{s_rk}^{2\,T} + \mathbf{D}_s^\dagger\right]^{-1}. \tag{3.140}$$

When no data is observed, $\hat{\mathbf{W}}_s^\dagger$ backs off to a unity transformation and when a large amount of data is observed, $\hat{\mathbf{W}}_s^\dagger$ converges to the MSE estimate. We propose calculating the diagonal term $\mathbf{D}_s^\dagger$ in a similar fashion to the MAPLR diagonal term (Equation 3.139), producing the equation

$$d_{si}^\dagger = \varpi\frac{\sum_{r=1}^{R}\sum_{k=1}^{K}\gamma_{s_rk}(\log\sigma_{s_rki}^2)^2}{\sum_{r=1}^{R}\sum_{k=1}^{K}\gamma_{s_rk}}, \tag{3.141}$$

where $d_{si}^\dagger$ is the ith term on the diagonal of $\mathbf{D}_s^\dagger$ and the overall prior weight scaling factor ϖ is shared with Equation 3.139.

3.4.3 Comparison of MLLR-MAP and MAP-MLLR

Figure 3.1 shows conceptually the difference between the MLLR-MAP and MAP-MLLR approaches. While MAP-MLLR controls the amount of adaptation the transformation can effect, MLLR-MAP uses the MLLR transformed models to seed prior distributions for MAP estimation.

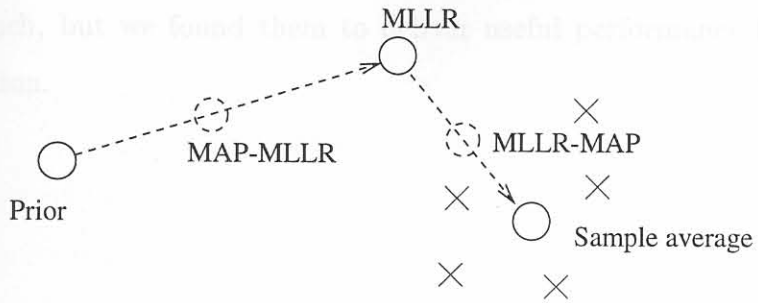

Figure 3.1: Graphical comparison of the working of the MAP-MLLR and MLLR-MAP techniques, showing adaptation of the Gaussian mean

Both techniques may be useful for cross-language adaptation in different ways. MLLR-MAP should provide better asymptotic performance than MLLR adaptation alone and should also improve performance if source and target language models are poorly matched by removing channel effects before commencing with MAP adaptation. MAP-MLLR may in particular improve the robustness of estimates for transformation classes with little data by decreasing the over-fitting effect when complex transformations are estimated from limited data. Both methods are experimented with in Chapter 6 and 7 and are shown to significantly improve performance.

3.5 Discussion

We have discussed the major classes of methods used for speaker adaptation, namely Bayesian and transformation-based methods, as well as combinations of these techniques. A new technique for full transformation of variance parameters in log-space and utilising MAP-like control over adaptation was proposed, specifically with cross-language model adaptation in mind. Some aspects regarding the application of the techniques for cross-language adaptation were mentioned, but will only be discussed in detail in Chapter 5. Experimental comparisons of Bayesian, transformation-based and combined techniques are given in Chapters 6 and 7).

In the next chapter we discuss a third class of methods applicable for acoustic adaptation, namely discriminative training methods. These methods are not generally used for speaker adaptation as such, but we found them to deliver useful performance for cross-language acoustic adaptation.

Chapter 4

Discriminative learning theory

Discriminative learning presents an alternative to the classical probabilistic interpretation of pattern recognition, which links a pattern classification task to a distribution estimation problem. Given an observation, the Bayes decision rule leads to the minimum misclassification probability when the true observation distribution is known, by selecting the model with the highest *a posteriori* probability. Unfortunately, the true form of the distribution is rarely known in classification problems and a parametric form is often assumed for computational tractability. Furthermore, the parameters of the assumed distribution have to be estimated from a limited amount of training data. These factors contribute to the sub-optimality of the distribution estimation strategy for classifier design and leads to the consideration of an alternative strategy that attempts only to discriminate between observations from different classes, rather than attempting to estimate the true distributions. The Bayes decision rule can still be applied if the models are used to implement discriminant functions rather than distribution estimators.

Discriminative learning has been researched for many decades for pattern classification purposes, but it is in conjunction with efficient methods for training artificial neural networks (ANNs) [82] that the most prominent research in this field has taken place. Most algorithms for training ANNs make use of supervised feedback of some error or reward (discriminative

measure) in computing derivatives for the weights of the ANN. In this chapter we focus on a discriminative approach that can be applied to the training of HMMs and in particular can be applied to the adaptation of HMMs as that is the prime reason for our interest in discriminative learning techniques.

4.1 Discriminative optimisation criteria

The most useful training or learning strategies include a criterion or function that in some way measures the quality of a particular model, given observations from the process that is modelled. Since the criterion determines subjectively the performance of any particular model, it prescribes which models will be considered "better" models and which will be considered "poorer" models. Usually, models are desired that maximise the criterion. The maximisation of a criterion, however, is generally obtained in closed-form solution only for the simplest criteria and models, e.g. a least squares linear estimation problem. Therefore, in addition to defining a criterion, a method for determining useful model parameters based on the criterion has to be established. The choice of optimisation function is thus influenced both by its intended purpose, e.g. increasing class separation or decreasing the misclassification rate, as well as by the ease with which it can be optimised.

Before we proceed to discuss discriminative criteria for optimisation, we first review the non-discriminative criterion that is most widely used for HMM training, namely the *maximum likelihood* criterion, $\arg\max_{\boldsymbol{\lambda}} f(\mathbf{X}|\boldsymbol{\lambda})$, of which the application was also discussed in Chapters 2 and 3. ML can be shown [83] to produce the best decoder if certain conditions are met, such as having a suitably large amount of data and knowing the true form of the distribution of the speech observations. Unfortunately, both conditions are not met for speech recognition problems. The ML criterion is functionally simple, but has the difficult goal of attempting to estimate the parameters of a distribution, rather than just a discriminant function. Even for ML estimation, closed form solutions do not exist for HMM parameter estimates and iterative methods such as the Baum-Welch procedure are

used to approximate the ML parameters. The Baum-Welch procedure guarantees increasing the ML criterion at each iteration until convergence in a locally optimal point, where discriminative optimisation techniques often deliver little in terms of guaranteeing increased performance and are prone to converge to (poor) local minima/maxima. ML model estimates are therefore almost always used as a starting point for discriminative training. After that, it is attempted to incrementally improve performance using discriminative optimisation techniques. We now proceed to discuss discriminative criteria applied to the training of HMMs.

4.1.1 Corrective training

Under the heading of corrective training we group various ad-hoc discriminative techniques that have been applied to the training of HMMs. Franco [84] suggested a heuristic discriminative training algorithm using a frame-level cost function to measure the degree of misalignment between a model and an observation sequence. The function measures at each time frame the squared error between the target function (set to one for the "correct state" of the true class models and zero otherwise) and the posterior state occupancy probability for the model at the time frame. The criterion function is thus a frame-level mean square error (MSE) function. Given an observation sequence, the method performs a gradient update on the parameters in the direction of a unity likelihood for the correct state of the correct model and in the direction of a likelihood of zero for any other state. The procedure theoretically stops when the correct state likelihoods are unity and the incorrect state (including all states from false class models) likelihoods are zero. Various heuristics were used to limit the degree of adaptation to avoid over-training. The method was shown to improve initial ML trained model performance on a phoneme recognition task.

Gauvain and Lee [58] proposed a simple modification to the statistics collection phase of the MAP algorithm to implement a heuristic version of corrective training. Training sentences that are incorrectly recognised are used as new data for updating model statistics and the state occupancy statistic, $\gamma_j(t)$ (Equation 2.11) is multiplied by -1 for an incorrect model

and by 1 for the correct model. A limited number of MAP adaptation iterations using the modified statistics are performed. The procedure has no explicit optimisation function, but was found to deliver better performance on connected digit and isolated word recognition tasks than ML estimated models.

Chen & Soong [85] suggested an N-best candidates-based discriminative training algorithm using a frame-level optimisation criterion. It improves on the method suggested by Franco [84] by not attempting to force zero-one state occupancy values for correct and incorrect classes respectively. The frame-level loss function comprises a half-wave rectified log-likelihood difference between the correct and competing hypotheses and is optimised on the training set by performing gradient descent on the HMM parameters. Performance improvement over ML trained HMMs was reported on connected digit and isolated word recognition tasks.

4.1.2 Maximum mutual information (MMI)

Maximum mutual information is an information-theoretic concept that provides a basis for the derivation of a discriminative training criterion. The following derivations closely follow McDermott [86]. The *conditional entropy* $H_{\Lambda}(\mathbf{C}|\mathbf{X})$ of the class random variable $\mathbf{C}$, given the observation random variable $\mathbf{X}$, is minimised in terms of the model parameters $\Lambda = (\lambda_1, ..., \lambda_M)$ when the *mutual information* is maximised for each of M classes. What this means is that the uncertainty associated with $\mathbf{C}$ given $\mathbf{X}$ is minimised when the model parameters Λ provide as much information as possible about the class random variable $\mathbf{C}$ given $\mathbf{X}$. This can be verified by noting that the mutual information between $\mathbf{C}$ and $\mathbf{X}$, $I_{\Lambda}(\mathbf{C};\mathbf{X}) = I_{\Lambda}(\mathbf{X};\mathbf{C})$ can be written as the difference between the entropy of $\mathbf{C}$ and the conditional entropy of $\mathbf{C}$ given $\mathbf{X}$:

$$I_{\Lambda}(\mathbf{C};\mathbf{X}) = H(\mathbf{C}) - H_{\Lambda}(\mathbf{C}|\mathbf{X}). \tag{4.1}$$

If the entropy $H(\mathbf{C})$, for a basic unit such as a word in a speech recognition task is expressed by a language model and may thus be considered a given [87] , minimisation of the conditional entropy is achieved by maximising the mutual information between $\mathbf{C}$ and $\mathbf{X}$.

A more useful form of Equation 4.1 will now be shown. With the entropy of a random variable $\mathbf{C}$ for model parameters $\mathbf{\Lambda}$ given by a summation over the class variable c

$$H(\mathbf{C}) = -\sum_{c} P(\mathbf{C}=c)\log P(\mathbf{C}=c) \tag{4.2}$$

and the conditional entropy of $\mathbf{C}$ given $\mathbf{X}$ for model parameters $\mathbf{\Lambda}$ given by a summation over c and the observation variable x by

$$H_{\mathbf{\Lambda}}(\mathbf{C}|\mathbf{X}) = -\sum_{c,x} P(\mathbf{C}=c,\mathbf{X}=x)\log P_{\mathbf{\Lambda}}(\mathbf{C}=c|\mathbf{X}=x), \tag{4.3}$$

it is convenient to examine the maximisation of Equation 4.1 expanded as follows [86]:

$$\begin{aligned}
I_{\mathbf{\Lambda}}(\mathbf{C};\mathbf{X}) &= -\sum_{c} P(\mathbf{C}=c)\log P(\mathbf{C}=c) + \sum_{c,x} P(\mathbf{C}=c,\mathbf{X}=x)\log P_{\mathbf{\Lambda}}(\mathbf{C}=c|\mathbf{X}=x) \\
&= -\sum_{c,x} P(\mathbf{C}=c,\mathbf{X}=x)\log P(\mathbf{C}=c) + \sum_{c,x} P(\mathbf{C}=c,\mathbf{X}=x)\log \frac{P_{\mathbf{\Lambda}}(\mathbf{C}=c,\mathbf{X}=x)}{P_{\mathbf{\Lambda}}(\mathbf{X}=x)} \\
&= \sum_{c,x} P(\mathbf{C}=c,\mathbf{X}=x)\log \frac{P_{\mathbf{\Lambda}}(\mathbf{C}=c,\mathbf{X}=x)}{P(\mathbf{C}=c)P_{\mathbf{\Lambda}}(\mathbf{X}=x)} \\
&= \sum_{c,x} P(\mathbf{C}=c,\mathbf{X}=x)\log \frac{P_{\mathbf{\Lambda}}(\mathbf{X}=x|\mathbf{C}=c)}{P_{\mathbf{\Lambda}}(\mathbf{X}=x)}.
\end{aligned} \tag{4.4}$$

Since the true likelihood $P(\mathbf{C}=c,\mathbf{X}=x)$ is unknown, training samples of $\mathbf{C}$ and $\mathbf{X}$ are assumed to be representative of the true distribution and the MMI criterion is maximised by the $\mathbf{\Lambda}$ that maximises

$$f_{\mathrm{MMI}}(\mathbf{\Lambda}) = \sum_{c,x} \log \frac{P_{\mathbf{\Lambda}}(\mathbf{X}=x|\mathbf{C}=c)}{\sum_{c'} P_{\mathbf{\Lambda}}(\mathbf{X}=x|\mathbf{C}=c')P(\mathbf{C}=c')}. \tag{4.5}$$

The MMI criterion differs from the ML criterion ($P_{\mathbf{\Lambda}}(\mathbf{X}=x|\mathbf{C}=c)$) since the MMI crite-

rion maximises the relative likelihood of the correct class, rather than simply maximising the absolute likelihood. This introduces discrimination into the training procedure. The correspondence between the MMI criterion and the *a posteriori* class probability is evident, but note that the likelihood functions associated with the model (Λ) that maximises the MMI criterion are of a discriminatory nature and do not implement density estimators.

Some applications of MMI are briefly mentioned. Cardin *et al.* [88] and also Normandin & Morgera [89] applied MMI estimation to the training of parameters of HMMs. Parameters were initialised with ML estimates and MMI was performed in an adaptive mode, with smoothing applied during the parameter update. Both studies showed improvement from ML trained models on the TI/NIST connected digit database. Kapadia *et al.* [90] achieved improved continuous phoneme recognition on the TIMIT [31] database using MMI estimation.

While the MMI criterion clearly improves on the ML criterion in terms of taking into account both the correct and incorrect model likelihoods, it still does not directly reflect the classification performance of a system. This topic is addressed in the next section.

4.1.3 Minimum error rate

The goal of a classifier is ultimately to achieve the minimum possible error rate, if equal cost is associated with each error. This minimum error rate is achieved with a Bayes classifier in which, for any observation, the discriminant function associated with the largest *a posteriori* probability has the largest value. The MMI criterion expresses the functional form of the *a posteriori* estimate, thereby increasing class separation, but does not expressly minimise the error rate association with the estimate. The most direct optimisation of the error rate can be achieved with a criterion that hard-limits the difference between the true class and the highest false class discriminant functions. Discontinuous criteria are hard to optimise, however, and therefore a continuous criterion that emulates the error rate should be considered. We discuss such a method that was implemented and extensively used in

this thesis in more detail in the following section.

4.2 Minimum classification error approach

The minimum classification error (MCE) approach suggested by Juang & Katagiri [91] provides a technique for designing a classifier that approaches the objective of minimum classification error more directly than the methods discussed so far in this section. This is achieved by providing a criterion that accurately reflects the error rate of a classifier, yet is continuous and thus differentiable, facilitating optimisation of the parameters of the classifier. The method in general does not lead to closed-form re-estimation solutions for parameters and is thus used in conjunction with a gradient-based optimisation scheme. We now proceed to discuss the criterion for optimisation used in MCE.

4.2.1 Optimisation criterion

The sample risk, represented by the number of misclassifications in the training set, is the simplest and most direct function representing the error rate. It is, however, a piece-wise constant function and thus very difficult to optimise numerically since its derivatives contain no information. MCE training attempts to overcome the difficulty of directly optimising the error rate of a classifier on a set of data by defining a smoothed version of the error rate for optimisation. There are two key problems that have to be addressed namely

- measuring the distance between a correct and multiple incorrect classes and
- measuring the loss associated with a classification.

Misclassification measure

The decision boundary for a two class classification problem, with classes C_1 and C_2, is easily described in terms of the *a posteriori* probabilities by $P(C_2|\mathbf{X}) = P(C_1|\mathbf{X})$. It is, however, not easily extended to provide a measure of the distance between the correct class and multiple incorrect classes. One way of defining such a *misclassification measure* for an observation $\mathbf{X}$ from class i in terms of the class conditional log-likelihood functions, using the notation $g_j(\mathbf{X};\mathbf{\Lambda}) = \log f(\mathbf{X}|\boldsymbol{\lambda}_j)$, where $f(\mathbf{X}|\boldsymbol{\lambda}_j)$ is the class conditional likelihood function for class j, is by [28]:

$$d_i(\mathbf{X};\mathbf{\Lambda}) = -g_i(\mathbf{X};\mathbf{\Lambda}) + \log\left[\frac{1}{M-1}\sum_{j,j\neq i}^{M} e^{g_j(\mathbf{X};\mathbf{\Lambda})\eta}\right]^{1/\eta}, \tag{4.6}$$

where η is a positive number. The *misclassification measure* is a continuous function of all the classifier parameters $\mathbf{\Lambda}$ and attempts to emulate the Bayes decision rule, i.e. that for an ith class utterance $\mathbf{X}$, $d_i(\mathbf{X};\mathbf{\Lambda}) < 0$ implies correct recognition and $d_i(\mathbf{X};\mathbf{\Lambda}) > 0$ implies incorrect recognition. The value of η controls the relative significance of false class likelihoods. When η is large the term in brackets approaches $\max_{j,j\neq i} g_j(\mathbf{X};\mathbf{\Lambda})$, which is exactly the Bayes decision rule. For smaller η, competing classes with relatively smaller likelihoods are also taken into account, thereby deviating from the Bayes decision rule in a well defined manner and creating a soft decision boundary.

The *averaging* of the incorrect classes in Equation 4.6 is perhaps easier to understand when it is expressed in terms of the class conditional likelihood functions

$$\left[\frac{1}{M-1}\sum_{j,j\neq i}^{M} f_j(\mathbf{X};\mathbf{\Lambda})^{\eta}\right]^{1/\eta}. \tag{4.7}$$

Note that the misclassification measure (Equation 4.6) therefore actually expresses the ratio of the incorrect to correct class likelihoods, just in the log domain. When working with HMMs this is sensible because the likelihood values have a very large range, making direct subtraction of likelihoods almost meaningless.

Loss function

The misclassification measure (Equation 4.6) improves on the MMI criterion (Equation 4.5) in the sense that it defines a threshold (in the region of zero) for misclassification, even though the threshold is soft. In order to achieve the goal of emulating the expected misclassification rate the misclassification measure is embedded in a smoothed zero-one function such as the sigmoid function. The resulting function is then called the *loss function* and is given by

$$l_i(\mathbf{X};\mathbf{\Lambda}) = \frac{1}{1+e^{-\gamma d_i(\mathbf{X};\mathbf{\Lambda})+\theta}} \tag{4.8}$$

with θ normally set to zero and $0 < \gamma \leq 1$. When $d_i(\mathbf{X};\mathbf{\Lambda})$ is much smaller than zero, which implies correct classification, virtually no loss is incurred, while a large value of $d_i(\mathbf{X};\mathbf{\Lambda})$ leads to a loss close to one.

The criterion for minimisation can then be defined for a given training data set consisting of O observation sequences $\mathbf{X}_1..\mathbf{X}_O$ from a total of M classes $\{C_1,...,C_M\}$ by the *empirical loss*

$$L(\mathbf{X}_1..\mathbf{X}_O,\mathbf{\Lambda}) = \sum_{o=1}^{O}\sum_{i=1}^{M} l_i(\mathbf{X}_o;\mathbf{\Lambda})\,1(\mathbf{X}_o \in C_i) \tag{4.9}$$

where $1(\varphi)$ is the indicator function, taking on the value 1 when φ is true and 0 when it is false. Use of the *expected loss* presents an alternative to using the empirical loss, but has the associated problem that since the true distributions are unknown, current distribution estimates must be used in an iterative procedure. This would, however, also imply that calculation of the expectation is dependent on the classifier parameters, further complicating the optimisation function, and thus we use the empirical loss as optimisation criterion.

4.2.2 Gradient descent optimisation

A simple solution to minimise the empirical loss defined in Equation 4.9 is to use the gradient descent technique with batch-mode parameter updates

$$\Lambda_{n+1} = \Lambda_n - \epsilon_n \sum_{o=1}^{O} \sum_{i=1}^{M} 1(\mathbf{X}_o \in C_i) \nabla l_i(\mathbf{X}_o; \Lambda)|_{\Lambda=\Lambda_n} \tag{4.10}$$

where ϵ_n is the update parameter in iteration n and is chosen to be a suitable decreasing function of n. Note that we calculate the gradient using all available samples, also termed a deterministic update, since this improves the estimate of the gradient. A block mode update may be computationally cheaper to perform, or even an on-line update can also be used for real-time purposes, but we have not further pursued these two options.

A problem with the gradient descent technique is that it is suitable only for unconstrained minimisation, while the parameters of an HMM have definite constraints. Chou *et al.* [92] suggested making use of parameter transformations that remove the constraints in the transformed parameter space and thus facilitate the use of gradient descent optimisation. Details of the parameter transformations are given in the next section, along with the application of the MCE approach for HMMs.

4.2.3 HMM parameter update

A procedure for applying MCE to the training of the parameters of continuous density HMMs was suggested by Chou *et al.* [92] under the name *segmental GPD* (generalised probabilistic descent). Use of the name GPD is derived from the original MCE paper [91] which proposed using *probabilistic descent*, i.e. minimising the *expected loss* rather than the *empirical loss*. For practical reasons, however, we optimise the empirical loss in implementations of the approach. A more detailed discussion of the application of MCE for HMM training was later published by Juang *et al.* [28] and forms the basis for the discussion in this section. We note, however, that previous publications [92, 28] did not take

into account false class derivatives, i.e. the derivative of the loss function with respect to competing classes. We therefore extend the derivations to include both true and false class derivatives as was suggested by Kwon & Un [93] for the special case of the discriminative state-weighted HMM. For completeness we also provide transition probability derivatives.

HMM likelihood functions

Given that HMMs have been selected as the framework for modelling speech features, the *class conditional log-likelihood function* $g_i(\mathbf{X};\mathbf{\Lambda})$, $i = 1,..,M$ takes the form

$$g_i(\mathbf{X};\mathbf{\Lambda}) = \log f_i(\mathbf{X};\mathbf{\Lambda}) = \log f(\mathbf{X}|\mathbf{A}^{(i)}, \{b_j^{(i)}\}_{j=1}^{N}) \tag{4.11}$$

where the superscript i denotes the parameter set associated with class i. The *segmental* training procedure uses the Viterbi state-aligned likelihood function, which calculates the likelihood of Equation 4.11 along the state sequence with the highest likelihood, producing the log-likelihood function given by

$$\begin{aligned} g_i(\mathbf{X};\mathbf{\Lambda}) &= \log\{\max_{\mathbf{q}} f_i(\mathbf{X},\mathbf{q};\mathbf{\Lambda})\} \\ &= \log f_i(\mathbf{X},\bar{\mathbf{q}};\mathbf{\Lambda}) \\ &= \sum_{t=1}^{T}[\log a_{\bar{q}_{t-1}\bar{q}_t}^{(i)} + \log b_{\bar{q}_t}^{(i)}(\mathbf{x}_t)] \end{aligned} \tag{4.12}$$

where $\bar{\mathbf{q}}$ is the sequence with the maximum likelihood. As discussed in Chapter 2.1.2, the observation density in each state is a Gaussian mixture distribution, given in extended form and diagonal covariance for model i by

$$b_j^{(i)}(\mathbf{x}_t) = \sum_{k=1}^{K} c_{jk}^{(i)} \mathcal{N}[\mathbf{x}_t, \boldsymbol{\mu}_{jk}^{(i)}, \mathbf{\Sigma}_{jk}^{(i)}] = \sum_{k=1}^{K} \frac{c_{jk}^{(i)}}{(2\pi)^{(D/2)} \prod_{l=1}^{D} \sigma_{jkl}^{(i)}} e^{-\frac{1}{2}\sum_{l=1}^{D}(\frac{x_{tl}-\mu_{jkl}^{(i)}}{\sigma_{jkl}^{(i)}})^2} \tag{4.13}$$

Parameter transformations

It is desirable to maintain the original parameter constraints in the HMMs when adaptation takes place such as $\sum_{j=1}^{N} a_{ij} = 1$, $a_{ij} \geq 0$, $\sum_{k=1}^{K} c_{jk} = 1$, $c_{jk} \geq 0$ and $\sigma_{jkl} > 0$. In order for the problem to remain an unconstrained problem that is suitable for direct optimisation by gradient descent (Equation 4.10), a transformation of the parameters is necessary. A set of transformed parameters $\bar{a}_{ij}$, $\bar{c}_{jk}$, $\bar{\mu}_{jkl}$ and $\bar{\sigma}_{jkl}$ can be calculated that will maintain the constraints on the original parameters [28]:

1. $a_{ij} \rightarrow \bar{a}_{ij}$, where $a_{ij} = \frac{e^{\bar{a}_{ij}}}{\sum_{j'} e^{\bar{a}_{ij'}}}$
2. $c_{jk} \rightarrow \bar{c}_{jk}$, where $c_{jk} = \frac{e^{\bar{c}_{jk}}}{\sum_{k'} e^{\bar{c}_{jk'}}}$
3. $\mu_{jkl} \rightarrow \bar{\mu}_{jkl} = \frac{\mu_{jkl}}{\sigma_{jkl}}$
4. $\sigma_{jkl} \rightarrow \bar{\sigma}_{jkl} = \log \sigma_{jkl}$.

The reverse transformations of a_{ij} and c_{jk} ensure that the coefficients remain positive and maintain the property of summing to one. The transformation of μ_{jkl} normalises the relative magnitude of the mean in each dimension by the variance of the component in that dimension. In the author's experience, this transformation is very important because without it the derivative of the loss function with respect to the mean contains the precision term (the inverse of the variance), rendering the mean update stable only for very small values of the update parameter ϵ. With the transform in place, the loss function derivative with respect to the mean is proportional to the variance, rendering the mean update stable for a much wider range of values of the update parameter. This can be understood intuitively by considering that the output values of a Gaussian are less sensitive to changes in the mean value in a dimension for which the variance is large than for changes in the mean value in a dimension for which the variance is small. Finally, the transformation of σ_{jkl} maintains the constraint that $\sigma_{jkl} > 0$ and also greatly reduces the sensitivity of the update for small values of the variance- thereby also helping to render the update stable for a greater range of update parameters. The transformation of σ_{jkl} essentially implies that it

is updated multiplicatively, making the magnitude of the update relative to the magnitude of σ_{jkl}.

Gradient descent update equations

From Equation 4.10 the following gradient descent update equations are derived (similar to [28]) for the transformed parameters belonging to class i:

$$\bar{\mu}_{jkl}^{(i)}(n+1) = \bar{\mu}_{jkl}^{(i)}(n) - \epsilon_n \sum_{o=1}^{O}\sum_{c=1}^{M} 1(\mathbf{X}_o \in C_c) \frac{\partial l_c(\mathbf{X}_o;\mathbf{\Lambda})}{\partial \bar{\mu}_{jkl}^{(i)}}\Big|_{\mathbf{\Lambda}=\mathbf{\Lambda}_n} \tag{4.14}$$

$$\bar{\sigma}_{jkl}^{(i)}(n+1) = \bar{\sigma}_{jkl}^{(i)}(n) - \epsilon_n \sum_{o=1}^{O}\sum_{c=1}^{M} 1(\mathbf{X}_o \in C_c) \frac{\partial l_c(\mathbf{X}_o;\mathbf{\Lambda})}{\partial \bar{\sigma}_{jkl}^{(i)}}\Big|_{\mathbf{\Lambda}=\mathbf{\Lambda}_n} \tag{4.15}$$

$$\bar{c}_{jk}^{(i)}(n+1) = \bar{c}_{jk}^{(i)}(n) - \epsilon_n \sum_{o=1}^{O}\sum_{c=1}^{M} 1(\mathbf{X}_o \in C_c) \frac{\partial l_c(\mathbf{X}_o;\mathbf{\Lambda})}{\partial \bar{c}_{jk}^{(i)}}\Big|_{\mathbf{\Lambda}=\mathbf{\Lambda}_n} \tag{4.16}$$

$$\bar{a}_{jj'}^{(i)}(n+1) = \bar{a}_{jj'}^{(i)}(n) - \epsilon_n \sum_{o=1}^{O}\sum_{c=1}^{M} 1(\mathbf{X}_o \in C_c) \frac{\partial l_c(\mathbf{X}_o;\mathbf{\Lambda})}{\partial \bar{a}_{jj'}^{(i)}}\Big|_{\mathbf{\Lambda}=\mathbf{\Lambda}_n}. \tag{4.17}$$

Calculation of derivatives

The derivatives of the loss with respect to the transformed parameters $\bar{\mu}_{jkl}$, $\bar{\sigma}_{jkl}$ and $\bar{c}_{jk}$, which appear in Equations 4.14 to 4.16, can be expanded as is now shown for the derivative of the mean (Equation 4.14) in the following equations [28]. First the derivative of the loss function (Equation 4.8) is expanded via the chain rule to include the misclassification

measure d (see also Equation 4.8)

$$\frac{\partial l_c(\mathbf{X}, \mathbf{\Lambda})}{\partial \bar{\mu}_{jkl}^{(i)}} = \frac{\partial l_c(\mathbf{X}, \mathbf{\Lambda})}{\partial d_c(\mathbf{X}, \mathbf{\Lambda})} \frac{\partial d_c(\mathbf{X}, \mathbf{\Lambda})}{\partial \bar{\mu}_{jkl}^{(i)}} \tag{4.18}$$

where

$$\frac{\partial l_c(\mathbf{X}, \mathbf{\Lambda})}{\partial d_c(\mathbf{X}, \mathbf{\Lambda})} = \gamma l_c(d_c(\mathbf{X}, \mathbf{\Lambda}))\big[1 - l_c(d_c(\mathbf{X}, \mathbf{\Lambda}))\big]. \tag{4.19}$$

Next the derivative of the misclassification measure (Equation 4.6) with respect to the transformed mean parameter is expanded, noting that it is dependent on whether $i = c$, i.e. whether the derivative is with respect to a true class or false class parameter (similar to [93])

$$\frac{\partial d_c(\mathbf{X}; \mathbf{\Lambda})}{\partial \bar{\mu}_{jkl}^{(i)}} = -1(i = c)\frac{\partial g_i(\mathbf{X}; \mathbf{\Lambda})}{\partial \bar{\mu}_{jkl}^{(i)}} + 1(i \neq c)\frac{e^{g_i(\mathbf{X};\mathbf{\Lambda})\eta}}{\sum_{f,f\neq c}^{M} e^{g_f(\mathbf{X};\mathbf{\Lambda})\eta}}\frac{\partial g_i(\mathbf{X}; \mathbf{\Lambda})}{\partial \bar{\mu}_{jkl}^{(i)}}. \tag{4.20}$$

Finally the derivative of the class discriminant function (Equation 4.12) with respect to the transformed mean parameter is expanded to include the observation log-probability derivative

$$\frac{\partial g_i(\mathbf{X}; \mathbf{\Lambda})}{\partial \bar{\mu}_{jkl}^{(i)}} = \sum_{t=1}^{T(\mathbf{X})}\sum_{j=1}^{N} 1(\bar{q}_t = j)\frac{\partial \log b_j^{(i)}(\mathbf{x}_t)}{\partial \bar{\mu}_{jkl}^{(i)}}. \tag{4.21}$$

In Equations 4.18 through 4.21 we have now detailed the procedure for calculating the derivative of the loss function with respect to the observation log-probability. All that remains is to calculate the derivatives of the log-observation probability function $\log b_j^{(i)}(\mathbf{x}_t)$ (Equation 4.13) with respect to the transformed parameters $\bar{\mu}_{jkl}^{(i)}$, $\bar{c}_{jk}^{(i)}$ and $\bar{\sigma}_{jkl}^{(i)}$. The derivatives are given by [28]:

$$\frac{\partial \log b_j^{(i)}(\mathbf{x}_t)}{\partial \bar{\mu}_{jkl}^{(i)}} = \frac{c_{jk}^{(i)}\mathcal{N}[\mathbf{x}_t, \boldsymbol{\mu}_{jk}^{(i)}, \mathbf{\Sigma}_{jk}^{(i)}]}{b_j^{(i)}(\mathbf{x}_t)}\left(\frac{x_{tl}}{\sigma_{jkl}^{(i)}} - \bar{\mu}_{jkl}^{(i)}\right), \tag{4.22}$$

$$\frac{\partial \log b_j^{(i)}(\mathbf{x}_t)}{\partial \bar{\sigma}_{jkl}^{(i)}} = \frac{c_{jk}^{(i)} \mathcal{N}[\mathbf{x}_t, \boldsymbol{\mu}_{jk}^{(i)}, \boldsymbol{\Sigma}_{jk}^{(i)}]}{b_j^{(i)}(\mathbf{x}_t)} \left[\left(\frac{x_{tl}}{\sigma_{jkl}^{(i)}} - \bar{\mu}_{jkl}^{(i)} \right)^2 - 1 \right] \tag{4.23}$$

and we calculate

$$\frac{\partial \log b_j^{(i)}(\mathbf{x}_t)}{\partial \bar{c}_{jk}^{(i)}} = c_{jk}^{(i)} \left[\frac{\mathcal{N}[\mathbf{x}_t, \boldsymbol{\mu}_{jk}^{(i)}, \boldsymbol{\Sigma}_{jk}^{(i)}]}{b_j^{(i)}(\mathbf{x}_t)} - 1 \right]. \tag{4.24}$$

The derivative of the loss function with respect to the transformed transition probability parameter $\bar{a}_{ij}$, which appears in Equation 4.17 is also given in a similar form to Equations 4.18-4.20. Finally the derivative of the class discriminant function with respect to the transformed transition probability is given by

$$\frac{\partial g_i(\mathbf{X}; \boldsymbol{\Lambda})}{\partial \bar{a}_{jj'}^{(i)}} = \sum_{t=1}^{T(\mathbf{X})} \sum_{s=1}^{N} 1(\bar{q}_{t-1} = j)\, 1(\bar{q}_t = s)[1(j' = s) - a_{jj'}^{(i)}]. \tag{4.25}$$

Detailed derivations of Equations 4.24 and 4.25 are given in Appendix C because our equations differ from those previously published [86], where mixture weight and transition probability parameter dependencies were not taken into account.

4.2.4 MCE training for HMMs

With the update equations fully specified, the training procedure is now discussed in more detail. MCE training is usually preceded by standard ML training of models, such as expectation-maximisation (EM) training. EM training has the desirable property that it guarantees increased data-likelihood during training and is less prone to converge to local optima than gradient-based techniques. In contrast, training using MCE does not guarantee decreased loss (it depends on the selection of a suitably small update parameter) and is prone to converge to local minima.

The gradient descent optimisation approach discussed in this section, also termed segmental

GPD training, is implemented as follows:

1. ML models are estimated using the EM algorithm,
2. observation sequences are aligned with the models, accumulating the derivative statistics (Equations 4.18 through 4.25),
3. transformed parameters are updated (Equations 4.14 through 4.17), and
4. the reverse transformation of the parameters completes the process, which is iteratively repeated from point 2.

One of the problems with using gradient descent optimisation on MCE is that over-training may occur and that, based on the training set only, there are no suitable stopping criteria. Previous research reporting results using MCE/GPD used a fixed number of iterations and a linearly decreasing update parameter that was determined empirically to work well [28]. There are, however, also other parameters such as the slope of the sigmoid γ and the offset of the sigmoid θ that need to be carefully selected as they influence the stability, speed of convergence and ultimately the recognition performance achieved with the method. These issues are discussed in Chapter 5 when the method is applied to cross-language adaptation.

An alternative training method for MCE, that uses the N-best candidates from a search, was suggested by Chou *et al.* [94]. The method is also known as string-level MCE [95] and is particularly useful for optimising continuous speech recognition performance when a large amount of data is available. We discuss string-level MCE in detail in Section 4.5.1, where we compare it with the standard approach that was detailed in this section (also termed phoneme-level or model-level MCE) when extended using a cost-based method for improving word recognition performance.

4.2.5 Applications

One of the first applications of the MCE approach for speech recognition used artificial neural networks to classify features in isolated word speech experiments [91]. Other early research on the application of MCE for speech recognition investigated improving the performance of dynamic time warping-based systems. The MCE criterion was employed for the discriminative optimisation of several parameters of dynamic time warping (DTW) systems, including trajectory weighting coefficients [96] and reference patterns [97, 98]. MCE was also applied to both DTW [99, 100] and HMM-based [101] word-spotting systems, as well as for utterance rejection [102]. MCE was applied for the optimisation of standard feature extraction parameters in speech recognition [103, 104] as well as for optimising dynamic (trajectory) features [105]. The unified framework that MCE provides for global optimisation of both the feature extraction front end, as well as the classification back end of a system was also researched [106, 107].

Applications that benefited from the use of MCE in the above-mentioned papers include vowel recognition, the E-set problem and connected digit recognition. Relatively few studies have targeted improving performance on continuous speech. An N-best-based MCE optimisation approach was shown to improve continuous speech recognition performance for the DARPA naval resource management (RM) task [94] compared to ML trained models. Another study [95] found string-level MCE to improve continuous phoneme recognition performance compared to ML trained models but found phoneme-level MCE to outperform string-level MCE for the specific task.

Adaptation using MCE

MCE has recently been applied specifically for the purpose of adapting pre-trained models to better fit new speech data. Matsui & Furui [29] compared the MAP and MCE techniques for adaptation of Gaussian mean and mixture weight parameters and found that the best results were obtained for a combination of the MAP and MCE methods. MAP was used to

find an initial estimate of the speaker adapted parameters and MCE was used to further fine-tune the results. The reason for first using MAP is that MCE is prone to converging to local minima and therefore achieved better results when starting with the improved MAP estimated models rather than with the speaker independent models. McDermott *et al.* [108] applied MCE to on-line adaptation and found it to outperform a segmental k-means approach. Laurila *et al.* [109] performed adaptation of only Gaussian mean parameters of HMMs to new speakers and environmental noise using MCE, MMI, MLLR and MAP methods. They report that MAP and MCE delivered very similar results and produced better recognition performance than the MMI and MLLR approaches.

In this section we detailed the basic approach to MCE optimisation of HMM parameters. In the following sections we propose a few specific extensions to MCE. We propose extending MCE to discriminatively adapt duration modelling parameters since it is expected that all parameters, including duration modelling parameters, may need to be optimised for a new language. A method for the discriminative optimisation of linear parameter transformations is proposed that may deliver better performance than ML estimated transformations. Finally, we propose a method to modify the MCE misclassification measure in order to associate a (language specific) cost with misclassifying a class as a certain other class. This enables MCE to focus on the adaptation of class boundaries that are important for recognition in the target language.

4.3 Discriminative optimisation of duration modelling parameters

Performance improvement may be obtained by the discriminative optimisation of the duration modelling parameters in addition to the discriminative optimisation of the HMM parameters described in Section 4.2. In Section 2.1.3 we detailed the approach to explicit duration modelling followed in this thesis. A gamma distribution function is used to model the distribution of the number of frames spent in each frame of the HMM. The parameters

of the gamma distributions, namely the α and β parameters, are simply estimated through their relationship to the expected mean and variance of the number of frames spent in each state. Use of this estimation procedure has been substantiated empirically, but it is not guaranteed to deliver optimal performance, especially since the true form of the duration p.d.f. is unknown.

In this section we therefore propose the discriminative optimisation of duration modelling parameters using the MCE framework and derive the equations for it. The state aligned HMM likelihood function in Equation 4.12 can easily be expanded to include explicit duration modelling and is then given by

$$g_i(\mathbf{X};\mathbf{\Lambda}) = +\sum_{t=1}^{T}[\log a^{(i)}_{\tilde{q}_{t-1}\tilde{q}_t} + \log b^{(i)}_{\tilde{q}_t}(\mathbf{x}_t)] + \sum_{j=1}^{N}\log \rho_j(\tau_j) \tag{4.26}$$

where τ_j is the number of discrete time frames spent in state j and $\log \rho_j()$ is the duration log-likelihood function in state j given by (refer to Equation 2.4)

$$\log \rho_j(\tau_j) = \alpha_j \log \beta_j - \log \Gamma(\alpha_j) + (\alpha_j - 1)\log \tau_j - \beta_j \tau_j. \tag{4.27}$$

We note that a model duration likelihood function can also be used in conjunction with the state duration likelihood function, but we have not incorporated a model duration likelihood function. Also the transition probability parameters can be left out if one considers them to be replaced by explicit duration modelling.

Considering the duration modelling parameters part of the HMM modelling parameter set Λ, gradient descent optimisation of the duration modelling parameters α and β in state j of model i is implemented by the update equations

$$\alpha_j^{(i)}(n+1) = \alpha_j^{(i)}(n) - \epsilon_n \sum_{o=1}^{O}\sum_{c=1}^{M} 1(\mathbf{X}_o \in C_c)\frac{\partial l_c(\mathbf{X}_o;\mathbf{\Lambda})}{\partial \alpha_j^{(i)}}\Big|_{\mathbf{\Lambda}=\mathbf{\Lambda}_n} \tag{4.28}$$

and

$$\beta_j^{(i)}(n+1) = \beta_j^{(i)}(n) - \epsilon_n \sum_{o=1}^{O}\sum_{c=1}^{M} 1(\mathbf{X}_o \in C_c) \frac{\partial l_c(\mathbf{X}_o;\mathbf{\Lambda})}{\partial \beta_j^{(i)}}\Big|_{\mathbf{\Lambda}=\mathbf{\Lambda}_n}. \tag{4.29}$$

The partial derivative of the loss function with respect to the class discriminant function is given by (refer to Equations 4.18-4.21)

$$\frac{\partial l_c(\mathbf{X},\mathbf{\Lambda})}{\partial g_i(\mathbf{X};\mathbf{\Lambda})} = \gamma l_c(d_c)\big[1 - l_c(d_c)\big]\Big[-1(i=c) + 1(i\neq c)\frac{e^{g_i(\mathbf{X};\mathbf{\Lambda})\eta}}{\sum_{f,f\neq c}^{M} e^{g_f(\mathbf{X};\mathbf{\Lambda})\eta}}\Big]. \tag{4.30}$$

The partial derivatives of the class discriminant function (Equation 4.26) with respect to the parameters α and β are given by

$$\frac{\partial g_i(\mathbf{X};\mathbf{\Lambda})}{\partial \alpha_j^{(i)}} = \log \beta_j^{(i)} - \frac{\Gamma'(\alpha_j^{(i)})}{\Gamma(\alpha_j^{(i)})} + \log \tau_j \tag{4.31}$$

and

$$\frac{\partial g_i(\mathbf{X};\mathbf{\Lambda})}{\partial \beta_j^{(i)}} = \frac{\alpha_j^{(i)}}{\beta_j^{(i)}} - \tau_j \tag{4.32}$$

where $\Gamma'(\alpha_j^{(i)})$ denotes the derivative of $\Gamma(\alpha_j^{(i)})$ with respect to $\alpha_j^{(i)}$ and is computed with numerical differentiation. Adaptation of the duration modelling parameters is thus relatively easily integrated into the MCE framework.

4.4 Discriminative optimisation of linear model transformations

Linear transformation for speech model adaptation usually follows the maximum likelihood approach, leading to the well known MLLR algorithm or variants of it. In contrast to this, when linear transformation is applied to the speech pre-processing or feature extraction

stage, linear discriminant analysis (LDA) [110], principal component analysis (PCA) [38] or discriminatively optimised linear transformations using MCE [107] are commonly used. We do not explore feature space reduction or discriminative optimisation of the feature extraction process, since the techniques are liable to be database, or at least language specific. We, however, are interested in the application of discriminative methods in the optimisation of the linear transformation of the HMM model parameters between languages, as this may improve on the performance of maximum likelihood transformation estimators.

Rathinavelu [111] proposed applying the MCE/GPD method in optimising the parameters of a linear transformation of the trajectory parameters of a non-stationary state HMM. We independently arrived at the same method for the transformation of the Gaussian mean components of a mixture observation density HMM. If the transformation of the Gaussian mean components is given by

$$\hat{\boldsymbol{\mu}}_{jk} = \mathbf{W}\boldsymbol{\mu}_{jk}, \tag{4.33}$$

the observation probability of a state in the transformed HMM becomes (for diagonal covariance)

$$b_j^{(i)}(\mathbf{x}_t) = \sum_{k=1}^{M} c_{jk}^{(i)} \mathcal{N}[\mathbf{x}_t, \mathbf{W}\boldsymbol{\mu}_{jk}^{(i)}, \boldsymbol{\Sigma}_{jk}^{(i)}] = \sum_{k=1}^{M} \frac{c_{jk}^{(i)}}{(2\pi)^{(D/2)} \prod_{l=1}^{D} \sigma_{jkl}^{(i)}} e^{-\frac{1}{2}\sum_{l=1}^{D}(\frac{x_{tl} - \mathbf{w}_l \mu_{jk}^{(i)}}{\sigma_{jkl}^{(i)}})^2}. \tag{4.34}$$

Derivation of the MCE loss function with respect to the transformation matrix $\mathbf{W}$ then proceeds in a similar fashion to Equations 4.18 through 4.21. These equations give the partial derivative of the loss function with respect to any parameter of the mixture distribution in terms of the derivative of the log-observation probability with respect to that parameter. Therefore all that remains is to compute the derivative of the state log-observation probability density with respect to the transformation matrix. This is given for the lth row of

W by

$$\frac{\partial \log b_j^{(i)}(\mathbf{x}_t)}{\partial \mathbf{w}_l} = \frac{1}{b_j^{(i)}(\mathbf{x}_t)} \frac{\partial}{\partial \mathbf{w}_l} \Big[\sum_{k=1}^{K} c_{jk}^{(i)} \mathcal{N}[\mathbf{x}_t, \mathbf{W}\boldsymbol{\mu}_{jk}^{(i)}, \boldsymbol{\Sigma}_{jk}^{(i)}] \Big]$$
$$= \frac{1}{b_j^{(i)}(\mathbf{x}_t)} \Big[\sum_{k=1}^{K} c_{jk}^{(i)} \mathcal{N}[\mathbf{x}_t, \mathbf{W}\boldsymbol{\mu}_{jk}^{(i)}, \boldsymbol{\Sigma}_{jk}^{(i)}] \, (\frac{x_{tl} - \mathbf{w}_l \boldsymbol{\mu}_{jkl}^{(i)}}{{\sigma_{jkl}^{(i)}}^2}) \boldsymbol{\mu}_{jk}^{(i)} \Big]. \quad (4.35)$$

Examination of Equation 4.35 shows that there is a ${\sigma_{jkl}^{(i)}}^2$ term in the denominator, which may cause the update to be unstable even for a small update parameter due to the extremely large range of gradient values associated with a small variance component. A solution to this is to reduce the quadratic form to first order, or even to drop the variance term in the denominator of the gradient altogether. This heuristic solution can be better expressed in the GPD framework proposed by Juang & Katagiri [91] that caters for a positive definite matrix U_n as part of the update equation, which is then given by

$$\mathbf{w}_l(n+1) = \mathbf{w}_l(n) - U_n \epsilon_n \sum_{o=1}^{O} \sum_{c=1}^{M} 1(\mathbf{X}_o \in C_c) \frac{\partial l_c(\mathbf{X}_o; \boldsymbol{\Lambda})}{\partial \mathbf{w}_l} \Big|_{\boldsymbol{\Lambda}=\boldsymbol{\Lambda}_n}. \quad (4.36)$$

Choosing U_n to be a diagonal scaling matrix, with the average component variance for the lth dimension taking the lth position on the diagonal, provides a way of normalising the influence of the variance in the update equation. Note that this could also have been used in the MCE Gaussian mean update equation (Equation 4.14), but was not necessary since using the transformed mean value has exactly the same effect.

A choice has to be made with respect to the initial value $\mathbf{W}(0)$ of the transformation matrix. Using an identity matrix presents one option, but in light of the tendency of the gradient descent procedure to converge to local optima, a better choice is perhaps to use an MLLR estimate for the initial value. The optimisation process, however, is perhaps not as sensitive to the initial value, number of iterations and update parameter values because of the simple nature of the linear transformation process and because relatively fewer parameters are optimised. Experiments with the discriminatively optimised linear transform used MLLR initial estimates for the transformation and achieved improved performance

over standard MLLR transformation. However, since both MLLR and the discriminatively optimised linear transform did not produce very good performance in isolation, experiments in Chapters 6 and 7 only detail MLLR-based transformation results, mainly for comparison with MLLR-MAP results.

In the next section we discuss modifying the MCE loss and misclassification measures to associate varying cost or loss with different misclassification errors. Associating a cost with a particular misclassification indicates the importance (or lack of importance) of the misclassification and can improve the performance achievable with discriminative phoneme model optimisation by focusing on phoneme errors that have a high probability of leading to word errors.

4.5 Cost-based MCE

The Bayesian framework for classifier design [25] allows for the specification of a cost or risk ζ_{ij} associated with classifying a sample from class i as belonging to class j. In this sense, the standard implementation of MCE uses only a true-false cost function, considering in the misclassification measure (Equation 4.6) only the class to which an observation belongs as the true class and treating all other classes equally as false classes. When the true goal of a classifier is to achieve minimum phoneme misclassification, use of a zero-one cost function makes sense. In this case the MCE loss function closely approximates the empirical misclassification rate and presents a suitable function for optimisation. Generally, however, the goal of a classifier may be better expressed in terms of a more useful property such as word accuracy in continuous speech, or even at a more abstract level, in terms of how accurately the meaning of a speech utterance is expressed by a recognised phrase. In this section we consider ways of improving the MCE loss function to more accurately reflect the goal of the classifier. We start off with a previously discussed approach that implements discriminative training by comparing competing hypotheses.

4.5.1 String-level MCE

In order to improve the performance of discriminative training techniques for continuous speech applications, research was performed by Chou *et al.* [92] on a minimum string error rate implementation of MCE using N-best candidate strings. Even though the method targets a string-level loss function, optimisation occurs at the subunit (word or phoneme model) level, thereby indirectly also optimising classification performance of the subunits. The method has been shown to work well for closed vocabulary problems such as connected digit recognition. This is to be expected because it implements a task dependent word error rate based minimisation that compares possible in-vocabulary errors to the correct alignment and computes the update accordingly. String-level MCE is very useful when speech data is available that has been transcribed, but not labelled, since alignment information is not needed for the method. The method also may have an advantage over the standard approach in that recognition units are automatically aligned in sentence context with each other during training and can therefore take into account insertion and deletion errors in addition to substitution errors. However, McDermott [95] points out that string-level MCE effectively only considers regions of the speech input frame where there are differences in segmentation between the correct and competing hypotheses. Since only a limited number of hypotheses are typically decoded for computational reasons, only limited regions of each input frame are used to increase discrimination, whereas with phoneme-level MCE adaptation, many or all competing hypotheses (single HMMs) are considered for every phoneme segment, thereby better utilising the available data to increase class separation.

String-level MCE will not necessarily deliver optimal performance at the subunit level, as has been found [95] for continuous phoneme recognition. This can be explained by considering exactly what the effect of string-level training on the basic modelling units are. In LVCSR systems, phoneme models are usually used as the building blocks for composite speech units such as words. String-level MCE uses an N-best search to find the best competing hypotheses that differ in terms of word sequence from the correct hypothesis. Adaptation occurs only for models associated with these strings, thus predicating the adap-

tation of phoneme models on word confusability. The cost in terms of word error rate for classifying one phoneme as another is thus determined based on the whole word training data and used to adjust the boundary between the two phonemes. This is, however, a simplified view of how the method works. Typically, an exact pronunciation dictionary is not available, and therefore a training speech utterance may not exactly match the phoneme sequence of the correct word sequence. The acoustic models are thus adapted to also exhibit phonemic properties. This happens anyway if forced-alignment training is done, and can surely improve task specific word recognition performance, but care should be taken in predicting performance a task with a different vocabulary or grammar.

We have now discussed how string-level training can use a string and thus in effect a word error-based cost in adapting phoneme models. The method we discuss next shows a way of directly integrating a phoneme misclassification cost into the MCE framework, without having to perform an N-best search-based word level alignment.

4.5.2 Incorporating cost into the loss function

Reasons for applying modification of the phoneme models at the phoneme level rather than at the string or word level include:

- greater efficiency is achieved with (phoneme) model-level MCE versus string (word) level MCE,
- better vocabulary independence can be achieved because the cost matrix can be manipulated directly and is not dependent on the training speech utterances and
- the adaptation of each model can be controlled separately.

If the cost ζ_{ij} associated with the classification of a sample from class i as being of class j is known and is treated as a risk, the design of a minimum risk classifier in the Bayes sense can be attempted (determination of ζ_{ij} is the topic of a following section).

McDermott [86] suggested a modification of the MCE loss function to integrate a model-level cost function that ensures that the overall loss function reflects the empirical risk. For each training token the method weights the contribution from each false class by the risk associated with that false class. To incorporate the method into the MCE misclassification measure, the weight of each incorrect class j with respect to the total contribution of the incorrect classes is first expressed as

$$\omega_j(\mathbf{X};\mathbf{\Lambda},c) = \frac{e^{g_j(\mathbf{X};\mathbf{\Lambda})\eta}}{\sum_{f,f\neq c}^{M} e^{g_f(\mathbf{X};\mathbf{\Lambda})\eta}}, \tag{4.37}$$

where the correct class is c. The cost-based loss function can then be expressed in terms of the cost ζ_{cj} and the contribution $\omega_j(\mathbf{X};\mathbf{\Lambda},c)$ of each incorrect class, summed over all the incorrect classes by

$$l_c^*(\mathbf{X};\mathbf{\Lambda}) = \Big[\sum_{j,j\neq c}^{M} \zeta_{cj}\,\omega_j(\mathbf{X};\mathbf{\Lambda},c)\Big]\; l_c(\mathbf{X};\mathbf{\Lambda}). \tag{4.38}$$

Figure 4.1 shows graphically for a two class problem how the loss varies as a function of the position of an observed value, when different cost values are associated with the misclassification. The loss is a function of the relative correct versus incorrect class likelihoods and also the cost associated with misclassifying the correct class as the incorrect class. Since Equation 4.38 expresses the overall empirical risk, optimisation of the equation minimises the risk. If suitable estimates of the individual risks ζ_{cj} are available, the method may approximate minimum risk classification.

Unfortunately, no further details of the implementation of the method were published in [86]. Since we are interested in implementing the method we compute the derivative of the cost-based loss function $l_c^*(\mathbf{X};\mathbf{\Lambda})$ with respect to the class discriminant function $g_i(\mathbf{X};\mathbf{\Lambda})$,

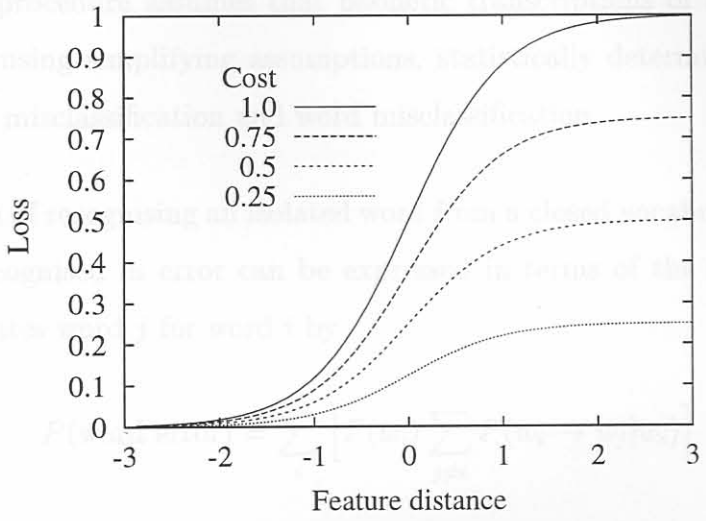

Figure 4.1: The loss l_c^* incurred as a function of the position of an observed value for various values of the misclassification cost when the Gaussian mean of the correct class is at -1 and the Gaussian mean of an incorrect class is at 1 and both Gaussians have unity variance

which is then given by

$$\frac{\partial l_c^*(\mathbf{X};\mathbf{\Lambda})}{\partial g_i(\mathbf{X};\mathbf{\Lambda})} = 1(i \neq c)\eta\omega_i(\mathbf{X};\mathbf{\Lambda},c)\left[\zeta_{ci} - \sum_{j,j\neq c}^{M} \zeta_{cj}\omega_j(\mathbf{X};\mathbf{\Lambda},c)\right] l_c(\mathbf{X};\mathbf{\Lambda}) + \left[\sum_{j,j\neq c}^{M} \zeta_{cj}\omega_j(\mathbf{X};\mathbf{\Lambda},c)\right]\frac{\partial l_c(\mathbf{X};\mathbf{\Lambda})}{\partial g_i(\mathbf{X};\mathbf{\Lambda})}, \quad (4.39)$$

where the partial derivative of the loss function with respect to the class discriminant function ($\frac{\partial l_c(\mathbf{X};\mathbf{\Lambda})}{\partial g_i(\mathbf{X};\mathbf{\Lambda})}$) is given by Equation 4.30. Since the derivation of the derivative of the discriminant function with respect to HMM, duration and transformation parameters have been given previously in this chapter, the process is now fully specified, except for the determination of the cost function itself, which is discussed next.

4.5.3 Estimating cost based on word error

We formulate a rather simple procedure for estimating the cost of misclassification for each phoneme pair that is based on the probability of the misclassification leading to a

word error. The procedure assumes that phonetic transcriptions of a large set of words are available and using simplifying assumptions, statistically determines the relationship between phoneme misclassification and word misclassification.

Given the problem of recognising an isolated word from a closed vocabulary, the probability that a word is recognised in error can be expressed in terms of the probability that the recogniser substitutes word j for word i by

$$P(\text{word error}) = \sum_i \left[P(w_i) \sum_{j \neq i} P(w_i \rightarrow w_j | w_i) \right] \tag{4.40}$$

where $P(w_i \rightarrow w_j | w_i)$ denotes the conditional probability that the substitution takes place and $P(w_i)$ is the *a priori* occurrence of word i. The next step is to condition the probability of a word substitution $w_i \rightarrow w_j$ on a specific phoneme substitution $\alpha_k \rightarrow \alpha_l$. The problem is greatly simplified by considering for each word pair only phoneme errors that change the first word along the optimal alignment path of the word pair to look more like the second word, hereafter termed *cross-word* phoneme errors. We define the phoneme misclassification cost ζ_{kl} to be the probability of a word error given that a specific substitution $\alpha_k \rightarrow \alpha_l$ of phoneme α_k by α_l occurs by:

$$\begin{aligned}
\zeta_{kl} &= P(\text{word error} | \alpha_k \rightarrow \alpha_l) \\
&= \frac{P(\text{word error}, \alpha_k \rightarrow \alpha_l)}{P(\alpha_k \rightarrow \alpha_l)} \\
&= \frac{\sum_i \left[P(w_i) \sum_{j \neq i} P(w_i \rightarrow w_j | w_i, \alpha_k \rightarrow \alpha_l)\, 1(\alpha_k \rightarrow \alpha_l \text{ in} \{w_i, w_j\}) \right]}{\sum_i \left[P(w_i) \sum_{j \neq i} 1(\alpha_k \rightarrow \alpha_l \text{ in} \{w_i, w_j\}) \right]} \\
&= \frac{\sum_i \left[P(w_i) \sum_{j \neq i} P(\text{word error} | \mathrm{d}(w_i, w_j), \#\text{substitutions} \geq 1)\, 1(\alpha_k \rightarrow \alpha_l \text{ in} \{w_i, w_j\}) \right]}{\sum_i \left[P(w_i) \sum_{j \neq i} 1(\alpha_k \rightarrow \alpha_l \text{ in} \{w_i, w_j\}) \right]}
\end{aligned} \tag{4.41}$$

where $1(\alpha_k \rightarrow \alpha_l \text{ in} \{w_i, w_j\})$ is 1 when the $\alpha_k \rightarrow \alpha_l$ substitution match occurs in the optimal alignment of w_i and w_j and 0 otherwise, $\mathrm{d}(w_i, w_j)$ is the number of insertions, deletions and substitutions in the optimal alignment of w_i and w_j and the probability of a

word error, given the distance between the words and the fact that at least one substitution takes place is given by $P(\text{word error}|\text{d}(w_i|w_j), \#\text{substitutions} \geq 1)$.

The probability of a word error is thus defined to depend only on the phonetic distance $\text{d}(w_i, w_j)$ between the words, i.e. on the number of insertions, deletions and substitutions necessary to convert one word to the other. Independence of the cross-word phoneme errors is assumed and the number of cross-word phoneme errors then assumes a binomial distribution. When more than 50% of cross-word phoneme errors occur, a word substitution is assumed to take place and when exactly 50% of the cross-word phoneme errors occur, a 50% chance of a word substitution error is assumed. The probability of a word substitution is expressed by the summation of the binomial probabilities that half or more of the cross-word phoneme errors occur, taking into account that at least one phoneme error has occurred. The word substitution probability can then be expressed in terms of the inter-word phonetic distance n and the cross-word phoneme error probability p by the function

$$b(n,p) = \begin{cases} \sum_{m=\lceil n/2 \rceil}^{n} \binom{n-1}{m-1} p^{m-1}(1-p)^{n-m} & n \text{ odd} \\ 1/2\binom{n-1}{n/2-1} p^{n/2-1}(1-p)^{n/2} + \sum_{m=n/2+1}^{n} \binom{n-1}{m-1} p^{m-1}(1-p)^{n-m} & n \text{ even.} \end{cases} \tag{4.42}$$

Figure 4.2 shows graphically how the word substitution probability varies as a function of the inter-word phonetic distance for a number of cross-word phoneme error probabilities. Since cross-word phoneme errors only include phoneme errors that change the first word according to the optimal alignment path with the second word, the use of a relatively small value for the cross-word phone error probability is therefore applicable. We selected to use a value of 0.1 for the cross-word phoneme error probability.

Conditioning the probability of a word error only on phoneme substitutions (Equation 4.41) is perhaps too simplistic. We therefore extended the method to also consider the effect of phoneme insertions and deletions, by regarding them in the same way as substitutions. An insertion before or after α_k of α_l is considered a possible misclassification of speech

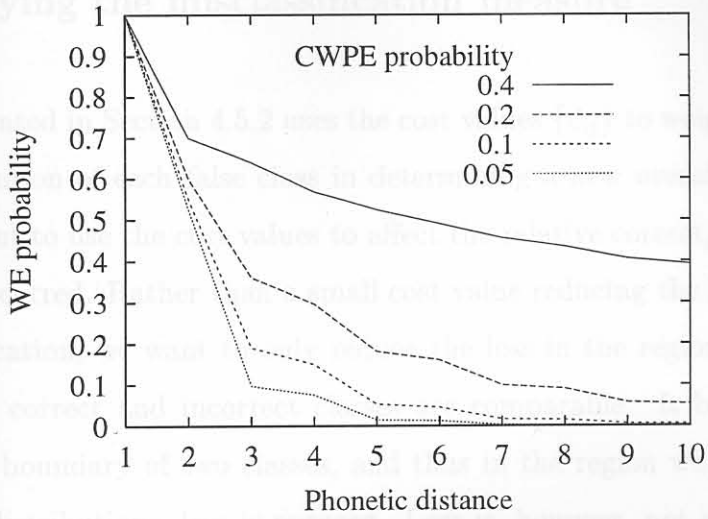

Figure 4.2: The word substitution error (WE) probability as a function of the inter-word phonetic distance for a number of values of the cross-word phoneme error (CWPE) probability, given that at least one substitution has occurred

data corresponding to α_k as α_l. Similarly, the deletion of α_k before or after α_l is also considered a possible misclassification of speech data corresponding to α_k as α_l. By associating a probability of $\frac{1}{2}$ with the mentioned insertions or deletions being caused by a $\alpha_k \rightarrow \alpha_l$ misclassification, the probabilities that these phoneme errors lead to word errors are incorporated into Equation 4.41 by the following equation:

$$\zeta_{kl} = P(\text{word error}|\alpha_k \rightarrow \alpha_l) + \frac{1}{2}P(\text{word error}|\alpha_k \uparrow \alpha_l) + \frac{1}{2}P(\text{word error}|\alpha_k \downarrow \alpha_l), \quad (4.43)$$

where $\alpha_k \uparrow \alpha_l$ means that α_l is inserted before or after α_k and $\alpha_k \downarrow \alpha_l$ means that α_k is deleted before or after α_l. Both $P(\text{word error}|\alpha_l \uparrow \alpha_k)$ and $P(\text{word error}|\alpha_k \downarrow \alpha_l)$ are computed in the same way as for $P(\text{word error}|\alpha_k \rightarrow \alpha_l)$ (Equation 4.41).

This concludes our discussion of the estimation of the word error-based phoneme misclassification cost. The resulting cost matrix can be used with the loss function method discussed in Section 4.5.2, or can be used with an alternative method that we discuss in the following section.

4.5.4 Modifying the misclassification measure

The method presented in Section 4.5.2 uses the cost values $\{\zeta_{ij}\}$ to weight for each training token the contribution of each false class in determining a new overall loss function. We may, however, want to use the cost values to affect the relative correct/incorrect likelihood at which loss is incurred. Rather than a small cost value reducing the total loss associated with a misclassification, we want to only reduce the loss in the region where the relative likelihoods of the correct and incorrect classes are comparable. It basically means that near the decision boundary of two classes, and thus in the region where overlap may occur between the distributions, loss is reduced. Loss is, however, not significantly reduced when the incorrect class likelihood is much higher than the correct class likelihood. This effectively shifts the loss function towards the incorrect class as the cost associated with a misclassification becomes lower.

Cost-based misclassification measure

In order to achieve the above, we present an approach to integrate the cost function (ζ_{kl}) into the MCE framework, based on a modification of the misclassification measure. The decision boundary of the misclassification measure is shifted with the value of the cost function. The new misclassification measure can then be expressed by

$$d_i^{\dagger}(\mathbf{X};\mathbf{\Lambda}) = -g_i(\mathbf{X};\mathbf{\Lambda}) + \log\left[\frac{1}{M-1}\sum_{j,j\neq i}^{M} e^{(\log\zeta_{ij}+g_j(\mathbf{X};\mathbf{\Lambda}))\eta}\right]^{1/\eta}. \tag{4.44}$$

The log cost $(\log\zeta_{ij})$ is added to the log-likelihood function, which is equivalent to multiplication of the likelihood function by the linear cost. Figure 4.3 shows graphically for a two class problem how the loss varies as a function of the position of an observed value, when different cost values are associated with the misclassification. It can be seen that the effect of the cost is to shift the loss function towards the incorrect class for lower cost.

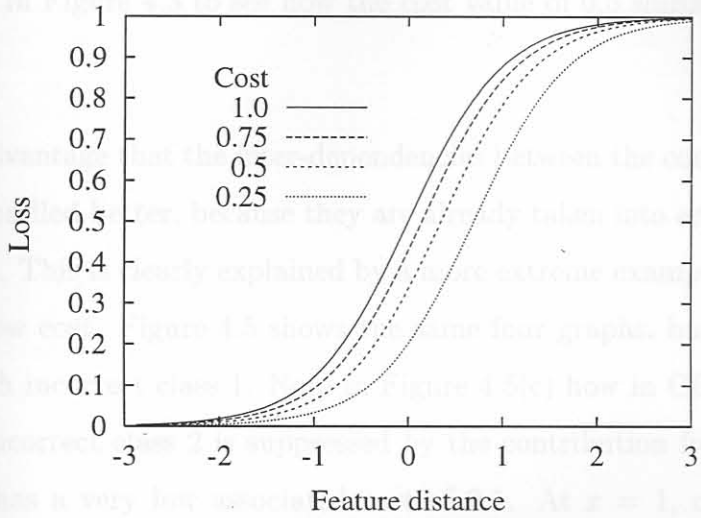

Figure 4.3: The loss $l_c^\dagger$ incurred as a function of the position of an observed value for various values of the misclassification cost used in the misclassification measure, when the Gaussian mean of the correct class is at -1 and the Gaussian mean of an incorrect class is at 1 and both Gaussians have unity variance

Comparison of modified misclassification measure with modified loss function approach

It is interesting to compare the working of the two approaches (Equations 4.44 and 4.38) to incorporating cost into the loss function. We refer to the approach of Section 4.5.2 that uses the cost to weight the loss function (4.38) as the cost-based loss function (CBLF) approach and to our approach (4.44) as the cost-based misclassification measure (CBMM) approach.

Figure 4.4 shows a three class problem with one correct class centred at -1 and two incorrect classes, centred at 1 and 3 and associated with misclassification costs of 0.5 and 1.0 respectively. Figure 4.4(a) shows the three Gaussian distributions, as well as the "average" incorrect class value as expressed by Equation 4.7 in the linear domain. Figure 4.4(b) also shows the three Gaussian distributions, but multiplied by their respective costs in the linear domain, as is effectively performed by CBMM in Equation 4.44 (by adding log-cost in the log domain). Figure 4.4(c) shows the shape of the loss function for the distributions in (a) achieved with CBLF (Equation 4.38). Figure 4.4(d) shows the loss CBMM associates with the class likelihood functions in (b). The contribution from incorrect class 1 can be

compared to that in Figure 4.3 to see how the cost value of 0.5 shifts the loss to the right in this case.

CBMM has the advantage that the inter-dependencies between the contribution of different false classes are handled better, because they are already taken into account in the misclassification measure. This is clearly explained by a more extreme example, in which incorrect class 1 has very low cost. Figure 4.5 shows the same four graphs, but with a cost of only 0.1 associated with incorrect class 1. Note in Figure 4.5(c) how in CBLF the contribution of the loss from incorrect class 2 is suppressed by the contribution from incorrect class 1, although class 1 has a very low associated cost of 0.1. At $x = 1$, one expects a loss in the region of 0.5 because the point is halfway between the true class and incorrect class 2, yet incorrect class 1 suppresses the loss. Figure 4.5(d) shows how in the CBMM approach the loss attributed to class 2 is only slightly suppressed by incorrect class 1 because the dependency between the incorrect classes in the misclassification measure is handled better.

Cost and reward-based misclassification measure

The goal of phoneme-level discriminative training should be the improvement of the overall system, of which the word error rate is a reasonably good measure. The method for estimating a word error-based phoneme misclassification cost and the integration of it into the discriminative training of phonemes provides a step in the right direction. String-level training, as we have discussed before, goes even a step further because it performs a degree of phonemic training - i.e. training based on what was supposed to have been said rather than for what was actually said. The method we propose next attempts to incorporate some phonemic information in the discriminative training procedure, while at the same time reducing the overall loss and thus the degree of adaptation that will occur.

The first step in the procedure is the extension of the word error-based phoneme misclassification cost estimation procedure to include a reward (negative cost) for a misclassification that may improve the word recognition rate. This is possible because different phonetic

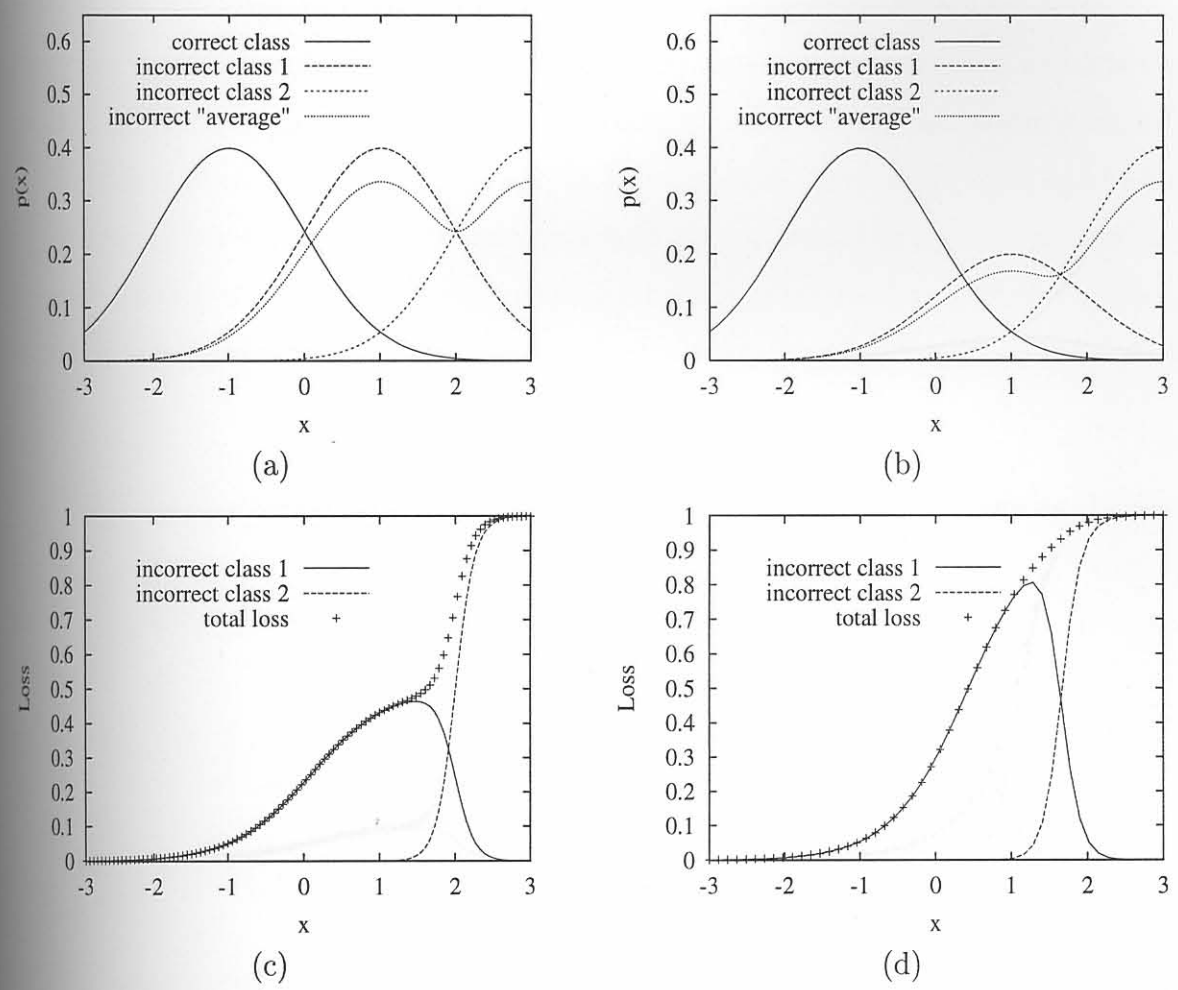

Figure 4.4: A comparison of two methods for computing the loss for a three class problem with a correct Gaussian mean at -1 and incorrect Gaussian means at 1 and 3 and with associated misclassification costs of 0.5 and 1 respectively, showing (a) the three distributions along with the "average" of the two incorrect classes, (b) the three distributions, modified according to the CBMM approach along with the "average" of the two incorrect classes, (c) the loss function according to the CBLF approach and (d) the loss function according to the CBMM approach

variants of the same word may occur in practice while the pronunciation dictionary for the word contains only a subset of the possibilities. The misclassification of one or more phonemes corresponding to the actual speech as the phonemes from the pronunciation dictionary may thus improve the overall word recognition rate. A procedure to estimate the expected reward associated with such misclassification can be derived by modifying

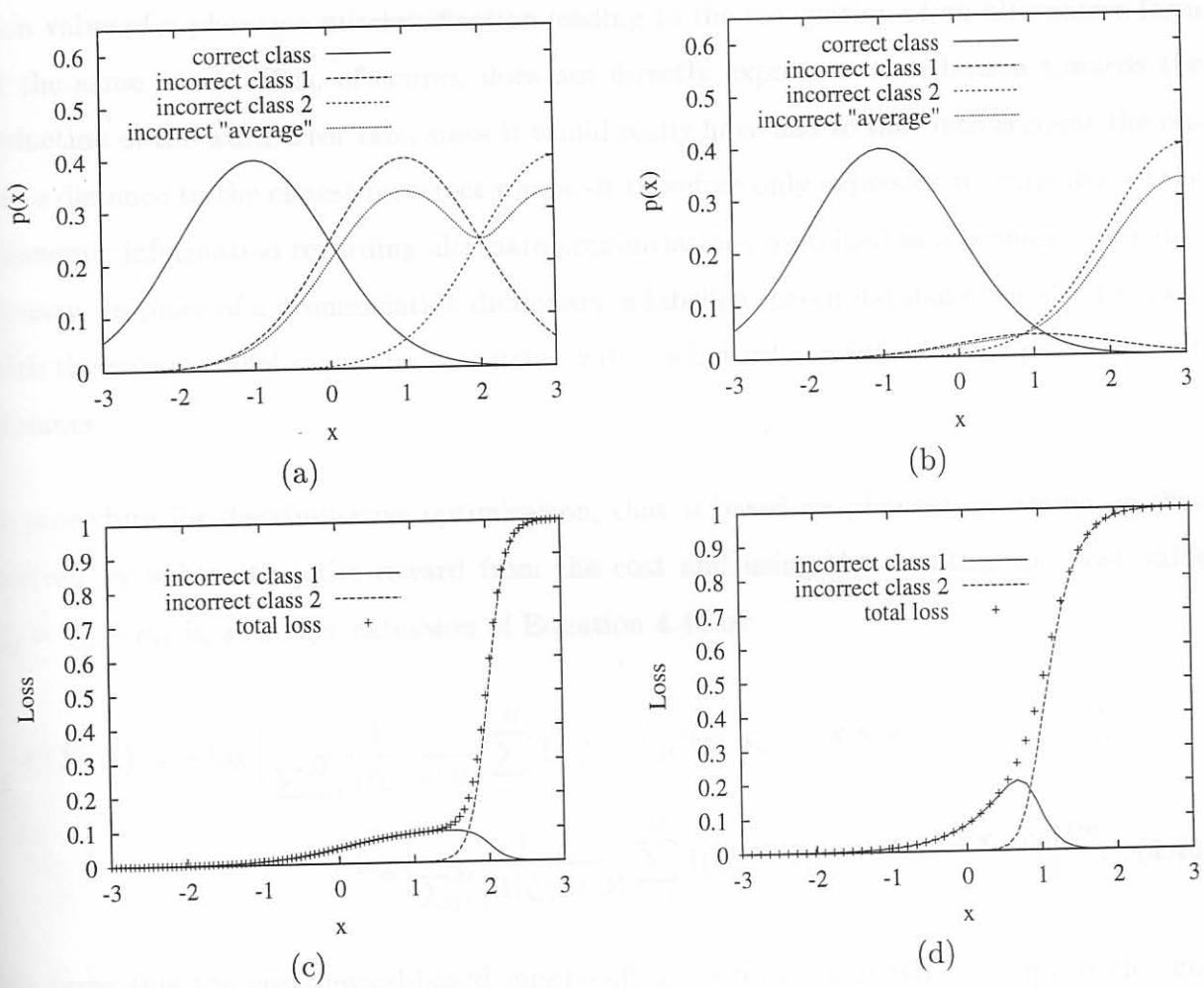

Figure 4.5: A comparison of two methods for computing the loss for a three class problem with a correct Gaussian mean at -1 and incorrect Gaussian means at 1 and 3 and with associated misclassification costs of 0.1 and 1 respectively, showing (a) the three distributions along with the "average" of the two incorrect classes, (b) the three distributions, modified according to the CBMM approach along with the "average" of the two incorrect classes, (c) the loss function according to the CBLF approach and (d) the loss function according to the CBMM approach

Equation 4.41 in the following way:

$$\kappa_{kl} = \frac{\sum_i \left[P(w_i) \sum_{j \neq i, w_j \in \Omega_i} P(\text{word subst.}|\mathrm{d}(w_i, w_j), \#\text{subst.} \geq 1)\, 1(\alpha_k \rightarrow \alpha_l \text{ in} \{w_i, w_j\}) \right]}{\sum_i \left[P(w_i) \sum_{j \neq i} 1(\alpha_k \rightarrow \alpha_l \text{ in} \{w_i, w_j\}) \right]} \tag{4.45}$$

where Ω_i represents the phonetic variants of word i. Equation 4.45 estimates the expecta-

tion value of a phoneme misclassification leading to the recognition of an alternative form of the same word. This, of course, does not directly express a contribution towards the reduction of the word error rate, since it would really have had to take into account the relative distance to the closest incorrect words. It therefore only expresses to some degree the phonemic information regarding alternate pronunciations contained in a pronunciation dictionary. In place of a pronunciation dictionary, a labelled speech database can also be used, with the various label sequences associated with each word considered to be pronunciation variants.

A procedure for discriminative optimisation, that is based on phonemic training, is then derived by subtracting the reward from the cost and using the resulting net cost value $\zeta^*_{ij} = \zeta_{ij} - \kappa_{ij}$ in a further extension of Equation 4.44 by

$$d_i^{\dagger}(\mathbf{X};\mathbf{\Lambda}) = -\log\left[\frac{1}{\sum_{j=1}^{M} 1(\zeta^*_{ij}<0)}\sum_{j=1}^{M} 1(\zeta^*_{ij}<0)e^{(\log(-\zeta^*_{ij})+g_j(\mathbf{X};\mathbf{\Lambda}))\eta}\right]^{1/\eta} + \log\left[\frac{1}{\sum_{j=1}^{M} 1(\zeta^*_{ij}\geq 0)}\sum_{j=1}^{M} 1(\zeta^*_{ij}\geq 0)e^{(\log(-\zeta^*_{ij})+g_j(\mathbf{X};\mathbf{\Lambda}))\eta}\right]^{1/\eta}. \quad (4.46)$$

We term this the cost-reward-based misclassification measure (CRBMM) approach. For phoneme pairs with a net reward or negative net cost, i.e. $\zeta^*_{kl} < 0$, values of the parameters of model j will be adapted to increase the likelihood of observations from class k, thereby effecting phonemic training. It should, however, be noted that this approach reduces the empirical loss and therefore less adaptation will likely take place than for zero-one cost functions. The reason why this approach works may thus be rooted not only in the fact that it performs phonemic training, but in the fact that it reduces the loss associated with errors that have some positive or little negative effect, thereby stopping the MCE approach from changing ML estimated models to enforce rigid acoustic separation between phonetic classes.

The working of the method is shown in Figure 4.6 for the same three class problem with Gaussians centred at -1, 1 and 3, but with an associated net cost of -1, -0.1 and 1.0

respectively. Figure 4.6(a) shows that the net reward of 1.0 (net cost of -1.0) associated with correct class 1 effectively means that its likelihood function is unaffected, while the 0.1 net reward associated with correct class 2 means that its likelihood function is multiplied by 0.1 in the linear domain. The likelihood function of class 3 (the incorrect class) is unaffected by its cost of 1.0. The "average" likelihood of the classes with net reward (correct classes) is compared to the net loss class likelihood in the misclassification measure. Figure 4.6(b) shows the total loss incurred, as well as the portion of the total loss attributed (in the component derivatives) to class 1 and class 2. It can be seen that for large x, loss is attributed to class 2, which is then adapted rather than class 1, which has almost no contribution to the loss for large x.

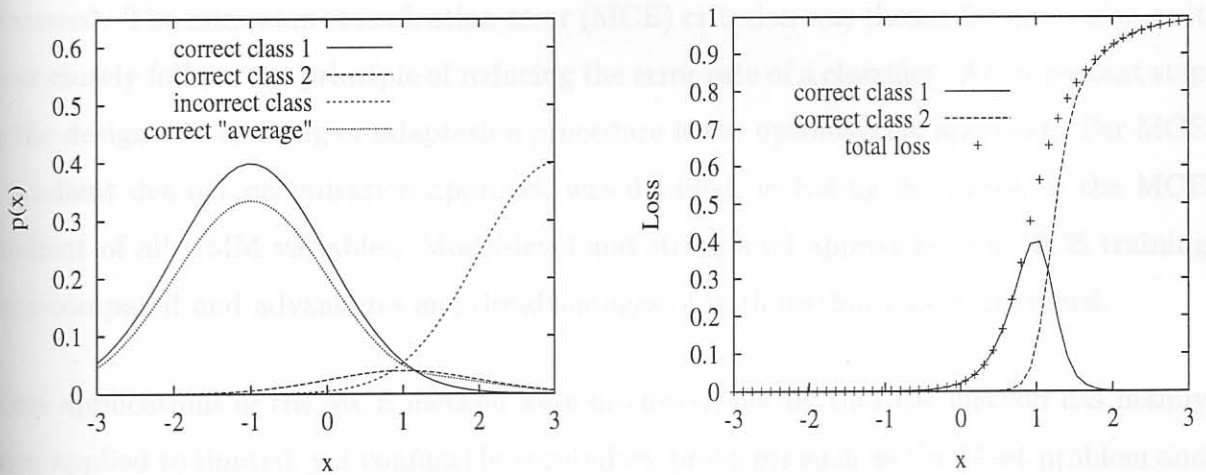

Figure 4.6: The CRBMM approach for computing the loss for a three class problem with Gaussian means at -1, 1 and 3 and with net rewards of 1.0 and 0.1 and a net loss of 1.0 respectively, indicating in (a) the three weighted distributions along with the average of the classes with net reward and in (b) the loss function as well as the contributions to the loss function by each of the classes with net reward

As far as the implementation of the procedure is concerned, the only derivation that changes is that of the misclassification measure. The derivative of the modified misclassification measure with respect to the class discriminant function becomes

$$\frac{\partial d_c^{\dagger}(\mathbf{X},\mathbf{\Lambda})}{\partial g_i(\mathbf{X};\mathbf{\Lambda})} = -\frac{1(\zeta_{ci}^* < 0)e^{[\log(-\zeta_{ci}^*)+g_i(\mathbf{X};\mathbf{\Lambda})]\eta}}{\sum_{j=1}^{M} 1(\zeta_{cj}^* < 0)e^{[\log(-\zeta_{cj}^*)+g_j(\mathbf{X};\mathbf{\Lambda})]\eta}} + \frac{1(\zeta_{ci}^* \geq 0)e^{[\log(-\zeta_{ci}^*)+g_i(\mathbf{X};\mathbf{\Lambda})]\eta}}{\sum_{j=1}^{M} 1(\zeta_{cj}^* \geq 0)e^{[\log(-\zeta_{cj}^*)+g_j(\mathbf{X};\mathbf{\Lambda})]\eta}}. \tag{4.47}$$

Note that this equation is also valid for the derivative of the misclassification measure of the CBMM approach (Equation 4.44) with respect to $g_i(\mathbf{X};\mathbf{\Lambda})$ by setting the correct class cost ζ_{ii} to -1.

4.6 Discussion

In this chapter we discussed the application of discriminative learning methods for the purpose of training and adapting parameters of speech recognition systems, continuous density HMMs in particular. The effect of the optimisation criterion on classifier design was discussed. The minimum classification error (MCE) criterion was chosen for discussion as it most closely follows the principle of reducing the error rate of a classifier. An important step in the design of a training or adaptation procedure is the optimisation approach. For MCE a gradient descent optimisation approach was detailed, including derivation of the MCE gradient of all HMM variables. Model-level and string-level approaches for MCE training were compared and advantages and disadvantages of both methods were discussed.

Some applications of the MCE method were discussed, noting that the method has mainly been applied to limited, yet confusable vocabulary problems such as the E-set problem and connected digit recognition. Only a limited number of medium to large vocabulary continuous speech recognition applications of MCE have been published. Adaptation performance of MCE was also discussed, with research indicating that better model initialisation, such as achieved by first performing MAP estimation, improves performance achieved with MCE and is better than ML adaptation in isolation.

Extensions to the standard MCE framework were presented, including discriminative adaptation of duration modelling variables, discriminative linear parameter transformation and word error-based phoneme adaptation approaches. The reason why the adaptation of all parameters, rather than say only Gaussian mean parameters are considered, is that cross-language adaptation may require significant adaptation, compared to perhaps the fine tun-

ing of models for a specific speaker. Alternative approaches for incorporating cost into the MCE framework were compared and an approach that also utilises reward in the misclassification measure was presented. The cost-based framework is of specific importance for cross-language adaptation since the phoneme inventory, context and acoustic separation between phonemes differ significantly between languages and adaptation should be able to address these issues efficiently for the target language.

In the next chapter we treat the issues involved in applying the techniques from speaker adaptation (Chapter 3) and discriminative learning (this chapter) for cross-language acoustic adaptation in detail.

Chapter 5

Cross-language acoustic adaptation issues

In this chapter we discuss the framework for the experiments detailed in the following two chapters. Practical aspects regarding the cross-language use of algorithms detailed in the previous two chapters are covered as part of this framework. Language and database issues are also discussed to place the experiments that were performed in the proper perspective.

Cross-language use of acoustic information attempts to exploit the acoustic-phonetic similarities between languages. These similarities are evident from the use of international phonetic inventories, such as the International Phonetic Alphabet[1] (IPA) and Speech Assessment Methods Phonetic Alphabet[2] (SAMPA), that serve to classify the sounds of many languages. There are still, however, differences with respect to the acoustic properties of sounds from different languages that share the same labels. Also, often a target language may contain sounds that do not occur in languages for which large databases are available. Labelling conventions, recording conditions and the type of speech recorded may also differ between databases, making cross-language and cross-database use of acoustic information

[1] http://www2.arts.gla.ac.uk/IPA/fullchart.html
[2] http://www/phon.ucl.ac.uk/home/sample/home.html

a formidable task. Aspects regarding language and database specific issues are discussed in this chapter to facilitate cross-language use of acoustic information.

We consolidate the application of methods discussed in the previous three chapters for using acoustic information across language boundaries. The strategies that have been used in previous research are (i) training on pooled multilingual data and (ii) adapting models trained on one language using data from another. There are, however, as we shall discuss in this chapter, other strategies that can be followed such as (iii) training models on pooled source and target language data and then adapting the models using only target language data and (iv) cross-language transformation of source data followed by training on the pooled target and transformed source data.

In addition to the different strategies for cross-language adaptation that will be discussed, the algorithms used for adaptation or transformation should also be examined to use them efficiently for the specific purpose. In previous chapters we already proposed new techniques to improve cross-language adaptation performance, such as the MSE log-variance transformation, the discriminative adaptation of duration parameters and use of a word error-based cost function in discriminative phoneme adaptation. These techniques are not part of the published repertoire of speaker adaptation algorithms, perhaps because speaker adaptation differs essentially from cross-language adaptation. In this chapter we therefore reconsider how to apply the standard adaptation methods together with our extensions thereof for cross-language adaptation.

5.1 Language and database issues

The acoustic-phonetic similarities between languages are well documented in international phonetic inventories. The existence of such standards is central to our goal of using acoustic information across language boundaries. The international phonetic inventories were developed by phoneticians using expert phonetic knowledge, however, and not by using acoustic

measurements or statistical techniques. There is thus no guarantee that these phonetic inventories represent an optimal classification of the acoustic properties of speech in different languages. Indeed, even for the same phonetic category, the acoustical realisation of the phoneme may differ between different languages for a number of reasons such as [48]:

- different phonetic context due to different phoneme sequence statistics and different phoneme inventories,
- different speaking styles,
- different prosodic features and
- different allophonic variations.

According to the principle of sufficient acoustic separation, the set of sounds in a language are kept acoustically distinct by its speakers to make it easy to distinguish between the sounds. Because the phonetic inventories of languages differ, the positions of the boundaries between phonemes are language dependent. In spite of the differences between languages with respect to the characteristics of speech of the same phonetic category, we still expect reasonable overlap between the phoneme feature distributions of different languages and that the phonetic categories give a good indication as to how the overlap occurs. This reasoning is supported by empirical evidence from systems with explicitly multilingual phone sets. In the next section we examine how differences with respect to phoneme inventories and context may influence the usefulness of speech data.

5.1.1 Phonetic inventories and context

Cross-language use of acoustic-phonetic information is limited to the overlap or junction of the phoneme inventories of the languages. When considering cross-language use of acoustic-phonetic information, it is therefore important to attempt to find languages that are as similar as possible. This ensures that maximal overlap of phonetic inventory as well as

overlap of phonetic context between the languages occur. Significant overlap of phonetic context ensures that monophone models contain to a large degree the same built-in context information and also facilitates the cross-language use of data to train context dependent phoneme models. In this thesis exclusive use was made of two Germanic languages, namely English and Afrikaans, mainly because (i) suitable databases were available for these languages and (ii) research on South African languages are of particular interest to us. Two different databases were used, one containing both South African English and Afrikaans, and one containing American English only.

As far as the specific context of the source language databases are concerned, it is better if the source language databases are phonetically diverse and contain a large variety of contexts. This reduces the specialisation of the source language acoustics for specific contexts and may improve the performance of models for recognition tasks containing speech from an entirely different context [112], such as is typically expected for a cross-language task. In our case, both speech databases are phonetically rich and the American English database also has diverse context. The phonetic context of the bilingual database, however, is not very diverse as a large number of utterances of only 60 different sentences are used.

5.1.2 Labelling conventions

In order to exploit acoustic-phonetic information, phonetically labelled databases are necessary in both source and target languages. Consistent labelling conventions should be followed and the selection of the set of phoneme labels to use should be taken with care. Although an international phoneme inventory (such as IPA) may be used, a subset of the inventory is usually selected that covers the expected occurrence of phonemes in the speech of the database. Using a limited number of phoneme categories has the advantages of simplifying the labelling process and possibly reducing the number of incorrectly assigned labels. On the other hand, a small number of labels may group together phonetic categories, which separately could provide useful information. If a database is created for the explicit purpose of multilingual speech recognition, then use of a larger set of labels that

suitably covers the phonetic variety over the combination of languages may facilitate the development of multilingual and cross-lingual application of the database. The bilingual database that is used in this thesis is of this nature, using a consistent set of labels for both languages. This is very convenient because it enables experiments to better quantify the effect of actual acoustical differences between the languages rather than possible artifacts of labelling differences.

An important aspect with respect to the accuracy of a database is the extent to which the database is labelled using phonetic or phonemic considerations. The purpose of labelling is usually to assign phonetic categories on an acoustical basis to the speech. This implies that phonetic labelling is attempted. Phonetic labelling is, however, a difficult and tedious task and it is often easier for the person performing the labelling to assign a label to a sound segment on a phonemic basis, i.e. on what was supposed to have been said, rather than on what was actually said. Perhaps the easiest way to assign labels to a speech database is by forced alignment, delivering a purely phonemic segmentation of the speech. For instance, in bootstrapping procedures this is the only solution because phonetic labelling of the target language database is not done. When source language models trained with forced alignment are used for a closed vocabulary task, or even for an open vocabulary same-language application this may not have a severe effect beyond the loss of acoustic resolution. It may even improve recognition performance in continuous speech when an imperfect pronunciation model is used. However, for a cross-language task, the loss of acoustic resolution, coupled with the incorporation of incorrect (source) language specific phonemic information, is likely to degrade performance for target language applications. For both databases used in this thesis, a phonetic labelling approach was used.

5.1.3 Phonetic mapping

Generally, if identical labelling conventions are used in the creation of the source and target databases, no work needs to be done at the phonetic level in determining how to implement cross-language use of the data. This is also the case for the bilingual database

used in this thesis. Several multilingual speech recognition studies have demonstrated the efficient re-use of acoustic-phonetic information across multiple languages [19, 20]. It is when differences exist with respect to labelling, i.e. a one-to-one mapping of phonemes does not exist, that difficulties are encountered. Two approaches for determining how to use the acoustic-phonetic information across language boundaries are generally applied namely

- phonetic knowledge-based and
- distance measure-based

methods for pairing phonemes or groups of phonemes from multiple languages. In our case the process is somewhat simplified since only a one-way mapping from source language(s) to target language is desired. Both approaches were attempted in this thesis and it was found that the phonetic knowledge-based mapping approach delivered better performance than a distance measure-based mapping approach. The experiments and results are discussed in Section 7.1.

Phonetic knowledge-based mapping

Expert phonetic knowledge can be used to determine a mapping from the phonetic inventory of a source language database to the phonetic inventory of a target language database. As previously mentioned, this is the only viable approach for research on bootstrapping of target language models, since target language data is not labelled. Research on explicit multilingual phoneme-based recognition often also makes use of phonetically derived sharing of acoustic parameters e.g. [19, 49]. For this thesis a phonetic expert determined a mapping from the American English database to the bilingual database, details of which are given in Appendix B.

Many multilingual systems use phonetically derived categories with multilingual scope, but then use statistical procedures to select whether to consider the same phoneme in the

different languages as one or whether to model them separately [30]. The procedures have also been extended to the creation of generalised triphone models of arbitrary complexity by using a decision tree clustering approach with both language and context questions in the splitting procedure. These approaches were not followed because the amount of data was deemed to be too limited.

Distance measure-based mapping

Research has shown the use of a metric such as the Bhattacharyya distance

$$D_{\text{Bhat}} = \frac{1}{8}(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)^T \left[\frac{\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2}{2}\right]^{-1} (\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1) + \frac{1}{2}\log \frac{\left|\frac{\boldsymbol{\Sigma}_1+\boldsymbol{\Sigma}_2}{2}\right|}{\sqrt{|\boldsymbol{\Sigma}_1||\boldsymbol{\Sigma}_2|}} \tag{5.1}$$

to measure the distance between Gaussian distributions representing phone classes of the same language for clustering purposes [113]. This metric has been used in a multilingual context to merge arbitrary phonemes from multiple languages [20] to reduce the complexity of the multilingual models. This entailed computing the distance between the phoneme models of the different languages and merging phonemes for which the distance measurement was below a pre-set threshold. Only a partial mapping of the phonemes was performed since some classes were not merged.

The question therefore still remains whether a distance measure can be efficiently used to perform a complete mapping of the phoneme set of a source language database to a target language database. An approach for automatic phoneme mapping is attempted in this thesis. Single state (single mixture) Gaussian distributions are estimated for each phoneme in both databases on CMS normalised data. For each target phoneme the list of closest source phonemes are then found using the Bhattacharyya distance (Equation 5.1).

A problem with an automatic approach to phoneme mapping is that it depends on the distance measure used, but more importantly, that the usefulness of the results depends on a close match between the acoustic properties of the speech in the relevant databases.

For example, say a significant (but mostly linear) bias in feature space exists between the databases. The knowledge-based phonetic approach to computing the mapping is independent of the recording properties of the database and thus should deliver a reasonable mapping, enabling the use of linear transformation to remove the bias. With the automatic approach, though, the mapping may be so poor that even iterative application of transformation and re-mapping may not converge to the optimal mapping. Iterative application of mapping and transformation is, however, not attempted in this thesis.

5.1.4 Database issues

The characteristics of databases used for speech recognition experiments influence to a large extent the expected results. Use of a multilingual database, or bilingual database, in our case, ensures that cross-language experiments using only the database can focus mainly on the acoustic differences between the languages. The characteristics of the database still determines to a large degree the type of experiments that can be performed and also the recognition performance expected when applying various techniques. Both databases used in this thesis contain read speech from many speakers. Read speech is easier to recognise than for example spontaneous speech. Both databases contain phonetically diverse speech, not limited to any particular topic or speaking style, thereby increasing the number of contexts each phoneme may occur in.

When attempting the transfer of acoustic information between databases, it may be important to compensate for differences between the recording conditions of the databases. For example, frequency range and even the frequency transfer functions imposed on recorded speech may differ. Linear frequency effects may be compensated for by cepstral mean subtraction (CMS), although it may be inaccurate to simply implement CMS over the complete databases if the phonetic contexts of the databases differ significantly. The approach that we follow to solve this problem is to train models on the source database and to then compute the maximum likelihood cepstral offset between the source models and the target data within the MLLR framework. The offset is applied to the source data to perform

CMS. Section 5.2.5 gives detail about a generalisation of this approach that performs a transformation of source data, as opposed to simply performing CMS. Experimental results are detailed in Section 7.7 of Chapter 7. Thus, while this thesis does not attempt to characterise the effect of using different databases in general, some experimentation is done to ascertain the relative influence of using the same source language (in this case English) from the same database and from a different database. In the next section we discuss various ways in which multilingual data sources can be used to create target language systems.

5.2 Strategies for using multilingual data sources

The typical position is that a large amount of data is available for one or more source languages and only a limited amount of data is available for the target language. The goal is to construct a recogniser that will achieve optimal performance on unseen data from the target language. We make the assumption that sensible use of all available data will lead to better recognition performance than using only the target language data. Baseline performance is thus set by training on target language data only and methods to improve on this performance are sought. The next sections discuss various ways of utilising the available data.

5.2.1 Data pooling

The simplest method of constructing a recogniser using all available data is to simply pool the data and train on the pooled data set. This technique is commonly used to construct explicitly multilingual systems. Previous studies [48, 19, 20, 30] have shown that for reasonably large amounts of data from each of the languages, a slight performance degradation is actually achieved by the multilingual system in comparison to the language specific recognisers because the accuracy of the models is decreased. If only a small amount of data for the specific language were available, then some improvement in performance is

possible with the multilingual system, simply because robust models cannot be trained with too little data. It is difficult to predict what amount of language specific data is necessary before multilingual pooling will degrade performance. This depends of course also on how close the match between the languages is - the closer the match, the better the chance that simple pooling will lead to desirable results.

5.2.2 Model combination

A simple alternative to the multilingual pooling method is to first train models separately on both source and target language data and to then select the specific models that perform the best on a separate cross-validation set. The reasoning behind this is that when a limited amount of training data is available, there may be enough data to train models for the phonemes that occur most, but not enough to train models for phonemes with low *a priori* occurrence. For these models the source language models can be used. For certain phonemes that occur only in the target language, the target language models are used irrespective of the amount of training data. This method is very simple to implement because pre-trained source language models can be used and training on the limited amount of target language data is computationally inexpensive. The method, however, does not make optimal use of all the available data.

5.2.3 Model adaptation

Adaptation of source language models using limited amounts of target language data has been previously researched, mostly for bootstrapping, but also for directly constructing target language recognisers. The assumption is that too little target language data is available for direct training, but that this data may be enough to adapt source language parameters to sufficiently improve target language performance. The adaptation task is complex compared to a typical speaker adaptation task and it is therefore expected that at least a reasonable amount of data will be necessary for adaptation to achieve good

performance. An argument in favour of the model adaptation approach is that complex source language models can be estimated by using the large amount of source language data. These models can then be adapted using only limited amounts of target language data. This can be achieved for example by transformation-based adaptation in which only the transformation parameters need to be estimated, which may comprise far fewer parameters than re-estimation of all model parameters. On the other hand, if a reasonable amount of target language data is available, a Bayesian or discriminative adaptation technique may utilise the available data more efficiently than a transformation-based technique.

5.2.4 Combined pooling and adaptation

Multilingual pooling and adaptation are considered separate approaches, but it may lead to more efficient use of the available data if training on pooled data were first done, followed by adaptation of the multilingual models using the target language data only. In this way robust models are trained to begin with and are "fine-tuned" using target language data. Adaptation on target language data may improve recognition compared to the multilingual models. This is because the accuracy of the models may be improved with the target language data without sacrificing the robustness of the multilingual models. The method may also outperform source language model adaptation with target language data because source language model adaptation only uses the target language data in the adaptation process, say in the estimation of a transformation. This process may not efficiently extract the available information, thereby leading to suboptimal performance. This may be especially apparent when a reasonable amount of data is available in the target language. Also, adaptation of source language models may "untrain" the acoustic characteristics of the source language in order to specialise the models for the target language - thereby degrading robustness of the models. On the other hand, combined pooling and adaptation is also prone to "untraining" of the source language acoustics, except that little adaptation is likely to be necessary, thereby decreasing that risk. Combined pooling and adaptation is likely to be most useful in conjunction with a technique that adapts the model parameters

only where necessary to effect a target language specific "fine-tuning".

5.2.5 Data augmentation

Augmentation usually comprises the transformation of data from one or more speakers to the space of a new speaker to augment the data from that speaker. It is therefore a form of data pooling, but only of transformed data. When using multiple databases for cross-language adaptation this may be of specific interest because differences, other than language, may also be removed as part of the process. When large differences exist between databases, the data augmentation approach may deliver an improvement in performance over the simple multilingual pooling approach.

Computation of the transformation

With the augmentation approach, a transformation is applied to the source data to alter it to better reflect the characteristics of the target data. The transformation can be computed in a number of ways, namely

- from source data to target data,
- from source models to target data,
- from source models to target models and
- the inverse of the transformation from target models to source data.

A data to data estimation approach for the transformation has the disadvantage that it is inevitably very simplistic since the elements of the transformation are not accurately identifiable. A model to data approach is more powerful since the model can be used to compute occupancy statistics, thereby artificially making the source and target elements of

the transformation identifiable, aiding in the estimation of multiple transformations. Model to data transformations can also be optimised easily for maximum likelihood or least square error criteria. A model to model transformation is again difficult to estimate since source and target model parameters are not identifiable, especially for mixture distribution models.

If a model to data transformation is thus preferred, the choice has to be made between using source models or target models. Since a larger amount of data is typically available for source models, its parameters can therefore be estimated more accurately and thus use of source models is preferable. Target data may not be fully representative and may make estimation of parameters such as variance inaccurate. A final reason why source model to target data is preferred has to do with the availability of a unidirectional phoneme mapping that is discussed in more detail in Section 5.3.2.

Application of the data transformation

The transformation that was computed from source models to target data is used to transform source data to more closely match the target data. Since the transform is applied to labelled speech tokens, the data can be grouped in a meaningful way and multiple regression classes can be identified for transformation. The transformations do not directly transform source data variance, except by its relationship to the scaling of the mean components. This may not present a problem since further adaptation is likely to be performed. The augmentative transformation of the data is mainly to remove the possibly large bias between source and target features and a single transformation may even be used for this purpose.

The transformed speech data is pooled with the target language speech data and used to train target language specific speech models. The ratio of transformed versus target language data may influence the results as too much transformed data may dominate the trained model parameters, degrading performance. A way to improve this situation is to again fine-tune models using target language data only and is discussed next.

5.2.6 Combined augmentation and adaptation

Models trained on the augmented data set consisting of the original target data along with transformed source data are still candidates for target language specific adaptation. This is due to the fact that the data transformation may be relatively simple, consisting of a single regression class and therefore perform mainly channel equalisation and frequency shifting between the databases and languages. Even a transformation with multiple regression classes will not compensate for the differences in variance between the phoneme data of the source and target languages. Another reason for performing further adaptation is that the amount of transformed data may be so much more than the original target data that the weight of the target data is not properly reflected by the pooling process.

Adaptation may therefore further improve performance by more efficiently utilising target language data in "fine-tuning" the models. At this stage, any adaptation method may be used of course, but it is expected that an approach that can adapt individual parameters efficiently may prove to be better than say transformation of a large number of tied parameters.

In this section we discussed various strategies for using the source and target data. A choice now exists between a number of algorithms to be used for the implementation of the strategies. Various implementation aspects of algorithms also bear discussion in order to ensure acceptable performance. These aspects are discussed in the next section.

5.3 Cross-language model adaptation issues

In this section we discuss how to apply various methods from the fields of speaker adaptation and discriminative training to cross-language acoustic adaptation. Cross-language adaptation of acoustic models is a more difficult and complex task than speaker adaptation for a number of reasons, including

- cross-language adaptation performs a SI to SI mapping, rather than an SI to SD mapping,
- acoustic variations across languages are expected to be far larger and more complex than same-language speaker variations, and
- for speaker adaptation the source models generally cover the acoustics of the target speaker well, but are just not very accurate - while for cross-language adaptation the source models may model the target acoustics poorly to start with.

According to the above criteria, dialect adaptation may be closely related to cross-language adaptation, except that the acoustic variations are not expected to be as large. A recent study on dialect adaptation found that more complex adaptation procedures delivered better performance than simpler procedures that perform well for speaker adaptation [114]. It is likely that the complex task of cross-language adaptation will benefit from even more complex adaptation procedures. In the previous two chapters we focussed on methods to utilise the available data as efficiently as possible and we examined adaptation of all HMM parameters including duration modelling parameters. In the following sections the specific application of Bayesian estimation, transformation-based adaptation and discriminative techniques for cross-language adaptation is discussed.

5.3.1 Bayesian adaptation

Bayesian methods exhibit the desirable property of asymptotic performance. This is especially applicable for cross-language adaptation since a reasonably large amount of data may be available in the target language - more than is typically available for speaker adaptation. The relatively slow adaptation performance of Bayesian techniques may also therefore not present a great problem, although it may necessitate a larger target language database than may be needed with alternative approaches. A problem that is related to the slow adaptation of Bayesian techniques, is the fact that with Bayesian techniques, only observed mixtures are adapted. This implies that if source and target language distributions overlap

partially, only the distributions in the overlap region are adapted and the other source model mixtures remain unadapted. The effect is reduced by performing a large number of adaptation iterations, but it may still not completely solve the problem.

A partial solution to the problem is to first perform linear transformation-based adaptation to increase the overlap of the distributions, thereby reducing the number of unadapted mixtures. This may, however, have the unwanted side-effect of changing the priors (seeded by the transformed source models) so much that they do not represent useful prior parameter distributions anymore. In this sense, even though transformation may improve overlap between prior model and target data distributions, it may cause the Bayesian adaptation process to be meaningless. An alternative to first performing model transformation is to use augmentative transformation of source data to improve correlation with target data. This should increase the overlap between augmented data and target data distributions, but may, similar to source model transformation, also degrade the usefulness of the models used to seed prior distributions in successive application of Bayesian adaptation techniques.

An alternative approach for estimating the prior distributions is to use data from as many languages as possible. In this way the inter-language variability is represented in the source models and can probably best express the expected uncertainty with respect to the parameters of a new target language. Such an approach, however, requires the availability of data from a number of languages and is not attempted in this thesis.

5.3.2 Transformation-based adaptation

Transformation-based adaptation is very efficient for correlated source and target distributions and needs relatively little data for robust estimation. For speaker adaptation, a motivation for using transformation-based adaptation is its efficiency in coping with spectral differences between speakers, such as vocal tract length differences. It is not known to what extent this applies to the cross-language adaptation scenario, but the differences may be larger and more complex for the cross-language case. Transformation-based adaptation

at least has the advantage over Bayesian techniques that a large level of mismatch between source and target distributions is not a problem by itself, the problem rather lies in whether a useful transformation exists and whether it can be estimated accurately.

In order to perform the complex adaptation expected to be necessary, relatively large amounts of data will typically be available in the target language. The question arises whether transformation-based techniques can efficiently use relatively large amounts of data since they do not guarantee asymptotic performance with respect to a target dependent system. An application in which transformation-based systems are expected to deliver an advantage over other approaches is when cross-database adaptation is performed as part of cross-language adaptation.

Effect of the mapping and transformation class grouping on the transformation

The fact that a phoneme mapping is used in the cross-language transformation can influence the transformation to a large degree. As we have discussed in Section 5.1, the mapping attempts to find the best source phoneme match for every target phoneme, but often there is no real counterpart and an approximate mapping may result. Source language phonemes may also be mapped to multiple target phonemes, of which some may present close matches, but others not. The transformation is computed from the statistics of a whole group of phonemes and the individual statistics from each phoneme mapping therefore influences the shared transformation. However, the shared transformation, as such, transforms each source model to only a single target model, and can therefore not discriminate at all between target classes seeded from the same source model. This is a serious disadvantage of the transformation-based approach, that, for example, does not present itself with Bayesian techniques where the parameters of each model are adapted independently. This problem may have to be addressed by post-transformation adaptation using Bayesian or discriminative adaptation methods.

A related problem caused by shared transformations is that inaccuracies in the mapping -

which are unavoidable - translate to bias in the adapted parameters. What this means is that "outlier" source models (mainly due to mapping inaccuracies) influence the shared transformations in an unpredictable and undesirable way. The method used to group phonemes into classes for separate transformations is therefore of importance since it determines which transformations should reasonably affect each other. The two grouping strategies that were discussed in Section 3.3.1 are phonetically motivated grouping and model clustering criteria. Grouping by phonetic category assumes that source-target correlation of distributions can be specified by phonetic category while clustering criteria assumes that it depends on position in feature space. Phonetically derived regression classes have been found to deliver better performance [80] than clustering procedures, perhaps because (speech production) information at a higher level than acoustics is used. The piece-wise linear feature space transformation implied by model clustering of regression classes is perhaps a too simplistic assumption for cross-language adaptation.

Adaptation of variance parameters

When the potentially large differences between the acoustic properties of languages are considered, the need for adaptation of variance parameters is obvious. The relationships between source model and target data variance may also be quite complex, necessitating the use of full transformation matrices. In this respect, the log variance transformation of Section 3.3.2 is applicable since constraints on variance parameters are maintained automatically and parameter accuracy is treated sensibly.

Implementation of variance adaptation entails first performing mean adaptation with MLLR, followed by a limited number of variance adaptation iterations. For speaker adaptation purposes often only a single iteration of MLLR is performed since the initial alignment is usually satisfactory. For cross-language adaptation, however, a reasonably large number of iterations may be necessary to achieve satisfactory alignment between the current model estimate and the target data. It is for this reason that estimation of the variance transform is preceded by mean transformation, otherwise very inaccurate initial variance estimates

may cause poor convergence.

Full, diagonal or block-diagonal transformation

For speaker adaptation it has been found that use of a full transformation matrix delivers better performance than use of either diagonal or block-diagonal matrices [27, 65]. For our more complex application of cross-language adaptation we therefore also expect full transformation matrices to deliver better performance. When computing the transformation(s) to be used for data augmentation, the less complex approach of block diagonal transformation may be useful.

We have mentioned in this section that transformation-based adaptation may for various reasons not produce ideal target models. The transformation, though, may be very useful as a first adaptation stage to deal with large overall differences between source and target language distributions. These transformed models can then be used to compute prior distributions for Bayesian adaptation. Bayesian adaptation may function more efficiently on transformed models than on source language models (since the fraction observable mixtures should improve) and may deliver good performance in the complex "fine-tuning" of distributions. In place of Bayesian adaptation, discriminative training may also be used to adapt transformed models or pooled data models. This is the topic of the next section.

5.3.3 Discriminative adaptation using MCE

The main advantage of a discriminative training technique such as MCE over distribution estimation strategies is usually explained in terms of the improved goal of the technique - namely to directly improve expected classification performance. Another advantage that is only really apparent in the implementation of discriminative training is that it may use the available information more efficiently in certain respects. Both Bayesian and discriminative adaptation approaches suffer from the problem of updating only observed mixtures since

every parameter is updated independently. However, with discriminative adaptation, both correct and false class tokens are used in computing the update for the parameters of a mixture, thereby greatly reducing the fraction of unobserved mixtures. This is especially applicable when cross-language adaptation is attempted, since the overlap between source and target distributions for some phoneme pairs may be poor.

Initial models for MCE training

The selection of the initial models to use is very important when MCE is applied because the approach is susceptible to local optima. Use of initial models that already achieve good performance, or that are expected to be robust under different testing circumstances is desirable. In Section 5.2, various strategies were discussed for using multilingual data and models. Some of these strategies produce models that are not necessarily optimal and can benefit from further MCE adaptation. Models that can serve as possible *initial models* for subsequent MCE adaptation can be produced by

- ML training on pooled multilingual data,
- source language models adapted using Bayesian and transformation-based techniques on target language data.
- multilingual models adapted using Bayesian techniques on target language data.

Explicitly multilingual models may be less accurate than target language specific models since model accuracy is decreased when data from multiple languages are used for training. However, these models may be more robust because more data is available for estimation and also because a larger set of contexts are represented. These models may therefore be suitable as initial models for MCE adaptation and performing MCE adaptation on these models may improve the accuracy of the models, while retaining some of the robustness achieved by the initial training that took place on a large, diverse training set.

Both source language models and multilingual models that have been adapted on target language data may produce models suitable for further MCE adaptation. Bayesian adaptation may deliver robust model estimates that already deliver good performance. Transformation-based adaptation may also deliver models that are good starting points for further adaptation if large, overall differences between source and target language distribution can be efficiently removed without severely impacting on the robust characteristics of these models. However, it should be taken into account that initial models that have already been adapted for improved target language performance may be specialised to the extent that they are not as robust as (unadapted) multilingual models.

Finally, the use of transformed source data to augment target data for model training may also produce good initial models for further MCE adaptation. The models should be more accurate than explicitly multilingual models since the transformation should at least partially compensate for differences between the languages.

MCE parameter optimisation

The use of the MCE method for the optimisation of the parameters of HMMs is relatively complicated since gradient-based optimisation has to be used. A further complicating factor is the existence of a number of parameters, namely η in the misclassification measure (Equation 4.6), γ and θ in the loss function (Equation 4.8) and ϵ in the parameter update (Equations 4.14-4.17). Since these parameters influence the results obtained with MCE, their importance is analysed at least in a qualitative manner.

The value of η determines the degree to which false classes contribute to the misclassification measure according to their likelihoods. We have elected to use $\eta = 4$ since this seems a reasonable trade-off between choosing the maximum incorrect class and averaging over the incorrect classes and was found empirically to deliver reasonable results. The value of γ scales the slope of the sigmoid and is important because it influences the size of the feature space region in which observations materially affect the update. Examination of

Equation 4.19 shows that for loss close to either 0 or 1, the derivative of the loss with respect to the misclassification measure becomes small. Smaller values of γ increase the region over which the derivate of the loss remains large, thereby taking into account more observations. High values of γ lead to the consideration of observations only in the immediate region of the decision boundary. In order to effectively use limited amounts of data and also to be able to significantly shift the current decision boundaries, for example when a mismatched seed model was used, it is expected that a reasonably small value of γ should be used. In implementation, we normalised the class conditional log-likelihood functions (the $g_i(\mathbf{X};|\Lambda)$ of Equation 4.12) by dividing it by the number of frames in X. This amounts to expressing each class log-likelihood on a per-frame basis, which greatly reduces the range of likelihood values that are observed. A value of $\gamma = 1$ was found empirically to deliver good performance and was used in experiments.

The value of the update parameter ϵ should also be selected. An update value of $\epsilon = 0.1$ was found empirically to deliver good performance and was used in experiments. On-line training can also be used, but we have selected to use gradient descent with batch-mode updates for simplicity. An approach whereby ϵ is a linearly decreasing function of the iteration count is commonly used with MCE adaptation and we selected to also decrease the update value as a function of the iteration count through $\epsilon_n = \epsilon_0(N-n)/N$ for iterations $n = 0, .., N-1$, with N typically set to 10. If a cross-validation set is available, it may be used as a stopping criterion for adaptation.

MCE cost function application

It can be reasoned that if a suitably large amount of data is available, that string-level MCE will optimally achieve the goal of minimum word and string error rate recognition. For a small vocabulary task such as CDR, the amount of data needed will probably be relatively small. For vocabulary independent adaptation, however, the amount of data needed to properly represent a reasonable percentage of phonetic contexts may be quite large. The use of phoneme-level MCE that implements word error-based phoneme misclassification

cost therefore presents an alternative. Estimation of the misclassification cost can be performed using pronunciation dictionaries as was detailed in Section 4.5.3 and can therefore be performed irrespective of the availability of speech databases. In fact, the cost-based MCE approaches can be applied directly on source language data, without using any target language data and may improve to some extent the class discrimination properties with respect to target language needs. However, we believe that the availability of at least some target language data is essential for the development of accurate speech recognition systems.

A side-effect of the cost-based methods, especially the reward-based method, is that the amount of adaption is decreased rather than increased. By reducing the loss associated with certain categories and even associating a reward with some categories, the overall loss and therefore indirectly the overall gradient in each iteration is reduced. This is desirable since over-specialisation can easily happen with MCE if too little target language data is available.

5.4 Discussion

In this chapter we discussed the issues involved with using data from multiple languages and databases in improving the recognition performance for a single target language. Strategies for cross-language use of data and models were proposed, as well as the implementation of these strategies via adaptation techniques. The suitability of implementing different strategies via specific adaptation techniques were discussed. Overall, the combination of the proposed strategies and their implementation in terms of adaptation techniques presents a framework for cross-language use of acoustic information.

Approaches from this framework are applied in the following two chapters on a multilingual database and on two different databases to empirically evaluate their performance for cross-language acoustic adaptation.

Chapter 6

Cross-language recognition on SUN Speech

This chapter details the cross-language experiments performed using the English and Afrikaans speech from the bilingual SUN Speech database. The experiments compare the recognition performance of the set of cross-language adaptation strategies and algorithms discussed in the previous chapter. The use of English source data in developing models for recognising Afrikaans speech is investigated in particular. The results empirically support the proposed extensions to speaker adaptation and discriminative training algorithms and also support the newly proposed strategies for using multilingual data.

The SUN Speech database is discussed first, setting the environment for the experimental work in this chapter. The experimental protocol is discussed next, covering the selection of the parameters of the system, including various adaptation algorithm parameters. Parameter selection is a difficult task in speech recognition because there are many parameters that can influence the results. The influence of a number of the more important parameters on the recognition results are therefore shown in the experimental sections, rather than selecting just a single value as part of the experimental protocol. The experimental protocol also covers the process used to measure the results of experiments. The following

sections discuss specific experiments that evaluate the various strategies for cross-language use of acoustic information. The chapter concludes with a comparison of the results from the different experiments.

6.1 The SUN Speech database

The SUN Speech database [12] contains phonetically labelled speech in both Afrikaans and English. Details of the database are given in Appendix A and only an brief overview is given here. The database contains read speech from 138 speakers totalling approximately 1500 utterances in English and 500 utterances in Afrikaans. The context of the database is limited since the English speech consists of only 40 different sentences and the Afrikaans speech of only 20 different sentences. The sentences were chosen to deliver a reasonable spread of the phonemes found in both languages. A total of 59 phonemes are used in the labelling of the database. They represent vowels, diphthongs, nasals, fricatives, affricates, glides, liquids, stops and an "other" category containing "silence" and "unknown" labels.

For the purpose of our experiments, the Afrikaans set is divided into a large training set, a smaller subset of the training set and a speaker and context independent test set, i.e. speech contained in the test set is from speakers not represented in the training set and the utterances for the test set differs from the utterances used in the training set. Details of the subdivision and composition of the database are given in Appendix A.2.

When all the available English speech data is used for training models, it amounts to 2 hours and 10 minutes of speech, which is approximately 5 times the amount of data contained in the Afrikaans training set (26 minutes of speech) and approximately 25 times the amount of data contained in the Afrikaans training subset (5 minutes of speech). For cross-language experiments it therefore makes sense to consider the English data as representing a source language with a relatively large amount of data and Afrikaans as the target language with a relatively small amount of adaptation data.

6.2 Experimental protocol

The goal of the experimental section is to evaluate the cross-language recognition performance of the various strategies and algorithms as fairly and accurately as possible, given the data that is available. For experimental purposes English is considered the source language and Afrikaans the target language, since a larger amount of labelled English speech is available in the SUN Speech database and because of the availability of other large English speech databases. Use of both the full Afrikaans training set and the training subset are evaluated to measure the effect of the amount of target language specific data on the recognition results. Initial experiments evaluate isolated phoneme recognition performance to focus on the performance of specific phonemes and classes of phonemes in the multilingual context. Later, more comprehensive experiments test continuous word recognition performance.

The results of the experiments are influenced by the parameters of the training, adaptation and recognition procedures. The selection of various parameters of the feature extraction, HMM modelling and training, duration modelling and adaptation processes is done so that system performance is nearly optimal, yet is not tuned to optimise results in favour of any particular approach.

6.2.1 General system setup

The system that is used for training and testing of the hidden Markov models was developed by the author and a colleague at the University of Pretoria. Details of the system are given in Chapter 2 and only a brief summary of the salient system parameters are given here.

Feature extraction computes 39 mel-scaled cepstral, delta and delta-delta features from 16 ms frames with a 10 ms frame advance. Continuous density hidden Markov models (HMMs) with Gaussian mixture distributions are used for modelling purposes. Strict left-to-right constraints are imposed on HMM transitions and three state HMMs are used to model each

phoneme. Training proceeds using 3 stages, namely initialisation, segmental training and Baum-Welch training. Mixture splitting with stopping criteria is used to enable training of complex mixture distributions. The number of mixtures allowed is varied and up to 10 mixtures per state are allowed. A variance floor of 10^{-4} is imposed. State duration is modelled with a Gamma distribution and the duration parameters are estimated after model training is done through use of segmental training.

As far as recognition is concerned, two main categories of experiments, namely phoneme recognition experiments and word recognition experiments were performed. The experimental protocol of the two recognition approaches is discussed next.

6.2.2 Phoneme recognition experiments

Phoneme recognition is performed using a subset of 47 phonemes, including silence, from the total set of 59 phonemes. The 47 phonemes represent the labels most commonly used in labelling the Afrikaans speech, and exclude the "unknown" category as well as categories that represent less than 0.1% percent of the Afrikaans speech labels. Context independent phoneme modelling is used throughout because of the increased computational expense of context dependent modelling and also because the context of the SUN Speech database is relatively limited and differs significantly between the Afrikaans training and testing sets.

Experiments perform isolated phoneme recognition to allow a comparison to be made between the confusions that occur between the phoneme classes in the recognition process. These results indicate comparatively how well different phoneme models are seeded by their cross-language counterparts.

It is useful to consider some measure of the statistical significance of phoneme recognition results on the entire test set of 9413 labelled phones. Under the assumption of independence, for expected phoneme recognition rates in the range of 40% to 65%, the 95% confidence interval starts at between 1.4% and 1.6% in absolute phoneme recognition rate. We do not

calculate confidence intervals for results from individual phonemes or phoneme groupings, as the experiments do not attempt to prove that the results differ (we are fairly sure that they should differ), but rather examine the type of differences encountered.

6.2.3 Word recognition experiments

The same 47 phoneme models that are reported on in the previous subsection are used to construct word models by connecting the phone HMMs according to a phonetic dictionary for all words occurring in utterances 11-20 of the speaker independent test set(see Appendix A for more detail). The phonetic dictionary is created by analysing the phoneme labels assigned to the speech of the 8 training subset speakers for utterances 11-20. Note that this speech does not form part of the Afrikaans test set. Multiple pronunciations of the same word are allowed, as long as at least two or more of the speakers used the given pronunciation. Using the pronunciation dictionary, in total 151 models for the 100 distinct words in the test utterances are created. In order to run a continuous speech recognition experiment, a small grammar was devised that allocates each word to one of 5 language categories comprising loosely verbs, nouns, adjectives, pronouns and conjunctives. A total of 18 transitions out of a possible 25 transitions between the 5 categories are allowed, limiting the possible sequences enough to deliver reasonable performance for continuous speech recognition in the absence of statistical language modelling.

Recognition results are obtained by aligning the output string from the recogniser with the true transcription and thereby identifying the insertions, deletions and substitutions that are needed to convert the transcription into the output string. Word *accuracy* is computed by subtracting the number of insertions, deletions, and substitutions from the number of words to be recognised and expressing this number as a fraction of the total number of words to be recognised.

It is useful to consider some measure of the statistical significance of word recognition results on the test set of 150 utterances, comprising 2096 distinct words. Under the assumption

of independence of word recognition, for expected word error rates in the range of 25% to 40%, the 95% confidence interval starts at between 2.2% and 2.8% in absolute word error rate.

6.3 Initial phoneme recognition experiments

Initial experiments are performed to evaluate baseline same language (Afrikaans train, Afrikaans test) and different language (English train, Afrikaans test) recognition performance, as well as to examine some aspects of using the SUN Speech database in speech recognition experiments. Testing is done on the Afrikaans test set, consisting of 9413 labelled phonemes in continuous speech, or approximately 12.5 minutes of speech data. More details regarding the SUN Speech database and its subdivision into training and testing sets are given in Appendix A. Experiments perform isolated phoneme recognition to allow examination of how the recognition performance of individual phonemes and phoneme classes are affected in the cross-lingual scenario. When no training tokens are available for a model, the model is not used in recognition, and all test samples from the phoneme category are misclassified.

6.3.1 Overall phoneme recognition performance

Figure 6.1 shows isolated phoneme classification performance on the Afrikaans test set as a function of the model complexity allowed, for models trained on either the Afrikaans training set, training subset, or on the entire English set. As expected, using Afrikaans training data delivers models that more closely match the Afrikaans testing data and delivers a peak correct classification rate of 62.5% (10 mixtures), which is 13.5% better than the peak correct classification rate of 49.0% (10 mixtures) achieved with models trained on the 5 times larger English set. Models trained on the Afrikaans training subset achieve a peak correct classification rate of 53.1% (4 mixtures), which is 4.1% better than that achieved

with models trained on the 25 times larger English set. The results indicate, at least as far as isolated phoneme recognition is concerned, that use of target language specific data may outperform using even a significantly larger amount of non-target language data.

Allowing a larger number of mixtures to be trained (allowing more mixture splitting in training), generally improves performance, as expected. Performance of models trained on the Afrikaans training set, and especially performance of models trained on the training subset, levels off at fewer mixtures than for models trained on the larger English set.

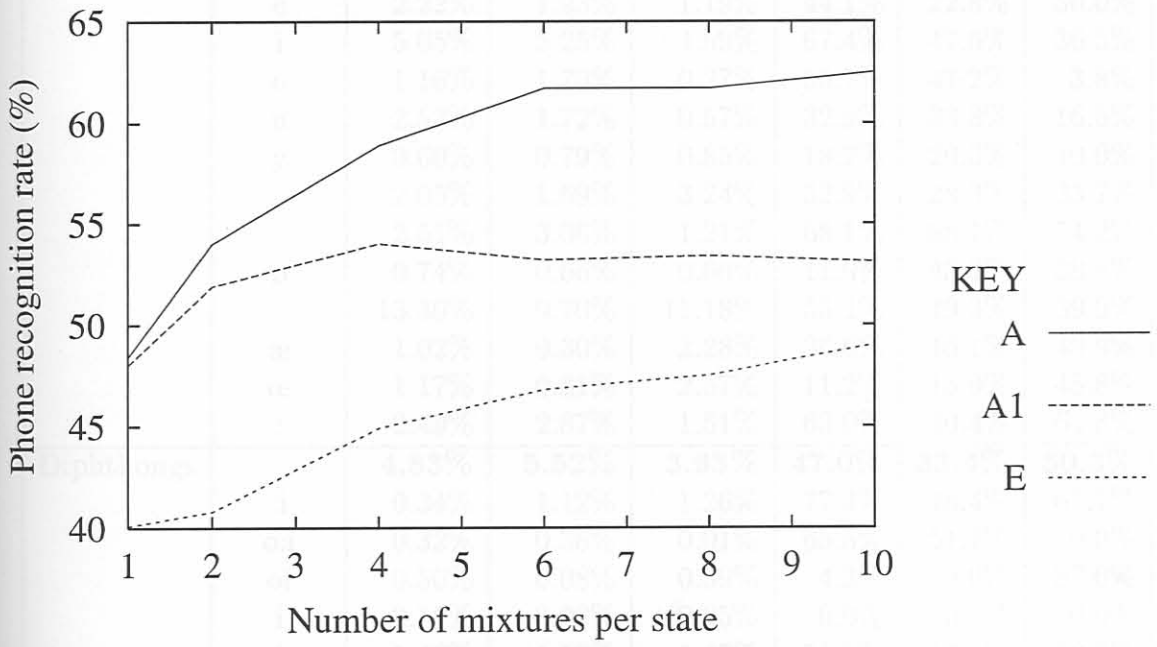

Figure 6.1: Isolated phone classification rate as a function of the number of HMM mixtures when training on the Afrikaans training set (A), the Afrikaans training subset (A1) and the entire English set (E) and testing on the Afrikaans test set

6.3.2 Individual phoneme recognition performance

The classification rates of Figure 6.1 give an overall view of relative phoneme classification rates, but it is of interest to study the classification rate of individual phoneme categories to compare their relative performance. Table 6.1 and 6.2 expands the 10 mixture per state results of Figure 6.1 by listing isolated phoneme recognition performance for each phoneme

class when models trained on the Afrikaans training set, Afrikaans training subset and the English set, are tested on the Afrikaans test set.

Table 6.1: Phoneme classification rates achieved on the Afrikaans test set for models trained on the Afrikaans training set (A), training subset (A1), and the English set (E), also showing relative frequency of phonemes in the Afrikaans train (F_{Atrain}), Afrikaans test (F_{Atest}) and the English set (F_E)

Category	Symbol	F_{Atest}	F_{Atrain}	F_E	A	A1	E
Vowels		**38.37%**	**34.22%**	**32.90%**	**53.0%**	**44.3%**	**41.0%**
	a	3.53%	3.74%	2.84%	59.3%	65.5%	38.5%
	e	2.22%	1.93%	1.19%	44.1%	22.8%	50.0%
	i	5.05%	5.25%	4.59%	67.4%	47.6%	36.5%
	o	1.16%	1.70%	0.27%	55.7%	47.2%	3.8%
	u	2.53%	1.72%	0.57%	32.9%	33.8%	16.5%
	y	0.60%	0.79%	0.85%	18.2%	20.0%	40.0%
		2.05%	1.69%	3.24%	52.9%	28.3%	33.7%
		2.51%	3.06%	1.21%	68.1%	68.1%	74.2%
	ø	0.74%	0.66%	0.60%	11.9%	43.3%	38.8%
		13.30%	9.70%	11.18%	55.1%	49.3%	39.9%
	æ	1.02%	0.30%	2.28%	36.6%	16.1%	40.9%
	œ	1.17%	0.81%	2.57%	11.2%	15.0%	45.8%
	:	2.49%	2.87%	1.51%	63.0%	30.4%	64.8%
Diphthongs		**4.83%**	**5.52%**	**3.93%**	**47.0%**	**33.4%**	**30.3%**
	:i	0.34%	1.12%	1.26%	77.4%	48.4%	67.7%
	o:i	0.32%	0.36%	0.01%	65.5%	51.7%	0.0%
	oi	0.50%	0.08%	0.39%	4.3%	0.0%	87.0%
	i	0.16%	0.00%	0.05%	0.0%	0.0%	0.0%
	i	1.46%	1.50%	1.47%	51.9%	14.3%	38.3%
	ui	0.46%	0.65%	0.00%	64.3%	66.7%	0.0%
	iu:	0.30%	0.49%	0.05%	59.3%	59.3%	3.7%
	œu	0.45%	0.46%	0.60%	26.8%	26.8%	24.4%
	œy	0.84%	0.86%	0.10%	50.6%	55.8%	14.3%
Nasals		**9.36%**	**10.71%**	**11.96%**	**66.9%**	**59.9%**	**65.3%**
	m	2.11%	2.92%	2.77%	58.3%	69.8%	73.4%
	n	5.69%	6.56%	7.82%	75.5%	58.4%	67.1%
		1.56%	1.23%	1.37%	47.2%	52.1%	47.9%

Table 6.2: Phoneme classification rates achieved on the Afrikaans test set for models trained on the Afrikaans training set (A), training subset (A1), and the English set (E), also showing relative frequency of phonemes in the Afrikaans train (F_{Atrain}), Afrikaans test (F_{Atest}) and the English set (F_E)

Category	Symbol	F_{Atest}	F_{Atrain}	F_E	A	A1	E
Fricatives		**14.55%**	**15.71%**	**14.04%**	**81.4%**	**77.5%**	**62.8%**
	f	2.91%	3.39%	2.18%	86.4%	81.5%	94.3%
	h	0.11%	0.71%	0.80%	0.0%	0.0%	0.0%
	s	6.25%	6.10%	5.67%	87.2%	86.7%	73.2%
	v	1.72%	1.75%	1.90%	56.7%	48.4%	64.3%
	x	2.59%	2.65%	0.01%	94.5%	89.8%	0.8%
	z	0.53%	0.67%	1.64%	47.9%	31.2%	72.9%
		0.30%	0.44%	1.48%	66.7%	55.6%	74.1%
		0.14%	0.00%	0.36%	0.0%	0.0%	61.5%
Affricates		**1.00%**	**0.74%**	**1.77%**	**34.0%**	**25.2%**	**53.8%**
	ts^h	0.32%	0.06%	0.46%	10.3%	0.0%	31.0%
	t^h	0.68%	0.68%	1.31%	45.2%	37.1%	64.5%
Liquids		**8.75%**	**8.16%**	**6.45%**	**70.5%**	**60.4%**	**32.6%**
	r	5.20%	4.33%	2.95%	85.0%	81.2%	16.9%
	l	2.97%	3.50%	2.97%	59.0%	35.8%	60.1%
		0.58%	0.33%	0.53%	0.0%	0.0%	32.1%
Glides		**1.92%**	**1.25%**	**2.05%**	**41.1%**	**21.1%**	**56.6%**
	j	1.03%	1.15%	0.47%	67.0%	38.3%	50.0%
	w	0.89%	0.10%	1.58%	11.1%	1.2%	64.2%
Stops		**19.08%**	**14.33%**	**15.62%**	**65.9%**	**52.3%**	**53.4%**
	b	2.93%	1.54%	1.50%	63.7%	58.4%	61.4%
	d	3.51%	3.95%	2.36%	75.9%	51.9%	46.9%
	g	1.02%	0.47%	0.66%	16.1%	16.1%	52.7%
	k	3.66%	2.79%	3.21%	72.2%	59.6%	72.5%
	p	2.47%	0.95%	2.04%	47.1%	23.1%	81.3%
	t	5.49%	4.63%	5.85%	74.2%	64.4%	28.0%
Other		**2.12%**	**3.72%**	**3.42%**	**90.2%**	**85.0%**	**89.6%**
	sil	2.12%	3.72%	3.42%	90.2%	85.0%	89.6%

Same-language recognition performance

We first discuss the recognition performance of models trained on the Afrikaans training set. Classification performance achieved on vowels is only 53.0%, compared to the overall isolated phoneme classification rate of 62.5%. This differs from what is reported in literature

[26], i.e. that for English speech at least, classification performance on vowels is generally much better than on non-vowels. To some degree this is explained by the large number of vowel classes (13 in all) that are used for labelling in the SUN Speech database, as well as the specific choice of vowel classes. Some classes, especially the rounded vowels [y] and [ø] are often confused for their unrounded counterparts [i] and [e]. In English speech this distinction is not important, but in Afrikaans speech the distinction is important e.g. [mi:r] versus [my:r] and [le:n] versus [løn]. The presence of the central vowel [] in the labelling also causes confusion as both front and back vowels are often confused with it. Exact distinction of the central vowel may not be very important for word recognition, yet it has been assigned to more than 13% of the total number of labels in the test set. Classification performance on diphthongs (47.0%) is also not very good as they are often confused with vowels. Better performance (66.9%) is achieved with the class of nasals, with most of the misclassified examples also being classified as one of the other nasal categories. Somewhat surprisingly, excellent performance (81.4%) is achieved on fricatives. This can be due in part to the fact that there are only four frequently found fricatives in Afrikaans ([f], [s], [v] and [x]) which are all relatively distinct. The other categories do not deliver major surprises. We note that phoneme classes not well represented in the training set often perform very poorly in classification, with four classes even achieving 0% correct classification. This would seem to indicate over-fitting, but inspection reveals that these models contain few mixtures (between one and three), which should limit the degree of specialisation. Still, the 21 Afrikaans phoneme classes that each have less than 1% of the total training samples, together comprise 9.9% of the training set, 13% of the test set, and achieve a combined correct recognition rate of only 24%. The performance of models trained on the English data is discussed next.

Cross-language recognition performance

The average correct classification rate for English models is 49.0% compared to 62.5% achieved when training with the smaller Afrikaans training set and 53.1% achieved with the even smaller Afrikaans training subset. For phoneme classes that contain no data in the

English set, no models are trained and all test tokens are classified incorrectly. Compared with the Afrikaans training set model performance, most categories show a decrease in performance, except for affricates, which improve from 34.0% to 53.8% and glides, which improve from 41.1% to 56.6% correct classification rate. Nasals show only a 1.6% drop in classification performance from 66.9% to 65.3%. Overall performance is not poor, however, and the only phoneme categories that achieve less than 50% recognition rate are the vowels, diphthongs and liquids. Relatively poor performance of the vowel and diphthong categories is to be expected due to the differences between the language in these categories. Poor performance of the liquids is due to the English [r] model (16.9% correct) not exhibiting the diverse allophonic variations found in Afrikaans. The [x] model also achieves very poor performance (0.8% correct) since it is not a sound which occurs naturally in English. Somehow, two [x] labels were assigned to English speech, enabling training of a simple single mixture per state model.

An interesting phenomenon can be observed by comparing the results from selected classes of Afrikaans (training set) and English trained models. When one considers the phoneme classes for which the English frequency of occurrence is at least twice the Afrikaans training set frequency of occurrence, the resulting set of phonemes is [æ], [œ], [oi], [z], [], [], [ts^h], [w] and [p]. For these phoneme classes there are at least 10 times more samples in the English training set than in the Afrikaans training set. It is not too surprising then, to notice that the recognition performance for each of these phoneme classes is higher in the experiments with the English trained models than in the experiments with the Afrikaans trained models. This is indicative of the general problem of estimating distributions of such high dimensionality (39 feature dimensions plus the dimension of time), i.e. that a large amount of data is necessary to obtain robust performance.

A comparison between (small) Afrikaans training subset and English model results also delivers interesting insights. Although the overall performance of the English models is 4.1% lower than that obtained with Afrikaans training subset models (49.0% versus 53.1% correct), for 27 out of 44 phonemes (more than 60% of phonemes) the English models deliver better performance. If the arithmetic average is computed of the classification percentages

of each phoneme (i.e. not taking into account the frequency of each phoneme in the test set), the English phoneme models average 45.5%, the Afrikaans trained models average 49.2% and the Afrikaans subset trained models average only 41.2%. The interpretation of isolated phoneme recognition results is problematic since the importance of individual phoneme misclassifications are not taken into account. It is therefore decided to perform continuous word recognition in subsequent experiments to represent the application of phoneme models for a useful purpose.

6.4 Multilingual data pooling

In this section we investigate multilingual data pooling in detail and perform experiments that measure word accuracy in continuous speech. Continuous word recognition experiments are performed as was discussed in Section 6.2.3 and evaluate the performance of monolingual and multilingual acoustic models for a real-world task.

Figure 6.2 shows the results achieved in continuous word recognition experiments on the Afrikaans test set of various monolingual and multilingual models. The best performance of 73.3% word accuracy is achieved by training on pooled English and Afrikaans data, followed by training on the Afrikaans training set (69.0% accuracy) and by training on the pooled English data and Afrikaans training subset (68.1% accuracy). Recognition using the English models peaks at 57.9% while training on only the Afrikaans training subset delivers an accuracy of only 45.0%. The results show a clear improvement in performance obtained by multilingual pooling versus using target language data only. Compared to using the Afrikaans training set, 4.3% absolute improvement in accuracy (73.3% versus 69.0%) is achieved by pooling with English data, and compared to using only the Afrikaans training subset, a large 23.1% absolute improvement in accuracy (68.1% versus 45.0%) is achieved by pooling with English data.

The baseline word accuracy results given in Figure 6.2 serve as reference for the results giv-

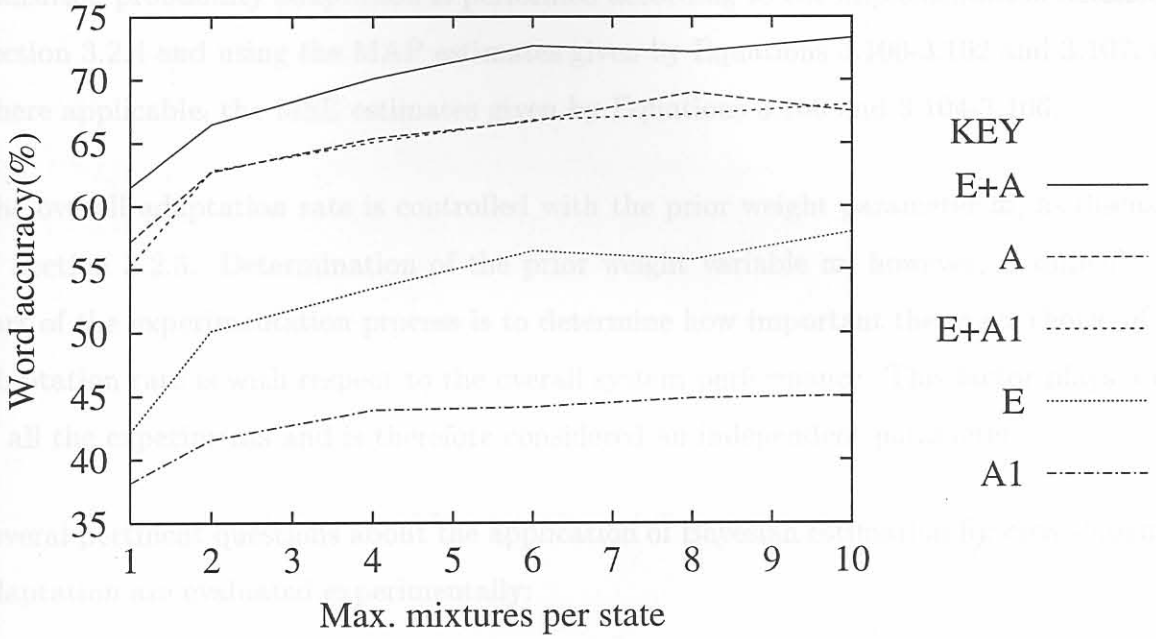

Figure 6.2: Word accuracy on the Afrikaans test set as a function of the maximum allowed number of mixtures per state for three-state HMMs trained on various monolingual and pooled multilingual data sets formed using the English set (E), the Afrikaans training set (A) and the smaller Afrikaans training subset (A1)

en in the following sections. It should be kept in mind that English and Afrikaans acoustics from the same database are used, explaining why even simple multilingual pooling delivers good results. The following sections detail experiments that evaluate the cross-language adaptation performance of techniques discussed in Section 5.3 for speaker independent speech recognition. Experiments using Bayesian, transformation-based and discriminative techniques are discussed. For all the experiments that follow, 3-state HMMs with a maximum of 10 mixtures per state are used as these represent the best performance for almost all models in Figure 6.2.

6.5 Bayesian adaptation

Experiments are performed to evaluate the application of Bayesian adaptation for cross-language adaptation as discussed in Section 5.3.1. Full mean, variance, mixture weight and

transition probability adaptation is performed according to the implementation detailed in Section 3.2.4 and using the MAP estimates given by Equations 3.100-3.102 and 3.107, and where applicable, the MSE estimates given by Equations 3.100 and 3.104-3.106.

The overall adaptation rate is controlled with the prior weight parameter ϖ, as discussed in Section 3.2.5. Determination of the prior weight variable ϖ, however, is difficult and part of the experimentation process is to determine how important the exact choice of the adaptation rate is with respect to the overall system performance. This factor plays a role in all the experiments and is therefore considered an independent parameter.

Several pertinent questions about the application of Bayesian estimation for cross-language adaptation are evaluated experimentally:

- How does the amount of target language data influence the results ?
- How does the performance of cross-language model adaptation compare with adapting multilingual models using target language data ?
- How important is adaptation of variance parameters ?
- How does the performance achieved with MAP and MSE Bayesian adaptation compare?

These questions form the basis for experiments discussed next.

6.5.1 Cross-language model adaptation

Figure 6.3 shows the performance achieved as a function of the adaptation rate for English prior models adapted on the Afrikaans training set and the Afrikaans training subset. Peak performance of 74.9% word accuracy is achieved when adapting on the full Afrikaans training set, which delivers an absolute 7.3% improvement over using only the Afrikaans training set (67.6% for 3 state, 10 mixture models). This performance (74.9%) also delivers

an absolute 1.6% improvement over using the pooled English and Afrikaans training set (73.3%). Even better relative performance is achieved by adaptation on the Afrikaans training subset, achieving peak performance of 70.2% word accuracy, which is 25.2% better than that achieved with the Afrikaans training subset alone (45.0%, not shown) and 2.1% better than that achieved with the pooled English and Afrikaans training subset (68.1%).

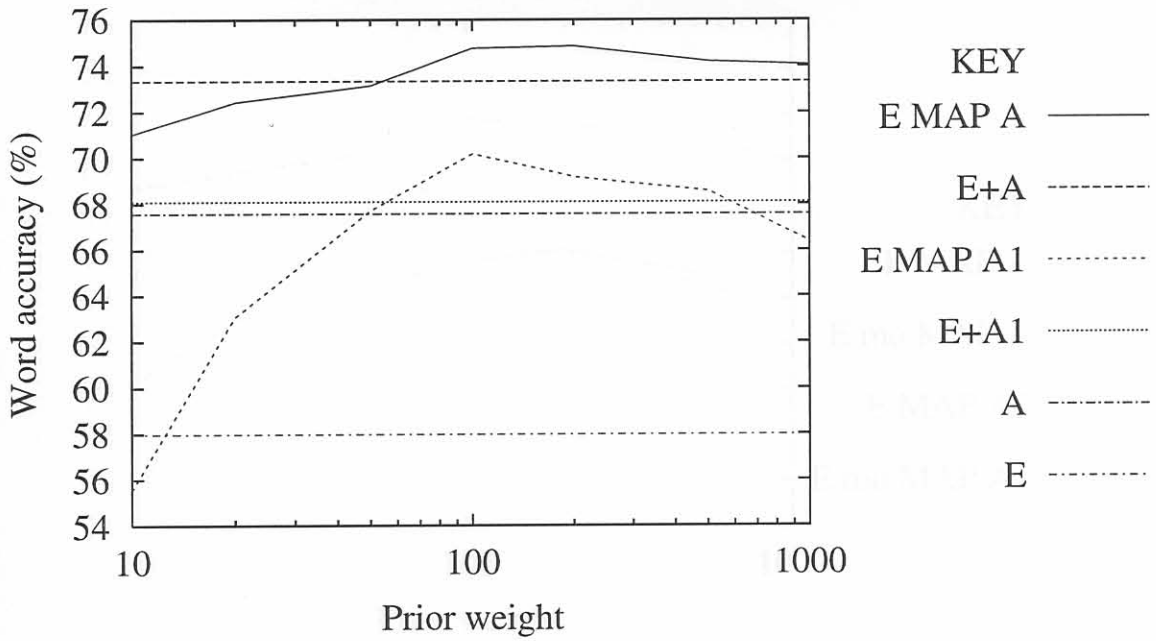

Figure 6.3: Word accuracy on the Afrikaans test set as a function of the adaptation rate for English models (E) adapted using MAP adaptation on the Afrikaans training set (A) and training subset (A1) with reference performance of monolingual and multilingual models also shown

The dependency between adaptation performance and the overall prior weight variable ϖ is apparent, with in particular, performance of adaptation with the small Afrikaans training subset suffering when ϖ is small. This is due to the fact that re-estimation on a small set delivers inaccurate estimates and therefore a larger weight should be associated with the prior to ensure good results.

6.5.2 Cross-language adaptation of variance

Figure 6.4 shows the effect of using mean-only MAP versus full MAP adaptation, which includes adaptation of the variance and mixture weight parameters when English prior models are adapted using the Afrikaans training set and training subset. It is apparent

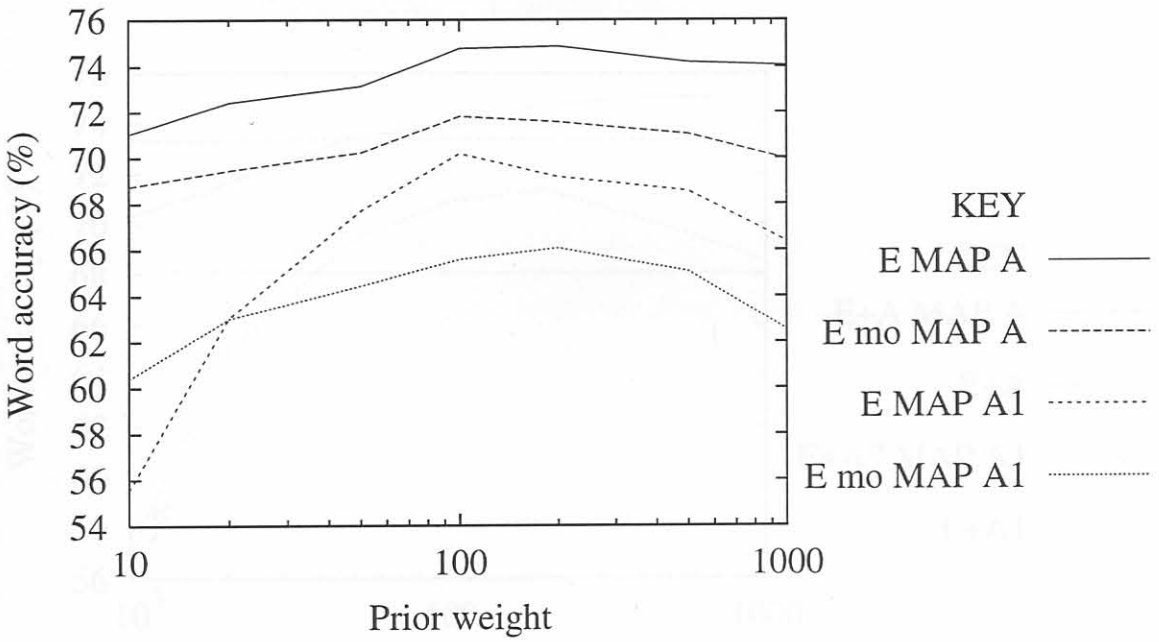

Figure 6.4: Comparison of word accuracy on the Afrikaans test set for mean-only (mo MAP) and full MAP (MAP) adaptation as a function of the adaptation rate for English models (E) adapted on the Afrikaans training set (A) and training subset (A1)

that significantly better performance is achieved when full adaptation is performed, with 3.1% degradation (74.9% versus 71.8%) in peak performance attributable to mean-only adaptation on the Afrikaans training set and 4.1% degradation (70.2% versus 66.1%) in peak performance attributable to mean-only adaptation on the Afrikaans training subset. The results indicate that adaptation of variance parameters is important to achieve good cross-language adaptation performance. However, for very small overall prior weight values ($\varpi < 20$) mean-only adaptation outperforms full adaptation on the Afrikaans training subset since variance re-estimation on little data is avoided.

6.5.3 Data pooling followed by adaptation

Figure 6.5 shows the performance achieved as a function of the adaptation rate for bilingual prior models trained on the pooled English and Afrikaans training sets and on the pooled English and Afrikaans training subsets, when adapted on the Afrikaans training set and the Afrikaans training subset respectively. Peak performance of 75.1% word accuracy is

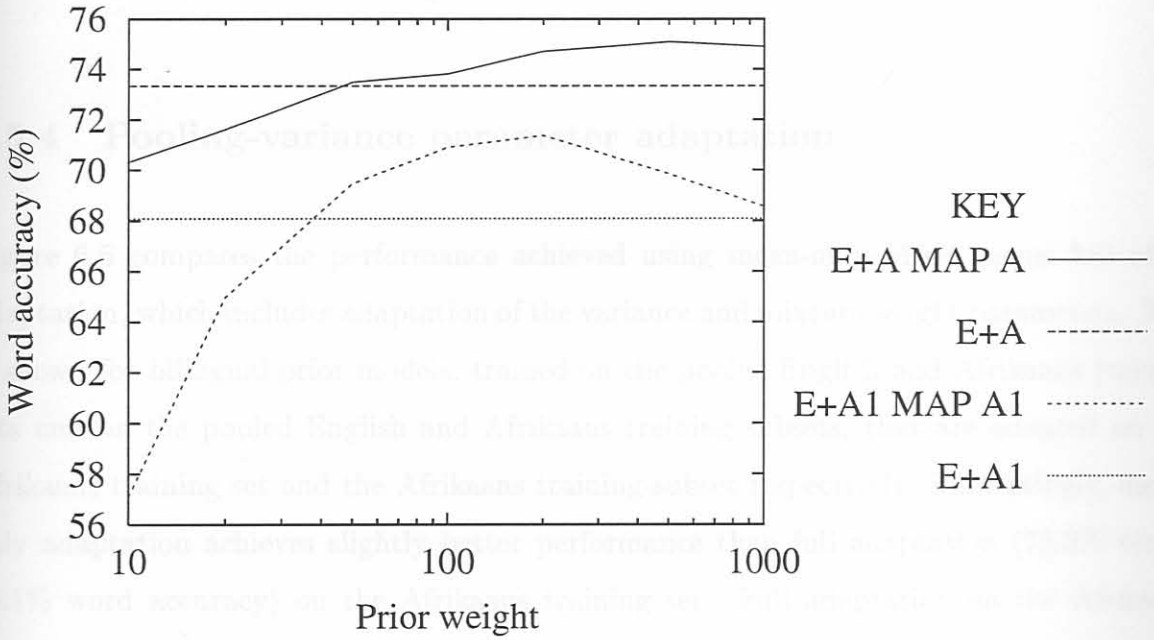

Figure 6.5: Word accuracy on the Afrikaans test set as a function of the adaptation rate for models trained on pooled English and Afrikaans training data (E+A) and pooled English and Afrikaans training subset data (E+A1) and adapted using MAP adaptation with reference performance of multilingual models also shown

achieved for pooling/adaptation on the full Afrikaans training set, which delivers an 1.8% improvement over data pooling (73.3%) and an additional 0.2% improvement over the direct cross-language adaptation of English source models (74.9%) in Section 6.5.1. Even better relative performance is achieved for pooling/adaptation on the Afrikaans training subset, delivering a 3.3% improvement in word accuracy (71.4% versus 68.1%) over data pooling and an additional 1.2% improvement over the direct cross-language adaptation of English source models (70.2%) in Section 6.5.1.

The general trend that the smaller Afrikaans data set benefits more from the sharing of

acoustic information with the English set than the larger Afrikaans set is to be expected, since with a sufficiently large Afrikaans data set we expect the benefit to asymptotically decrease to zero. It is interesting to observe that peak performance is achieved with larger prior weight values ($200 < \varpi < 500$) compared to direct cross-language models adaptation ($\varpi \approx 100$). This is indicative that less adaptation is required for peak performance when multilingual priors (which include the target language) are used compared to using priors from a single different source language.

6.5.4 Pooling-variance parameter adaptation

Figure 6.6 compares the performance achieved using mean-only MAP versus full MAP adaptation, which includes adaptation of the variance and mixture weight parameters. This is shown for bilingual prior models, trained on the pooled English and Afrikaans training sets and on the pooled English and Afrikaans training subsets, that are adapted on the Afrikaans training set and the Afrikaans training subset respectively. Interestingly, mean-only adaptation achieves slightly better performance than full adaptation (75.3% versus 75.1% word accuracy) on the Afrikaans training set. Full adaptation on the Afrikaans training subset, however, outperforms mean-only adaptation by 1.8% (71.4% versus 69.6%). This indicates that, when a reasonably large target language specific data set forms part of the pooled multilingual data set, variance adaptation may not be important. However, when a small amount of target specific data is used in pooling, variance adaptation may be necessary because training on pooled data may not represent the variance characteristics accurately enough.

6.5.5 MAP versus MSE estimation

So far in Section 6.5 we have been using MAP estimates, and in particular the proposed variance estimate of Equation 3.107. The next experiment compares the performance achieved with this method with the performance achieved using the biased MAP variance estimate

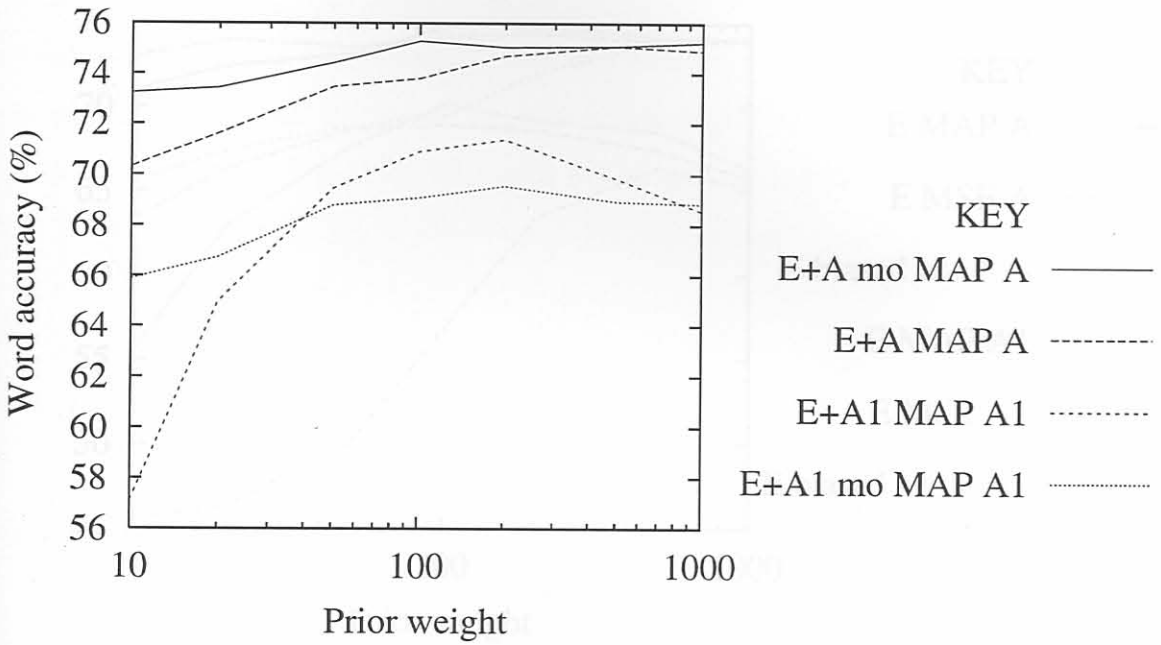

Figure 6.6: Comparison of word accuracy on the Afrikaans test set for mean-only (mo MAP) and full MAP (MAP) adaptation as a function of the adaptation rate for pooled English-Afrikaans models adapted on the Afrikaans training set (A) and training subset (A1)

of Equation 3.103 and the performance achieved with MSE variance estimation (Equations 3.100 and 3.104-3.106). Figure 6.7 shows word accuracy on the Afrikaans test set for the adaptation of English prior models on the Afrikaans training set and training subset. When adapting English priors on the Afrikaans training set, best performance of 74.9% is achieved with the proposed MAP estimate (Equation 3.107), 74.4% word accuracy is achieved with an MSE Bayes estimate, and a peak accuracy of 74.0% is achieved with the biased MAP estimate (Equation 3.103). For adaptation on the Afrikaans training subset, best performance of 70.2% is achieved with the proposed MAP estimate, 68.9% word accuracy is achieved with an MSE Bayes estimate and peak accuracy of 66.4% is achieved with the biased MAP estimate. For smaller prior weighting, the MSE estimator delivers the best performance (for $\varpi < 50$), while the performance achieved with the biased MAP estimate degrades significantly (for $\varpi < 200$).

To understand the better performance of the MSE estimate for small ϖ, we consider that the feature dimension D is basically a lower bound on the weight the MSE estimate attaches

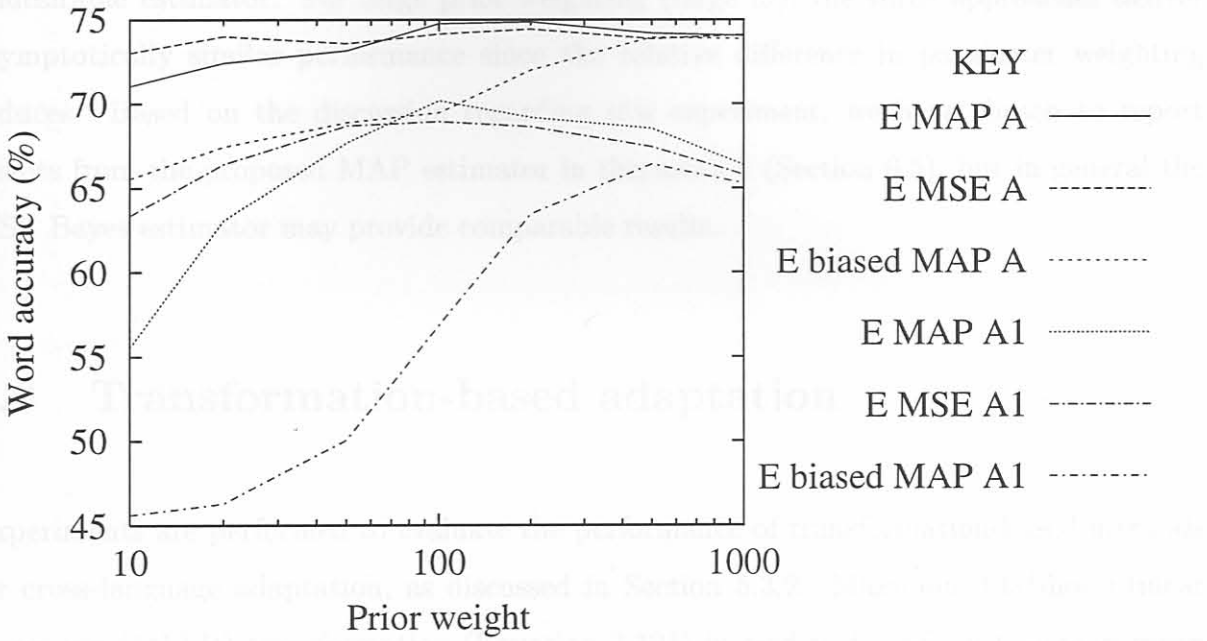

Figure 6.7: Comparison of word accuracy on the Afrikaans test set for MAP, biased MAP (using Equation 3.103 for variance estimation) and MSE Bayesian adaptation as a function of the adaptation rate for English models adapted on the Afrikaans training set (A) and training subset (A1)

to the prior variance $\hat{\Sigma}$ (see Equation 3.106), which in this case leads to improved performance since the degree to which target dependent variance re-estimation occurs is reduced. If the effective offset (of D) in ϖ is ignored, the only difference between the MAP and MSE approaches is that the MSE estimate effectively attaches less importance to the difference between the prior mean and the posterior mean estimate ($\mathbf{m}_{ik} - \hat{\mathbf{m}}_{ik}$). The reason for the proposed MAP estimate achieving better peak performance than the MSE estimate for this experiment must therefore be that it attaches greater weight to the difference between the prior mean and the posterior mean. This may have a positive influence on recognition performance because relatively larger displacements in mean position are likely to be incurred for poorly seeded distributions, in turn increasing posterior variance. Increased variance is applicable for poorly seeded models to the degree that limited target data does not allow for accurate estimation of the posterior mean.

The poor performance achieved with the standard MAP estimator for small ϖ can be attributed to its highly biased estimate of the variance for small ϖ, which makes it an

undesirable estimator. For large prior weighting (large ϖ), the three approaches deliver asymptotically similar performance since the relative difference in parameter weighting reduces. Based on the discussion regarding this experiment, we have chosen to report results from the proposed MAP estimator in this section (Section 6.5), but in general the MSE Bayes estimator may provide comparable results.

6.6 Transformation-based adaptation

Experiments are performed to evaluate the performance of transformation-based methods for cross-language adaptation, as discussed in Section 5.3.2. Maximum likelihood linear regression (MLLR) transformation (Equation 3.121) is used to transform Gaussian mean parameters. In order to comprehensively adapt source model parameters, we also experiment with the adaptation of Gaussian variance parameters. The techniques used for the adaptation of the Gaussian variance parameters include:

- no adaptation,
- direct re-estimation (on only the target data),
- linear transformation with MSE criterion (Equation 3.129), and
- log-domain transformation with MSE criterion (Equation 3.136).

Relative to speaker adaptation, it is expected that cross-language adaptation will necessitate a more complex and comprehensive adaptation of source language models. In order to estimate a complex mapping, models are grouped into regression classes, with a separate transformation being calculated for each class. Grouping into classes is done according to broad phonetic groupings, i.e. for a two-class subdivision vowels/diphthongs are separated from the rest, a five-class subdivision separates vowels, diphthongs, fricatives/affricates, stops and nasals/glides/liquids and for a eight-class subdivision all mentioned categories

are treated as distinct regression classes. Grouping transformations into classes has the advantage that each class of similar phonemes share a transformation, which is different from that used to transform the other classes. The assumption is that the distributions of the acoustic parameters for the target language exhibit correlation within each class. In experiments the effect of both the number of regression classes, as well as the method used for variance compensation are evaluated. We first experiment with cross-language transformation of English models using the Afrikaans sets and then evaluate the effect of transforming bilingual (pooled) models.

6.6.1 Cross-language model adaptation

Figure 6.8 shows word accuracy as a function of the number of regression classes when English models are transformed using the Afrikaans training subset only. Best performance of 65.7% is achieved with a 2-class MLLR mean/MSE log-variance transformation and second best performance is achieved with mean-only MLLR transformation, delivering peak word accuracy of 62.7%. The other techniques that adapt variance in addition to performing MLLR mean transformation perform significantly poorer and do not even improve on baseline (untransformed) English model performance. As expected, variance re-estimation performs poorly on the small Afrikaans training subset. The relatively poor results for the 5 and 8-class transformations indicate that there is not enough data to perform complex transformations contained in many regression classes. As the number of regression classes increases, performance degrades even below that achieved with direct training on the Afrikaans training subset. This happens because the English source models were trained on a large amount of data and therefore typically contain many mixtures per state (up to a maximum of 10). They are therefore easily over-fitted by the transformation on the limited amount of target data. In Section 6.7.2 we experiment with a technique that solves this problem to some degree by combining MAP with MLLR transformation.

Overall, the results in Figure 6.8 show that MLLR transformation of the Gaussian means, with or without log-variance transformation, at least delivers an improvement on baseline

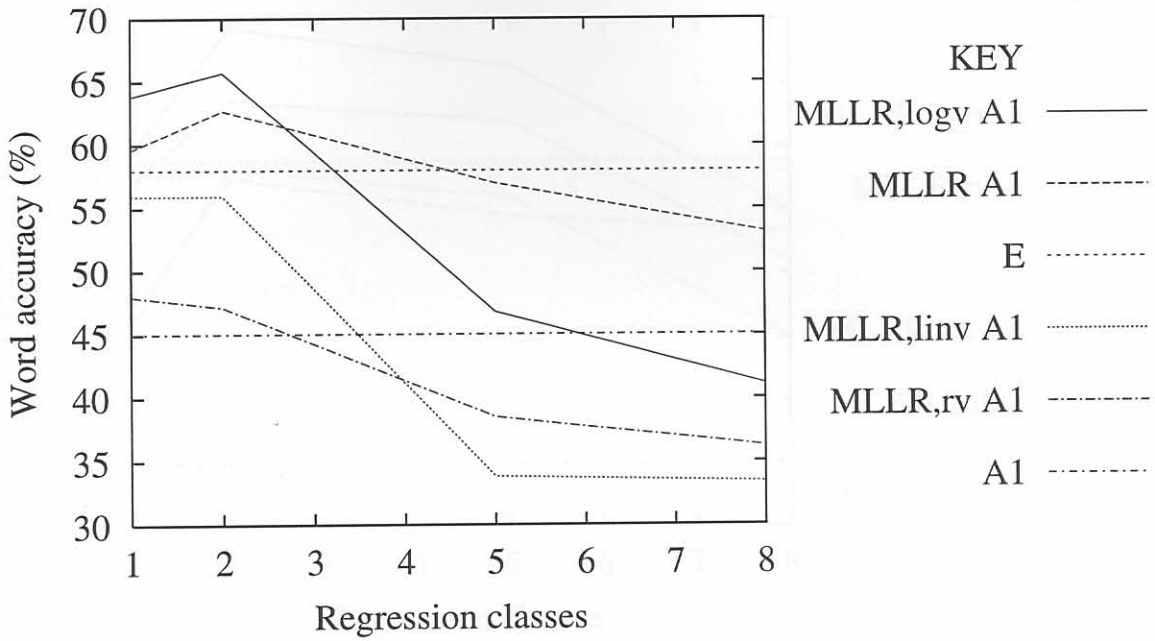

Figure 6.8: Comparison of word accuracy on the Afrikaans test set for MLLR transformation of Gaussian means combined with different variance transformation techniques: mean-only, log variance (logv), linear variance (linv) and variance re-estimation (rv) as a function of the number of regression classes for English models adapted on the Afrikaans training subset (A1)

English model performance (58.0% word accuracy) for few regression classes, but does not attain the performance achieved with bilingual pooling (68.1% word accuracy) in Section 6.4 or with cross-language MAP adaptation (peak word accuracy of 70.2%) in Section 6.5.1.

Figure 6.9 shows word accuracy as a function of the number of regression classes when English models are transformed using MLLR transformation of mean parameters combined with various techniques to adapt variance parameters on the (full) Afrikaans training set. Comparing the overall results with that of Figure 6.8, it is apparent that performance using more regression classes has improved since more target data is available.

Best performance of 71.8% is achieved with an MSE log-variance transformation when two regression classes are used. MLLR mean transformation with variance re-estimation also delivers good results (69.5% word accuracy), probably due to the fact that the Afrikaans training set is large enough for re-estimation to deliver reasonable estimates. Mean-only

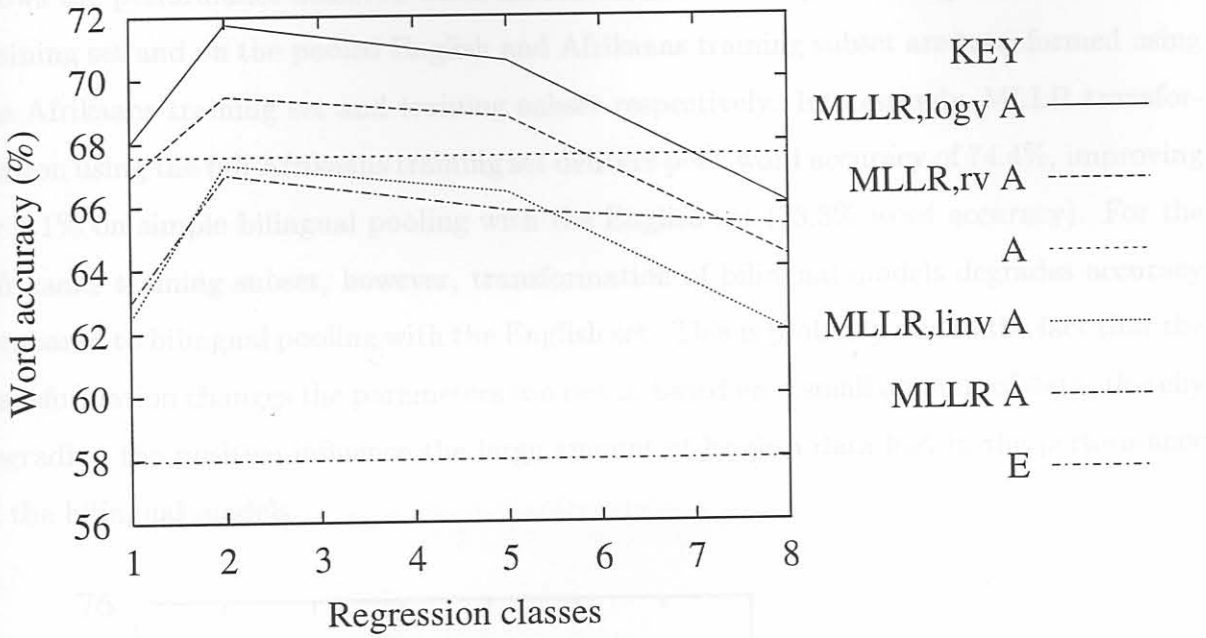

Figure 6.9: Comparison of word accuracy on the Afrikaans test set for MLLR transformation of Gaussian means combined with different variance transformation techniques: mean-only, log variance (logv), linear variance (linv) and variance re-estimation (rv) as a function of the number of regression classes for English models adapted on the Afrikaans training set (A)

(67.0%) and linear variance transformations (67.3%) show performance improvement over the baseline English models, but do not exceed the performance obtained with training directly on the Afrikaans training set. The peak word accuracy of 71.8% achieved with log-variance transformation is better than that achieved using English-only or Afrikaans-only training sets (58.0% and 69.0% respectively), but is still less than the word accuracy achieved with bilingual models (73.3%) in Section 6.4 or with cross-language MAP adaptation (74.9%) in Section 6.5.1.

6.6.2 Data pooling followed by adaptation

It was decided to evaluate the performance of transformation-based adaptation of bilingual models for the purpose of comparing the results with MAP adaptation under the same circumstances, even though the meaning of such a procedure is not intuitive. Figure 6.10

shows the performance achieved when models trained on the pooled English and Afrikaans training set and on the pooled English and Afrikaans training subset are transformed using the Afrikaans training set and training subset respectively. Interestingly, MLLR transformation using the full Afrikaans training set delivers peak word accuracy of 74.4%, improving by 1.1% on simple bilingual pooling with the English set (73.3% word accuracy). For the Afrikaans training subset, however, transformation of bilingual models degrades accuracy compared to bilingual pooling with the English set. This is probably due to the fact that the transformation changes the parameters too much, based on a small amount of data, thereby degrading the positive influence the large amount of English data had in the performance of the bilingual models.

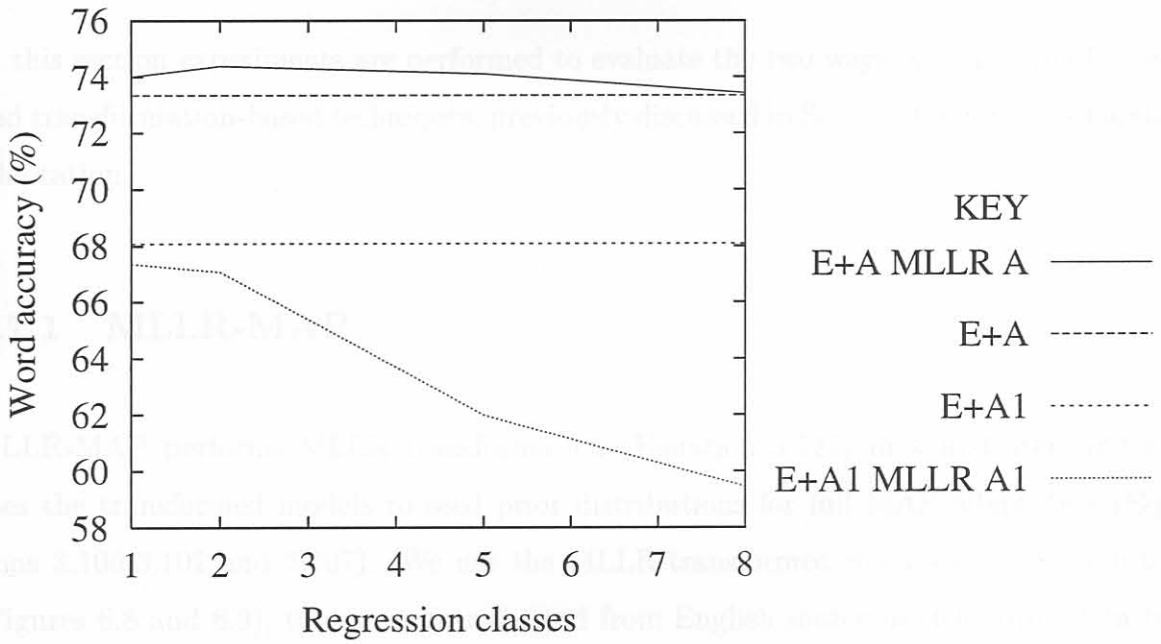

Figure 6.10: Word accuracy on the Afrikaans test set for (mean only) MLLR transformation of Gaussian means as a function of the number of regression classes for pooled English-Afrikaans models adapted on the Afrikaans training set (A) and Afrikaans training subset (A1)

Results for methods that perform variance compensation are not shown because they were found to degrade performance for bilingual model adaptation. This agrees with the results for MAP adaptation, where it was found that variance compensation of the bilingual models was less important than for the English models.

The peak word accuracy of 74.4% achieved with MLLR transformation is less than the 75.3% achieved with mean-only MAP adaptation of bilingual models in Section 6.5.4. Furthermore, transformation of the bilingual models presents a risk since it may degrade performance if too little target data is available, such as is the case for transformations calculated using the Afrikaans training subset. In the next section experiments are performed in an attempt to combine some of the advantages of both Bayesian and transformation-based adaptation.

6.7 Combined transformation-Bayesian adaptation

In this section experiments are performed to evaluate the two ways of combining Bayesian and transformation-based techniques, previously discussed in Section 3.4, for cross-language adaptation.

6.7.1 MLLR-MAP

MLLR-MAP performs MLLR transformation (Equation 3.121) in a first step and then uses the transformed models to seed prior distributions for full MAP adaptation (Equations 3.100-3.102 and 3.107). We use the MLLR-transformed models from Section 6.6.1 (Figures 6.8 and 6.9), that were transformed from English source models using data from the Afrikaans training set and training subset. These MLLR-transformed models are used as seed models for further MAP adaptation on the respective Afrikaans sets.

Figure 6.11 shows the word accuracy achieved on the Afrikaans test set as a function of the MAP prior weight when the MLLR transformed models are adapted using full MAP adaptation. Results are shown for single regression class mean-only MLLR transformations as this delivered the best performance, achieving peak performance of 74.8% word accuracy for Afrikaans training set adaptation and 69.9% word accuracy for Afrikaans training

subset adaptation. The performance is slightly below that achieved with MAP adaptation of the English priors, indicating that the additional use of MLLR transformation does not improve performance in this case. This was expected since both the English and Afrikaans data are from the same database and the ability of MLLR to remove overall mismatch is not important. The results are of interest, though, for comparison with results in Chapter 7, where we show that MLLR-MAP is very useful for cross-database adaptation when significant differences exist with respect to the databases.

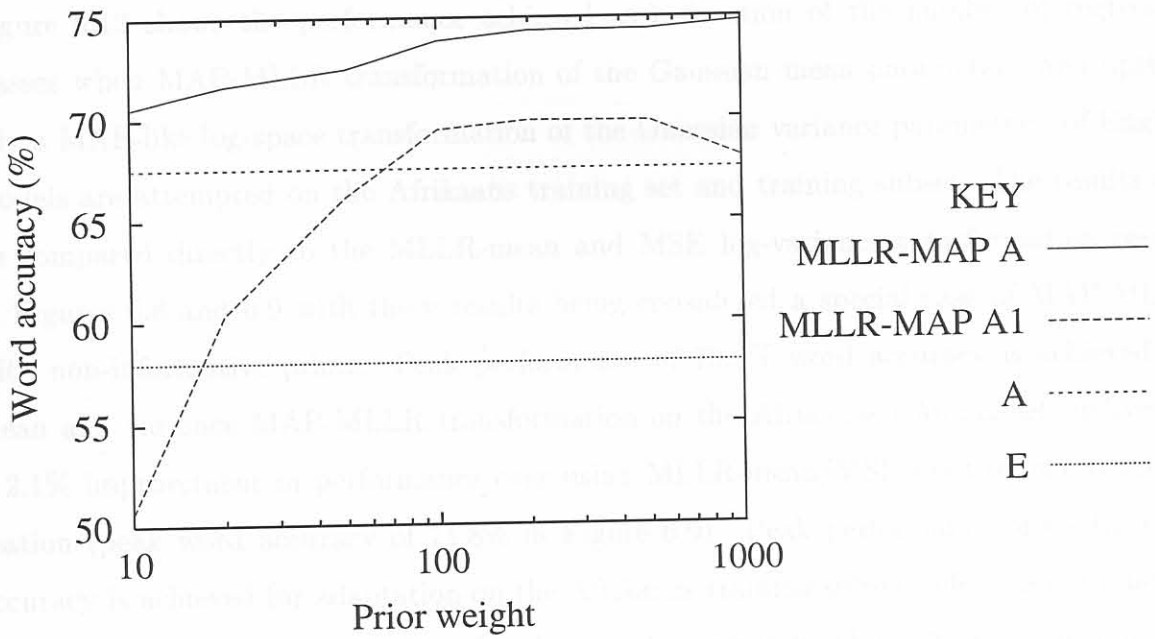

Figure 6.11: Word accuracy on the Afrikaans test set as a function of the adaptation rate for English models adapted using MLLR-MAP adaptation on the Afrikaans training set (A) and training subset (A1) with reference performance of monolingual models also shown

6.7.2 MAP-MLLR

MAPLR provides a second way of combining Bayesian and transformation-based adaptation and attempts to determine the linear regression parameters that deliver the maximum *a posteriori* probability model estimate. We have combined MAPLR (Equations 3.138 and 3.139) with a MAP-like variance adaptation technique (Equations 3.140 and 3.141) and

group the combination of the techniques under the name MAP-MLLR. The MAP-MLLR transformations converge to unity transformations as the amount of adaptation data available decreases and converge to the MLLR (for Gaussian means) and log-variance MSE (for Gaussian variance) estimates as the amount of adaptation data available increases. The amount of adaptation that source models incur under the transformation can be controlled and performance can therefore be improved by decreasing the degree of over-fitting, especially for complex transformations that span many regression classes.

Figure 6.12 shows the performance achieved as a function of the number of regression classes when MAP-MLLR transformation of the Gaussian mean parameters, and optionally a MAP-like log-space transformation of the Gaussian variance parameters, of English models are attempted on the Afrikaans training set and training subset. The results can be compared directly to the MLLR-mean and MSE log-variance transformation results in Figures 6.8 and 6.9 with those results being considered a special case of MAP-MLLR with non-informative priors. Peak performance of 73.9% word accuracy is achieved for mean and variance MAP-MLLR transformation on the Afrikaans training set, delivering a 2.1% improvement in performance over using MLLR-mean/MSE log-variance transformation (peak word accuracy of 71.8% in Figure 6.9). Peak performance of 65.9% word accuracy is achieved for adaptation on the Afrikaans training subset, which is 0.2% better than using a non-informative prior for the transformation (peak 65.7% word accuracy in Figure 6.8). The results in Figure 6.12 also show that variance transformation (in addition to mean transformation) significantly outperforms mean-only transformation, by 5.9% on the Afrikaans training set (73.9% versus 68.0% word accuracy) and by 2.6% on the Afrikaans training subset (65.9% versus 63.3% word accuracy).

6.8 Discriminative adaptation

Experiments are performed to evaluate the application of discriminative adaptation for cross-language adaptation, as was discussed in Section 5.3.3. A major consideration when

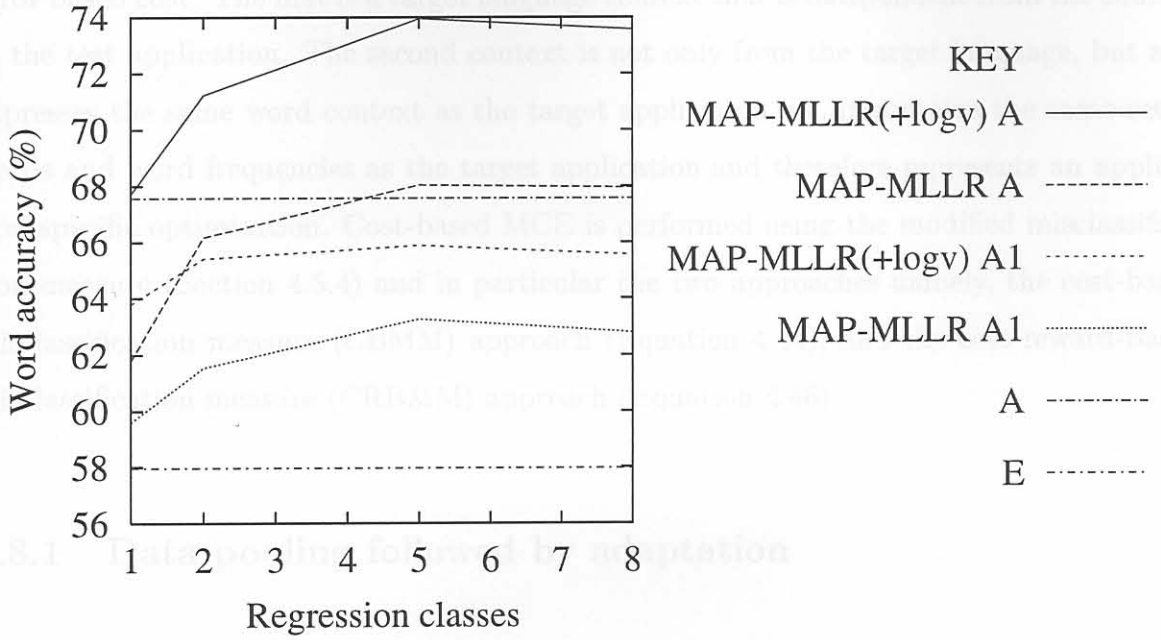

Figure 6.12: Comparison of word accuracy on the Afrikaans test set for MAP-MLLR transformation of Gaussian means, optionally combined with a MAP-like log-space (+logv) MSE transformation of Gaussian variance parameters, computed as a function of the number of regression classes for English models adapted on the Afrikaans training set (A) and training subset (A1)

applying discriminative adaptation is the selection of initial model parameters, as discriminative techniques are prone to converge to local minima (in terms of the loss function). The initial model must therefore be selected to exhibit desirable characteristics and discriminative optimisation is performed only to "fine-tune" the characteristics for the target language. We experiment with initial models that are trained on pooled multilingual data, as well as with models that are the product of other adaptation techniques, such as MAP adaptation, and therefore have already been specialised to some extent for improved target language performance.

The MCE framework for discriminative training from Chapter 4 is used and in particular experiments are performed to determine the performance of the cost-based extensions to MCE that we proposed in Section 4.5. A method from Section 4.5.3 is used to calculate the word error-based cost associated with each phoneme misclassification (in particular Equation 4.43). In experiments, two sets of word contexts were used to derive the word

error-based cost. The first is a target language context that is independent from the context in the test application. The second context is not only from the target language, but also expresses the same word context as the target application, i.e. it contains the same set of words and word frequencies as the target application and therefore represents an application specific optimisation. Cost-based MCE is performed using the modified misclassification measure (Section 4.5.4) and in particular the two approaches namely, the cost-based misclassification measure (CBMM) approach (Equation 4.44), and the cost-reward-based misclassification measure (CRBMM) approach (Equation 4.46).

6.8.1 Data pooling followed by adaptation

MCE adaptation is performed on models trained on pooled English and Afrikaans data. The multilingual models are trained on a large amount of data, ensuring robust parameter estimation, although the models will be biased towards the source language since it represents most of the training data. Discriminative adaptation is performed, using target language data only, to adapt the multilingual models with the aim of improving performance specifically for the target language.

Figure 6.13 shows word accuracy on the Afrikaans test set as a function of the number of adaptation iterations when models trained on the pooled English and Afrikaans training subset (68.1% word accuracy) are used as initial models for MCE adaptation on the Afrikaans training subset. Peak performance of 71.3% word accuracy is achieved with target context CBMM MCE adaptation, which is 3.2% better than the performance of the baseline multilingual models. Similar peak performance is achieved with target context CRBMM MCE adaptation and (independent context) CBMM (both achieve 71.2% word accuracy). MCE adaptation (without a modified misclassification measure) achieves peak performance of 70.6%, which is still 2.5% better than the performance of the multilingual initial models. The best performance of 71.3% word accuracy is, however, 0.1% below the 71.4% word accuracy achieved with MAP adaptation of multilingual models in Figure 6.5. The results indicate that the CRBMM approach to MCE adaptation does not offer im-

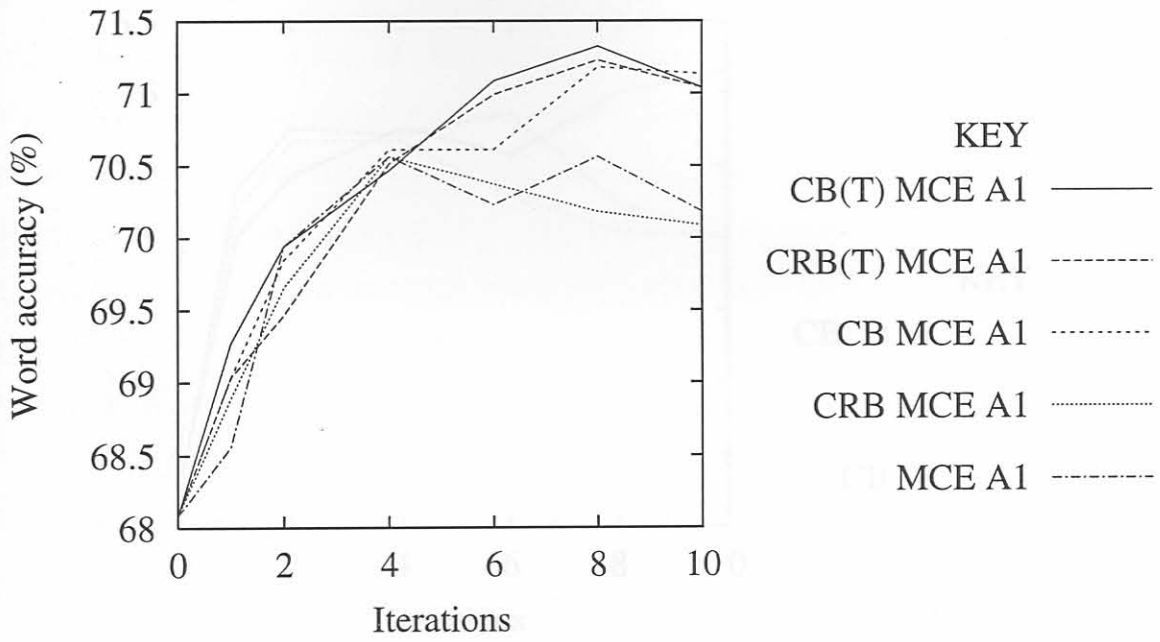

Figure 6.13: Comparison of word accuracy on the Afrikaans test set for the MCE adaptation of pooled English-Afrikaans models on the Afrikaans training subset, also including use of cost-based (CB) and cost-reward-based (CRB) misclassification measures, optionally designed specifically for the target context (T)

proved performance over the CBMM approach (this is also generally the case for the other experiments) and in following experiments we therefore discuss only the CBMM approach for incorporating cost into MCE adaptation.

Figure 6.14 shows word accuracy on the Afrikaans test set as a function of the number of adaptation iterations when models trained on the pooled English and Afrikaans training set (73.3% word accuracy) are used as initial models for MCE adaptation on the Afrikaans training set. Peak performance of 76.1% word accuracy is achieved with target context CBMM MCE adaptation, which is 2.8% better than the performance of the multilingual initial models and is also 0.8% better than the best performance previously reported in this chapter, namely the 75.3% word accuracy achieved with MAP adaptation of multilingual models in Figure 6.6. MCE adaptation (without CBMM) delivers peak performance of 75.9% and CBMM MCE delivers peak performance of 75.8% word accuracy. For Afrikaans training set adaptation, all of the MCE adaptation techniques therefore achieve better performance than the best performing non-MCE techniques that were evaluated.

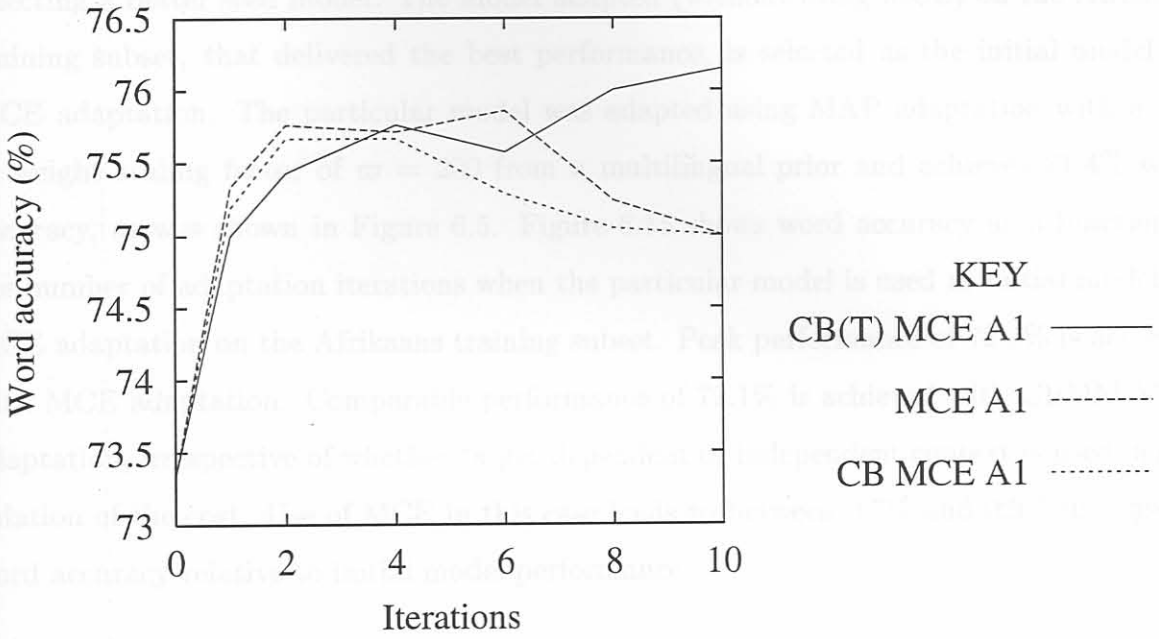

Figure 6.14: Comparison of word accuracy on the Afrikaans test set for the MCE adaptation of pooled English-Afrikaans models on the Afrikaans training set, also including use of a cost-based (CB) misclassification measure, optionally designed specifically for the target context (T)

The experiments with the different approaches to MCE adaptation show significant improvements in performance compared to the performance of multilingual initial models used for adaptation. Target context cost-based MCE adaptation delivers the best performance for multilingual model adaptation, increasing performance by 3.2% for Afrikaans training subset adaptation and by 2.8% for full Afrikaans training set adaptation. The results achieved on the Afrikaans training set are the best results that are reported. The results on the Afrikaans training subset are 0.1% lower than the best non-MCE adaptation method and an attempt is made in the next section to improve on this performance.

6.8.2 Improving best performing models

The performance achieved with MCE adaptation on the Afrikaans training subset (Figure 6.13) is less than the best performance achieved without using MCE. An experiment is performed to test whether performance of the MCE technique can be improved upon by

selecting a better seed model. The model adapted (without using MCE) on the Afrikaans training subset, that delivered the best performance, is selected as the initial model for MCE adaptation. The particular model was adapted using MAP adaptation with a prior weight scaling factor of $\varpi = 200$ from a multilingual prior and achieves 71.4% word accuracy, as was shown in Figure 6.5. Figure 6.15 shows word accuracy as a function of the number of adaptation iterations when the particular model is used as initial model for MCE adaptation on the Afrikaans training subset. Peak performance of 72.3% is achieved with MCE adaptation. Comparable performance of 72.1% is achieved with CBMM MCE adaptation, irrespective of whether target dependent or independent context is used in calculation of the cost. Use of MCE in this case leads to between 0.7% and 0.9% increase in word accuracy relative to initial model performance.

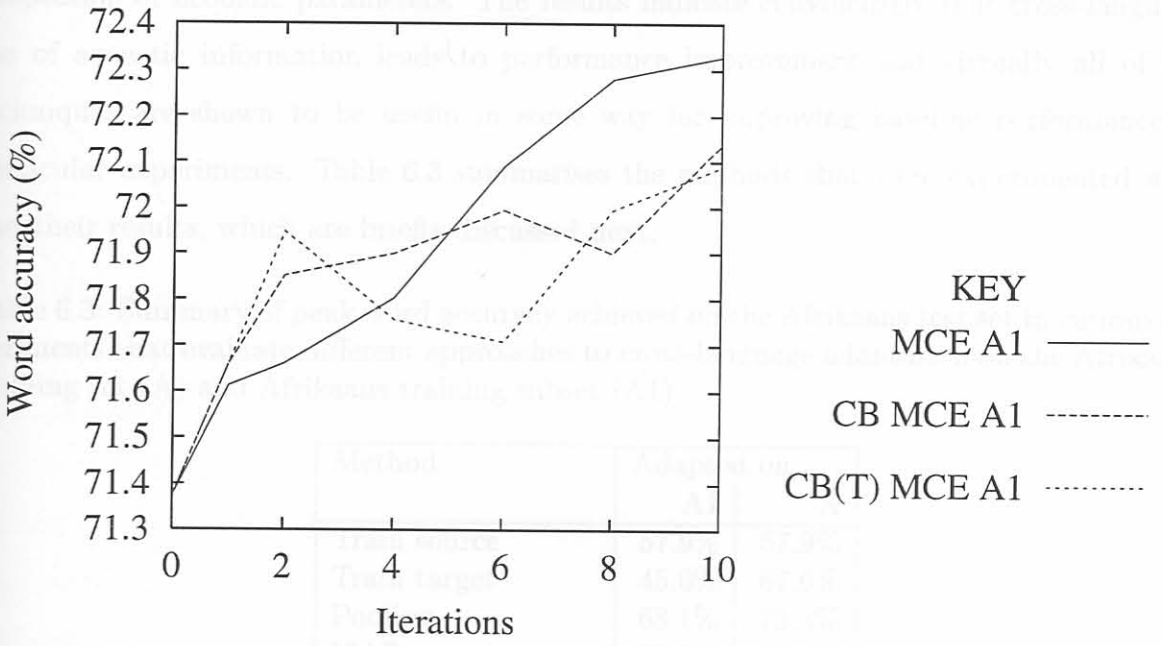

Figure 6.15: Comparison of word accuracy on the Afrikaans test set for the MCE adaptation of models that have already been optimised for performance on the Afrikaans training subset, also including use of cost-based (CB) and cost-reward-based (CRB) misclassification measures, optionally designed specifically for the target context (T)

The same approach, namely to use an initial model that delivers better performance than the multilingual models trained on pooled data, is attempted for the full Afrikaans training set adaptation. However, use of the MAP adapted model (from Figure 6.6) that produced

the best results, as initial model for subsequent MCE adaptation does not deliver improved performance. Initial model performance of 75.3% word accuracy on the Afrikaans test set is only degraded by further MCE adaptation on the Afrikaans training set and is therefore not shown graphically. The result suggests that use of an initial model that achieves better word accuracy does not necessarily ensure that better final performance will be achieved.

6.9 Discussion of results

The experiments in this chapter covered application of the major categories of speaker adaptation techniques, as well as extensions and combinations of them, to cross-language adaptation of acoustic parameters. The results indicate convincingly that cross-language use of acoustic information leads to performance improvement and virtually all of the techniques are shown to be useful in some way for improving baseline performance in particular experiments. Table 6.3 summarises the methods that were experimented with and their results, which are briefly discussed next.

Table 6.3: Summary of peak word accuracy achieved on the Afrikaans test set in various experiments that evaluate different approaches to cross-language adaptation on the Afrikaans training set (A) and Afrikaans training subset (A1)

Method	Adapted on	
	A1	A
Train source	57.9%	57.9%
Train target	45.0%	67.6%
Pooling	68.1%	73.3%
MAP	70.2%	74.9%
Pooling-MAP	71.4%	75.3%
Transformation	65.7%	71.8%
MLLR-MAP	69.9%	74.8%
MAP-MLLR	65.9%	73.9%
Pooling-MCE	71.3%	**76.1%**
Pooling-MAP-MCE	**72.3%**	75.3%

The relatively good results achieved with English (source) language models, as well as

the good results achieved with simple multilingual pooling should be seen in light of the "closeness" of the match between the SUN Speech English and Afrikaans data sets. Because multilingual data from a single database is used, there are no differences with respect to recording conditions between the data sets and also a consistent set of labels were used. This situation facilitates easy cross-language use of speech data. Cross-language MAP adaptation delivers good results, improving even further when adaptation is done from multilingual models, and achieves the best results of the non-discriminative adaptation approaches.

Cross-language transformation-based adaptation does not deliver very good performance and in isolation does not even achieve the level of performance achieved by the multilingual (pooling approach) models. MLLR-MAP delivers good performance, but performance is still less than that achieved with MAP adaptation in isolation - meaning that even the simplest transformation degrades the priors. MAP-MLLR improves upon using MLLR alone, allowing transformation-based adaptation to exceed pooling performance on the Afrikaans training set.

MCE adaptation delivers the best overall performance on both the Afrikaans training set and training subset, irrespective of whether the CBMM approach is used. Use of target context CBMM, in particular, achieves improved performance when adapting multilingual initial models and achieves the best overall performance of 76.1% word accuracy for Afrikaans training set adaptation. For adaptation on the Afrikaans training subset, MCE adaptation of multilingual initial models previously adapted with MAP adaptation (denoted pooling-MAP-MCE in Table 6.3) delivers the best performance of 72.3% word accuracy. MCE-based adaptation of best-performing Afrikaans training set adapted models does, however, not deliver any further improvement in performance.

This chapter discussed experiments performed to evaluate different strategies and techniques for cross-language use of acoustic information. In particular the use of English data from the SUN Speech database in addition to Afrikaans data, also from SUN Speech, was investigated for the purpose of improving recognition performance on an indepen-

dent Afrikaans test set. The results indicate that significant performance improvement is attained by use of the English data in addition to the Afrikaans data, achieving an improvement of 27.3% (72.3% versus 45% word accuracy), or a 50% relative reduction in word error rate over using the Afrikaans training subset alone and an improvement of 8.5% (76.1% versus 67.6% word accuracy), or a 26% relative reduction in word error rate compared with using only the Afrikaans training set. Use of English data in addition to a small amount of Afrikaans data (the training subset) outperforms using five times more Afrikaans data (the full training set) by 3.3% (72.3% versus 67.6%). In the next chapter we investigate to what extent this gain in performance extends to use of acoustic information across different databases.

Chapter 7

Cross-language TIMIT - SUN Speech recognition

This chapter details cross-language, cross-database experiments performed using American English speech from the TIMIT [31] database in conjunction with Afrikaans speech from the SUN Speech [12] database to improve speech recognition performance on Afrikaans. Experiments compare word recognition performance when the set of cross-language adaptation strategies from Chapter 5 are applied. The results can be compared with results in Sections 6.4-6.8 from the previous chapter since the experiments described in this chapter also perform continuous word recognition as described in Section 6.2.3. Experiments in both chapters (Chapters 6 and 7) use reasonably large amounts of English source data in conjunction with smaller amounts of Afrikaans target data and test performance on the same speaker independent Afrikaans test set. Results should therefore give a good indication of the expected variation in performance of different techniques when multilingual data is used from the same database (i.e. the same recording conditions and labelling process) versus using data from different databases. It is expected that the performance achieved with cross-language use of the TIMIT database will be less than that achieved with using English speech from the SUN Speech database in recognising Afrikaans speech from SUN Speech, due to the fact that the characteristics and labelling of the databases differ, but

also because the acoustics of South African English may match the acoustics of Afrikaans more closely than American English. On the other hand, the fact that TIMIT contains approximately 80% more speech data than is contained in the English part of SUN Speech, 9 times more speech data than the Afrikaans training set and 45 times more speech data than the Afrikaans adaptation set, may positively influence performance.

The layout of the chapter is as follows. Some characteristics of the TIMIT database, as well as the mapping of the phoneme labels from TIMIT to SUN Speech are discussed first. Experiments then follow, discussing bilingual data pooling, Bayesian adaptation, transformation-based adaptation, combined Bayesian and transformation-based adaptation, discriminative adaptation and finally data augmentation experiments.

7.1 TIMIT - SUN Speech phonetic mapping

The TIMIT [31] database contains read speech in English from a large number of speakers from various dialect regions in the USA. Utterances are labelled phonetically and contain diverse phonetic content. TIMIT is easily available and has been used in previous research for seeding cross-lingual acoustic models [14, 16]. It is therefore well suited for use as a source language database, especially in our case since it allows some evaluation of the effect of database characteristics on cross-language use of acoustic information.

In order to use the TIMIT database with the SUN Speech database, it is necessary to determine a mapping from TIMIT phoneme labels to the SUN Speech phoneme labels. In Chapter 5 we discussed two methods of determining the phoneme mapping, namely a phonetic knowledge-based approach and an automatic approach to determining a phoneme mapping that uses the Bhattacharyya distance. A phonetic knowledge-based mapping from TIMIT phonemes to SUN Speech phonemes was performed by a phonetic expert, details of which are given in Appendix B. The two mappings agree (i.e. list the same TIMIT label for a given SUN Speech label) on 20 out of the 47 phoneme pairs that are used in

recognition experiments (see Tables 6.1 and 6.2 in Section 6.3.1 for the list of phonemes used in experiments). The automatically determined mapping assigns a smaller subset of the TIMIT phonemes in the mapping process, i.e. only 29 different TIMIT phonemes compared to the 38 different TIMIT phonemes listed as the first entry for the phonetically determined mapping.

Continuous word recognition experiments were performed to compare the performance achieved with the two techniques. Results for models trained on TIMIT data and tested on the Afrikaans test set deliver poor performance, achieving -2.6% word accuracy for the automatic approach and -5.9% accuracy for the phonetic approach. It is not surprising that the automatic approach delivers better performance for direct training, since it selects the "closest" source models, thereby reflecting to some extent the channel differences between the source and target data in its choice. In TIMIT/SUN Speech pooling experiments, however, pooling with models determined by the phonetic approach delivers 55.3% and 45.0% word accuracy, versus 50.9% and 32.7% for the automatic mapping approach, when pooling is done with the Afrikaans training set and subset respectively. Also, MAP adaptation of pooled data models indicates that the phonetically derived mapping produces better final results, with word accuracies of 67.7% and 57.0% achieved versus 66.8% and 54.7% for the automatic mapping approach, when adaptation is done on the Afrikaans training set and subset respectively.

The comparative results indicate that better performance is achieved by using the phonetic mapping approach and therefore results are reported only for the phonetically derived mapping in the rest of the chapter. The phonetically derived mapping associates a quality figure with each source/target phoneme pair, indicating qualitatively how accurate each mapping is expected to be, providing extra information which may be useful for seeding prior weight values for adaptation. We, however, did not experiment with using the quality figures.

7.2 Multilingual data pooling

This section evaluates the performance achieved by models trained on pooled speech data from more than one language and from different databases. Figure 7.1 shows word accuracy achieved on the Afrikaans test set when pooled data consisting of the entire TIMIT database in addition to the SUN Speech Afrikaans training set and training subset are used to train phoneme models. Performance is also shown for models trained on the data sets in isolation. Best performance of 69.0% is achieved by using the Afrikaans training set in isolation. Pooling of the Afrikaans training set with the TIMIT set degrades performance to 55.3% word accuracy. Peak performance of models trained on the Afrikaans training subset and the pooled TIMIT plus Afrikaans training subset both round off to 45.0%. Performance of models trained only on the TIMIT database perform poorly on the Afrikaans test set, achieving peak performance of only -3.7% word accuracy. The poor results indicate

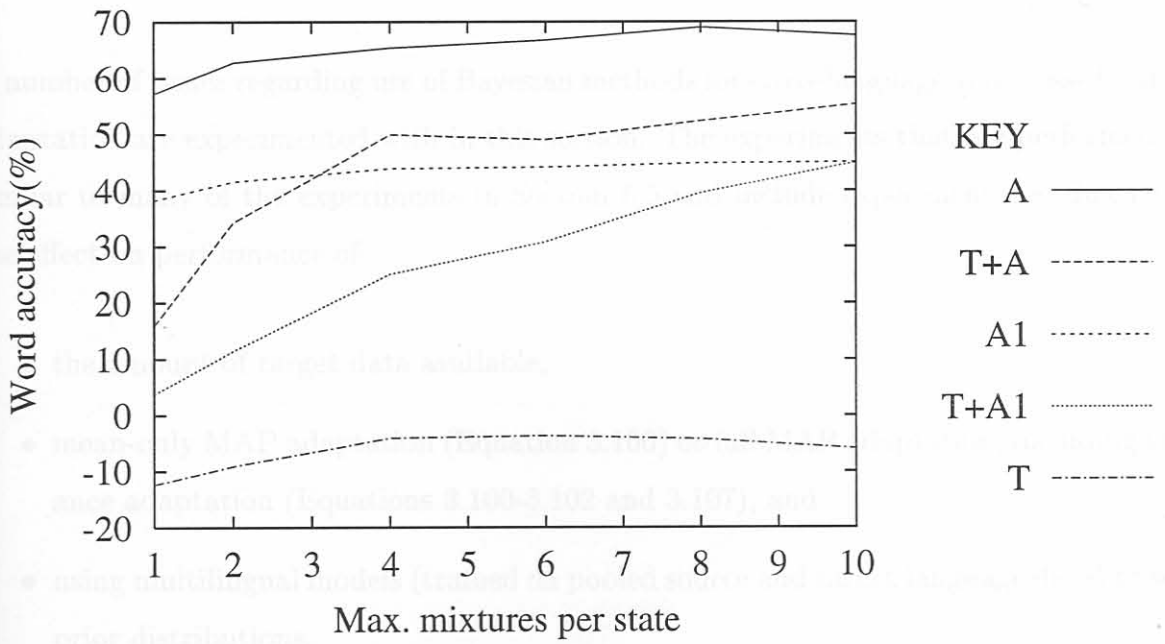

Figure 7.1: Word accuracy as a function of the maximum allowed number of mixtures per state for three state HMMs trained on various monolingual and pooled multilingual data sets using the TIMIT database (T), the Afrikaans training set (A) and the smaller Afrikaans training subset (A1) and tested on the Afrikaans test set

that a large mismatch exists between the TIMIT and SUN Speech databases, especially

if the results are compared to the results of same-database experiments in Section 6.4, where pooling of English and Afrikaans data from the SUN Speech database delivered better results than using Afrikaans data alone. In Section 7.7 we experiment with a data augmentation approach that attempts to improve upon the results achieved with simple data pooling.

For both the pooling approaches (TIMIT pooled with either the Afrikaans training set or training subset), we expect that the performance of pooled-data models will improve if more complex models are trained, i.e. if more than 10 mixtures per state are allowed, but we restrict our attention to using techniques that improve model performance without increasing model complexity.

7.3 Bayesian adaptation

A number of issues regarding use of Bayesian methods for cross-language and cross-database adaptation are experimented with in this section. The experiments that are performed are similar to many of the experiments in Section 6.5 and include experimental evaluation of the effect on performance of:

- the amount of target data available,
- mean-only MAP adaptation (Equation 3.100) or full MAP adaptation, including variance adaptation (Equations 3.100-3.102 and 3.107), and
- using multilingual models (trained on pooled source and target language data) to seed prior distributions.

All experiments also evaluate the influence on performance of the overall weight associated with the prior distribution as this value is determined empirically. The experiments all perform Bayesian adaptation, using the MAP estimation equations from Sections 3.2.3-3.2.5 and in particular Equation 3.107 for variance estimation.

7.3.1 Adaptation performance

Figure 7.2 shows the performance achieved as a function of the adaptation rate for TIMIT English prior models adapted on the SUN Speech Afrikaans training set and the Afrikaans training subset. Peak performance of 67.7% word accuracy is achieved when adapting on the full Afrikaans training set, which delivers an absolute 0.1% improvement over using only the Afrikaans training set (67.6% word accuracy for 3 state, 10 mixture models). Adaptation on the Afrikaans training subset achieves peak performance of 57.0% word accuracy, which is 12.0% better than that achieved by models trained on the Afrikaans training subset alone (45.0%) or by models trained on the pooled TIMIT/Afrikaans training subset (also 45.0% word accuracy).

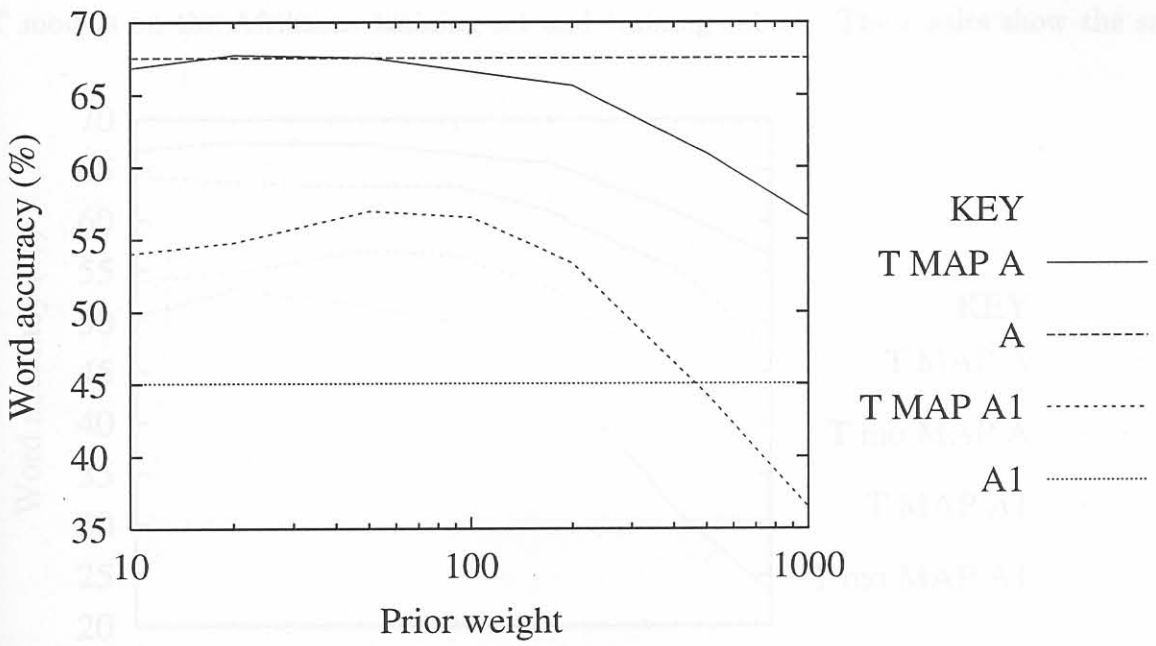

Figure 7.2: Word accuracy on the Afrikaans test set as a function of the adaptation rate for TIMIT models (T) adapted using MAP adaptation on the Afrikaans training set (A) and training subset (A1) with reference performance of monolingual models also shown

The results using English prior models trained on TIMIT are significantly poorer than corresponding results obtained using English priors trained on SUN Speech (67.7% versus 74.9% word accuracy for the Afrikaans training set and 57.0% versus 70.2% word accuracy

for the Afrikaans training subset). Peak performance for TIMIT priors is also achieved for smaller prior weighting ($20 < \varpi < 50$) than the weighting that delivers peak performance for the SUN Speech English prior models ($100 < \varpi < 200$), indicating that the TIMIT priors are less informative than the SUN Speech English priors. The disparity in performance between using TIMIT priors and SUN Speech English priors is expected since the pooling results (Sections 6.4 and 7.2) also show that the English SUN Speech data matches the SUN Speech Afrikaans data more closely than is the case for the TIMIT data.

7.3.2 Variance parameter adaptation

Mean-only and full MAP adaptation are compared in Figure 7.3 for the adaptation of TIMIT models on the Afrikaans training set and training subset. The results show the same

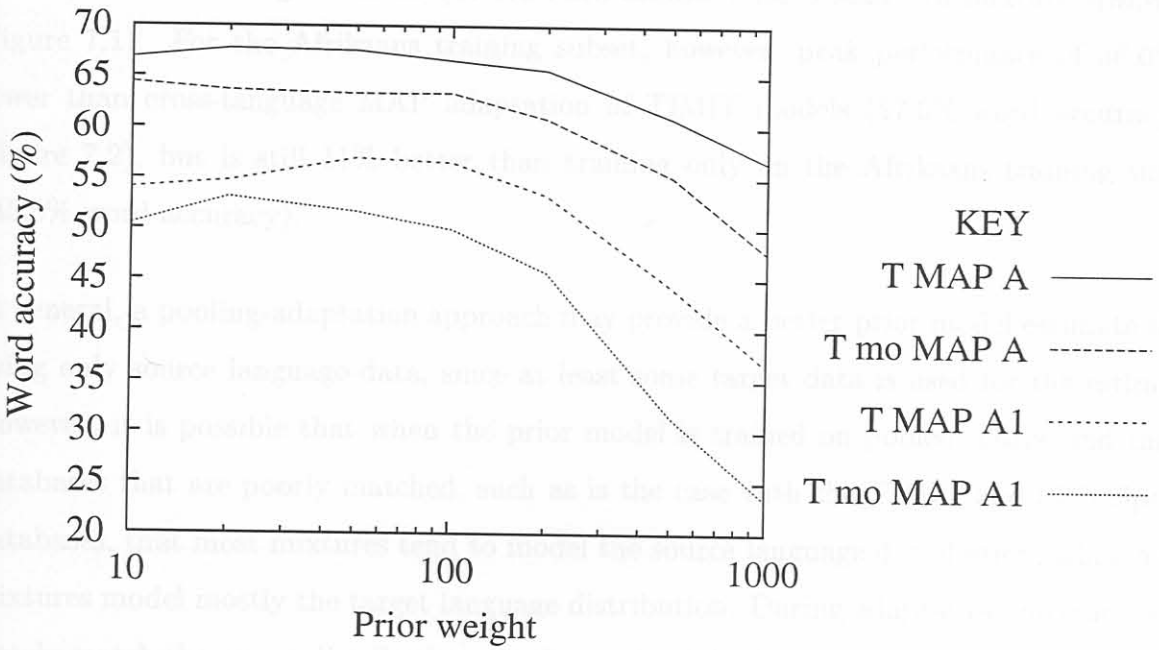

Figure 7.3: Comparison of word accuracy on the Afrikaans test set for mean-only (mo MAP) and full MAP (MAP) adaptation as a function of the adaptation rate for TIMIT models (T) adapted on the Afrikaans training set (A) and training subset (A1)

trend as was shown for adaptation of English prior models from SUN Speech, namely that better cross-language adaptation performance is achieved for full MAP adaptation than for

mean-only MAP adaptation. A 4.1% degradation in peak word accuracy (63.6% versus 67.7%) is attributable to mean-only adaptation versus full adaptation on the Afrikaans training set and a 3.9% degradation in peak word accuracy (53.1% versus 57.0%) is attributable to mean-only versus full adaptation on the Afrikaans training subset.

7.3.3 Pooling-adaptation performance

Figure 7.4 shows the performance achieved when models trained on pooled TIMIT and SUN Speech Afrikaans data set are adapted using full MAP adaptation on the respective Afrikaans data sets. Peak performance of 69.0% is achieved when adapting on the Afrikaans training set, which is 1.3% better than that achieved by cross-language MAP adaptation of TIMIT models (67.7% word accuracy in Figure 7.2) and 1.4% better than training on the Afrikaans training set alone (67.6% word accuracy for 3 state, 10 mixture HMMs in Figure 7.1). For the Afrikaans training subset, however, peak performance of 56.0% is lower than cross-language MAP adaptation of TIMIT models (57.0% word accuracy in Figure 7.2), but is still 11% better than training only on the Afrikaans training subset (45.0% word accuracy).

In general, a pooling-adaptation approach may provide a better prior model estimate than using only source language data, since at least some target data is used for the estimate. However, it is possible that when the prior model is trained on pooled source and target databases that are poorly matched, such as is the case with the TIMIT and SUN Speech databases, that most mixtures tend to model the source language distribution, while a few mixtures model mostly the target language distribution. During adaptation, mixtures that closely match the target distribution are observed, while the large fraction of mixtures that modelled the source language distribution in the initial model are not observed and are therefore also not adapted - negatively influencing performance. This can possibly explain why the pooling-adaptation approach does not necessarily deliver better performance than simply using source language priors for MAP adaptation.

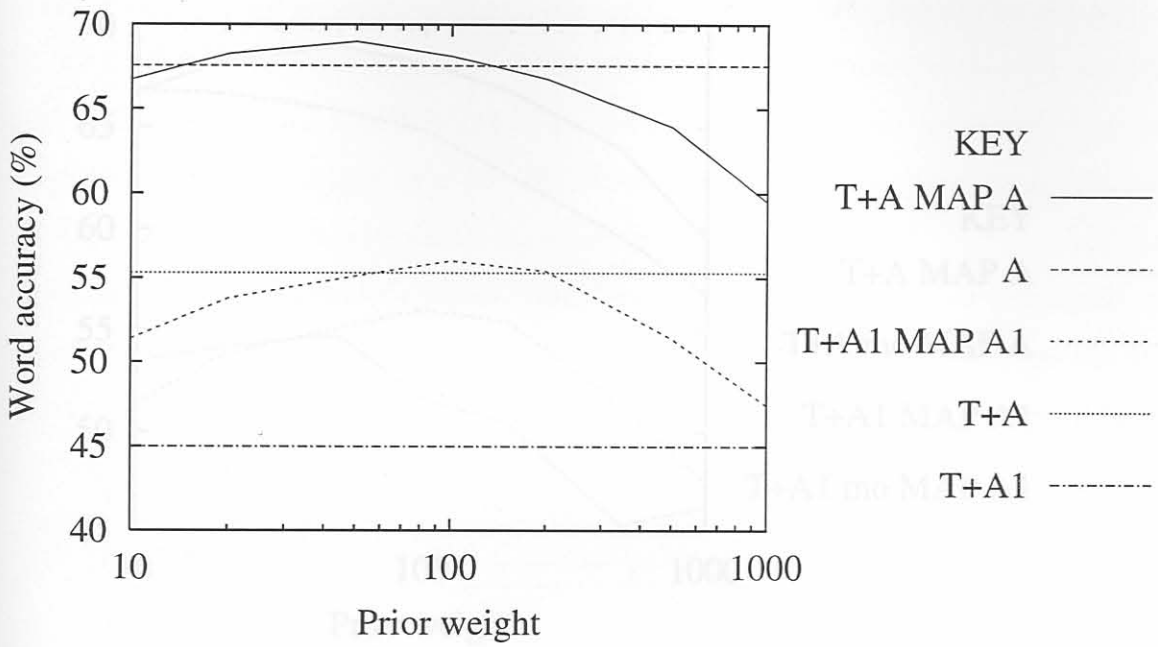

Figure 7.4: Word accuracy on the Afrikaans test set as a function of the adaptation rate (ϖ) for models trained on pooled TIMIT and Afrikaans training data (T+A) and pooled TIMIT and Afrikaans training subset data (T+A1) and adapted using MAP adaptation with reference performance of monolingual and multilingual models also shown

7.3.4 Pooling-variance parameter adaptation

A comparison between results achieved with mean-only and full MAP adaptation of models trained on the pooled TIMIT/Afrikaans data set is given in Figure 7.5. In contrast to the pooled-model MAP adaptation results of the previous chapter (see Figure 6.6), full adaptation outperforms mean-only adaptation for both Afrikaans sets, achieving 2.2% improvement for the Afrikaans training set (69.0% versus 66.8%) and 1.2% improvement for the Afrikaans training subset (56.0% versus 54.8% word accuracy). This may be due in part to the fact that the TIMIT set is even larger than the SUN Speech English set, thereby dominating the pooled model parameters to a larger extent and necessitating variance adaptation since the Afrikaans speech characteristics are not adequately represented in the pooled models.

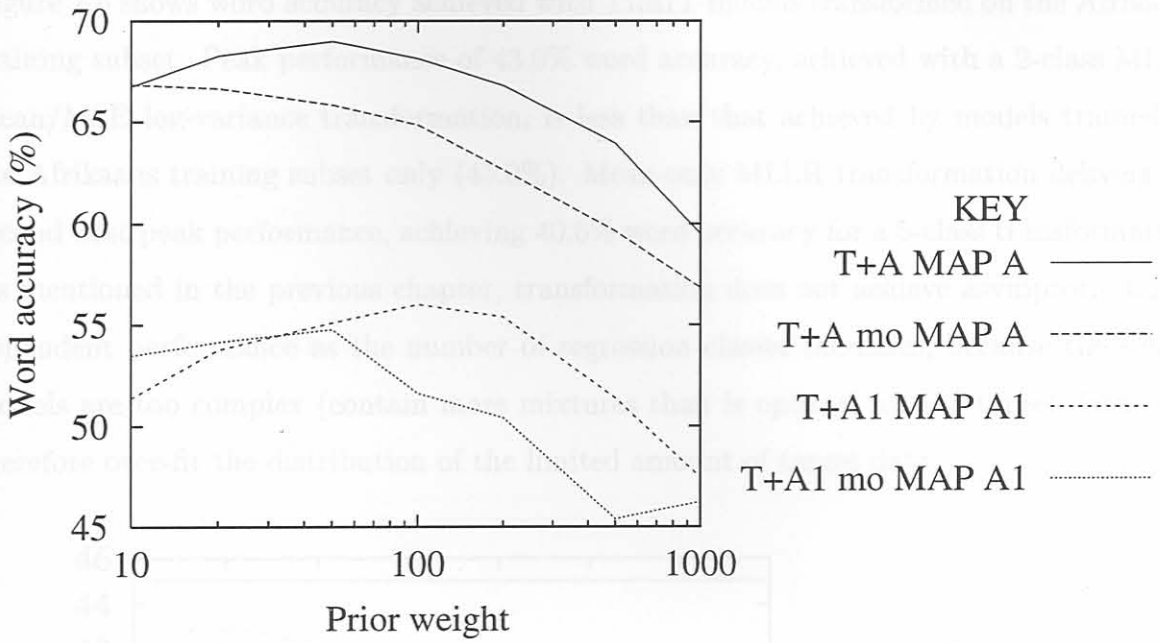

Figure 7.5: Comparison of word accuracy on the Afrikaans test set for mean-only (mo MAP) and full MAP (MAP) adaptation as a function of the adaptation rate for pooled TIMIT-Afrikaans models adapted on the Afrikaans training set (A) and training subset (A1)

7.4 Transformation-based adaptation

Experiments are performed to evaluate the performance of parameter transformation for cross-language and cross-database adaptation, as discussed in Section 5.3.2. Maximum likelihood linear regression (MLLR) transformation (Equation 3.121) is used to transform Gaussian mean parameters and various methods are experimented with for adaptation of Gaussian variance parameters, including:

- no adaptation,
- direct re-estimation (on only the target data),
- linear transformation with MSE criterion (Equation 3.129), and
- log-domain transformation with MSE criterion (Equation 3.136).

Figure 7.6 shows word accuracy achieved with TIMIT models transformed on the Afrikaans training subset. Peak performance of 43.0% word accuracy, achieved with a 2-class MLLR mean/MSE log-variance transformation, is less than that achieved by models trained on the Afrikaans training subset only (45.0%). Mean-only MLLR transformation delivers the second best peak performance, achieving 40.6% word accuracy for a 5-class transformation. As mentioned in the previous chapter, transformation does not achieve asymptotic target dependent performance as the number of regression classes increases, because the source models are too complex (contain more mixtures than is optimal for the target data) and therefore over-fit the distribution of the limited amount of target data.

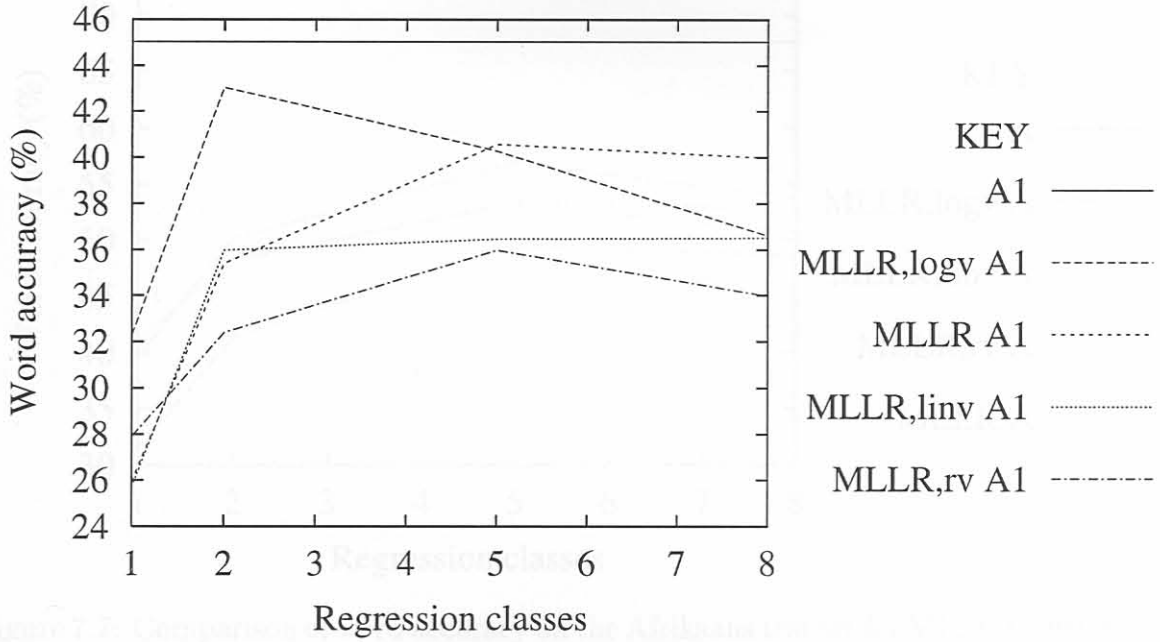

Figure 7.6: Comparison of word accuracy on the Afrikaans test set for MLLR transformation of Gaussian means combined with different variance transformation techniques: mean-only, log variance MSE (logv), linear variance MSE (linv) and variance re-estimation (rv) as a function of the number of regression classes for TIMIT models adapted on the Afrikaans training subset (A1)

Figure 7.7 shows word accuracy achieved with TIMIT models transformed on the Afrikaans training set. Peak performance of 56.9% word accuracy is achieved with a 5-class MLLR mean/MSE log-variance transformation, but is still less than the 67.6% accuracy achieved by models trained on the Afrikaans training set. Linear variance (53.5%) and variance re-

estimation (53.1%) deliver poorer performance, with poorest performance (49.6%) achieved with mean-only MLLR transformation. Other transformation approached were attempted, including block-diagonal transformation which computes separate transformations for cepstral, delta and delta-delta coefficients, as well as a diagonal transformation, which transforms each feature dimension independently. Use of these simpler transformations allows the use of a larger number of regression classes (up to 47 regression classes were used with the diagonal transformation), but did not improve upon the performance achieved with full transformation matrices (results not shown).

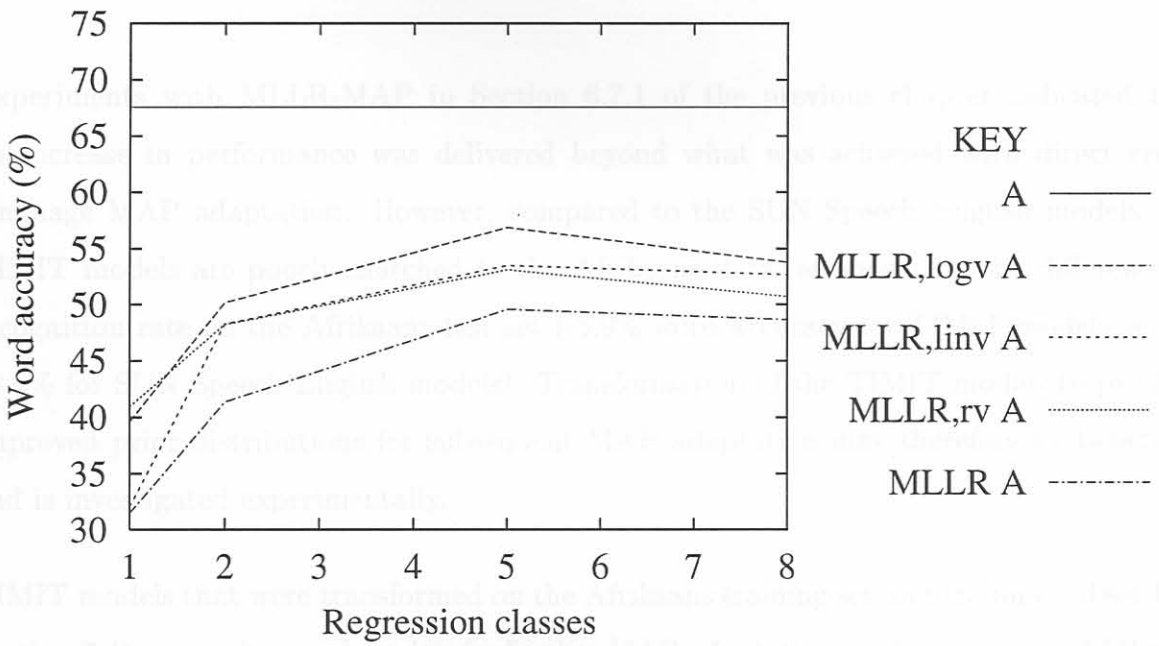

Figure 7.7: Comparison of word accuracy on the Afrikaans test set for MLLR transformation of Gaussian means combined with different variance transformation techniques: mean-only, log variance MSE (logv), linear variance MSE (linv) and variance re-estimation (rv) as a function of the number of regression classes for TIMIT models adapted on the Afrikaans training set (A)

It is, of course, meaningless to perform transformation-based adaptation if the performance achieved is less than that achieved by direct training on the target language data. However, in the next section we show that very good performance is obtained when these transformed models are used for further adaptation.

7.5 Combined transformation-Bayesian adaptation

Experiments are performed to evaluate the two ways of combining Bayesian and transformation-based techniques from Section 3.4 for cross-language and cross-database adaptation and show how MLLR-MAP in particular can lead to greatly improved performance over either MLLR or MAP approaches in isolation.

7.5.1 MLLR-MAP

Experiments with MLLR-MAP in Section 6.7.1 of the previous chapter indicated that no increase in performance was delivered beyond what was achieved with direct cross-language MAP adaptation. However, compared to the SUN Speech English models, the TIMIT models are poorly matched to the Afrikaans data, as shown by the difference in recognition rate on the Afrikaans test set (-5.9% word accuracy for TIMIT models versus 58.0% for SUN Speech English models). Transformation of the TIMIT models to produce improved prior distributions for subsequent MAP adaptation may therefore be beneficial and is investigated experimentally.

TIMIT models that were transformed on the Afrikaans training set and training subset (see Section 7.4) are used as seed models for further MAP adaptation on the respective Afrikaans sets. Figure 7.8 shows the word accuracy achieved as a function of the adaptation rate when the MLLR transformed models are adapted using MAP adaptation. Peak performance of 72.0% word accuracy is achieved when a 2-class (mean-only) MLLR transformation is followed by full MAP adaptation on the Afrikaans training set. This peak performance is 4.4% better than achieved with training on the Afrikaans set only (72.0% versus 67.6%) and 3.0% better than the best MAP adaptation results using TIMIT (69.0% word accuracy for adaptation of bilingual models in Figure 7.4). When the Afrikaans training subset is used for adaptation purposes, peak performance of 64.1% is achieved when a single class MLLR transformation is followed by MAP adaptation. This peak performance is

19.1% better than the 45.0% word accuracy achieved with models trained on the Afrikaans training subset only and 7.1% better than the 57.0% word accuracy achieved with MAP adaptation on the Afrikaans training subset in Figure 7.2. Use of the MSE log-variance

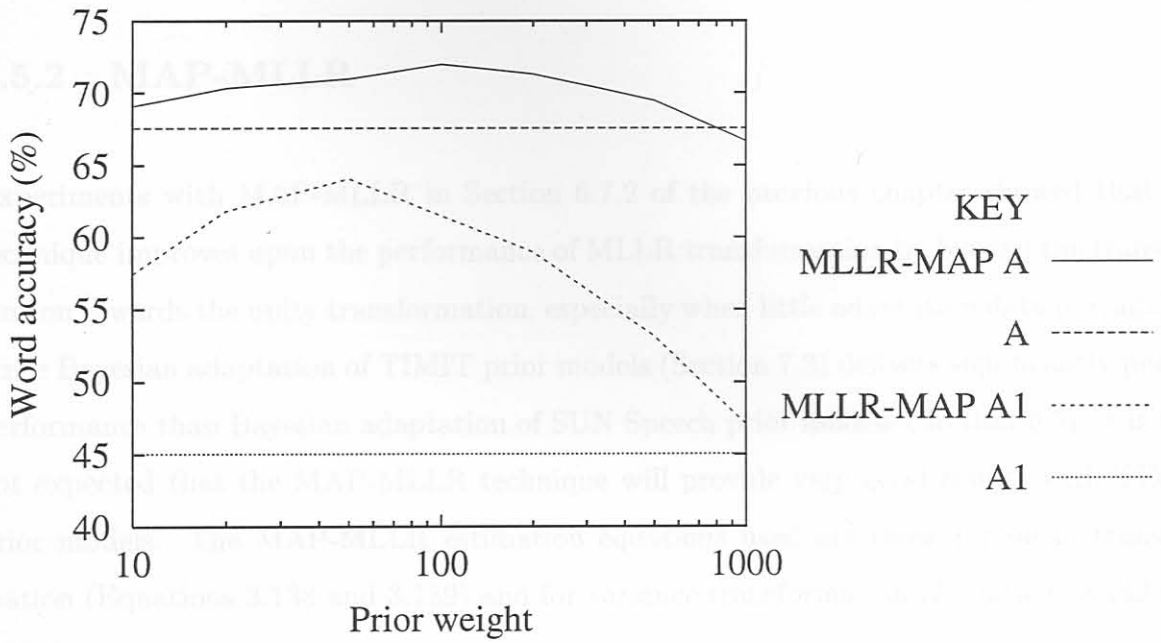

Figure 7.8: Word accuracy on the Afrikaans test set as a function of the adaptation rate for TIMIT models adapted using MLLR-MAP adaptation on the Afrikaans training set (A) and training subset (A1) with reference performance of monolingual models also shown

transformation in the first step of MLLR-MAP produces almost the same performance as using MLLR transformation (71.8% for a single regression class versus 72.0% for the 2-class MLLR transformation).

The results of Figure 7.8 show useful increases in performance by using the TIMIT database to improve the Afrikaans recogniser, indicating that the MLLR-MAP strategy is well suited for cross-database, cross-language adaptation. Best performance is not achieved by using the MLLR models that deliver the best performance (this would imply using 2-class and 5-class transformed models), but by using models transformed with simpler transformations (single and 2-class transformations). The transformation step acts to improve the priors by using correlation between the source language feature distribution and the target language feature distribution. The transformation step should therefore not necessarily be optimised

for transformed model performance because a too complex transformation may over-fit the target data, decreasing the usefulness of the transformed model in seeding the priors for subsequent MAP adaptation.

7.5.2 MAP-MLLR

Experiments with MAP-MLLR in Section 6.7.2 of the previous chapter showed that the technique improves upon the performance of MLLR transformation by biasing the transformation towards the unity transformation, especially when little adaptation data is available. Since Bayesian adaptation of TIMIT prior models (Section 7.3) delivers significantly poorer performance than Bayesian adaptation of SUN Speech prior models (Section 6.5), it is also not expected that the MAP-MLLR technique will provide very good results with TIMIT prior models. The MAP-MLLR estimation equations used are those for mean transformation (Equations 3.138 and 3.139) and for variance transformation (Equations 3.140 and 3.141).

Figure 7.9 shows the performance achieved as a function of the number of regression classes when MAP-MLLR transformation of the Gaussian mean parameters and, optionally, a MAP-like log-space transformation of the Gaussian variance parameters of TIMIT models are attempted on the Afrikaans training set and training subset. Peak performance of 56.1% word accuracy is achieved for mean and variance MAP-MLLR transformation on the Afrikaans training set. This performance is for a prior weight scaling factor of 10 ($\varpi = 10$), and is less than the peak word accuracy of 56.9% achieved with MLLR-mean/MSE log-variance transformation in Figure 7.7, i.e. when a prior weight scaling factor of zero is used ($\varpi = 0$), and is also significantly less than the performance achieved with models trained directly on the Afrikaans training set (67.6% word accuracy). The best performance on the Afrikaans training subset is 45.1% (also using $\varpi = 10$), at least slightly improving on direct training on the Afrikaans training subset (45.0% word accuracy) and also improving on using zero prior weighting (43.0% word accuracy).

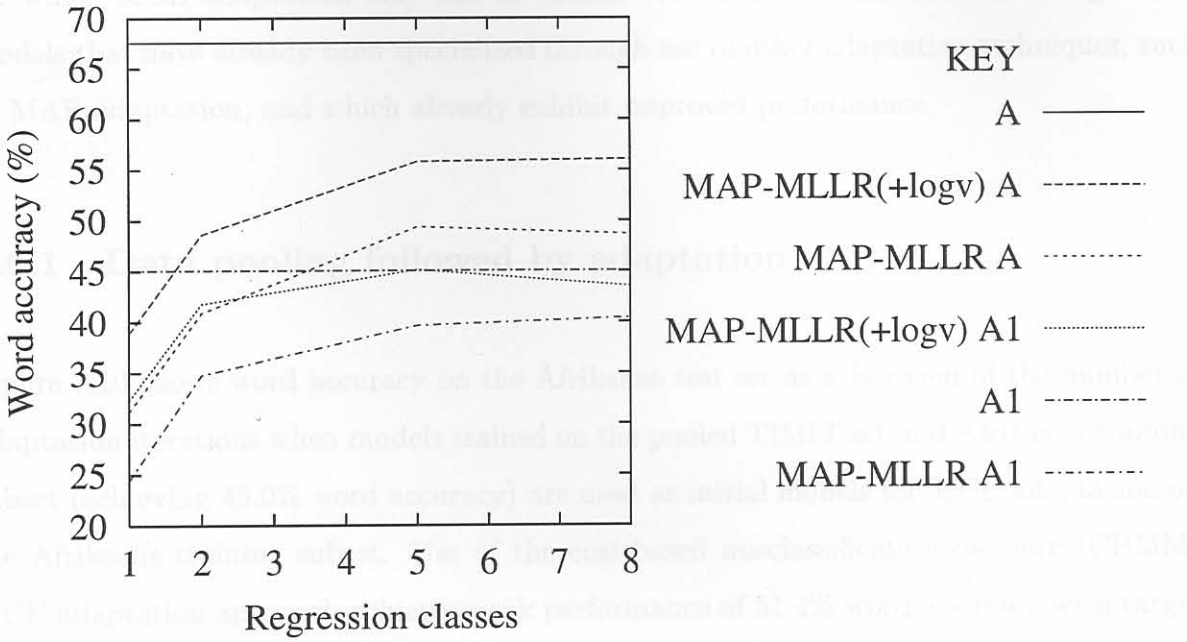

Figure 7.9: Comparison of word accuracy on the Afrikaans test set for MAP-MLLR transformation of Gaussian means, optionally combined with a MAP-like log-space (+logv) MSE transformation of Gaussian variance parameters, computed as a function of the number of regression classes for TIMIT models adapted on the Afrikaans training set (A) and training subset (A1)

7.6 Discriminative adaptation

Experiments are performed to evaluate the application of discriminative adaptation for cross-language and cross-database adaptation, as was discussed in Section 5.3.3. MCE adaptation experiments in Section 6.8 of the previous chapter delivered good performance when using initial models trained on pooled multilingual data and we therefore also attempt the same approach in experiments in this section. The reasoning behind using multilingual initial models is that they are robustly estimated from a large amount of data (pooled multilingual data) and may need only a degree of language specific "fine-tuning" to deliver good target language performance. It should, however, be taken into account that pooling Afrikaans data from SUN Speech with data from the TIMIT database did not improve on using the Afrikaans data alone (see Section 7.2) and may therefore produce initial models for MCE adaptation that need significant adaptation to reach a reasonable level of performance,

for which MCE adaptation may not be suited. We therefore also consider using initial models that have already been specialised through use of other adaptation techniques, such as MAP adaptation, and which already exhibit improved performance.

7.6.1 Data pooling followed by adaptation

Figure 7.10 shows word accuracy on the Afrikaans test set as a function of the number of adaptation iterations when models trained on the pooled TIMIT set and Afrikaans training subset (achieving 45.0% word accuracy) are used as initial models for MCE adaptation on the Afrikaans training subset. Use of the cost-based misclassification measure (CBMM) MCE adaptation approach achieves peak performance of 51.4% word accuracy with target context cost, compared to 49.4% word accuracy with target independent cost. Performance using MCE adaptation (without CBMM) is not as good, and only improves by 0.2% on the initial model performance (45.2% versus 45.0% word accuracy). The best performance of 51.4%, achieved with MCE adaptation using the Afrikaans training subset (obtained with CBMM MCE), is 6.4% better than that achieved with the Afrikaans training subset alone (45.0% word accuracy), delivering useful cross-language adaptation performance. The performance (51.4%) is, however, below the peak accuracy of 64.1% achieved with MLLR-MAP adaptation on the Afrikaans training subset.

For comparison purposes, MCE adaptation experiments were also performed using an initial model trained directly on the Afrikaans training set. Adaptation using target context CBMM achieves peak performance of 47.4%, which improves on the baseline 45.0% performance achieved with direct training on the Afrikaans training subset, but which is less than that achieved by adapting the multilingual (TIMIT and Afrikaans training subset) initial model.

Figure 7.11 shows word accuracy as a function of the number of adaptation iterations when models trained on the pooled TIMIT and Afrikaans training set (achieving 55.3% word accuracy) are used as initial models for MCE adaptation on the Afrikaans training set.

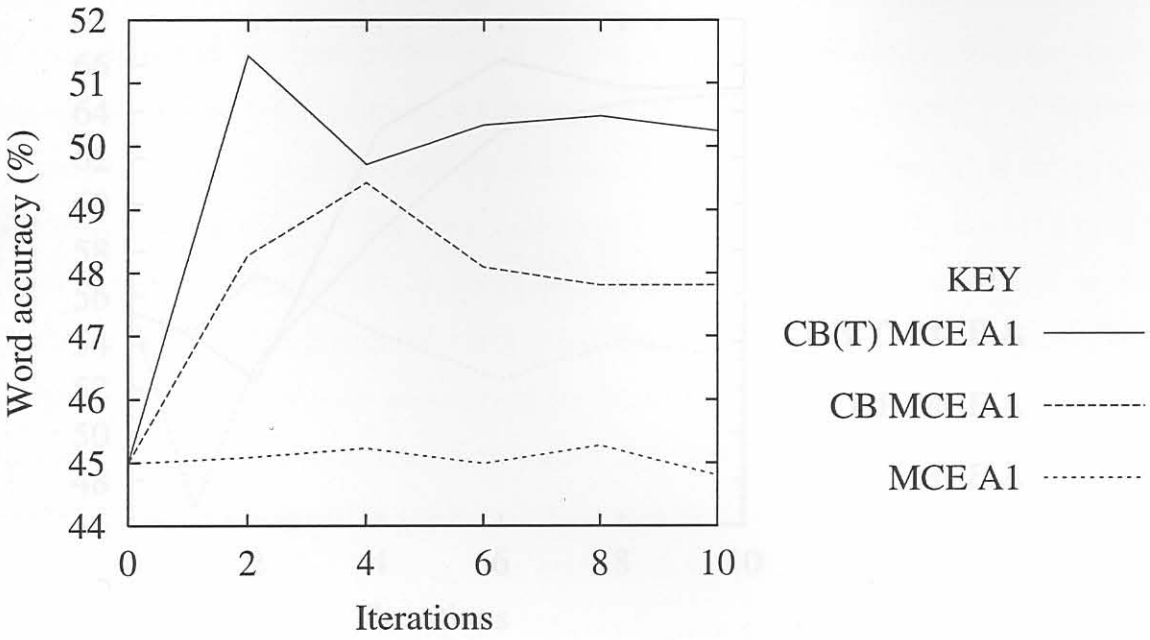

Figure 7.10: Comparison of word accuracy on the Afrikaans test set for MCE adaptation of pooled TIMIT-Afrikaans models on the Afrikaans training subset, also showing results using of a cost-based (CB) misclassification measure, optionally designed specifically for the target context (T)

CBMM MCE adaptation achieves peak performance of 66.2% word accuracy with target context cost and 64.7% word accuracy with target independent cost. As in the previous experiment, performance of MCE adaptation without CBMM is not as good as MCE with CBMM, and achieves only 57.0% peak word accuracy.

The best performance of 66.2% (achieved with target context CBMM MCE) is still below the baseline 67.6% achieved with direct training on the Afrikaans training set. The result illustrates a drawback of the pooling-adaptation approach. If the performance of the models trained on pooled multilingual data is far below that of models trained directly on the target language data only (e.g. in this case 55.3% for multilingual models versus 67.6% for target language only models), then the tendency of discriminative adaptation techniques to converge to local minima may result in a poorer final model than simple target language training.

The experiments with MCE adaptation show reasonable improvements in performance com-

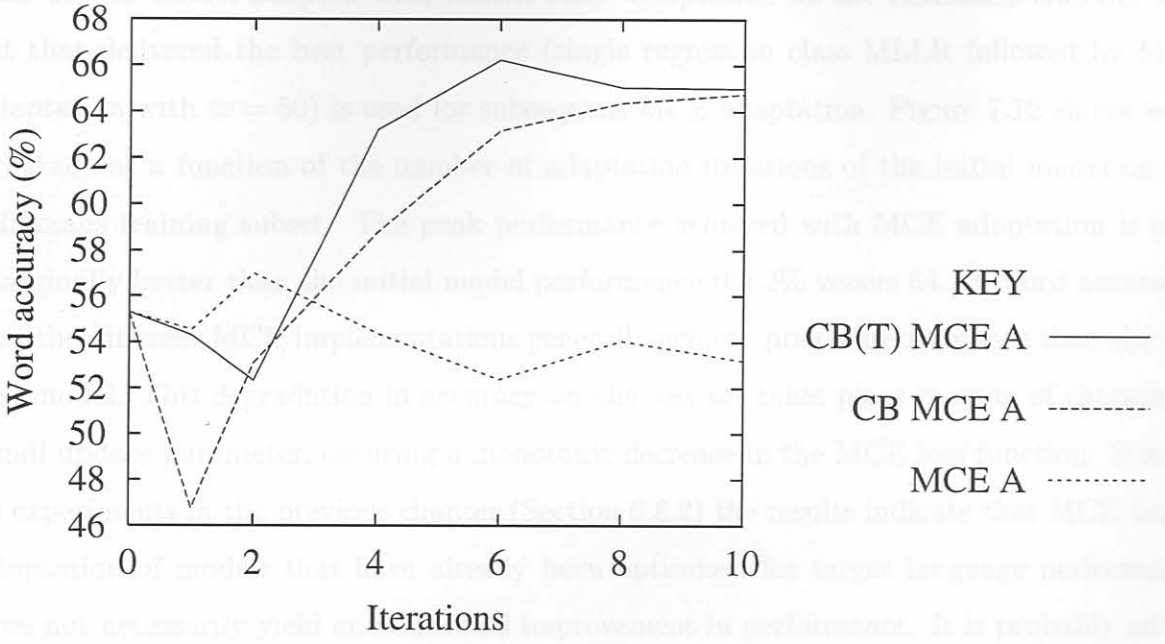

Figure 7.11: Comparison of word accuracy on the Afrikaans test set for the MCE adaptation of pooled TIMIT-Afrikaans models on the Afrikaans training set, also including use of a cost-based (CB) misclassification measure, optionally designed specifically for the target context (T)

pared to the performance of the initial models before adaptation. Of the MCE approaches, target context cost-based MCE adaptation delivers the best performance for multilingual model adaptation, increasing performance by 6.4% for Afrikaans training subset adaptation (51.4% versus 45.0% word accuracy). For Afrikaans training set adaptation, peak performance of 66.2% is achieved, which is still below that achieved with the Afrikaans training set in isolation (67.6%).

7.6.2 Improving best performing models

The performance achieved with MCE adaptation on both the Afrikaans training set and subset (Figures 7.10 and 7.11) is less than the best performance achieved with other adaptation techniques. The adapted models that delivered the best performance in previous experiments are now used as initial models in an attempt to improve performance with MCE adaptation.

The TIMIT model adapted with MLLR-MAP adaptation on the Afrikaans training subset that delivered the best performance (single regression class MLLR followed by MAP adaptation with $\varpi = 50$) is used for subsequent MCE adaptation. Figure 7.12 shows word accuracy as a function of the number of adaptation iterations of the initial model on the Afrikaans training subset. The peak performance achieved with MCE adaptation is only marginally better than the initial model performance (64.2% versus 64.1% word accuracy) and the different MCE implementations generally achieve poorer performance than the initial model. This degradation in accuracy on the test set takes place in spite of choosing a small update parameter, ensuring a monotonic decrease in the MCE loss function. Similar to experiments in the previous chapter (Section 6.8.2) the results indicate that MCE-based adaptation of models that have already been optimised for target language performance does not necessarily yield an additional improvement in performance. It is probably advisable to use a cross-validation set with such adaptation since a decrease in the MCE loss function does not guarantee improved performance on the test set.

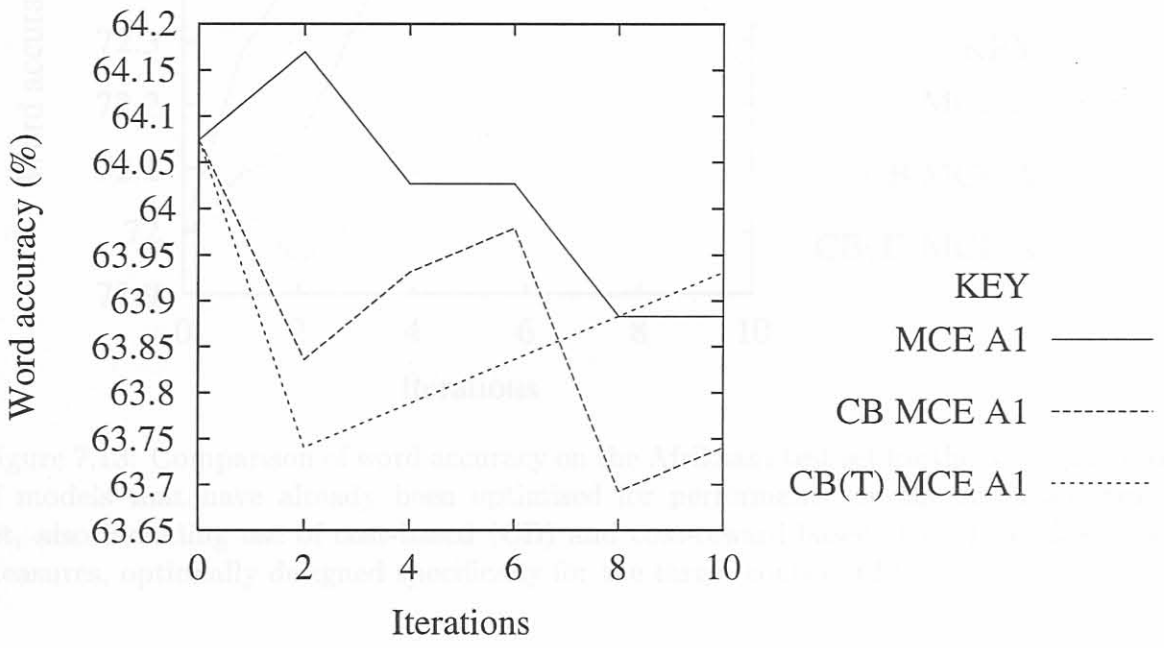

Figure 7.12: Comparison of word accuracy on the Afrikaans test set for the MCE adaptation of models that have already been optimised for performance on the Afrikaans training subset, also including use of cost-based (CB) and cost-reward-based (CRB) misclassification measures, optionally designed specifically for the target context (T)

The next experiment attempts to improve on MCE adaptation performance on the Afrikaans training set. The TIMIT model adapted with MLLR-MAP adaptation on the Afrikaans training set that delivered the best performance (2-class MLLR followed by MAP adaptation with $\varpi = 100$) is used for subsequent MCE adaptation. Figure 7.13 shows word accuracy as a function of the number of adaptation iterations when the initial model is adapted on the Afrikaans training set. Peak performance of 72.7% word accuracy is achieved with MCE adaptation (without using a cost function), which is 0.7% better than (unadapted) initial model performance (72.0% word accuracy). CBMM MCE adaptation delivers peak performance of 72.5% word accuracy and target dependent CBMM MCE delivers only 72.2% peak word accuracy.

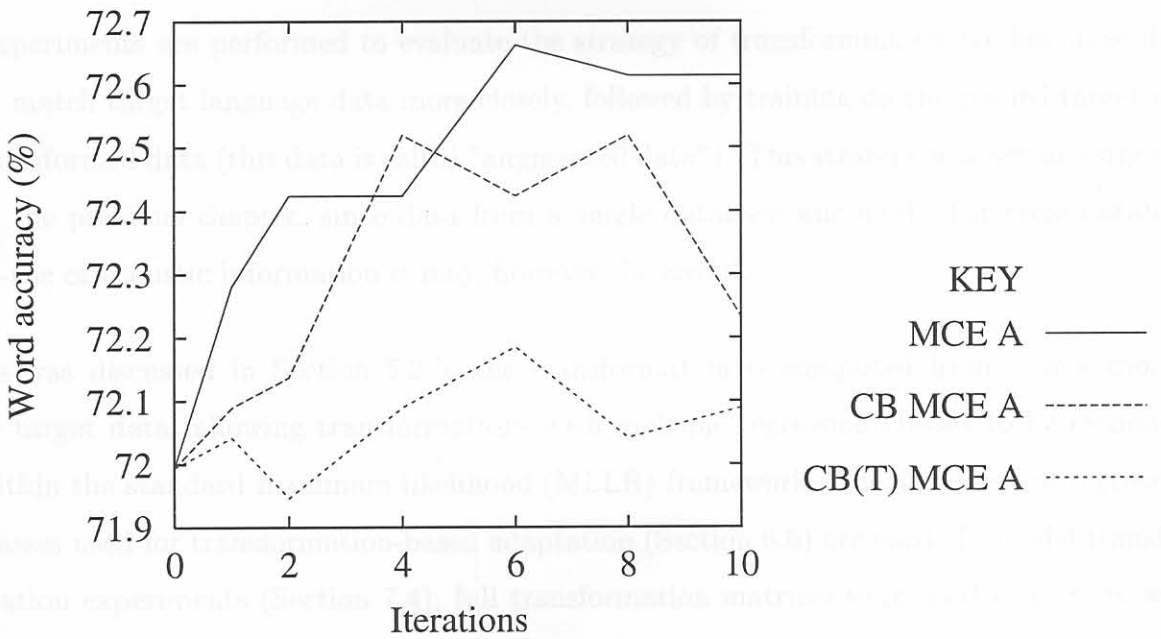

Figure 7.13: Comparison of word accuracy on the Afrikaans test set for the MCE adaptation of models that have already been optimised for performance on the Afrikaans training set, also including use of cost-based (CB) and cost-reward-based (CRB) misclassification measures, optionally designed specifically for the target context (T)

Similar to Section 6.8.2, experimental results for discriminative adaptation of models already optimised for target language dependent performance are not conclusive. When adapted initial models are further adapted with MCE, performance improves for Afrikaans training set adaptation, but does not improve for Afrikaans training subset adaptation. If

this approach is therefore followed, use of a cross-validation set is recommended to ensure that discriminatively adapted models are not used if their performance is less than that achieved with the initial models.

Throughout this chapter different methods of adapting model parameters were investigated. In the next section we investigate a technique for the transformation of source language data.

7.7 Data augmentation

Experiments are performed to evaluate the strategy of transforming source language data to match target language data more closely, followed by training on the pooled target and transformed data (this data is called "augmented data"). This strategy was not investigated in the previous chapter, since data from a single database was used. For cross-database re-use of acoustic information it may, however, be useful.

As was discussed in Section 5.2.5, the transformation is computed from source models to target data, allowing transformations with multiple regression classes to be estimated within the standard maximum likelihood (MLLR) framework. The same sets of regression classes used for transformation-based adaptation (Section 6.6) are used. In model transformation experiments (Section 7.4), full transformation matrices were used since they were found to deliver better performance than diagonal or block-diagonal matrices. For data augmentation, use of diagonal and block-diagonal transformations are reconsidered, especially if only the mel-cepstral coefficients are transformed (i.e. not including time derivative components). Single state HMMs with a maximum of 10 mixtures per state are trained on the TIMIT database and used as source models in the computation of the transformation. A single state source model is used in order to apply a single transformation to data from a particular class. Using multiple transformations per speech segment leads to discontinuities at the alignment points of the data (assuming Viterbi-alignment is done to segment source

data) and can degrade performance when time derivative feature components are calculated afterwards.

Table 7.1 summarises the performance achieved when training on the pooled transformed and target data. Results for zero regression classes indicate pooling with the (untransformed) source data. Diagonal transformation matrices (with offset) of all features, including time derivative features, were used as they were found to deliver better performance than using either block-diagonal or full transformation matrices. The results indicate that

Table 7.1: Word accuracy achieved on the Afrikaans test set for models trained on data from TIMIT that is transformed to better match the respective Afrikaans set and also pooled with the respective Afrikaans set

Set used for adaptation	Regression classes					
	0	1	2	5	8	15
Afr. training set	55.3%	53.3%	48.3%	49.3%	45.8%	40.3%
Afr. training subset	45.0%	31.0%	28.9%	29.4%	25.9%	22.8%

no gain in performance is achieved by first transforming the TIMIT data before pooling it with the SUN Speech data for the training of multilingual models. The transformation improves the likelihood of the source model (on the target data) when it is transformed, and therefore probably improves the match between the transformed data and the target data. However, it should be kept in mind that increased overlap between the (transformed) source data distribution and that of the target data does not necessarily imply improved model performance, since the degree of class confusability may also be increased.

The augmentation results of Table 7.1 show no reason to use the approach, but in the next section we put models trained on the augmented data to good use.

7.8 Augmentation followed by adaptation

Models trained on augmented data (source data transformed using single regression class transformations together with the target data) are used for subsequent MAP adaptation on target language data. Figure 7.14 shows word accuracy achieved on the Afrikaans test set as a function of the prior weight (ϖ) for MAP adaptation on the Afrikaans training set and training subset. Peak accuracy of 71.8% is achieved for MAP adaptation of models trained on augmented data and adapted on the Afrikaans training set, compared to 69.0% word accuracy (see Figure 7.4) achieved by adapting models trained on pooled data. For Afrikaans training subset adaptation, peak word accuracy of 61.8% is achieved when priors trained on augmented data are used, compared to 56.0% word accuracy for priors trained on pooled data. The results clearly show the benefit of using augmented data priors versus using priors trained simply on pooled data.

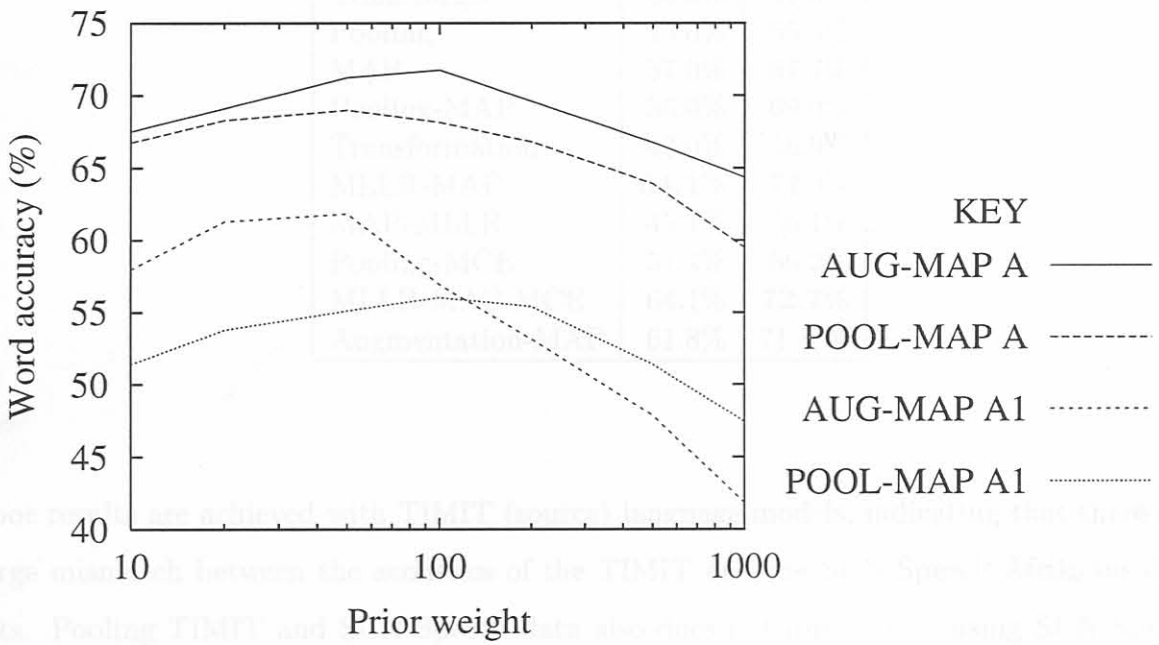

Figure 7.14: Comparison of the word accuracy on the Afrikaans test set as a function of the adaptation rate for models trained on transformed TIMIT data in addition to Afrikaans data (AUG), or on pooled TIMIT and Afrikaans data, when the Afrikaans training set (A) or training subset (A1) is used

7.9 Discussion of results

The experiments in this chapter covered application of the major categories of speaker adaptation techniques, as well as extensions and combinations of them, to cross-language and cross-database adaptation of acoustic parameters, combining data and models associated with both the TIMIT and SUN Speech databases. A number of approaches are shown to deliver useful cross-language adaptation performance. Table 7.2 summarises the methods that were experimented with and their results, which are briefly discussed next.

Table 7.2: Summary of peak word accuracy achieved on the Afrikaans test set in various experiments that evaluate different approaches to cross-language adaptation on the Afrikaans training set (A) and Afrikaans training subset (A1)

Method	Adapted on	
	A1	A
Train source	-5.9%	-5.9%
Train target	45.0%	67.6%
Pooling	45.0%	55.3%
MAP	57.0%	67.7%
Pooling-MAP	56.0%	69.0%
Transformation	43.0%	56.9%
MLLR-MAP	**64.1%**	72.0%
MAP-MLLR	45.1%	56.1%
Pooling-MCE	51.4%	66.2%
MLLR-MAP-MCE	64.1%	**72.7%**
Augmentation-MAP	61.8%	71.8 %

Poor results are achieved with TIMIT (source) language models, indicating that there is a large mismatch between the acoustics of the TIMIT and the SUN Speech Afrikaans data sets. Pooling TIMIT and SUN Speech data also does not improve on using SUN Speech data alone, due to the large mismatch between the databases. This is in contrast with results in the previous chapter (Section 6.4) where simple pooling of SUN Speech English and Afrikaans data improved on using only the Afrikaans data.

Cross-language MAP adaptation delivers reasonably good results, showing large improve-

ment over using the Afrikaans training subset only (57.0% versus 45.0%), but showing little improvement (0.1%) over using the Afrikaans training set only. Using multilingual priors for MAP adaptation (pooling-MAP) improves performance for Afrikaans training set adaptation, but degrades performance for the Afrikaans training subset.

Transformation-based adaptation (MLLR mean and log-variance MSE transformation) does not deliver useful performance by itself, achieving poorer performance than is achieved with direct training on target data. However, use of MLLR transformed models to seed prior distributions for subsequent MAP adaptation delivers the best overall results (except for subsequent MCE adaptation) achieving 64.1% word accuracy using the Afrikaans training subset and 72.0% when using the full Afrikaans training set. The results on the Afrikaans training subset are the best achieved in conjunction with the TIMIT database and represent an improvement of 19.1% (absolute) in word accuracy, or a 35% relative reduction of the word error rate, over using the Afrikaans training subset in isolation. The MLLR-MAP approach works well for relatively simple MLLR transformations (one or two regression classes), since simple transformations remove consistent bias between the source models and target data without over-specialising the models, thereby improving estimation of the prior distributions. Subsequent MAP adaptation efficiently uses the relatively large amounts of adaptation data to deliver robust parameter estimates. MAP-MLLR adaptation improves on the sensitivity of MLLR performance with respect to the number of regression classes (see Figure 7.9), but does not deliver consistently better peak performance than MLLR adaptation.

MCE adaptation of models trained on pooled multilingual data improves performance of the models, but performance for Afrikaans training set adaptation is still below that achieved with direct training on the training set. Performing MCE adaptation on models previously adapted with MLLR-MAP adaptation on the Afrikaans training set delivers the best performance of 72.7% on the Afrikaans training set. This represents an improvement of 5.1% in word accuracy, or a 16% relative reduction of the word error rate, over using the Afrikaans training set in isolation. MCE-based adaptation of best-performing Afrikaans training subset MLLR-MAP adapted models, however, does not deliver any further improvement in

performance.

The last approach for cross-language use of acoustic information that was investigated is the data augmentation approach. Models trained on target data augmented with transformed source data did not deliver improved performance. The approach is not without merit though, since models trained on the augmented data provide good prior models for MAP adaptation, achieving 61.8% word accuracy for Afrikaans training subset adaptation and 71.8% word accuracy for Afrikaans training set adaptation, which is better than that achieved with either cross-language MAP or pooling-MAP approaches.

Overall, use of the TIMIT database in addition to the Afrikaans data from SUN Speech delivers a significant improvement in performance, achieving peak improvement of between 16% and 35% reduction in relative word error rate, depending on the amount of target language data available.

Chapter 8

Conclusion

In this thesis several general strategies for cross-language use of acoustic information were proposed. The strategies incorporate specific techniques from the field of speaker adaptation and attempt to use relatively large amounts of source language data to improve the performance of recognisers for a new target language in which a limited amount of speech data is available. Extensions to current speaker adaptation techniques were also presented, with the particular aim of improving the performance of these techniques for cross-language adaptation. Experimental results vindicate the new approaches for cross-language use of acoustic information, also indicating improved performance for the proposed extensions to speaker adaptation techniques when applied to cross-language adaptation.

A fundamental question answered by the research performed in this thesis is whether cross-language use of acoustic information can be useful for the purpose of improving target language specific recognition. Previous research on this subject has been inconclusive, with studies of multilingual speech recognition [48, 19, 30] generally reporting that sharing of acoustic information degrades performance, and a few studies [20, 21] reporting marginal improvements by either multilingual pooling or mean-only MAP adaptation. The results in Chapters 6 and 7 of this thesis show significant reductions in word error rate (between 16% and 50%) for continuous speech recognition in a target language by use of speech data from

a source language, showing that cross-language use of acoustic information can significantly improve target language specific recognition.

Once it is ascertained that cross-language use of acoustic information may be useful, the question arises as to the approach that should be followed to deliver good results. The strategies presented in this thesis, allied with specific adaptation techniques, create a framework for research into cross-language acoustic adaptation. The strategies that are presented include:

- cross-language adaptation of hidden Markov acoustic models,
- multilingual speech data pooling followed by acoustic model adaptation, and
- cross-language speech data augmentation, followed by acoustic model adaptation.

All three general strategies were shown to deliver useful results, with the newly-proposed *multilingual pooling-adaptation* strategy (Section 5.2.4) being especially suited when source and target data are closely matched i.t.o. recording conditions and labelling conventions, while the newly-proposed *cross-language augmentation-adaptation* strategy (Sections 5.2.5-5.2.6) provides improved performance when bias exists between source and target databases, such as exist between the TIMIT and SUN Speech databases.

These general strategies, incorporating adaptation techniques, form different approaches for cross-language acoustic use of information. The most prominent methods from the major categories of HMM adaptation techniques were used, including

- Bayesian techniques, in particular maximum *a posteriori* (MAP) estimation (Section 3.2),
- transformation-based adaptation techniques, in particular maximum likelihood linear regression (MLLR) transformation (Section 3.3), and
- discriminative learning techniques, in particular minimum classification error (MCE) adaptation (Chapter 4).

Within the cross-language adaptation framework, a large number of experiments were performed, using two speech databases namely the SUN Speech and TIMIT databases, to evaluate the relative performance of different approaches. Interestingly, all three adaptation techniques were found to contribute to achieving the best performance achievable for particular cross-language adaptation tasks. For experimental purposes, English was considered a source language, with relatively large amounts of English speech data available in the TIMIT and SUN Speech databases. Afrikaans was considered the target language, with a relatively smaller amount of data available in the SUN Speech database. For the relatively closely matched English and Afrikaans data from the SUN Speech database, training on pooled models, followed by either MAP, MCE, or a combination of MAP and MCE adaptation delivered the best results. When the English SUN Speech data set was used in conjunction with a 25 times smaller Afrikaans data set, a pooling-MAP-MCE approach (Sections 6.5.3 and 6.8.2) achieved a 50% reduction in word error rate over using only the Afrikaans data. When the English speech data was used in conjunction with a 5 times smaller Afrikaans data set, a pooling-MCE approach (Section 6.8.1) achieved a 26% reduction in word error rate over using only the Afrikaans data. For experiments with the TIMIT and SUN Speech databases, cross-language MLLR-MAP transformation delivered the best results, with further MCE adaptation achieving additional improvement. When the TIMIT database was used in conjunction with a 45 times smaller Afrikaans data set, an MLLR-MAP approach (Section 7.5.1) achieved a 35% reduction in word error rate over using only the Afrikaans data. When the TIMIT database was used in conjunction with a 9 times smaller Afrikaans data set, an MLLR-MAP-MCE approach (Sections 7.5.1 and 7.6.2) achieved a 16% reduction in word error rate over using only the Afrikaans data.

Extensions to current speaker adaptation techniques were proposed, in an effort to improve the performance of these techniques for cross-language acoustic adaptation and are briefly reviewed next. MSE Bayesian estimation equations were derived (Sections 3.2.2-3.2.3), and both the MAP and MSE prior parameter estimation equations were specified in terms of seed model parameters (Section 3.2.5). The choice of prior parameter initialisation was found to materially influence recognition results, but the choice between MAP and MSE

estimation was found to provide little difference in performance (Section 6.5.5).

A technique was proposed to perform full (i.e. considering parameter dependencies) MSE transformation of the diagonal Gaussian variance parameters, in addition to using MLLR for the transformation of Gaussian mean parameters (Section 3.3.2). The variance transformation is implemented in log-space to ensure that parameter constraints are maintained, as well as to ensure optimisation of the relative variance error. Use of the proposed variance transformation technique was found to outperform MLLR transformation for cross-language adaptation, achieving between 8% and 18% reduction in word error rate for same-database experiments (Section 6.6.1) and between 4% and 14% reduction in word error rate for cross-database experiments (Section 7.4) over using mean-only MLLR transformation. An extension of MAP-MLLR for variance adaptation was proposed, incorporating a MAP-like term into the variance transformation (Section 3.4.2). Experiments showed MAP-MLLR to improve sensitivity of the transformation with respect to the number of regression classes, but to not significantly improve peak performance (Sections 6.7.2 and 7.5.2).

The application of MCE adaptation for cross-language adaptation was improved, firstly by extending the MCE framework to include adaptation of all model parameters, including duration modelling parameters (Section 4.3), and secondly by defining a method for including the cost of phoneme errors into the misclassification measure (Section 4.5.4). Methods to estimate useful cost matrices were proposed (Section 4.5.3) and cost-based MCE adaptation was shown to outperform standard MCE on adaptation of multilingual prior models, delivering useful cross-language adaptation performance (Sections 6.8.1 and 7.6.1).

In conclusion, several strategies, including extensions of current speaker adaptation and discriminative learning techniques, were presented, providing a framework for cross-language use of acoustic information. Approaches from the framework were found to deliver good performance under a reasonable variety of conditions, exceeding the performance achieved with previously published approaches to cross-language adaptation and showing significant gain from cross-language use of acoustic information.

8.1 Future research

The research performed in this thesis evaluated a wide scope of adaptation techniques, but on a fairly limited set of languages and databases. Research should be performed to study in particular

- use of more data sets to study application of the concepts to other databases and languages than those used in this study,
- use of mappings from more than one source language to a target language as this may allow improved prior model estimation,
- cross-language adaptation of context dependent phoneme models for target language LVCSR system development,
- use of target data to estimate the optimal complexity for each phoneme model, followed by training of models with the specified complexity on either source data or on pooled multilingual data,
- automatic phoneme mapping procedures - iterative use of a distance metric-based source-target phoneme mapping in conjunction with estimation of a source-target acoustic transformation may provide improved automatic phoneme mapping.

Use of the cross-language techniques proposed here, for accent adaptation, or even for speaker adaptation may provide interesting results. Another, more general research topic that follows from discussions in this thesis is the application of the integral over HMM parameter space (Equations 3.1, 3.2) to perform Bayesian estimation. Research into optimisation of MCE training, as well as improving the generalisation achieved with MCE training, perhaps using bootstrapping or Bayesian techniques, also warrants further research. Research regarding the use of MCE for adaptation, rather than training, may also provide interesting results.

Appendix A

SUN Speech database

A.1 Description

The SUN Speech database [12] was compiled by the Department of Electrical and Electronic Engineering of the University of Stellenbosch containing phonetically labelled speech in both Afrikaans and English. Speech data was recorded under controlled circumstances with 12 bit resolution and a 16kHz sampling rate. Details of the number of speakers and the number of sentences spoken by each group of speakers are given in Table A.1. The 60 sentences comprising the four sentence sets were chosen to exhibit the diversity of phonemes in the two languages.

Table A.1: Description of SUN Speech database: number of male and female speakers and total number of speakers for each sentence set

Language	Sentence set	Number of speakers			Number of sentences
		Male	Female	Total	
Afrikaans	1	24	16	40	10
	2	18	12	30	10
English	3	33	17	50	20
	4	22	4	26	20

A.2 Subdivision into training and test sets

The database is not entirely consistent in that some speakers, but not all, spoke sentences from more than one sentence set. For experimental purposes it is desired to obtain as much invariance as possible between the training set and the test set. A speaker independent, sentence independent division of the Afrikaans data can be obtained by using data from the first sentence set for training or adaptation and data from the second sentence set for testing. If data from the same speaker is available for both sentence sets, then data from only one of the sentence sets are used for either training or testing. A subset of the Afrikaans training set is also defined, containing the first 10 utterances from a group of 8 speakers who spoke all 20 Afrikaans sentences, with the second 10 sentences of these speakers making up the speaker dependent test set. The speaker dependent test set is not used in experiments reported in this thesis, except to create a pronunciation dictionary, but has been used previously by the authors for cross-language speaker and multispeaker adaptation experiments [33]. Details of the composition of the various subdivisions of the database given in Table A.2.

Table A.2: Subdivision of SUN Speech database into an Afrikaans training set, training subset, speaker dependent test set and speaker independent test set, as well as an English set

Language	Set	Speakers			Sentence numbers	Label count	Duration (seconds)
		Male	Female	Total			
Afrikaans	train	23	16	39	1-10	17251	1555
	train subset	2	6	8	1-10	3466	316
	SD test	2	6	8	11-20	5128	441
	SI test	14	1	15	11-20	9413	745
English		55	21	76	21-60	93778	7757

A.3 Phonetic content and labelling

A total of 59 phonetic categories, including both a *silence* and *unknown* category, were used to segment both the Afrikaans and the English speech. It was attempted to assign the labels phonetically, i.e. according to the sound produced, rather than phonologically assigning the labels, i.e. according to what was supposed to be said. The complete list of symbols along with the database numerical representation, computer phonetic representation, examples and frequency of occurrence is given in Tables A.3 and A.4. The frequency of occurrence is useful to evaluate the phonetic composition and diversity of the database, as well as the match between the two languages and also the match between the Afrikaans training and test sets.

It is evident from Tables A.3 and A.4 that many of the phonemes with no given English examples occur relatively frequently as labels in the English speech. Even though these phonemes do not possess accurately representative examples in English, they may occur due to particular pronunciations of certain words. This happens especially when rounding of the front vowels in English occurs, such as when [e] is rounded to form [ø] and [i] is rounded to form [y], which are not usually associated with English speech. The significant exceptions (more than 0.1% of all occurrences) and the words most commonly containing these labels are listed next, in order of decreasing occurrence.

y : educational, to, reputation, dreary

: dead, dreary, various, yesterday

ø : guilty, beautiful, annual, continues

œ : to, of, will, a

œy : educational, motivate, reputation, observation

As far as the Afrikaans speech is concerned, there are also a few of the phonetic labels

Table A.3: Phonetic classes, labels, SUN Speech numbering and computer phonetic labels with English and Afrikaans examples, as well as the percentage occurrence relative to all labels of the label in the English data (F_E) and the Afrikaans training set(F_{Atrain}) and Afrikaans testing set(F_{Atest})

Category	Symbol	Numeric	Code	English	Afrikaans	F_E	F_{Atrain}	F_{Atest}
Vowels	a	97	a	dug	kat	2.84%	3.74%	3.53%
	e	101	e	fear	lees	1.19%	1.93%	2.22%
	i	105	i	meet	tier	4.59%	5.25%	5.05%
	o	111	o	poor	oop	0.27%	1.70%	1.16%
	u	117	u	boot	soek	0.57%	1.72%	2.53%
	y	121	y		nuut	0.85%	0.79%	0.60%
	:	130	eh:		sê	0.31%	0%	0%
		131	eh	met	met	3.24%	1.69%	2.05%
		132	ao	paw	kos	1.21%	3.06%	2.51%
	:	133	ao:	bore	môre	0.64%	0%	0%
	ø	142	iax		kleur	0.60%	0.66%	0.74%
		143	ax	ago	is	11.18%	9.70%	13.30%
	:	144	ax:	flower	wîe	0.63%	0.01%	0%
	æ	145	ae	bat	ek	2.28%	0.30%	1.02%
	œ	149	oe		nut	2.57%	0.81%	1.17%
	œ:	150	oe:	fur	brûe	0.97%	0%	0%
	:	247	aa	bar	aan	1.51%	2.87%	2.49%
Diphthongs	:i	126	a:i	bite	saai	1.26%	1.12%	0.34%
	o:i	128	o:i		mooi	0.01%	0.36%	0.32%
	oi	134	oi	boy		0.39%	0.08%	0.50%
	i	140	ehi		bedjie	0.05%	0%	0.16%
	i	151	axi	fate	ys	1.47%	1.50%	1.46%
	ui	153	ui		moeite	0.00%	0.65%	0.46%
	iu:	210	iu:	due	leeu	0.05%	0.49%	0.30%
	œu	211	oeu	goat	oud	0.60%	0.46%	0.45%
	œy	217	oey		lui	0.10%	0.86%	0.84%
	õ:	245	aw	brow		0.32%	0%	0%
Nasals	m	109	m	mat	mat	2.77%	2.92%	2.11%
	n	110	n	net	net	7.82%	6.56%	5.69%
		205	ng	sing	sing	1.37%	1.23%	1.56%

Table A.4: Phonetic classes, symbols, Sunspeech labels and computer phonetic labels with English and Afrikaans examples, as well as the percentage occurrence relative to all labels of the label in the English data (F_E) and the Afrikaans training set(F_{Atrain}) and Afrikaans testing set(F_{Atest})

Category	Symbol	Numeric	Code	English	Afrikaans	F_E	F_{Atrain}	F_{Atest}
Fricatives	f	102	f	fat	vars	2.18%	3.39%	2.91%
	h	104	h	hat	huis	0.80%	0.71%	0.11%
	s	115	s	sit	slim	5.67%	6.10%	6.25%
	v	118	v	van	was	1.90%	1.75%	1.72%
	x	120	x		gaan	0.01%	2.65%	2.59%
	z	122	z	zip	soem	1.64%	0.67%	0.53%
	θ	171	th	thin		0.53%	0%	0%
		172	dh	then		1.53%	0%	0%
		188	sh	ship	Sjina	1.48%	0.44%	0.30%
		195	zh	vision	genre	0.36%	0%	0.14%
Affricates	ts^h	181	ts	cats		0.46%	0.06%	0.32%
	dz	184	dz	cads		0.12%	0%	0%
	t^h	191	ch	chin		1.31%	0.68%	0.68%
	d	193	jh	jam		0.82%	0%	0.01%
Liquids	r	114	r	rat		2.95%	4.33%	2.88%
	R	82	r		rooi	0%	4.15%	2.54%
		94	r		berge	0%	0%	0.26%
	l	108	l	lot	lou	2.97%	3.50%	2.97%
		218	/	refers to a flap		0.53%	0.33%	0.58%
Glides	j	106	j	yet	jas	0.47%	1.15%	1.03%
	w	119	w	win	kwes	1.58%	0.10%	0.89%
Stops	b	98	b	bat	bed	1.50%	1.54%	2.93%
	d	100	d	dog	dam	2.36%	3.95%	3.51%
	g	103	g	go	berge	0.66%	0.47%	1.02%
	k	107	k	kit	kar	3.21%	2.79%	3.66%
	p	112	p	pet	pos	2.04%	0.95%	2.47%
	t	116	t	tip	taal	5.85%	4.63%	5.49%
Other		42	sil	silence		3.42%	3.72%	2.12%
	?	63	?	Unknown		1.97%	0.47%	0.42%

that do not have representative examples, but yet occur in the database. The significant examples are listed next, in order of decreasing occurrence.

oi : t**oyi**t**oyi**, b**oi**kotters, m**ooi**, r**ooi**

ts^h : **ts**e**ts**evlieg, maa**ts**kappy, **S**uid-Afrika, **tj**inkering**tj**ees

t^h : **tj**inkering**tj**ees, **Ch**arles, Gorba**tsj**of, **tj**elloversameling

The set of phonemes used to label the SUN Speech Database represent the union of the phonemes found in Afrikaans and English. The labels were assigned phonetically and there should be close correspondence between data in the two languages with the same phonetic labels. Because such an expanded set of phonemes were used for labelling, many of the phonemes that do not usually appear in English phonetic transcriptions do appear as phone labels in the English transcriptions of this database. This has the advantage that for almost all the phonemes found in Afrikaans there are labelled examples in English, with the exception of the phonemes [R] and [] that do not appear in the English part of the database. The [r], [R] and [] categories are combined into a single [r] category since the [R] and [] categories are not well represented in either of the sets used for training, and because the distinction between the three phones is not important for word recognition purposes. For a few of the phonemes the number of examples that appear in the English transcriptions are very little. They are the phonemes(with the number of occurrences in brackets) : [o:i] (11), [ui] (2) and [x] (8).

Appendix B

TIMIT - SUN Speech phonetic mapping

This appendix (Appendix B) presents almost verbatim work that was performed by Dr. Hendrik Boshoff in his capacity as phonetic expert and is included in this thesis for completeness and because it has not been published elsewhere.

A mapping from SUN Speech symbols to those of TIMIT was required, in order to do cross-database training of phonetic models.

Mapping from one set of phonetic symbols to another is fraught with difficulty, especially when more than one language is involved. Vowels are especially problematic, as dynamic features contribute to subtle differences. In the present case, SUN Speech already contains English and Afrikaans speech, but some problems remain.

A few significant differences in approach between the databases must be mentioned:

- TIMIT views stops as potentially two segments, closure and release. An intervocalic stop of [t] for example, is always transcribed as 'TCL T.' In other positions, the transcription depends upon the actual realization. This allows the affricate [ts] to

be rendered as 'TCL S.' SUN Speech segments all phases of the stop together, and provides separate symbols for all affricates.

- SUN Speech makes provision for front rounded vowels, and when judged appropriate, English vowels are also transcribed using these symbols. This is a somewhat more 'phonetic' approach, versus that of TIMIT, which is more 'phonemic' with respect to vowels.
- TIMIT groups all vocalic sounds together, and does not indicate diphthongization. SUN Speech has an extensive set of diphthongs, and also labels quantity to some extent.
- TIMIT explicitly indicates beginning and end of speech, and sometimes primary and secondary stress. Both these types of transcription are absent in SUN Speech.

It was assumed that every symbol of SUN Speech had to be mapped to one of TIMIT and vice versa. In some cases this was highly artificial, and a 'matching quality' figure was introduced. This ranges from 1 to 3, with the following meanings. 1: The phonemes indicated by the symbols match closely, and some allophones are likely to be identical across the databases. 2: The phonemes are not identical, but are 'neighbours' in phonetic space. 3: The match is poor, but some features are similar, eg place or manner of articulation.

Following are two tables according to the SUN Speech organisation, with the preferred equivalents from TIMIT.

Table B.1: Mapping from SUN Speech to TIMIT symbols (vocoids)

SUN Speech						Match	TIMIT	
Category	Sym	Num	Code	Eng	Afr	quality	code	word
Vowels	a	97	a	dug	kat	1	ah	but
						2	ax-h	suspect
	e	101	e	fear	lees	3	ey	bait
	i	105	i	meet	tier	1	iy	beet
	o	111	o	poor	oop	2	ow	boat
	u	117	u	boot	soek	1	uw	boot
						2	ux	toot
						2	uh	book
	y	121	y		nuut	3	ux	toot
						3	uw	boot
	:	130	eh:		sê	2	eh	bet
		131	eh	met	met	1	eh	bet
		132	ao	paw	kos	1	ao	bought
	:	133	ao:	bore	môre	2	ao	bought
	ø	142	iax		kleur	3	ey	bait
						3	uw	boot
		143	ax	ago	is	1	ax	about
						1	ix	debit
						1	ih	bit
	:	144	ax:	flower	wîe	3	axr	butter
						3	er	bird
	æ	145	ae	bat	ek	1	ae	bat
	œ	149	oe		nut	2	ih	bit
						2	ix	debit
	œ:	150	oe:	fur	brûe	3	er	bird
	:	247	aa	bar	aan	2	aa	bott
Diphthongs	:i	126	a:i	bite	saai	1	ay	bite
	o:i	128	o:i		mooi	3	oy	boy
	oi	134	oi	boy		1	oy	boy
	i	140	ehi		bedjie	3	eh	bet
						3	ey	bait
	i	151	axi	fate	ys	1	ey	bait
	ui	153	ui		moeite	3	ey	bait
						3	ux	toot
	iu:	210	iu:	due	leeu	3	ow	boat
						3	ey	bait
	œu	211	oeu	goat	oud	1	ow	boat
	œy	217	oey		lui	3	ey	bait
						3	oo	toot
	õ:	245	aw	brow		1	aw	bout

Table B.2: Mapping from SUN Speech to TIMIT symbols (contoids)

SUN Speech						Match	TIMIT	
Category	Sym	Num	Code	Eng	Afr	quality	code	word
Nasals	m	109	m	**m**at	**m**at	1	m	**m**om
						1	em	bott**om**
	n	110	n	**n**et	**n**et	1	n	**n**oon
						1	en	butt**on**
		205	ng	sing	sing	1	ng	sing
						1	eng	Washington
Fricatives	f	102	f	**f**at	**v**ars	1	f	**f**in
	h	104	h	**h**at	**h**uis	1	hh	**h**ay
						1	hv	a**h**ead
	s	115	s	**s**it	sli**m**	1	s	**s**ea
	v	118	v	**v**an	**w**as	1	v	**v**an
	x	120	x		ga**a**n	3	hh	**h**ay
						3	k	**k**ey
	z	122	z	**z**ip	**s**oem	1	z	**z**one
	θ	171	th	**th**in		1	th	**th**in
		172	dh	**th**en		1	dh	**th**en
		188	sh	**sh**ip	**Sj**ina	1	sh	**sh**e
		195	zh	vi**s**ion	ge**n**re	1	zh	a**z**ure
Affricates	ts^h	181	ts	**ts**ats		3	t	**t**ea
						3	s	**s**ea
	dz	184	dz	**ds**ads		3	d	**d**ay
						3	z	**z**one
	t^h	191	ch	**ch**in		1	ch	**ch**oke
	d	193	jh	**j**am		1	jh	**j**oke
Glides	j	106	j	**y**et	**j**as	1	y	**y**acht
	w	119	w	**w**in	**kw**es	1	w	**w**ay
Liquids	r	114	r	**r**at		1	r	**r**ay
	R	82	r		**r**ooi	2	r	**r**ay
		94	r		ber**g**e	2	r	**r**ay
	l	108	l	**l**ot	**l**ou	1	l	**l**ay
						1	el	bott**le**
		218	/	(flap)		1	dx	mu**dd**y
						2	nx	wi**nn**er
Stops	b	98	b	**b**at	**b**ed	1	b	**b**ee
	d	100	d	**d**og	**d**am	1	d	**d**ay
	g	103	g	**g**o	ber**g**e	1	g	**g**ay
	k	107	k	**k**it	**k**ar	1	k	**k**ite
						3	q	ba**t**
	p	112	p	**p**et	**p**os	1	p	**p**ea
	t	116	t	**t**ip	**t**aal	1	t	**t**ea

Appendix C

MCE update derivations

C.1 Mixture weight derivative

$$\frac{\partial}{\partial \bar{c}_{jk}^{(i)}} \log b_j^{(i)}(\mathbf{x}_t) = (b_j^{(i)}(\mathbf{x}_t))^{-1} \frac{\partial b_j^{(i)}(\mathbf{x}_t)}{\partial \bar{c}_{jk}^{(i)}} \tag{C.1}$$

$$= (b_j^{(i)}(\mathbf{x}_t))^{-1} \sum_{k'}^{M} \frac{\partial b_j^{(i)}(\mathbf{x}_t)}{\partial c_{jk'}^{(i)}} \frac{\partial c_{jk'}^{(i)}}{\partial \bar{c}_{jk}^{(i)}} \tag{C.2}$$

$$= \sum_{k'}^{M} \frac{\mathcal{N}[\mathbf{x}_t, \boldsymbol{\mu}_{jk'}^{(i)} \Sigma_{jk'}^{(i)}]}{b_j^{(i)}(\mathbf{x}_t)} \frac{\partial c_{jk'}^{(i)}}{\partial \bar{c}_{jk}^{(i)}} \tag{C.3}$$

where

$$\frac{\partial c_{jk'}^{(i)}}{\partial \bar{c}_{jk}^{(i)}} = \frac{\partial}{\partial \bar{c}_{jk}^{(i)}} \left[\frac{e^{\bar{c}_{jk'}^{(i)}}}{\sum_l e^{\bar{c}_{jl}^{(i)}}} \right] \tag{C.4}$$

$$= \delta(k = k') \frac{\partial}{\partial \bar{c}_{jk}^{(i)}} \left[\frac{e^{\bar{c}_{jk}^{(i)}}}{\sum_l e^{\bar{c}_{jl}^{(i)}}} \right] + \delta(k \neq k') \frac{\partial}{\partial \bar{c}_{jk}^{(i)}} \left[\frac{e^{\bar{c}_{jk'}^{(i)}}}{\sum_l e^{\bar{c}_{jl}^{(i)}}} \right] \tag{C.5}$$

$$= \delta(k = k') \frac{e^{\bar{c}_{jk}^{(i)}} \sum_l e^{\bar{c}_{jl}^{(i)}} - e^{\bar{c}_{jk}^{(i)}} e^{\bar{c}_{jk}^{(i)}}}{(\sum_l e^{\bar{c}_{jl}^{(i)}})^2} + \delta(k \neq k') \frac{-e^{\bar{c}_{jk'}^{(i)}} e^{\bar{c}_{jk}^{(i)}}}{(\sum_l e^{\bar{c}_{jl}^{(i)}})^2} \tag{C.6}$$

$$= \delta(k = k') \frac{e^{\bar{c}_{jk}^{(i)}}}{\sum_l e^{\bar{c}_{jl}^{(i)}}} \frac{\sum_{l \neq k} e^{\bar{c}_{jl}^{(i)}}}{\sum_l e^{\bar{c}_{jl}^{(i)}}} + \delta(k \neq k') c_{jk'} c_{jk} \tag{C.7}$$

$$= \delta(k = k') c_{jk}(1 - c_{jk}) + \delta(k \neq k') c_{jk'} c_{jk} \tag{C.8}$$

$$= c_{jk}^{(i)} \delta(k' - k) - c_{jk}^{(i)} c_{jk'}^{(i)}. \tag{C.9}$$

and therefore

$$\frac{\partial}{\partial \bar{c}_{jk}^{(i)}} \log b_j^{(i)}(\mathbf{x}_t) = \sum_{k'}^{M} \frac{\mathcal{N}[\mathbf{x}_t, \boldsymbol{\mu}_{jk'}^{(i)}, \Sigma_{jk'}^{(i)}]}{b_j^{(i)}(\mathbf{x}_t)} \left[c_{jk}^{(i)} \delta(k' - k) - c_{jk}^{(i)} c_{jk'}^{(i)} \right] \tag{C.10}$$

$$= c_{jk}^{(i)} \sum_{k'}^{M} \frac{\mathcal{N}[\mathbf{x}_t, \boldsymbol{\mu}_{jk'}^{(i)}, \Sigma_{jk'}^{(i)}]}{b_j^{(i)}(\mathbf{x}_t)} \left[\delta(k' - k) - c_{jk'}^{(i)} \right] \tag{C.11}$$

$$= \frac{c_{jk}^{(i)}}{b_j^{(i)}(\mathbf{x}_t)} \left[\mathcal{N}[\mathbf{x}_t, \boldsymbol{\mu}_{jk}^{(i)}, \Sigma_{jk}^{(i)}] - \sum_{k'}^{M} c_{jk'}^{(i)} \mathcal{N}[\mathbf{x}_t, \boldsymbol{\mu}_{jk'}^{(i)}, \Sigma_{jk'}^{(i)}] \right] \tag{C.12}$$

$$= c_{jk}^{(i)} \left[\frac{\mathcal{N}[\mathbf{x}_t, \boldsymbol{\mu}_{jk}^{(i)}, \Sigma_{jk}^{(i)}]}{b_j^{(i)}(\mathbf{x}_t)} - 1 \right]. \tag{C.13}$$

C.2 Transition probability derivative

$$\frac{\partial}{\partial \bar{a}_{jj'}^{(i)}} g_i(X; \Lambda) = \sum_{t=1}^{T(X)} \sum_{s=1}^{N} \delta(\bar{q}_{t-1} - j) \delta(\bar{q}_t - s) \frac{\partial}{\partial \bar{a}_{jj'}^{(i)}} \log a_{js}^{(i)} \tag{C.14}$$

$$= \sum_{t=1}^{T(X)} \sum_{s=1}^{N} \delta(\bar{q}_{t-1} - j) \delta(\bar{q}_t - s) \frac{1}{a_{js}^{(i)}} \frac{\partial a_{js}^{(i)}}{\partial \bar{a}_{jj'}^{(i)}} \tag{C.15}$$

where the derivative of $a_{js}^{(i)}$ with respect to $\bar{a}_{jj'}^{(i)}$ is similar to the derivation in Equations C.4-C.9, giving

$$\frac{\partial a_{js}^{(i)}}{\partial \bar{a}_{jj'}^{(i)}} = a_{jj'}^{(i)} \delta(j' - s) - a_{jj'}^{(i)} a_{js}^{(i)} \tag{C.16}$$

and therefore

$$\frac{\partial}{\partial \bar{a}_{jj'}^{(i)}} \log g_i(X;\Lambda) = \sum_{t=1}^{T(X)} \sum_{s=1}^{N} \delta(\bar{q}_{t-1}-j)\delta(\bar{q}_t-s) \frac{a_{jj'}^{(i)}\delta(j'-s) - a_{jj'}^{(i)} a_{js}^{(i)}}{a_{js}^{(i)}} \tag{C.17}$$

$$= \sum_{t=1}^{T(X)} \sum_{s=1}^{N} \delta(\bar{q}_{t-1}-j)\delta(\bar{q}_t-s)[\delta(j'-s) - a_{jj'}^{(i)}]. \tag{C.18}$$

Bibliography

[1] L. Rabiner, "An introduction to hidden Markov models," *IEEE Signal Processing Magazine*, vol. 5, pp. 4–16, Jan. 1988.

[2] S. Young, "A review of large-vocabulary continuous-speech recognition," *IEEE Signal Processing Magazine*, vol. 13, pp. 45–57, Sep. 1996.

[3] L. Lamel, R. Kassel, and S. Seneff, "Speech database development: design and analysis of the acoustic-phonetic corpus," in *Proc. DARPA Speech Recognition Workshop*, Report No. SAIC-86/1546, pp. 100–109, Feb. 1986.

[4] J. Godfrey, E. Holliman, and J. McDaniel, "SWITCHBOARD: telephone speech corpus for research and development," in *Proc. ICASSP '92*, (San Francisco, CA), March 1992.

[5] D. Paul and J. Baker, "The design for the Wall Street Journal-based CSR corpus," in *Proc. DARPA Speech Natural Language Workshop*, pp. 357–362, Feb. 1992.

[6] A. Nakamura, S. Matsunaga, T. Shimizu, M. Tonomura, and Y. Sagisaka, "Japanese speech databases for robust speech recognition," in *Proc. ICSLP '96*, Vol. 4, (Philadelphia, PA), pp. 2199–2202, Oct. 1996.

[7] J.-L. Gauvain, L. F. Lamel, and M. Eskénazi, "Design considerations and text selection for BREF, a large French read-speech corpus," in *Proc. ICSLP '90*, (Kobe, Japan), pp. 1097–1100, Nov. 1990.

[8] D. Langmann, R. Haeb-Umbach, L. Boves, and E. den Os, "FRESCO: The French telephone speech data collection - part of the European SpeechDat(m) project," in *Proc. ICSLP '96*, Vol. 3, (Philadelphia, PA), pp. 1918–1921, Oct. 1996.

[9] K. J. Kohler, "Labelled data bank of spoken standard German: The Kiel corpus of read/spontaneous speech," in *Proc. ICSLP '96*, Vol. 3, (Philadelphia, PA), pp. 1938–1941, Oct. 1996.

[10] T. Bub and J. Schwinn, "VERBMOBIL: The evolution of a complex large speech-to-speech translation system," in *Proc. ICSLP '96*, Vol. 4, (Philadelphia, PA), pp. 2371–2374, Oct. 1996.

[11] H. Höge, C. Draxler, H. van den Heuvel, F. Johansen, E. Sanders, and H. Tropf, "SpeechDat multilingual speech databases for teleservices: Across the finish line," in *Proc. Eurospeech '99*, (Budapest, Hungary), pp. 2699–2702, Sep. 1999.

[12] T. Waardenburg, J. du Preez, and M. Coetzer, "The automatic recognition of stop consonants using hidden Markov models," in *Proc. ICASSP '92*, (San Francisco, CA), pp. 1585–1588, March 1992.

[13] F. de Wet and E. Botha, "Towards speech technology for South African languages: Automatic speech recognition in Xhosa," Accepted for publication in the *South African Journal of African Languages*, Mar. 1999.

[14] J. Glass, G. Flammia, D. Goodine, M. Phillips, J. Polifroni, S. Sakai, S. Seneff, and V. Zue, "Multilingual spoken-language understanding in the MIT Voyager system," *Speech Communication*, vol. 17, pp. 1–18, Aug. 1995.

[15] J. Billa, K. Ma, J. W. McDonough, G. Zavaliagkos, D. R. Miller, K. N. Ross, and A. El-Jaroudi, "Multilingual speech recognition: The 1996 Byblos callhome system," in *Proc. Eurospeech '97*, (Rhodes, Greece), pp. 363–366, Sep. 1997.

[16] B. Wheatley, K. Kondo, W. Anderson, , and Y. Muthusamy, "An evaluation of cross-language adaptation for rapid HMM development in a new language," in *Proc. ICASSP '94*, (Adelaide, Australia), pp. I–237 – I–240, Apr. 1994.

[17] T. Schultz and A. Waibel, "Fast bootstrapping of LVCSR systems with multilingual phoneme sets," in *Proc. Eurospeech '97*, (Rhodes, Greece), pp. 371–374, Sep. 1997.

[18] C. Wang, J. Glass, H. Meng, J. Polifroni, S. Seneff, and V. Zue, "Yinhe: A Mandarin Chinese version of the Galaxy system," in *Proc. Eurospeech '97*, (Rhodes, Greece), pp. 351–354, Sep. 1997.

[19] F. Weng, H. Bratt, L. Neumeyer, and A. Stolcke, "A study of multilingual speech recognition," in *Proc. Eurospeech '97*, (Rhodes, Greece), pp. 359–362, Sep. 1997.

[20] P. Bonaventura, F. Gallocchio, and G. Micca, "Multilingual speech recognition for flexible vocabularies," in *Proc. Eurospeech '97*, (Rhodes, Greece), pp. 355–358, Sep. 1997.

[21] J. Köhler, "Language adaptation of multilingual phone models for vocabulary independent speech recognition tasks," in *Proc. ICASSP '98*, (Seattle, USA), pp. 417 – 420, May 1998.

[22] U. Bub, J. Köhler, and B. Imperl, "In-service adaptation of multilingual hidden Markov models," in *Proc. ICASSP '97*, (Munich, Germany), pp. 1451 – 1454, Apr. 1997.

[23] R. M. Stern and M. J. Lasry, "Dynamic speaker adaptation for feature-based isolated word recognition," *IEEE Trans. Acoustics, Speech and Signal Processing*, vol. 35, pp. 751–763, June 1987.

[24] C.-H. Lee, C.-H. Lin, and B.-H. Juang, "A study on speaker adaptation of the parameters of continuous density hidden Markov models," *IEEE Trans. Signal Processing*, vol. 39, pp. 806–841, Apr. 1991.

[25] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis.* New York, NY: John Wiley & Sons, 1973.

[26] L. Rabiner and B. Juang, *Fundamentals of Speech Recognition.* Englewood Cliffs, NJ: Prentice-Hall, 1993.

[27] C. Leggetter and P. Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models," *Computer Speech and Language*, vol. 9, pp. 171–185, Apr. 1995.

[28] B.-H. Juang, W. Chou, and C.-H. Lee, "Minimum classification error rate methods for speech recognition," *IEEE Transactions on Speech and Audio Processing*, vol. 5, pp. 257–265, May 1997.

[29] T. Matsui and S. Furui, "A study of speaker adaptation based on minimum classification error training," in *Proc. Eurospeech '95*, (Madrid, Spain), pp. 81–84, Sep. 1995.

[30] T. Schultz and A. Waibel, "Language independent and language adaptive large vocabulary speech recognition," in *Proc. ICSLP '98*, Vol. 5, (Sydney, Australia), pp. 1819–1822, Nov. 1998.

[31] ARPA, "The DARPA TIMIT Acoustic-Phonetic Continuous Speech Corpus," NIST Speech Disc CD1-1.1, Dec. 1990.

[32] C. Nieuwoudt and E. Botha, "Cross-language use of acoustic information for automatic speech recognition," Submitted to *Speech Communication*, May 2000.

[33] C. Nieuwoudt and E. Botha, "Adaptation of acoustic models for multilingual recognition," in *Proc. Eurospeech '99*, (Budapest, Hungary), pp. 907–910, Sep. 1999.

[34] C. Nieuwoudt and E. Botha, "Multilingual training of acoustic models in automatic speech recognition," Accepted for publication in the *South African Computer Journal*, Oct. 1999.

[35] C. Nieuwoudt and E. Botha, "Connected digit recognition in Afrikaans using hidden Markov models," *South African Computer Journal*, pp. 85–91, Jul. 1999.

[36] L. R. Rabiner, J. G. Wilpon, and F. K. Soong, "High performance connected digit recognition using hidden Markov models," *IEEE Trans. Acoustics, Speech and Signal Processing*, vol. ASSP-37, no. ASSP-37, p. 1214, 1989.

[37] Y. Gotoh, M. M. Hochberg, and H. F. Silverman, "Using MAP estimated parameters to improve HMM speech recognition performance," in *Proc. ICASSP '94*, (Adelaide, Australia), pp. I–229 – I–232, Apr. 1994.

[38] X. Wang, *Incorporating Knowledge on Segmental Duration in HMM-Based Continuous Speech Recognition.* PhD thesis, University of Amsterdam, Amsterdam, The Netherlands, Apr. 1997.

[39] J. Du Preez, "Modelling durations in hidden Markov models with application to word spotting," in *Proc. IEEE South African symposium on Communications and Signal Processing*, (Fourways, South Africa), pp. 1–5, Aug. 1991.

[40] D. Burshtein, "Robust parametric modeling of durations in hidden Markov models," in *Proc. ICASSP '95*, (Detroit, MI), pp. 548 – 551, May 1995.

[41] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society*, vol. 39, no. 39, pp. 1–38, 1977.

[42] A. Viterbi, "Error bounds for convolutional codes and an asymptotically optimal decoding algorithm," *IEEE Trans. Information Theory*, vol. IT-13, pp. 260–269, Apr. 1967.

[43] F. K. Soong and E. Huang, "A tree-trellis based fast search for finding the N best sentence hypotheses in continuous speech recognition," in *Proc. ICASSP '91*, (Toronto, Canada), pp. 705–708, May 1991.

[44] M. Zissman, "Comparision of four approaches to automatic language identification of telephone speech," *IEEE Transactions on Speech and Audio Processing*, vol. 4, pp. 31–44, Jan. 1996.

[45] E. Barnard and Y. Yan, "Toward new language adaptation for language identification," *Speech Communication*, vol. 21, pp. 245–254, 1997.

[46] A. Lavie, A. Waibel, L. Levin, D. Gates, M. Gavalda, T. Zeppenfeld, P. Zhan, and O. Glickman, "Translation of conversational speech with JANUS-II," in *Proc. ICSLP '96*, Vol. 4, (Philadelphia, PA), pp. 2375–2378, Oct. 1996.

[47] T. Schultz, D. Koll, and A. Waibel, "Japanese LVCSR on the spontaneous scheduling task with JANUS-3," in *Proc. Eurospeech '97*, (Rhodes, Greece), pp. 367–370, Sep. 1997.

[48] J. Köhler, "Multi-lingual phoneme recognition exploiting acoustic-phonetic similarities of sounds," in *Proc. ICSLP '96*, Vol. 4, (Philadelphia, PA), pp. 2195–2198, Oct. 1996.

[49] U. Uebler, M. Schussler, and H. Niemann, "Bilingual and dialectal adaptation and retraining," in *Proc. ICSLP '98*, Vol. 5, (Sydney, Australia), pp. 1815–1818, Nov. 1998.

[50] X. Huang and K.-F. Lee, "On speaker-independent, speaker-dependent, and speaker-adaptive speech recognition," in *Proc. ICASSP '91*, (Toronto, Canada), pp. 877–880, May 1991.

[51] S. Furui, "Research on individuality features in speech waves and automatic speaker recognition techniques," *Speech Communication*, vol. 5, pp. 183–197, 1986.

[52] C. Leggetter, *Improved acoustic modelling for HMMs using linear transformations.* PhD thesis, Cambridge University, 1995.

[53] H. Wakita, "Normalisation of vowels by vocal tract length and its application to vowel identification," *IEEE Trans. Acoustics, Speech and Signal Processing*, vol. 25, no. 25, pp. 183–192, 1977.

[54] H. C. Choi and R. W. King, "On the use of spectral transformation for speaker adaptation in HMM based isolated-word speech recognition," *Speech Communication*, vol. 17, pp. 131–144, Aug. 1995.

[55] L. Uebel and P. Woodland, "An investigation into vocal tract length normalisation," in *Proc. Eurospeech '99*, (Budapest, Hungary), pp. 2527–2530, Sep. 1999.

[56] Y. Zhao, "An acoustic-phonetic-based speaker adaptation technique for improving speaker-independent continuous speech recognition," *IEEE Transactions on Speech and Audio Processing*, vol. 2, pp. 380–394, July 1994.

[57] M. H. DeGroot, *Probability and Statistics.* Reading, MA: Addison-Wesley, 1975.

[58] J.-L. Gauvain and C. Lee, "Bayesian learning for hidden Markov models with Gaussian mixture state observation densities," *Speech Communication*, vol. , pp. 205–213, June 1992.

[59] M. H. DeGroot, *Optimal Statistical Decisions.* New York, NY: McGraw-Hill, 1970.

[60] C.-H. Lee and J.-L. Gauvain, "Speaker adaptation based on MAP estimation of HMM parameters," in *Proc. ICASSP '93*, (Minneapolis, MN), pp. II–558–II–561, Apr. 1993.

[61] J.-L. Gauvain and C.-H. Lee, "Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains," *IEEE Transactions on Speech and Audio Processing*, vol. 2, pp. 291–298, Apr. 1994.

[62] L. Liporace, "Maximum likelihood estimation for multivariate observations of Markov sources," *IEEE Trans. Information Theory*, vol. IT-28, pp. 729–734, Sep. 1982.

[63] C. Leggetter and P. Woodland, "Speaker adaptation of HMMs using linear regression," Tech. Report CUED/F-INFENG/TR. 181, Cambridge University Engineering Department, Trumpington Street, Cambridge, CB2 1PZ, England, June 1994.

[64] H. Matsukoto and H. Inoue, "A piecewise linear spectral mapping for supervised speaker adaptation," in *Proc. ICASSP '92*, (San Francisco, CA), pp. 449–452, March 1992.

[65] A. Sankar, L. Neumeyer, and M. Weintraub, "An experimental study of acoustic adaptation algorithms," in *Proc. ICASSP '95*, (Detroit, MI), pp. 713 – 716, May 1995.

[66] V. Nagesha and L. Gillick, "Studies in transformation-based adaptation," in *Proc. ICASSP '97*, (Munich, Germany), pp. 1031 – 1034, Apr. 1997.

[67] M. Padmanabhan, L. R. Bahl, D. Nahamoo, and M. Picheny, "Speaker clustering and transformation for speaker adaptation in speech recognition systems," *IEEE Transactions on Speech and Audio Processing*, vol. 6, pp. 71–77, Jan. 1998.

[68] A. J. Hewett, *Training and speaker adaptation in template-based speech recognition.* PhD thesis, Cambridge University, 1989.

[69] S. Cox, "A speaker adaptation technique using linear regression," in *Proc. ICASSP '95*, (Detroit, MI), pp. 700 – 703, May 1995.

[70] C. Leggetter and P. Woodland, "Flexible speaker adaptation using maximum likelihood linear regression," in *Proc. ARPA Spoken Language Technology Workshop*, (Barton Creek), 1995.

[71] V. Digalakis, D. Rtischev, and L. Neumeyer, "Speaker adaptation using constrained estimation of Gaussian mixtures," *IEEE Transactions on Speech and Audio Processing*, vol. 3, pp. 357–366, Sep. 1995.

[72] M. Gales and P. Woodland, "Mean and variance adaptation with the MLLR framework," *Computer Speech and Language*, vol. 10, pp. 249–264, Oct. 1996.

[73] M. Gales, "Maximum likelihood linear transformations for HMM-based speech recognition," *Computer Speech and Language*, vol. 12, pp. 75–98, Apr. 1998.

[74] V. Abrash, A. Sankar, H. Franco, and M. Cohen, "Acoustic adaptation using nonlinear transformations of HMM parameters," in *Proc. ICASSP '96*, (Atlanta, GA), pp. 729 – 732, May 1996.

[75] J. R. Bellegarda, P. V. de Souza, A. J. Nádas, D. Nahamoo, M. A. Picheny, and L. R. Bahl, "Robust speaker adapation using a piecewise linear acoustic mapping," in *Proc. ICASSP '92*, (San Francisco, CA), pp. 445–448, March 1992.

[76] J. R. Bellegarda, P. V. de Souza, D. Nahamoo, M. Padmanabhan, M. A. Picheny, and L. R. Bahl, "Experiments using data augmentation for speaker adaptation," in *Proc. ICASSP '95*, (Detroit, MI), pp. 692 – 695, May 1995.

[77] J. Ishii and M. Tonomura, "Speaker normalization and adaptation based on linear transformation," in *Proc. ICASSP '97*, (Munich, Germany), pp. 1055 – 1058, Apr. 1997.

[78] V. Digalakis and L. Neumeyer, "Speaker adaptation using combined transformation and Bayesian methods," in *Proc. ICASSP '95*, (Detroit, MI), pp. 680–683, May 1995.

[79] V. Digalakis and L. Neumeyer, "Speaker adaptation using combined transformation and Bayesian methods," *IEEE Transactions on Speech and Audio Processing*, vol. 4, pp. 294–300, July 1996.

[80] E. Thelen, X. Aubert, and P. Beyerlein, "Speaker adaptation in the Philips system for large vocabulary continuous speech recognition," in *Proc. ICASSP '97*, (Munich, Germany), pp. 1035 – 1038, Apr. 1997.

[81] W. Chou, "Maximum a posterior linear regression with elliptically symmetric matrix variate priors," in *Proc. Eurospeech '99*, (Budapest, Hungary), pp. 1–4, Sep. 1999.

[82] D.E.Rumelhart, G.E.Hinton, and R.J.Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533–536, 1986.

[83] 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. Acoustics, Speech and Signal Processing*, vol. ASSP-31, no. ASSP-31, p. 814, 1983.

[84] H. Franco and A. Serralheiro, "Training HMMs using a minimum recognition approach," in *Proc. ICASSP '91*, (Toronto, Canada), pp. 357–360, May 1991.

[85] J.-K. Chen and F. Soong, "An N-best candidates-based discriminative training for speech recognition applications," *IEEE Transactions on Speech and Audio Processing*, vol. 2, pp. 206–216, Jan. 1994.

[86] E. McDermott, *Discriminative Training for Speech Recognition*. PhD thesis, Waseda University, Japan, March 1997.

[87] A. Nadas, D. Nahamoo, and M. A. Picheny, "On a model-robust training method for speech recognition," *IEEE Trans. Acoustics, Speech and Signal Processing*, vol. ASSP-36, no. ASSP-36, p. 1432, 1988.

[88] R. Cardin, Y. Normandin, and R. D. Mori, "High performance connected digit recognition using maximal mutual information estimation," in *Proc. ICASSP '91*, (Toronto, Canada), pp. 533–536, May 1991.

[89] Y. Normandin and S. Morgera, "An improved MMIE training algorithm for speaker-independent, small vocabulary, continuous speech recognition," in *Proc. ICASSP '91*, (Toronto, Canada), pp. 537–540, May 1991.

[90] S. Kapadia, V. Valtchev, and S. J. Young, "MMI training for continuous phoneme recognition on the TIMIT database," in *Proc. ICASSP '93*, (Minneapolis, MN), Apr. 1993.

[91] B.-H. Juang and S. Katagiri, "Discriminative learning for minimum error classification," *IEEE Transactions on Signal Processing*, vol. 40, pp. 3043–3054, December 1992.

[92] W. Chou, B.-H. Juang, and C.-H. Lee, "Segmental GPD training of HMM based speech recognizer," in *Proceedings of the International Conference on Acoustics Speech and Signal Processing*, (San Francisco), IEEE, pp. 473–476, 1992.

[93] O. Kwon and C. Un, "Performance of HMM-based speech recognizers with discriminative state-weights," *Speech Communication*, vol. 19, pp. 197–205, 1996.

[94] W. Chou, C.-H. Lee, and B.-H. Juang, "Minimum error rate training based on N-best string models," in *Proc. ICASSP '93*, (Minneapolis, MN), pp. I–652 – I–655, Apr. 1993.

[95] E. McDermott and S. Katagiri, "String-level MCE for continuous phoneme recognition," in *Proc. Eurospeech '97*, (Rhodes, Greece), pp. 123–126, Sep. 1997.

[96] P.-C. Chang, S.-H. Chen, and B.-H. Juang, "Discriminative analysis of distortion sequences in speech recognition," in *Proc. ICASSP '91*, (Toronto, Canada), pp. 549–552, May 1991.

[97] P.-C. Chang and B.-H. Juang, "Discriminative template training for dynamic programming speech recognition," in *Proc. ICASSP '92*, (San Francisco, CA), pp. 493–496, March 1992.

[98] T. Komori and S. Katagiri, "Application of a generalized probabilistic descent method to dynamic time warping-based speech recognition," in *Proc. ICASSP '92*, (San Francisco, CA), pp. 497–500, March 1992.

[99] E. McDermott and S. Katagiri, "Prototype-based MCE/GPD training for word spotting and connected word recognition," in *Proc. ICASSP '93*, (Minneapolis, MN), pp. II–291 – II–294, Apr. 1993.

[100] T. Komori and S. Katagiri, "An optimal learning method for minimizing spotting errors," in *Proc. ICASSP '93*, (Minneapolis, MN), pp. II–271 – II–274, Apr. 1993.

[101] R. Sukkar and J. Wilpon, "A two pass classifier for utterance rejection in keyword spotting," in *Proc. ICASSP '93*, (Minneapolis, MN), pp. II–451 – II–454, Apr. 1993.

[102] R. A. Sukkar, "Rejection for connected digit recognition based on GPD segmental discrimination," in *Proc. ICASSP '94*, (Adelaide, Australia), pp. I–393 – I–396, Apr. 1994.

[103] A. Biem and S. Katagiri, "Feature extraction based on minimum classification error/generalized probabilistic descent method," in *Proc. ICASSP '93*, (Minneapolis, MN), pp. II–275 – II–278, Apr. 1993.

[104] A. Biem and S. Katagiri, "Filter bank design based on discriminative feature extraction," in *Proc. ICASSP '94*, (Adelaide, Australia), pp. I–485 – I–488, Apr. 1994.

[105] C. Rathinavelu and L. Deng, "Use of generalized dynamic feature parameters for speech recognition: Maximum likelihood and minimum classification error approaches," in *Proc. ICASSP '95*, (Detroit, MI), pp. 373 – 376, May 1995.

[106] H. Watanabe, T. Yamaguchi, and S. Katagiri, "Discriminative metric design for pattern recognition," in *Proc. ICASSP '95*, (Detroit, MI), pp. 3439 – 3442, May 1995.

[107] C. Rathinavelu and L. Deng, "HMM-based speech recognition using state-dependent linear transforms on mel-warped DFT features," in *Proc. ICASSP '96*, (Atlanta, GA), pp. 9 – 12, May 1996.

[108] E. McDermott, E. Woudenberg, and S. Katagiri, "A telephone-based directory assistance system adaptively trained using minimum classification error/generalised probabilistic descent," in *Proc. ICASSP '96*, (Atlanta, GA), pp. 3347 – 3350, May 1996.

[109] K. Laurila, M. Vasilache, and O. Viikki, "A combination of discriminative and maximum likelihood techniques for noise robust speech recognition," in *Proc. ICASSP '98*, (Seattle, USA), pp. 85 – 88, May 1998.

[110] R. Haeb-Umbach and H. Ney, "Linear discriminant analysis for improved large vocabulary speech recognition," in *Proc. ICASSP '92*, (San Francisco, CA), pp. I–13–I–16, March 1992.

[111] C. Rathinavelu, "Minimum classification error linear regression (MCELR) for speaker adaptation using HMM with trend functions," in *Proc. Eurospeech '97*, (Rhodes, Greece), pp. 2343–2346, Sep. 1997.

[112] D. B. Paul, "On the interaction between between true source, training and testing language models," in *Proc. ICASSP '91*, (Toronto, Canada), pp. 569–572, May 1991.

[113] B. Mak and E. Barnard, "Phone clustering using the Bhattacharyya distance," in *Proc. ICSLP '96*, Vol. 4, (Philadelphia, PA), pp. 2005–2008, Oct. 1996.

[114] V. Diakoloukas and V. Digalakis, "Maximum-likelihood stochachastic-transformation adaptation of hidden Markov models," *IEEE Transactions on Speech and Audio Processing*, vol. 7, pp. 177–187, March 1999.