Discriminative Complexity Control and Linear Projections for Large Vocabulary Speech Recognition

Xunying Liu

Clare Hall
University of Cambridge

September 2005

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

Summary

Selecting the optimal model structure with the “appropriate” complexity is a standard problem for training large vocabulary continuous speech recognition (LVCSR) systems, and machine learning in general. State-of-the-art LVCSR systems are highly complex. A wide variety of techniques may be used which alter the system complexity and word error rate (WER). Explicitly evaluating systems for all possible configurations is infeasible. Automatic model complexity control criteria are needed. Most existing complexity control schemes can be classified into two types, Bayesian learning techniques and information theory approaches. An implicit assumption is made in both that increasing the likelihood on held-out data decreases the WER. However, this correlation is found to be quite weak for current speech recognition systems. Hence it is preferable to employ discriminative methods for complexity control. In this thesis a novel discriminative model selection technique, the marginalization of a discriminative growth function, is presented. This is a closer approximation to the true WER than standard likelihood based approaches. The number of Gaussian components and feature dimensions of an HMM based LVCSR system is controlled. Experimental results on a wide rage of LVCSR tasks showed that marginalized discriminative growth functions outperformed the best manually tuned systems using conventional complexity control techniques, such as BIC, in terms of WER.

Another important aspect of a speech recognition problem is to derive a good and compact feature representation for the data. This should contain sufficient discriminant information to distinguish between linguistic units. Features consisting of non-discriminating information should be removed. One category of such techniques are linear projection schemes. For these scheme the linear projections are normally estimated using the maximum likelihood (ML) criterion. It is well known that certain incorrect modeling assumptions are made in current HMM based speech recognition systems. Hence, in addition to a discriminative selection of number of subspace dimensions, it is also preferable to use discriminative criteria to estimate these projections. The commonly used extended Baum-Welch (EBW) algorithm provides an efficient, iterative, EM-like optimization scheme for discriminative criteria. However, using this algorithm the forms of model parameters that can be optimized are fairly restricted. Hence, it is useful to have a more general approach to discriminatively train a variety of forms of model parameters. In this thesis the recently proposed weak-sense auxilary function approach is used for discriminative estimation of linear projection schemes. Experimental results on a range of LVCSR tasks show that discriminative training of linear projections may be useful for improving the performances of current LVCSR systems.

Keywords

Speech Recognition, acoustic modeling, complexity control, discriminative growth functions, linear projection schemes, discriminative training, hidden Markov models.

Declaration

This thesis is the result of my own work carried out at the Cambridge University Engineering Department; it includes nothing which is the outcome of any work done in collaboration. Reference to the work of others is specifically indicated in the text where appropriate. Some of the material has been presented at international conferences and workshops [73, 71, 70, 72, 76], and technical reports [75, 74]. To my best knowledge, the length of this thesis including footnotes and appendices is approximately 53000 words.

Acknowledgment

First, I would like to thank my supervisor Mark Gales for his guidance and support throughout the work for this thesis. His experienced suggestions, well targeted questions and endless encouragement have been invaluable to me. It is a privilege to have worked with him.

Special thanks go to Phil Woodland and Steve Young for their support and leading the speech research group of the Machine Intelligence Laboratory. Outstanding computing facilities are provided by the group and I must also thank Patrick Gosling and Anna Langley for maintaining the computer systems.

I am grateful to Phil Woodland for kindly providing funding support throughout my study. Particular thanks must go to Sarah Airey and Khe Chain Sim for proof reading this thesis. There are also many people who have helped me in various ways and make the group a interesting place to work in. are many individuals to acknowledge but my thanks go to, no particular order, Gunnar Evermann, Thomas Hain, Dan Povey, Do Yeong Kim, Nathan Smith, Mat Stuttle, Yulan He, Hui Ye, Lang Wang, Kai Yu, Ho Yin Chan and David Mrva.

I am indebted to Steve Young, Phil Woodland, Mark Gales, Thomas Hain, Gunnar Evermann, Anna Langley and all other contributers to CU-HTK for providing and maintaining this excellent research toolkit. Without them this work would have been impossible.

Finally the biggest thank you to my family to whom I always owe everything!

Mathematical Notations:

$p(\cdot)$	probability density function
$P(\cdot)$	probability mass or prior distribution
$P(\cdot\|\cdot)$	conditional probability distribution
$\{\cdot\}^\top$	transpose of a matrix
$\|\cdot\|$	determinant of a square matrix
$\{\cdot\}^{-1}$	inverse of a square matrix
$\texttt{diag}(\cdot)$	diagonal elements of a square matrix
$\nabla\{\cdot\}$	gradient of a function
$\partial\{\cdot\}$	partial derivative of a function
$\mathcal{N}(\cdot;\boldsymbol{\mu},\boldsymbol{\Sigma})$	multivariate Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$

HMM Notations:

a_{ij}	discrete state transition probability
$b_j(\boldsymbol{o}_\tau)$	observation density given hidden state j
$\alpha_j(\tau)$	forward probability associated with hidden state j and time instance τ
$\beta_j(\tau)$	backward probability associated with hidden state j and time instance τ
$\gamma_j(\tau)$	posterior distribution of hidden state j given the observation sequence $\mathcal{O}$
$\boldsymbol{\mu}^{(j)}$	mean vector of hidden state j
$\boldsymbol{\Sigma}^{(j)}$	covariance matrix of hidden state j

General Model and Complexity Control Notations:

$\mathcal{M}$	model structural configuration
$\mathcal{O}$	sequence of observations with finite length
$\boldsymbol{o}_\tau$	n dimensional acoustic observation at a time instance τ
$\mathcal{T}$	total number of frame samples in the training data
λ	set of arbitrary model parameters
$\tilde{\lambda}$	set of current parameter estimates
$\hat{\lambda}$	set of optimal parameter estimates
$\mathcal{W}$	reference word sequence
$\tilde{\mathcal{W}}$	arbitrary sequence of words
$\mathcal{F}$	arbitrary training criterion
$\mathcal{G}$	discriminative growth function
$\mathcal{S}$	discrete hidden state
$\boldsymbol{\psi}$	sequence of hidden states
$\mathcal{Q}$	auxiliary function
$\mathcal{L}$	lower bound

$\mathcal{P}(\boldsymbol{\psi}, \lambda)$	a hidden state sequence posterior or variational distribution

Linear Projection Notations:

$\boldsymbol{A}$	$n \times n$ square linear transform
$\boldsymbol{A}_{[p]}$	$p \times n$ non-square linear projection with p rows where $p < n$
$\boldsymbol{B}$	between class covariance
$\boldsymbol{\Sigma}$	within class covariance
r	index of transformation classes for multiple projections
$\boldsymbol{\mu}^{(g,r)}$	global mean vector for class r
$\boldsymbol{\Sigma}^{(g,r)}$	global covariance matrix for class r
$\check{\boldsymbol{\mu}}^{(j)}$	transformed Gaussian mean
$\check{\boldsymbol{\Sigma}}^{(j)}$	transformed Gaussian covariance
$\boldsymbol{a}_i$	ith row of a linear transform
$\boldsymbol{c}_i$	cofactor vector of the ith row of a square linear transform

List of Acronyms:

AIC	Akaike information criterion
ASR	Automatic speech recognition
BIC	Bayesian information criterion
BN	Broadcast news
CER	Character error rate
CML	Conditional maximum likelihood
CN	Confusion network
CTS	Conversational telephone speech
EBW	Extended Baum-Welch
EM	Expectation maximization
GFunc	Growth function
GMM	Gaussian mixture model
HLDA	Heteroscedastic linear discriminant analysis
HMM	Hidden Markov model
LDA	Linear discriminant analysis
LDC	Linguistic data consortium
LVCSR	Large vocabulary continuous speech recognition
MCE	Minimum classification error
MCMC	Markov chain Monte Carlo
MDL	Minimum description length
MAP	Maximum a posteriori
ML	Maximum likelihood
MLLR	Maximum likelihood linear regression
MMI	Maximum mutual information
MML	Minimum message length
MWE	Minimum word error
MPE	Minimum phone error
PLP	Perceptual linear prediction
ROVER	Recognizer output voting error reduction
STC	Semi-tied covariances
VTLN	Vocal tract length normalization
WER	Word error rate

Contents

List of Figures

List of Tables

1

Introduction

Automatic speech recognition (ASR) has been the subject of active research for the past three decades. As the commercial and military interest has grown, investigation of ASR tasks has progressed to increasing difficulty and large scales. There have been significant advances in speech recognition technology in these years. Many techniques have been developed to improve the performance of speech recognition systems. The most significant technical breakthrough was made in the 1970s when hidden Markov models (HMMs) were introduced for speech recognition [5, 58]. In the following years hidden Markov models gradually became the dominant technique for acoustic modeling. These approaches have been applied to adapt them to a wide range of speech recognition tasks. ASR research has been applied to tasks ranging from clean and well controlled environments, such as Wall Street Journal (WSJ), to spontaneous, noisy and limited bandwidth domains, such as broadcast news (BN) and conversational telephone speech (CTS). As the complexity of the task has increased, the amount of date required for "good" performances is also increasing. Thousands of hours of audio data are being used for the training of state-of-the-art large vocabulary continuous speech recognition (LVCSR) systems. On the other hand, the rapid development of computing power in terms of speed and storage capability has further boosted the use of large amounts of training data. For these reasons state-of-the-art LVCSR systems are becoming more and more complex.

Many challenging problems still remain unsolved in speech recognition research. The performance of current speech recognition systems is still worse than human recognition. The performance of current ASR systems degrades rapidly as the level of background noise increases. In addition, the optimal complexity, or number of parameters, in a speech recognition system also affects the performance. This is the main area investigated in this thesis. Like many other pattern classification tasks, the correct model complexity, or structural configuration, needs to be determined to yield a good generalization to unseen data. For current speech recognition systems, especially on large vocabulary tasks, explicitly building and evaluating all possible systems is infeasible. Hence, automatic model complexity control criteria are needed. Another challenging problem in speech recognition research is how to extract a compact set of features that contain the most discriminant information. They should contain no redundant information, and more importantly should improve the classification accuracy. In this thesis the automatic complexity control and feature selection problems for HMM based speech recognition systems are investigated.

1.1 Speech Recognition Systems

A speech recognition system is normally decomposed into individual parts. The basic structure of a typical ASR system is shown in figure 1.1. The first stage involves the front-end processing of the speech waveforms. The speech signals are compressed into streams of acoustic feature vectors. These extracted feature vectors are assumed to contain sufficient information for the classification of speech patterns. An *acoustic model*, *language model* and *lexicon* are used to infer the most likely hypothesis for the spoken utterance given this set of acoustic features. The language model represents the syntactic and semantic information of the spoken sentence. The acoustic model maps each streams of acoustic feature vectors into individual words, or sub-word units. For LVCSR tasks the lexicon, or commonly referred to as *dictionary*, provides a mapping between words and sub-word units, reflecting the pronunciation variation of each word in the vocabulary. A wide range of techniques, such as parameter tying and discriminative training schemes, may be employed to improve the performance of speech recognition systems. These techniques may interact with each other. Hence, the development of an ASR system is complex and requires careful analysis, design and implementation of its individual parts.

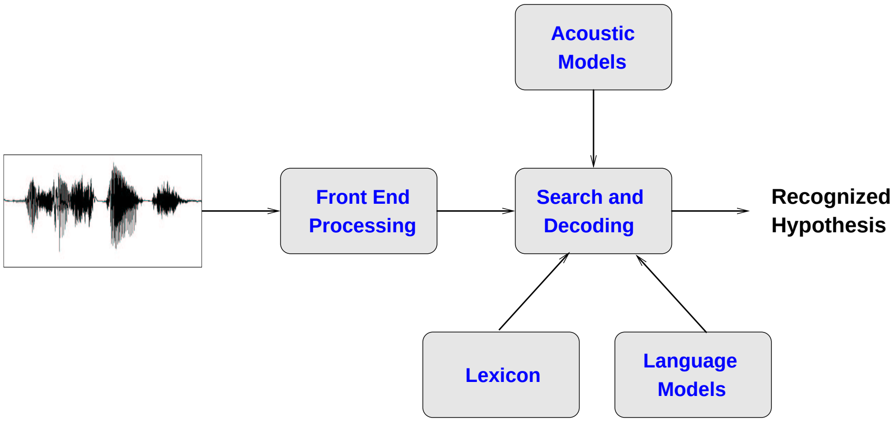

Figure 1.1 *An overview of a speech recognition system*

A statistical framework is usually used for speech recognition. The problem may be expressed as finding the most likely word sequence $\mathcal{W}$, given a sequence of acoustic observation vectors, $\mathcal{O} = \{\boldsymbol{o}_1, ..., \boldsymbol{o}_\tau, ..., \boldsymbol{o}_T\}$, where $\boldsymbol{o}_\tau$ denote the acoustic observation at some time instance τ. This may be written as

$$\mathcal{W} = \arg\max_{\tilde{\mathcal{W}}} \left\{ P(\tilde{\mathcal{W}}|\mathcal{O}) \right\}. \tag{1.1}$$

Applying Bayes rule yields

$$\begin{aligned}\mathcal{W} &= \arg\max_{\tilde{\mathcal{W}}} \left\{ \frac{p(\mathcal{O}|\tilde{\mathcal{W}})P(\tilde{\mathcal{W}})}{p(\mathcal{O})} \right\} \\ &= \arg\max_{\tilde{\mathcal{W}}} \left\{ p(\mathcal{O}|\tilde{\mathcal{W}})P(\tilde{\mathcal{W}}) \right\} \qquad (1.2)\end{aligned}$$

since the most likely word sequence is not dependent on the probability of the acoustic observations $p(\mathcal{O})$. The calculation of the optimal word sequence consists of two probability distributions: the probability of the acoustic vectors given a word sequence, $p(\mathcal{O}|\tilde{\mathcal{W}})$, given by the acoustic model; and the prior probability of a given word sequence, $P(\tilde{\mathcal{W}})$, given by the language model. This thesis is only concentrated on the complexity control problem for acoustic models and the selection of front-end features.

1.2 Model Complexity Control

Selecting the model structure with the "appropriate" complexity is a standard problem when training LVCSR systems and for machine learning in general. Systems with the optimal complexity have a good generalization to unseen data. For speech recognition systems, this generalization is usually measured by the word error rate (WER). Unfortunately, state-of-the-art LVCSR systems are highly complex. A wide range of techniques may be used which alter the system complexity and affect the WER performance. Examples of these techniques are using mixtures of Gaussians as state distributions, dimensionality reduction schemes, decision tree based state tying and linear transforms based speaker adaptation. Explicitly evaluating the WER for all possible model structural configurations is infeasible. It is therefore necessary to find a criterion that accurately predicts the WER ranking order, without explicitly requiring all the systems to be built and evaluated.

Most existing complexity control schemes can be classified into two types. In *Bayesian* techniques the model parameters are treated as random variables. The likelihood is integrated over the model parameters as random variables. This yields the Bayesian evidence [2, 122, 41]. In the *information theory* approaches the complexity control problem is viewed as finding a minimum code length for an underlying data generation process [16, 6, 96, 54]. These two approaches are closely related to each other. They asymptotically tend to the Bayesian information criterion (BIC) [104] first order expansion, or Laplace's approximation for second order expansion [122] with increasing amounts of data. These approximation schemes have been previously studied for various complexity control problems for speech recognition systems. For instance, they have been applied to determine the number of states in a decision tree based clustering [12, 13, 15, 59, 105, 107, 117, 130], or the number of linear transforms for speaker and environment adaptation [106]. An implicit assumption is made in both sets of schemes that increasing the likelihood on held-out data will decrease the WER. However, this correlation has been found to be weak for current speech recognition systems [71, 70]. This is due to two well known incorrect modeling assumptions with the HMM based framework: the observation independence assumption and the quasi-stationary assumption. Thus it would be preferable to use a complexity control scheme that is more closely related to WER. Discriminative measure has previously been used for building speech recognition systems. In [4, 88, 85], it was used as a method of incrementally splitting Gaussian mixture components. However, no stopping criterion was provided to penalize over-complex model structures.

This thesis presents a novel complexity control technique that uses the marginalization of a discriminative measure, rather than using the likelihood as in standard Bayesian approaches. Due to sensitivity to outliers, the direct marginalization of discriminative criteria, such as maximum mutual information (MMI) [3], is inappropriate for complexity control. Instead a related *discriminative growth function* is marginalized. This growth function retains certain attributes of the original discriminative criterion but has reduced sensitivity to outliers. The calculation of the "discriminative evidence" is still impractical for LVCSR systems. Hence, for efficiency Laplace's approximation is used for the integration of discriminative growth functions. The growth functions proposed in this thesis are based on the MMI and minimum phone error (MPE) [93, 62] criteria.

This work uses ASR systems built from HMM based acoustic models that have mixtures of Gaussians as the state output distributions and multiple linear feature projections. Two forms of system complexity attributes are to be investigated, the number of components per state and the number of dimensions for each projection. In addition to a discriminative selection of the dimensionality, a second area investigates in this thesis is the discriminative estimation of linear projection schemes.

1.3 Discriminative Linear Projection Schemes

In common with other pattern classification tasks, an important aspect of the speech recognition problem is to derive a good, compact, feature representation for the data. This should contain sufficient discriminant information to distinguish between classes. Features consisting of non-discriminating information should be removed. One family of such techniques used in speech recognition systems are linear projection schemes. Standard linear projection schemes, such as linear discriminant analysis (LDA) [26, 121] and its heteroscedastic extensions [66, 102, 34], attempt to generate one or more uncorrelated subspaces within the maximum likelihood (ML) framework. When using multiple projections, a consistent likelihood comparison may be ensured across different subspaces associated with each projection. However, it is well known that certain incorrect modeling assumptions are made in current HMM based speech recognition systems. Hence, in addition to a discriminative control of the number of subspace dimensions, it is also preferable to use discriminative criteria to estimate linear projections.

Most state-of-the-art LVCSR systems are built using discriminative training techniques [124, 51, 23, 64]. Usually the extended Baum-Welch (EBW) algorithm is used as it provides an efficient iterative EM-like optimization scheme for discriminative training criteria. However, using the EBW algorithm the forms of model parameters that may be optimized are restricted to standard

HMM parameters, such as Gaussian means, covariances. Gradient descent based numerical techniques are expensive for LVCSR training and have difficulty guaranteeing convergence in practice. Recently the weak-sense auxiliary function approach was introduced. This method provides a flexible and intuitive derivation of the EBW algorithm [91, 89, 93]. In this thesis weak-sense auxiliary functions are used to discriminatively optimize linear projections.

1.4 Thesis Structure

This thesis is structured as follows: In the following chapter the basic theory of using hidden Markov models for speech recognition, and the maximum likelihood training scheme are presented. Other details of the development of a large vocabulary recognizer, including the parameterization of human speech, selection of recognition units and parameter tying, language and pronunciation modeling are also briefly reviewed. Then the basic search and decoding algorithms are briefly described. Finally, two categories of acoustic modeling techniques widely used in state-of-the-art speech recognition systems, linear feature projection schemes and speaker adaptation techniques, are presented.

Chapter 3 presents standard complexity control techniques. First, the word error rate is the most widely used performance evaluation metric for current ASR tasks, hence minizing the WER on test data may be viewed as the ultimate aim, or a zero risk complexity control criterion, for speech recognition. In standard complexity control techniques a model correctness assumption is made that the likelihood on unseen speech data is strongly correlated with the systems' WER. Under this general likelihood based framework, two major categories of model selection schemes, Bayesian learning techniques and information theory methods, are outlined. This is followed by a brief review on existing complexity control research for speech recognition. Finally, the limitations of likelihood based complexity control schemes are discussed.

Chapter 4 presents standard discriminative training techniques for speech recognition. In this chapter several commonly used discriminative criteria are presented first, followed by a discussion on the optimization schemes for discriminative training. In particular, the extended Baum-Welch (EBW) algorithm, and a recently introduced weak-sense auxiliary function based approach are presented.

In chapter 5 a novel discriminative model complexity control technique is presented. First, some previous work on discriminative complexity control is reviewed. Then issues with a direct marginalization of discriminative criteria for complexity control are discussed. Due to the sensitivity to outliers, direct marginalization of discriminative training criteria is inappropriate for complexity control. Instead the criteria are transformed into a closely related discriminative growth function to be marginalized over. A discriminative growth function retains certain attributes of the original criterion and has reduced sensitivity to outliers. In this chapter two forms of growth functions based on the MPE and MMI criteria are presented. This is followed by a discussion on implementation issues when using growth functions for complexity control. Detailed derivations for discriminative growth functions can be found in appendix A and B.

In chapter 6 the discriminative training algorithms for linear projections schemes are presented. First, an introduction and motivation of the work is presented. Then previous research on the discriminative training of linear transformations for speech recognition is reviewed. This is followed by an investigation of using weak-sense auxiliary functions for discriminative training of linear projection schemes. Some implementations issues are also discussed in this chapter. Some detailed derivations of using weak-sense auxiliary functions to derive the update algorithms can be found in appendix C.

In chapter 7 experimental results are presented for model complexity control using marginalized discriminative growth functions. Initially, complexity control schemes are used to optimize multiple model complexity attributes on a global level. This allows all systems to be trained and evaluated explicitly. The correlation with WER and the performance ranking error is examined for a variety of complexity control schemes. This is followed by the optimization of multiple complexity attributes on a local level for an LVCSR task on CTS English data. The generalization to two other LVCSR tasks is also investigated using marginalized discriminative growth functions. The interaction with discriminative training and speaker adaptation techniques is also investigated. Finally, the performances of complexity controlled systems are evaluated within a state-of-the-art 10 time real-time LVCSR system.

Chapter 8 presents the performances of discriminatively trained linear projections on LVCSR tasks. Initially, experimental results for CTS English data are presented. Then the generalization to two other LVCSR tasks are investigated. This is followed by an investigation of using matched lattices for the discriminative training of standard HMM parameters after linear projections are estimated. Finally, the optimization of both model complexity and parameter are integrated into a consistent, discriminative, framework. The complexity of discriminatively trained model structures is optimized for CTS tasks.

In chapter 9 a summary of the work in this thesis is presented. Potential future directions of research are also discussed.

2

Fundamentals of Speech Recognition

In this chapter the basic theory of using Hidden Markov models for speech recognition is outlined. The standard maximum likelihood training of these models is presented. In addition, the parameterization of speech, the selection of recognition units and parameter tying, language and pronunciation modeling, and the decoding algorithm are briefly described. Finally two categories of techniques that are widely used in state-of-the-art speech recognition systems are presented. The two categories are linear feature projection schemes and speaker adaptation techniques.

2.1 HMMs as Acoustic Models

Currently the most popular and successful approach for modeling the variations of speech signals is to use hidden Markov models (HMM). Since their introduction in the 1970's HMMs have been applied to a wide range of speech recognition tasks [5, 58]. In this section the basic concepts of HMMs are presented. The structural assumptions that underly HMMs are also discussed.

2.1.1 Model Topology

Speech production is a non-stationary process. Precisely modeling all the complexities of the signals is impossible. When using HMMs to model speech signals, certain simplifying assumptions are made about the nature of speech. Although HMMs have been the most successful form of acoustic models for ASR systems, they are not the *correct* models for modeling speech patterns. When using HMMs the following assumptions are made about the nature of the speech signals:

- Speech signals may be split into discrete states in which the waveform is stationary and transitions between states are instantaneous. This is often referred to as the *quasi-stationary* assumption.
- The probability of an acoustic observation is only conditionally dependent on the vector and the current hidden state. Each observation vector is conditionally independent of the

sequence of vectors preceding and following it, given the current state. This is commonly referred to as the *observation independence* assumption.

Neither of these two assumptions are true for speech signals. The first assumption is not valid because speech production is a non-stationary process. The second assumption is not true for multiple reasons. For instance, the dynamics of speech articulator constrain its trajectory to be continuous, rather than discrete. Furthermore, techniques like the use of overlapping frames in speech parameterization may also introduce correlation between acoustic observations. These assumptions are further discussed in later sections.

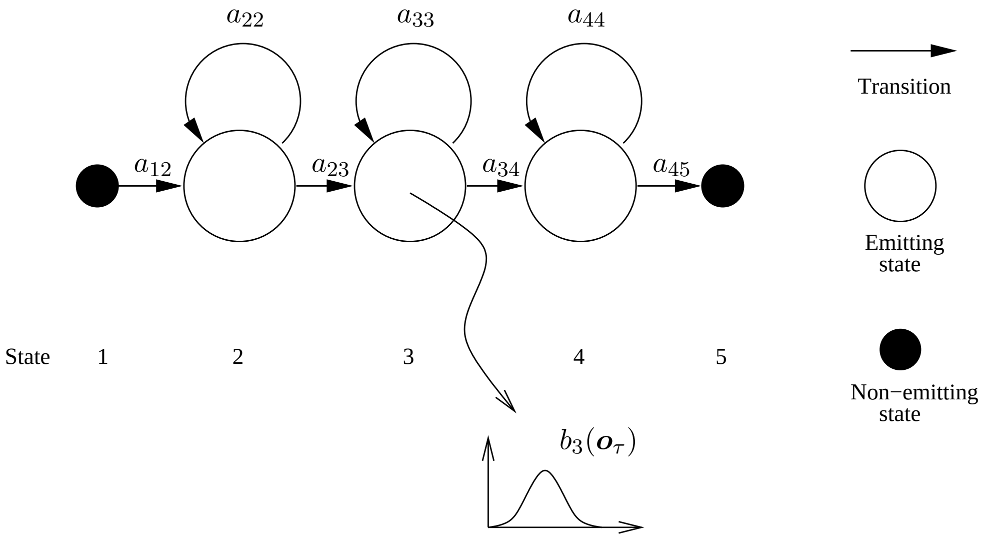

Figure 2.1 *An HMM with a left-to-right topology and three emitting states*

Under these assumptions speech signals that are expressed as a sequence of n dimensional acoustic observations of finite length, $\mathcal{O} = \{\boldsymbol{o}_1, ..., \boldsymbol{o}_{\mathcal{T}}\}$, are assumed to be generated by a Markov model as is shown in figure 2.1.1. Here self-loop transitions are allowed. In the figure a simple left-to-right model topology is used. There are a total of five states, including three emitting states and non-emitting entrance and exit states. Let λ denote the model parameters and $\boldsymbol{\psi}$ an arbitrary hidden state sequence. The model parameters describe the probability density function (PDF) associated with each emitting state and transition probabilities associated with each pair of states. In the figure an observation PDF, $b_j(\boldsymbol{o}_\tau) = p(\boldsymbol{o}_\tau|\boldsymbol{\psi}_\tau = \mathcal{S}_j, \lambda)$, is associated with each emitting state. Here $\boldsymbol{\psi}_\tau = \mathcal{S}_j$ indicates that at time instance τ, an acoustic observation $\boldsymbol{o}_\tau$ was generated by a hidden state j. In addition, a transition probability, $a_{ij} = P(\boldsymbol{\psi}_\tau = \mathcal{S}_j|\boldsymbol{\psi}_{\tau-1} = \mathcal{S}_i, \lambda)$, is associated with each pair of states. For any state, the transition probabilities satisfies a sum-to-one constraint, $\sum_j a_{ij} = 1$. Note that self-looping transitions are not allowed for non-emitting states. These non-emitting states allows multiple HMMs to be simply concatenated together to form a composite model.

2.1.2 State Output Distributions

The state emission PDF may have a variety of forms of distribution. Its form depends on the front-end feature extraction for speech signals. A more detailed discussion of frond-end processing techniques for speech recognition may be found in section 2.3.1. If the input speech data is discrete, or the data has been vector quantized, then discrete state PDFs may be used. However the majority of the current speech recognition systems use continuous acoustic features. A commonly used form is a multivariate Gaussian distribution given by

$$\begin{aligned} b_j(\boldsymbol{o}_\tau) &= \mathcal{N}\left(\boldsymbol{o}_\tau; \boldsymbol{\mu}^{(j)}, \boldsymbol{\Sigma}^{(j)}\right) \\ & (2\pi)^{-\frac{n}{2}} \left|\boldsymbol{\Sigma}^{(j)}\right|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}\left(\boldsymbol{o}_\tau - \boldsymbol{\mu}^{(j)}\right)^\top \boldsymbol{\Sigma}^{(j)-1}\left(\boldsymbol{o}_\tau - \boldsymbol{\mu}^{(j)}\right)\right\} \end{aligned} \tag{2.1}$$

where $\boldsymbol{\mu}^{(j)}$ and $\boldsymbol{\Sigma}^{(j)}$ are the Gaussian mean and covariance respectively.

Using full covariances for large HMM systems is computationally expensive. Let n denotes the dimensionality of the acoustic space. The number of covariance parameters is increased by $\mathcal{O}(n^2)$ as n increases. The number of HMM states in an LVCSR system can be in the thousands. In order to obtain robust parameter estimates, the training of full covariance Gaussians may also require a large amount of data. To overcome this problem, diagonal matrices may be used. However for complex patterns like speech such an approximation may be poor. Alternatively, more complicated methods may be used. These techniques include linear projection schemes that attempt to remove the spatial correlation, and advanced forms of covariance parameter tying. These techniques are discussed in more detail in later sections.

By using a Gaussian distribution it is assumed that the state emission distribution has a single mode at the mean. However, the characteristics of speech may vary substantially depending on the speaker and acoustic environment. This may result in a mismatch between models and data. Hence, instead of using diagonal covariance Gaussian distributions, Gaussian mixture models (GMM) are widely used as the state emission PDFs [69]. A GMM based state emission PDF is given by,

$$b_j(\boldsymbol{o}_\tau) = \sum_{m=1}^{M_j} c_{jm}\mathcal{N}\left(\boldsymbol{o}_\tau; \boldsymbol{\mu}^{(jm)}, \boldsymbol{\Sigma}^{(jm)}\right) \tag{2.2}$$

where M_j the number of mixture components for state j, and $\mathcal{N}(\cdot)$ denotes a multivariate Gaussian distribution of the form given in equation 2.1. The component prior c_{jm} satisfies a sum-to-one constraint, $\sum_{m=1}^{M_j} c_{jm} = 1$, to ensure that $b_j(\boldsymbol{o}_\tau)$ is a valid PDF. Usually diagonal covariance matrices are used for each component. Using GMMs the spatial correlation in the acoustic space may be implicitly accounted for. Alternatively, other more complicated forms of covariances may be used [31, 45, 99, 98, 108].

There are two issues when using GMMs as state distributions for an HMM based speech recognition system. First, the number of Gaussian components in each GMM affects the overall complexity of the system and needs to be determined. This may be manually tuned by explicitly building and evaluating all possible systems. However, this is only applicable when the same number of components is assigned to all states in the system. A more complicated scenario is that the complexity is locally varied across different states. In these cases automatic model complexity control techniques are required. Second, the number of Gaussian components in LVCSR systems can be in the millions. A significant portion of the run time is consumed by likelihood calculation on mixture component level. To achieve efficiency, appropriate caching and pruning of Gaussian probabilities may be used [32].

2.2 Maximum Likelihood Training of HMMs

Maximum likelihood (ML) training is a standard machine learning scheme. The underlying model is assumed to be close to the "correct" one so that increasing the likelihood of the training data will decrease the classification error on the unseen data. For an HMM based speech recognition system, the aim is to find the optimal parameter estimates, $\hat{\lambda}$, such that the log likelihood of the given observation sequence is maximized. This may be expressed as

$$\hat{\lambda} \;=\; \arg\max_{\lambda} \left\{ \log p(\mathcal{O}|\mathcal{W},\lambda) \right\} \tag{2.3}$$

where $\mathcal{W}$ is the reference transcription. Directly maximizing equation 2.3, for example by setting the gradient with respect to λ to zero, is non-trivial. This is because the likelihood may be expressed as a marginalization over a set of unknown hidden state sequences $\{\boldsymbol{\psi}\}$, allowed by the reference transcription,

$$\begin{aligned} \hat{\lambda} &= \arg\max_{\lambda} \left\{ \log \sum_{\boldsymbol{\psi}} p(\mathcal{O},\boldsymbol{\psi}|\mathcal{W},\lambda) \right\} \\ &= \arg\max_{\lambda} \left\{ \log \sum_{\boldsymbol{\psi}} \prod_{\tau} P(\boldsymbol{\psi}_{\tau}|\boldsymbol{\psi}_{\tau-1},\lambda) p(\boldsymbol{o}_{\tau}|\boldsymbol{\psi}_{\tau},\lambda) \right\} \end{aligned} \tag{2.4}$$

where $\boldsymbol{\psi}_{\tau}$ denotes the hidden state an acoustic observation at time instance τ was generated from. For HMMs the *expectation maximization* algorithm [19] is normally used to maximize the log-likelihood of the training data.

2.2.1 EM Algorithm

The EM algorithm is a standard optimization scheme for statistical models which may contain latent variables. An HMM is an example of these models. Its hidden states may be viewed as latent variables. Rather than directly maximizing the log likelihood in equation 2.4, the following strict lower bound of the training data log likelihood, derived using Jensen's inequality, will be optimized,

$$\begin{aligned} \log \sum_{\boldsymbol{\psi}} p(\mathcal{O},\boldsymbol{\psi}|\mathcal{W},\lambda) &= \log \sum_{\boldsymbol{\psi}} P(\boldsymbol{\psi}|\mathcal{O},\mathcal{W},\tilde{\lambda}) \frac{p(\mathcal{O},\boldsymbol{\psi}|\mathcal{W},\lambda)}{P(\boldsymbol{\psi}|\mathcal{O},\mathcal{W},\tilde{\lambda})} \\ &\geq \sum_{\boldsymbol{\psi}} P(\boldsymbol{\psi}|\mathcal{O},\mathcal{W},\tilde{\lambda}) \log \frac{p(\mathcal{O},\boldsymbol{\psi}|\mathcal{W},\lambda)}{P(\boldsymbol{\psi}|\mathcal{O},\mathcal{W},\tilde{\lambda})} \end{aligned} \tag{2.5}$$

where $\tilde{\lambda}$ is the *current* estimate of model parameters. Applying Jensen's inequality requires that the hidden state sequence posteriors $P(\boldsymbol{\psi}|\mathcal{O},\mathcal{W},\tilde{\lambda})$ satisfies a non-negative and sum-to-one constraint. As $P(\boldsymbol{\psi}|\mathcal{O},\mathcal{W},\tilde{\lambda})$ is a probability, this constraint holds. This lower bound may be re-arranged as

$$\log p(\mathcal{O}|\lambda,\mathcal{W}) \quad \geq \quad \log p(\mathcal{O}|\tilde{\lambda},\mathcal{W}) + \mathcal{Q}_{\mathtt{ml}}(\lambda,\tilde{\lambda}) - \mathcal{Q}_{\mathtt{ml}}(\tilde{\lambda},\tilde{\lambda}) \tag{2.6}$$

where the auxiliary function $\mathcal{Q}_{\mathtt{ml}}(\lambda,\tilde{\lambda})$ is given by

$$\mathcal{Q}_{\mathtt{ml}}(\lambda,\tilde{\lambda}) \quad = \quad \sum_{\boldsymbol{\psi}} P(\boldsymbol{\psi}|\mathcal{O},\mathcal{W},\tilde{\lambda}) \log p(\mathcal{O},\boldsymbol{\psi}|\lambda) \tag{2.7}$$

The EM algorithm is performed in an iterative fashion. In the E-step, the hidden state sequence posteriors, $P(\boldsymbol{\psi}|\mathcal{O},\mathcal{W},\tilde{\lambda})$, are computed given the current parameters estimates, $\tilde{\lambda}$, obtained from the previous iteration. In the M-step, the lower bound in equation 2.6 is optimized given the fixed statistics computed in the E-step. Note that equation 2.6 becomes an equality when $\lambda = \tilde{\lambda}$. Maximizing the lower bound given in equation 2.6 is guaranteed not to decrease the log likelihood of the training data. During the M-step, this is equivalent to maximizing the auxiliary function, $\mathcal{Q}(\lambda,\tilde{\lambda})$, given the fixed statistics. One limitation with the EM algorithm is that it can only find a local optimum for the model parameters when the log likelihood converges.

2.2.2 Forward-backward Algorithm and Parameter Re-estimation

Using the observation independence assumption discussed in section 2.1.1, the EM auxiliary function in equation 2.7 may be written as the following for HMMs,

$$\mathcal{Q}_{\mathtt{ml}}(\lambda,\tilde{\lambda}) \quad = \quad \sum_{j,\tau} \gamma_j(\tau) \log b_j(\boldsymbol{o}_\tau) + \sum_{j,i,\tau} \xi_{ij}(\tau) \log a_{ij} \tag{2.8}$$

where the hidden state posterior probability,

$$\gamma_j(\tau) \quad = \quad P(\boldsymbol{\psi}_\tau = \mathcal{S}_j|\mathcal{O},\mathcal{W},\tilde{\lambda}) \tag{2.9}$$

and the pairwise hidden state transition posterior.

$$\xi_{ij}(\tau) \quad = \quad P(\boldsymbol{\psi}_{\tau-1} = \mathcal{S}_i, \boldsymbol{\psi}_\tau = \mathcal{S}_j|\mathcal{O},\mathcal{W},\tilde{\lambda}) \tag{2.10}$$

Here $\boldsymbol{\psi}_\tau = \mathcal{S}_j$ denotes that an acoustic observation vector was generated at time instance τ by hidden state j.

These two hidden state posterior probabilities are usually computed using the *forward* and *backward* probabilities. The forward probability is defined as the joint likelihood of the partial observation sequence up to time instance τ and frame $\boldsymbol{o}_\tau$ is emitted from state $\mathcal{S}_j$. This is expressed as

$$\alpha_j(\tau) \quad = \quad p(\boldsymbol{o}_1,...,\boldsymbol{o}_\tau,\boldsymbol{\psi}_\tau = \mathcal{S}_j|\mathcal{W},\tilde{\lambda}) \tag{2.11}$$

Using the observation independence assumption the forward probability may be computed by

$$\alpha_j(\tau) \;=\; \begin{cases} 1 & j=1, \quad \tau=1 \\ a_{1j}b_j(\boldsymbol{o}_\tau) & 1<j<N_s, \quad \tau=1 \\ \sum_{i=2}^{N_s-1}\alpha_i(\tau-1)a_{ij}b_j(\boldsymbol{o}_\tau) & 1<j<N_s, \quad 1<\tau\leq\mathcal{T} \\ \sum_{i=2}^{N_s-1}\alpha_i(\tau)a_{ij} & j=N_s, \quad \tau=\mathcal{T} \end{cases} \tag{2.12}$$

where N_s is the number of states in each HMM, including the non-emitting entry and exit states.

The backward probability, defined as

$$\beta_j(\tau) \;=\; p(\boldsymbol{o}_{\tau+1},......,\boldsymbol{o}_{\mathcal{T}}|\boldsymbol{\psi}_\tau=\mathcal{S}_j,\mathcal{W},\tilde{\lambda}), \tag{2.13}$$

is also recursively calculated for the partial observation sequence from time instance $\tau+1$ up to $\mathcal{T}$.

$$\beta_j(\tau) \;=\; \begin{cases} \sum_{i=2}^{N_s-1}a_{1i}b_i(\boldsymbol{o}_1)\beta_i(1) & j=1, \quad \tau=1 \\ \sum_{i=2}^{N_s-1}a_{ji}b_i(\boldsymbol{o}_{\tau+1})\beta_i(\tau+1) & 1<j<N_s, \quad 1\leq\tau<\mathcal{T} \\ a_{jN_s} & j=N_s, \quad \tau=\mathcal{T} \end{cases} \tag{2.14}$$

Using the forward and backward probabilities, the hidden state posterior probability, $\gamma_j(\tau)$, and the transition posterior, $\xi_{ij}(\tau)$, may be efficiently computed using

$$\begin{aligned} \gamma_j(\tau) &= \frac{\alpha_j(\tau)\beta_j(\tau)}{p(\mathcal{O}|\mathcal{W},\tilde{\lambda})} \\ \xi_{ij}(\tau) &= \frac{\alpha_i(\tau-1)a_{ij}b_j(\boldsymbol{o}_\tau)\beta_j(\tau)}{p(\mathcal{O}|\mathcal{W},\tilde{\lambda})} \end{aligned} \tag{2.15}$$

The total likelihood of the complete observation sequence may be calculated as

$$p(\mathcal{O}|\mathcal{W},\tilde{\lambda}) \;=\; \alpha_{N_s}(\mathcal{T}), \tag{2.16}$$

or

$$p(\mathcal{O}|\mathcal{W},\tilde{\lambda}) \;=\; \beta_1(1). \tag{2.17}$$

For HMMs using GMMs as state emission PDFs, Gaussian mixture component may be treated as hidden variables. The component posteriors, $\gamma_{jm}(\tau)$, are required as sufficient statistics for re-estimating the parameters. This is given by

$$\gamma_{jm}(\tau) \;=\; \frac{\sum_{i=2}^{N_s-1}\alpha_i(\tau-1)a_{ij}c_{jm}b_{jm}(\boldsymbol{o}_\tau)\beta_j(\tau)}{p(\mathcal{O}|\mathcal{W},\tilde{\lambda})} \tag{2.18}$$

Given these sufficient statistics the parameter re-estimation formula for HMM may be derived. For the state transition probabilities, the update formula is given by

$$a_{ij} \;=\; \begin{cases} \gamma_j(1) & i=1, \quad 1<j<N_s \\ \frac{\sum_{\tau=2}^{\mathcal{T}}\xi_{ij}(\tau)}{\sum_{\tau=2}^{\mathcal{T}}\gamma_i(\tau-1)} & 1<i<N_s, \quad 1<j<N_s \\ \frac{\gamma_i(\mathcal{T})}{\sum_{\tau=2}^{\mathcal{T}}\gamma_i(\tau-1)} & 1<i<N_s, \quad j=N_s \end{cases} \tag{2.19}$$

The re-estimation formula for the weights, means and covariances of the component m of emitting state j are given by

$$\begin{aligned} c_{jm} &= \frac{\sum_{\tau} \gamma_{jm}(\tau)}{\sum_{m,\tau} \gamma_{jm}(\tau)} \\ \boldsymbol{\mu}^{(jm)} &= \frac{\sum_{\tau} \gamma_{jm}(\tau)\boldsymbol{o}_{\tau}}{\sum_{\tau} \gamma_{jm}(\tau)} \\ \boldsymbol{\Sigma}^{(jm)} &= \frac{\sum_{\tau} \gamma_{jm}(\tau) \left(\boldsymbol{o}_{\tau} - \boldsymbol{\mu}^{(jm)}\right) \left(\boldsymbol{o}_{\tau} - \boldsymbol{\mu}^{(jm)}\right)^{\top}}{\sum_{\tau} \gamma_{jm}(\tau)} \end{aligned} \qquad (2.20)$$

In the above update the re-estimation of full covariance Gaussians requires the second order moments to be stored as full matrices for each component. Again the computational requirement during training is dramatically increased as the feature dimensionality increases. Hence, it is preferable to use more complicated forms of covariance modeling techniques.

One limitation with ML training is that no prior knowledge about the model parameters is considered. This leads to unreliable estimates when the training data is limited. Prior knowledge about model parameters may be incorporated, for example, in *maximum a-posteriori* (MAP) [36] training and *Bayesian learning* [2]. Thus uncertainty about model parameters may be more robustly handled. Furthermore, in ML training the underlying statistical model is assumed to be the “correct” one. For current ASR systems using HMMs, this model correctness assumption may be too strong due to the two structural assumptions explained in section 2.1.1. Hence it is preferable to employ training schemes that explicitly aim to reduce the classification error rate, such as discriminative training criteria.

2.3 Recognition of Speech Using HMMs

In this section the application of HMMs for recognizing speech is outlined. First, the parameterization of speech signals as the input for HMMs is presented. Then the selection of recognition units and parameter tying is discussed. This is followed by an outline of the usage of language models and the modeling of pronunciation variants. Finally, the search and decoding algorithms are briefly described.

2.3.1 Parameterization of Speech

When using HMMs for speech recognition several assumptions are made about the nature of the speech signals, as described in section 2.1.1. One assumption is that speech waveforms may be partitioned into series of quasi-stationary discrete segments, or frames. The standard front-end processing schemes are based on this assumption. The spectral envelope of the signals is extracted for each frame, which contains most of the useful information of speech [18]. Two types of speech parameterization are widely used in current speech recognition systems, *Mel-frequency cepstral coefficients* (MFCC) [17] and *perceptual linear prediction* (PLP) [55]. In both cases the frame length is fixed by a predefined widowing function, for example, at 10 ms. For each frame an acoustic observation vector is produced using cepstral analysis for a segment of speech. The span of the widowing function is often set as 25ms. The widowing functions may be over-lapping over adjacent frames.

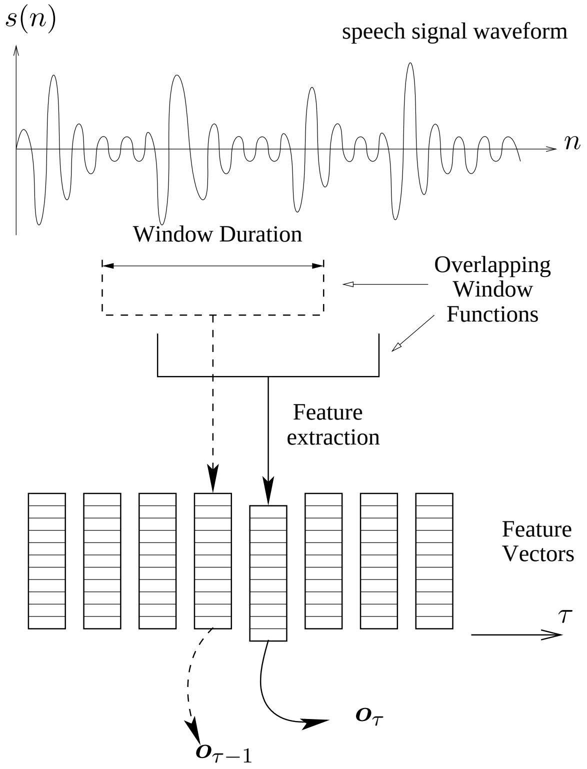

Figure 2.2 *Extraction of acoustic features using over-lapping widowing functions*

Figure 2.3.1 illustrates the extraction of acoustic features using over-lapping widowing functions. The first stage is to apply a windowing function, such as Hamming or Hanning window [18]. Both aim at smooth the over-lapping regions of speech signals that belong to different frames, so that the boundary effects may be reduced. For each frame a short term analysis of the speech signals is performed using a Fourier transform to obtain the frequency domain power spectrum. The linear frequency scale is then warped. For MFCC front-ends a Mel-frequency scale is used. This is given by

$$f_{\mathtt{mel}} \quad = \quad 1125 \log\left(1 + \frac{f_{\mathtt{Hz}}}{625}\right) \tag{2.21}$$

where $f_{\mathtt{mel}}$ denotes the warped frequency on the Mel scale. The power spectrum is down-sampled using a bank of triangular filters, for instance 24. The log amplitudes of the down-sampled spectrum are then transformed using a *discrete cosine transform* (DCT) to reduce the spatial correlation between filter bank amplitudes. The DCT transform is given by

$$o_{\tau,i} \quad = \quad \sqrt{\frac{2}{B}} \sum_{b=1}^{B} \log\left(x_{\tau,b}\right) \cos\left(\frac{i(b-0.5)\pi}{B}\right) \tag{2.22}$$

where $x_{\tau,b}$ is the amplitude of filter bank b at time instance τ, and B is the total number of Mel scale filters. The cepstral coefficients used are often the lower 12. A 13 dimensional acoustic feature vector is constructed by further including either the zeroth order cepstra or the normalized log energy. Higher order cepstras represent the high frequency range variation in the spectrum and little information about speech, and hence may be removed. For PLP front-ends, the following Bark-frequency scale is used to warp the spectrum.

$$f_{\texttt{bark}} = \log\left\{\left[\left(\frac{f_{\texttt{Hz}}}{600}\right)^2 + 1\right]^{\frac{1}{2}} + \frac{f_{\texttt{Hz}}}{600}\right\} \tag{2.23}$$

where $f_{\texttt{bark}}$ denotes the warped frequency on the Bark scale. Critical band filters are then used for spectrum down-sampling. Equal-loudness, pre-emphasis and intensity-loudness power law are then applied. Finally *linear prediction* (LP) analysis is performed and the LP coefficients are transformed to the cepstral domain. In common with MFCC features, the order of LP analysis is often set as 12.

The observation independence assumption of HMMs ignores the temporal correlation of speech signals. Acoustic feature vectors are assumed independently against one another. Hence it is desirable to incorporate more information of the correlation between frames. One widely adopted approach is to include dynamic coefficients into the feature vector [27]. The first order dynamic coefficients, $\Delta\boldsymbol{o}_\tau$, or the delta coefficients are calculated by

$$\Delta\boldsymbol{o}_\tau = \frac{\sum_{d=1}^{D_r} d\left(\boldsymbol{o}_{\tau+d} - \boldsymbol{o}_{\tau-d}\right)}{2\sum_{d=1}^{D_r} d^2} \tag{2.24}$$

where $2D_r + 1$ is the size of the regression widow. The second order dynamic coefficients, $\Delta^2\boldsymbol{o}_\tau$, or the delta-delta features, are calculated in the same fashion as equation 2.24, by replacing the static parameters with the deltas features. Appending both the delta and delta-delta coefficients to the standard feature constructs a 39 dimensional acoustic vector. If D_r is set to 2, then the regression widow size for the delta-delta coefficients will span across a total of 9 consecutive frames.

Using dynamic coefficients, the observation independence assumption of HMMs may be compensated for to some degree without changing the model structure. However, the use of over-lapping widowing functions may introduce correlation between frames of speech samples. Hence some correlation may be introduced to the feature space when using dynamic features computed in equation 2.24. In this case, using diagonal Gaussian covariances may be a poor choice.

2.3.2 Recognition Units and Tying

For speech recognition tasks using a very small vocabulary, it is possible to use HMMs to model individual words. However, when the vocabulary size is increased, it is difficult to obtain sufficient data to robustly estimate HMM parameters for each word in the dictionary. In addition, the appropriate HMM topology needs to be determined for each word. The standard approach to solve this problem is to split words into smaller sub-word units, *phones* [57]. A phone may be a linguistic unit, such as a *phoneme* or *syllable*. Phonemes are the smallest atomic sub-units of speech. They are elementary sound units and represent the smallest distinct elements of speech. Syllables are the intermediate units between phonemes and words. Models based on phonemes are more commonly used than syllable models and often referred to as phone models. The selection of the phone set may depend on the amount of training data available. A phone set may not contain every single phoneme in the language being considered, and in practice often includes silence and short pause. A dictionary, or lexicon, contains the mapping from words to sub-word units. It is used to obtain the corresponding sequences of sub-word units given a word sequence. For continuous speech recognition all sub-word level HMMs are concatenated to form a composite model to represent words and sentences.

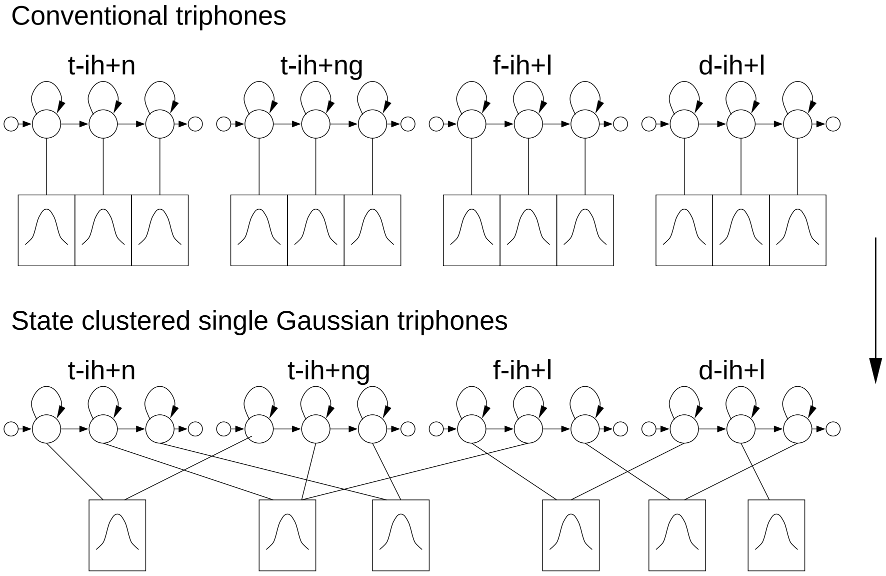

Figure 2.3 *State level tying for single Gaussian triphone HMMs*

When HMMs are used to model the basic phone set, without taking phonetic contexts into account, they are normally referred to as context *independent* or *monophone* models. Due to the co-articulatory effect, the acoustic realization of the same phone can vary substantially depending on the surrounding phonetic contexts. To model these variations, context *dependent* phones are often used. One commonly used type of context dependent phone is *triphone*, which considers both the preceding and following phones. It is possible to build up larger contexts using more phones on either side of the current phone, for instance, quinphone units [51], but only triphones are considered in this work. Triphones may be further split into two categories depending on the spanning of the phonetic contexts. Cross word triphones span across word boundaries, while word internal triphones do not. For word internal triphone systems, *biphones* are used to model the start and end phones at the word boundaries. For systems using context dependent phone models, given limited training data, parameter tying may be used to robustly estimate the model parameters [132, 133]. The tying of parameters can be flexible. It may be performed on different levels, such as phones, states or Gaussian components [53]. One commonly used approach for LVCSR systems is to perform state level parameter tying, such that certain states will share the same output distribution [125, 51]. Figure 2.3 shows an example of state level tying for four triphone HMMs with the same center phone **/ih/**. A triphone with the central phone **/ih/**, the left context **/t/**, and right context **/n/** is written as **/t-ih+n/**. After the tying there are a total of 6 distinct state distributions shared among 12 states.

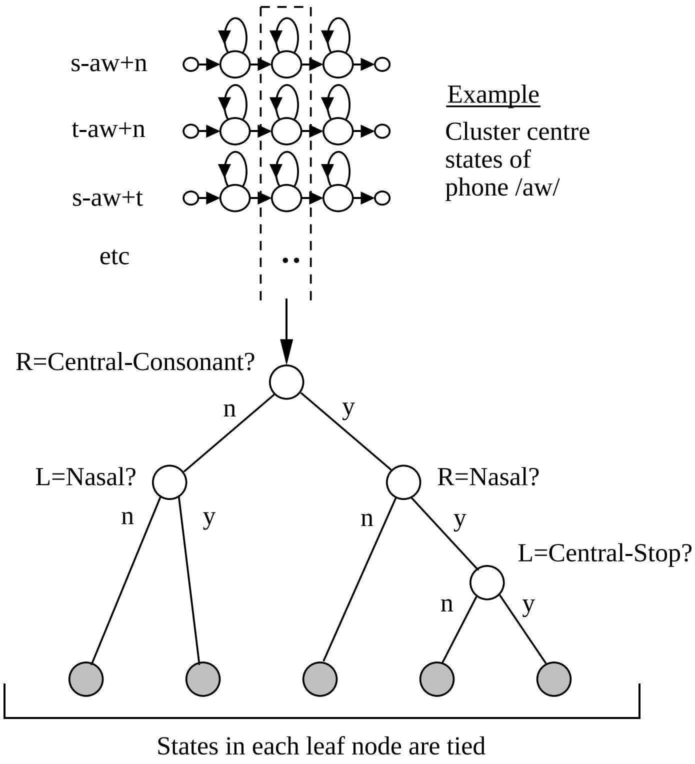

Figure 2.4 *Clustering of central states for triphones with center phone* **/aw/**

In order to perform state tying appropriate clustering schemes are required. One standard approach is to use a phonetic decision tree [132, 133]. A phonetic decision tree is a binary tree with a set of ”yes” or ”no” questions at each node related to the context surrounding each model. Figure 2.4 shows an example section of a phonetic decision tree for triphone models with the center phone **/aw/**. The clustering proceeds in a top-down fashion, with all states clustered together at the root node of the tree. The state clusters are then split based on the questions in the tree. The questions used are chosen to locally maximize the likelihood of the training data whilst ensuring that each clustered state also has a minimum amount of data observed. This ensures that rarely seen or unseen contexts may be robustly handled. In the final stage, tree nodes are merged if the likelihood loss is beneath a given threshold, until no such nodes can be found.

One disadvantage of decision tree based clustering is that the cluster splits are only local maximization, and not all questions that could split the state clusters are considered. Another issue with this method is that the complexity of the final tied HMM system is only manually controlled. Two thresholds require manual tuning: the minimum amount of training data associated with each tree node during splitting clusters, and the minimum likelihood loss when merging tree nodes. The setting of these two thresholds is often heuristic and largely on an empirical basis. Hence the optimal cut for the decision tree can not be automatically determined.

2.3.3 Pronunciation Modeling

In HMM based speech recognition systems the mapping between words and phones is provided by the lexicon, or dictionary. Characteristics of speech may vary substantially depending on the linguistic "environments". For example, differences in accents may lead to different phoneme realizations of the same word. Spontaneous speech may also introduce variability in the speaking style. Hence appropriate modeling of pronunciation variability is an important part of current speech recognition systems. The commonly used approach to model such variability is to include multiple pronunciation variants for each word in the dictionary. For instance, the English word "the" may have two pronunciation variants to choose, depending on the first phone of the following word. These variants are often generated automatically using a rule based system and then corrected manually [37]. The use of multiple pronunciation variants may increase the confusion between words, because the distance in pronunciation between words may become smaller. Thus the benefit from adding new variants has to be balanced with added confusability. One approach to solve this problem is to assign a probability to each variant. For most state-of-the-art LVCSR systems probabilities for pronunciation variants are estimated from the alignment of the training data [51, 126].

As discussed in section 2.3.2, state-of-the-art speech recognition systems make use of context dependent phones and parameter tying techniques. Note that a variety of tying schemes for HMM parameters may also be viewed as implicit ways to model the pronunciation variability [52]. These techniques include the phonetic decision tree based state clustering discussed in section 2.3.2, the use of tied-mixture models [8] and soft tying of states by sharing Gaussian components [103]. A more general form of stochastic tying of HMM parameters, the hidden model sequence HMMs proposed in [53], may also be viewed as an implicit modeling of pronunciation variation. For this reason there is no exact boundary between acoustic modeling and pronunciation modeling. However implicit pronunciation modeling using parameter tying are not considered in this thesis. Standard multiple pronunciation dictionaries with variant probabilities are used in the experiments.

2.3.4 Language Modeling

As discussed in section 1.1, the prior probability of a word sequence in a speech recognizer, $P(\mathcal{W})$, is given by a language model. Using the chain rule, the probability of a sequence of L words, $\mathcal{W} = \{w_1, w_2, ..., w_L\}$, may be decomposed into a product of conditional probability of individual words given its history.

$$P(\mathcal{W}) \;=\; \prod_{l=1}^{L} P(w_l|w_{l-1}, w_{l-2}, \ldots\ldots, w_1) \tag{2.25}$$

For LVCSR systems the vocabulary size is too big to allow a robust estimate of $P(\mathcal{W})$ for every possible word sequence. Thus it is necessary to reduce the parameter space to obtain a reasonable coverage and reliable probability estimation. This can be achieved by clustering the set of possible word histories into equivalent classes $h(w_{l-1}, w_{l-2}, \ldots\ldots, w_1)$. Once an appropriate set of equivalence classes has been defined, the probability of a word sequence $\mathcal{W}$ in equation 2.25, may be written as

$$P(\mathcal{W}) \;=\; \prod_{l=1}^{L} P(w_l|h(w_{l-1}, w_{l-2}, \ldots\ldots, w_1)) \tag{2.26}$$

N-gram language models are one standard approach to cluster histories into equivalence classes. For N-gram language models, word histories may be defined by how many words they are truncated before the current word. For example in case of a *tri-gram* language model equivalence classes are constrained as the set of all possible word pairs.

$$h(w_{l-1}, w_{l-2}, \ldots\ldots, w_1) \;\approx\; (w_{l-1}, w_{l-2}) \tag{2.27}$$

Using this approximation, it is straightforward to obtain ML estimates for tri-gram language models. These are are given by

$$P(w_l|w_{l-1}, w_{l-2}) \;=\; \frac{N(w_l, w_{l-1}, w_{l-2})}{\sum_w N(w, w_{l-1}, w_{l-2})} \tag{2.28}$$

where $N(w_l, w_{l-1}, w_{l-2})$ denotes the frequency counts of the word triplet observed in the training data. In order to robustly estimate these probabilities, sufficient coverage of possible word triplets in the training data is required. For a vocabulary of V words the number of possible tri-grams is V^3. Complete coverage for all tri-grams in the observed data is infeasible.

To obtain robust estimates of N-gram probabilities, smoothing approaches are commonly used. One category of techniques smooth the N-gram probability estimates by allocating a certain amount of the overall probability mass to those unseen events. These methods are referred to as *discounting* schemes. The portion of probability mass re-distributed is controlled by a discounting factor. Popular discounting techniques include Good-Turing discounting [46, 63], Witten-Bell discounting [123] and absolute discounting [83]. Another type of techniques is *back-off*. Instead of allocating a certain amount of probability mass to all possible histories, including those that are highly unlikely, back-off makes use of distributions with shorter histories and thus can be estimated more robustly. These distributions are called back-off distributions. The probabilities for unseen and rare events are taken from the back-off distributions after proper normalization. In practice a hierarchical back-off may be used. For example, a hierarchy might back-off 4-gram distributions to tri-gram, bi-gram and ultimately uni-gram distributions. A third category of smoothing techniques is deleted interpolation. For instance, uni-gram, bi-gram and tri-gram distributions are interpolated using weights. These weights may be tuned on held-out data.

2.3.5 Decoding Algorithms

In a speech recognition system, *decoding* or *search* refers to the process of finding the most probable word sequence, $\mathcal{W}$, given an observation sequence, $\mathcal{O}$. This can be expressed as

$$\begin{aligned}\mathcal{W} &= \arg\max_{\tilde{\mathcal{W}}} \left\{ P(\tilde{\mathcal{W}}|\mathcal{O},\lambda) \right\} \\ &= \arg\max_{\tilde{\mathcal{W}}} \left\{ p(\mathcal{O}|\tilde{\mathcal{W}},\lambda) P(\tilde{\mathcal{W}}) \right\}. \end{aligned} \tag{2.29}$$

The word sequence with the highest posterior probability given the observation sequence and model parameters is selected. As discussed in section 1.1, a speech recognition system may be split into three components: the acoustic model based on HMMs, the pronunciation model and the language model. A word sequence may have more than one phone representations associated with it, due to the presence of multiple pronunciation variants. Meanwhile a sequence of HMM phone models may correspond to more than one possible hidden state sequences. Hence equation 2.29 may be expanded as a hierarchical marginalization over all possible sequences of HMM models $\{\boldsymbol{\theta}\}$ given a sequence of words, and then all possible sequences of hidden states $\{\boldsymbol{\psi}\}$ given a sequence of HMMs.

$$\mathcal{W} = \arg\max_{\tilde{\mathcal{W}}} \left\{ P(\tilde{\mathcal{W}}) \sum_{\boldsymbol{\theta}} P(\boldsymbol{\theta}|\tilde{\mathcal{W}}) \sum_{\boldsymbol{\psi}} p(\mathcal{O},\boldsymbol{\psi}|\boldsymbol{\theta},\tilde{\mathcal{W}},\lambda) \right\} \tag{2.30}$$

Here the prior probability of a word sequence, $P(\tilde{\mathcal{W}})$, is given by the language model. The conditional probability of a HMM model sequence given a string of words, $P(\boldsymbol{\theta}|\tilde{\mathcal{W}})$, is provided by the pronunciation model, and the joint conditional probability of an observation sequence and a state sequence, $p(\mathcal{O},\boldsymbol{\psi}|\boldsymbol{\theta},\tilde{\mathcal{W}},\lambda)$, is determined by the acoustic model.

Direct evaluation of equation 2.30 is very expensive and rapidly becomes impractical as the sentence length increases. To overcome this problem, the summation over all HMMs and state sequences may be approximated by a maximum.

$$\mathcal{W} = \arg\max_{\tilde{\mathcal{W}}} \left\{ P(\tilde{\mathcal{W}}) \max_{\boldsymbol{\theta}} P(\boldsymbol{\theta}|\tilde{\mathcal{W}}) \max_{\boldsymbol{\psi}} p(\mathcal{O},\boldsymbol{\psi}|\boldsymbol{\theta},\tilde{\mathcal{W}},\lambda) \right\} \tag{2.31}$$

The selection of the most likely word sequence is based on the ML state sequence.

Finding the ML state sequence for an HMM using equation 2.31 is realized via the *Viterbi* algorithm [57]. Let $\phi_j(\tau)$ denote the maximum likelihood of the partial observation sequence, $\{\boldsymbol{o}_1, ..., \boldsymbol{o}_\tau\}$, staying in state j at time instance τ. $\phi_j(\tau)$ may be computed using the following recursion

$$\begin{aligned}\phi_j(\tau) &= \max_i \{\phi_i(\tau-1)a_{ij}\} b_j(\boldsymbol{o}_\tau) \\ \phi_{N_s}(\mathcal{T}) &= \max_i \{\phi_i(\tau-1)a_{iN_s}\} \end{aligned} \tag{2.32}$$

where N_s denotes the number of states in an HMM and

$$\begin{aligned} \phi_1(1) &= 1 \\ \phi_j(1) &= a_{1j} b_j(\boldsymbol{o}_1) \end{aligned} \tag{2.33}$$

for any state $1 < j < N_s$.

An implementation of the Viterbi algorithm for continuous speech recognition is the *token passing* algorithm [131]. Each state has one or more tokens associated with each time instance. The token contains a word-end link and the value of the partial likelihood $\phi_j(\tau)$. These tokens are updated for each time instance and the most likely token at the end of each HMM model is propagated onto all connecting models. At the end of the utterance, the token with the highest log probability can be traced back to give the most likely sequence of words. The number of connecting models will be considerably increased if phone models with long cross word contexts are used. Using a language model can also expand the size of the decoding network. This is because tokens can only be merged if the word histories are identical. If an N-gram language model is used, the word probabilities may depend on previous word histories and there must be a separate path through the network for each distinct word history. The search cost may be reduced by *pruning*, or removing the tokens which fall below a given threshold. The threshold, or *beam-width*, is set as a certain likelihood loss below the current most likely path. All active tokens with a likelihood below that level will be deleted. Pruning may also be performed at the end of words when the language model is applied with a more punitive threshold. If the pruning beam-width is too tight, the most likely path could be pruned before the token reaches the end of the utterance. This will result in a search error. The choice of pruning beam-width is a trade off between avoiding search errors and reducing the computational cost. The efficient implementation of large vocabulary decoders is in active research.

One problem with the use of language and pronunciation models is that there is a considerable mismatch between the dynamic ranges of those two models and the acoustic model. This is partly because the probabilities from the acoustic model can often be very small due to the assumptions of HMMs as described in section 2.1.1. To handle this problem, the language and pronunciation model probabilities are scaled. The scaling factor may be empirically set and fixed for a particular task. Another related issue is the use of word insertion penalties. They penalize a higher number of words in a sentence. This is desirable as a significant proportion of recognition errors stem from the insertion of short words. These short words tend to have higher acoustic likelihood and frequencies of occurrence in the text copora used for language model training. Similar to the language model and pronunciation probability scaling, insertion penalties can be manually tuned to improve the balance of word insertions versus deletions on specific tasks. Now equation 2.31 may be modified as

$$\mathcal{W} \;=\; \arg\max_{\tilde{\mathcal{W}}} \left\{ \alpha \log P(\tilde{\mathcal{W}}) + \beta \max_{\boldsymbol{\theta}} \log p(\boldsymbol{\theta}|\tilde{\mathcal{W}}) \; + \max_{\boldsymbol{\psi}} \log p(\mathcal{O}, \boldsymbol{\psi}|\boldsymbol{\theta}, \tilde{\mathcal{W}}, \lambda) + \gamma L \right\} \tag{2.34}$$

where α and β are the language model and pronunciation probability scaling factors, γ the word insertion penalty, and L is the length of word sequence $\tilde{\mathcal{W}}$.

2.4 Linear Projection Schemes

For any pattern recognition task it is important to derive a good, compact, feature representation. The feature set should contain sufficient discriminant information to distinguish between classes. Features consisting of non-discriminating information should be removed. As discussed in section 2.3.1, current speech recognition systems often use 39 dimensional MFCC or PLP cepstral features including dynamic parameters. Although they have been widely adopted in current speech recognition systems, it is still unclear whether such a feature representation is the best choice. First, the use of dynamic features, computed using equation 2.24, further introduces correlation between static and dynamic coefficients in the acoustic space. Second, the correlation between low, and high order cepstral coefficients is not completely removed after the DCT transform is applied [77]. Hence it is preferable to appropriately model this correlation.

Various techniques for this purpose have been proposed over the years. They can be roughly classified into two main categories: covariance modeling and linear projection schemes. In *covariance modeling*, or *precision matrix modeling*, various tying of covariance parameters are used to allow Gaussian components to effectively have full covariance matrices without dramatically increase the model complexity [31, 45, 99, 98, 108]. In *linear projection* schemes, the original acoustic space is projected into one or more un-correlated subspaces. Within each subspace, diagonal Gaussian covariances may still be used. In this section several forms of linear subspace projection schemes are briefly reviewed. They are discussed within the linear discriminant analysis (LDA) framework.

2.4.1 Linear Discriminant Analysis

Linear discriminant analysis (LDA) is a standard dimensionality reduction scheme [26, 121]. A $p \times n$ linear transform $\boldsymbol{A}_{[p]}$ projects the original n dimensional feature space to a lower dimensional, uncorrelated subspace. The projected feature vector, $\check{\boldsymbol{o}}_{\tau[p]}$, is given by

$$\check{\boldsymbol{o}}_{\tau[p]} = \boldsymbol{A}_{[p]}\boldsymbol{o}_{\tau} \tag{2.35}$$

The matrix transform $\boldsymbol{A}_{[p]}$ is estimated by maximizing the ratios of the projected between class covariance, $\boldsymbol{B}$, and the average within class covariance $\boldsymbol{\Sigma}$.

$$\hat{\boldsymbol{A}}_{[p]\texttt{lda}} = \arg\max_{\boldsymbol{A}_{[p]}} \left\{ \frac{\left|\texttt{diag}\left(\boldsymbol{A}_{[p]}\boldsymbol{B}\boldsymbol{A}_{[p]}^{\top}\right)\right|}{\left|\texttt{diag}\left(\boldsymbol{A}_{[p]}\boldsymbol{\Sigma}\boldsymbol{A}_{[p]}^{\top}\right)\right|} \right\} \tag{2.36}$$

Both the between class, $\boldsymbol{B}$, and within class covariance, $\boldsymbol{\Sigma}$, are constrained to be diagonal in the projected subspace. For HMM based speech recognition systems, the definition of a “class” may correspond either to individual states or Gaussian components. In this work, Gaussian components are considered as classes. The between class covariance, $\boldsymbol{B}$, is then computed as the average distance between the global and component specific means. The within class covariance, $\boldsymbol{\Sigma}$, is computed as the average of component specific full covariances.

It can be shown that a closed form solution for the LDA transform is the Eigen vectors associated with top p Eigen values of $\boldsymbol{\Sigma}^{-1}\boldsymbol{B}$ [26, 121]. A maximum likelihood based estimation of LDA was proposed in [10]. The ML estimation of LDA requires optimizing a $n \times n$ square linear transform $\boldsymbol{A}$. Using this transform, the complete acoustic space is partitioned into two parts: a *useful* subspace associated with $\boldsymbol{A}_{[p]}$ and a nuisance subspace associated with $\boldsymbol{A}_{[n-p]}$, where Gaussian means and diagonal covariances are globally tied. This is given by

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{A}_{[p]} \\ \boldsymbol{A}_{[n-p]} \end{bmatrix} \tag{2.37}$$

and the transformed acoustic vector in the complete feature space, $\check{\boldsymbol{o}}_\tau$, may be expressed as the following

$$\check{\boldsymbol{o}}_\tau = \begin{bmatrix} \boldsymbol{A}_{[p]}\boldsymbol{o}_\tau \\ \boldsymbol{A}_{[n-p]}\boldsymbol{o}_\tau \end{bmatrix}. \tag{2.38}$$

The ML estimation of the LDA transform, $\boldsymbol{A}$, requires maximizing

$$\hat{\boldsymbol{A}}_{\texttt{lda}} = \arg\max_{\boldsymbol{A}} \left\{ \sum_{j,m,\tau} \gamma_{jm}(\tau) \left(\log|\boldsymbol{A}|^2 - \log\left|\check{\boldsymbol{\Sigma}}\right| \right) \right\} \tag{2.39}$$

where $\gamma_{jm}(\tau)$ is the Gaussian posterior occupancy given in equation 2.18, and $\check{\boldsymbol{\Sigma}}$ is the transformed average within class covariance in the complete feature space of $\boldsymbol{A}$. For LDA, $\check{\boldsymbol{\Sigma}}$ is constrained to be diagonal.

Gaussian likelihood calculation is efficient for LDA in the projected subspace of $\boldsymbol{A}_{[p]}$, as the Jacobian of the global transform may be ignored. However LDA suffers from a strong assumption that the within class covariances for all components are restricted to be the same. This assumption may be too strong for LVCSR systems which contain thousands of Gaussian components.

2.4.2 Heteroscedastic LDA

The uniform within class covariance assumption of standard LDA is strong. It may be a poor assumption for speech recognition systems containing a large number of Gaussian mixture components. To overcome this problem, two forms of heteroscedastic extensions to standard LDA have been proposed in recent years.

The first is an intuitive extension of the LDA objective function given in equation 2.36, allowing the within class covariances to vary across Gaussian components. This is referred to as the heteroscedastic discriminant analysis (HDA) [102]. The HDA projection is estimated by optimizing the following objective function

$$\hat{\boldsymbol{A}}_{[p]\texttt{hda}} = \arg\max_{\boldsymbol{A}_{[p]}} \left\{ \sum_{j,m,\tau} \gamma_{jm}(\tau) \log \left(\frac{\left|\texttt{diag}\left(\boldsymbol{A}_{[p]}\boldsymbol{B}\boldsymbol{A}_{[p]}^\top\right)\right|}{\left|\texttt{diag}\left(\boldsymbol{A}_{[p]}\boldsymbol{\Sigma}^{(jm)}\boldsymbol{A}_{[p]}^\top\right)\right|} \right) \right\}. \tag{2.40}$$

Compared with the LDA objective function in equation 2.36, the average within class covariance, $\boldsymbol{\Sigma}$, is replaced by Gaussian component specific covariances, $\boldsymbol{\Sigma}^{(jm)}$. Hence the uniform within class covariance assumption is removed. Unfortunately, HDA does not have a maximum likelihood interpretation like LDA. This is because the Jacobian normalization term for the HDA projection, $\left|\boldsymbol{A}_{[p]}\right|$, can not be computed for likelihood calculation. Hence there is no simple EM based optimization scheme for HDA. Numerical methods must be used to estimate the transform parameters. This can be very expensive due to the iterative computation of the objective function and its gradient [73].

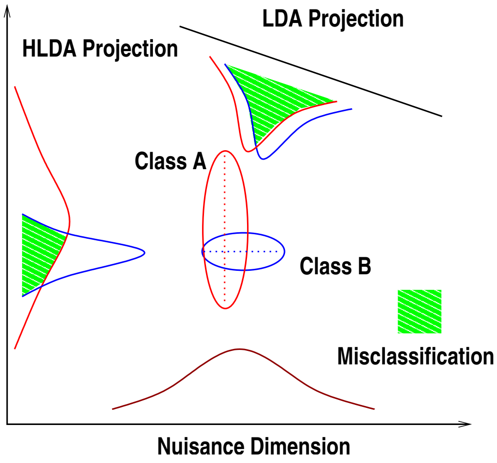

Figure 2.5 *HLDA and LDA projection*

Another form of heteroscedastic extension to standard LDA is the heteroscedastic linear discriminant analysis (HLDA) [66]. This method is widely used in LVCSR systems training [51, 64, 23, 127]. In contrast to HDA, HLDA has an ML interpretation and an efficient EM based optimization scheme is available [31]. As with the ML interpretation of LDA in section 2.4.1, HLDA may be viewed as a square, $n \times n$, linear transform. The complete acoustic space is also partitioned into two parts. The difference from LDA is that in the useful subspace means and diagonal covariances are Gaussian component specific. In figure 2.5 an example of HLDA is shown. Under the uniform within class variance assumption, the standard LDA chooses a projection in which the between class confusion is considerably stronger than HLDA.

The HLDA transform parameters are estimated by maximizing the following objective function

$$\hat{\boldsymbol{A}}_{\texttt{hlda}} = \arg\max_{\boldsymbol{A}} \left\{ \sum_{j,m,\tau} \gamma_{jm}(\tau) \left(\log|\boldsymbol{A}|^2 - \log\left|\check{\boldsymbol{\Sigma}}^{(jm)}\right| \right) \right\} \tag{2.41}$$

where $\check{\boldsymbol{\Sigma}}^{(jm)}$ is the transformed component covariances in the complete feature space of $\boldsymbol{A}$. For HLDA, again $\check{\boldsymbol{\Sigma}}^{(jm)}$ is constrained to be diagonal. HLDA is closely related to semi-tied covariances (STC) [31]. The two are equivalent to one another when the STC transform is globally shared and all feature dimensions are retained by HLDA. An efficient iterative optimization scheme proposed for STC may also be used maximize the objective function for HLDA [31, 34]. For both LDA and HLDA, the number of useful dimensions retained significantly affects the overall complexity of the underlying HMM system. This is an important issue that must be resolved using appropriate complexity control techniques.

2.4.3 Multiple Subspace Projection Schemes

For complex patterns, such as human speech, multiple sets of feature representation may be required to incorporate more class specific information. The acoustic realization of speech signals may be better modeled in different subspaces, depending on whether, for instance, a vowel or constant is generated. This is particularly important for state-of-the-art LVCSR systems. Context dependent phone models and a large number of Gaussian components are typically used in these systems, as discussed in section 2.3.2. Therefore a local projection of the speech signals may yield performance gains over a global one. For multiple linear projection schemes, it is important that the likelihood calculation in different subspaces be directly comparable. The Jacobian terms associated with each projection must be computed for this purpose. Unfortunately, HDA does not have an ML interpretation because a non-square linear projection is used, as discussed in section 2.4.2. Hence HDA can not be extended to have multiple projections. In contrast, an ML interpretation is available for both LDA and HLDA. Since a square linear transform is used, the likelihood calculation may be performed in the complete feature space. Both standard LDA and HLDA may be extended to have multiple projections that are shared locally [34] and are referred to as multiple HLDA and multiple LDA.

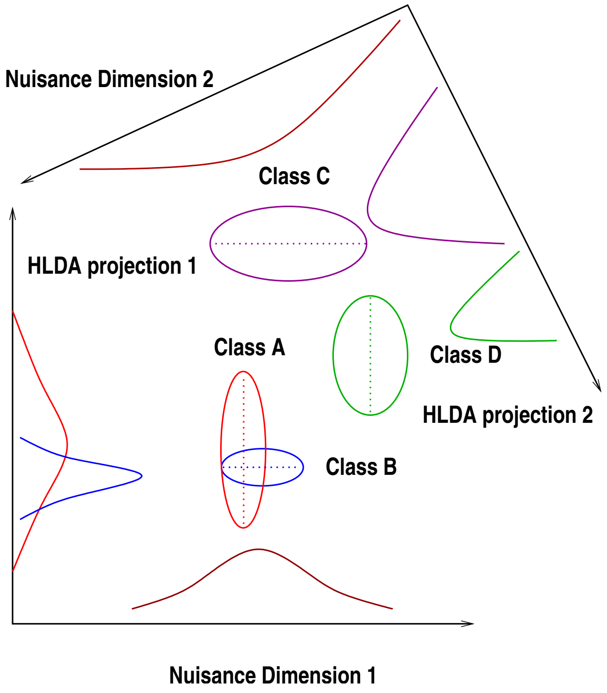

Figure 2.6 *multiple HLDA projections*

Multiple HLDA is a simple extension to standard HLDA. It allows multiple useful and nuisance subspaces to be locally shared in the system. Let $\boldsymbol{A}^{(r)}$ denote the rth HLDA projection, and the transformed feature vector, $\check{\boldsymbol{o}}_{\tau}^{(r)}$, is given by

$$\check{\boldsymbol{o}}_{\tau}^{(r)} = \begin{bmatrix} \boldsymbol{A}_{[p]}^{(r)}\boldsymbol{o}_{\tau} \\ \boldsymbol{A}_{[n-p]}^{(r)}\boldsymbol{o}_{\tau} \end{bmatrix}. \tag{2.42}$$

An example of multiple HLDA is shown in figure 2.6. In the figure there are two HLDA projections. Model parameters in both the useful and nuisance subspaces are locally shared. The presence of multiple nuisance subspaces means that likelihood calculation for the nuisance dimensions can not be discarded as in standard LDA or HLDA.

The estimation of multiple HLDA transforms requires maximizing the following objective function

$$\hat{\boldsymbol{A}}_{\texttt{mhlda}}^{(r)} = \arg\max_{\boldsymbol{A}^{(r)}} \left\{ \sum_{j,m\in r,\tau} \gamma_{jm}(\tau) \left(\log\left|\boldsymbol{A}^{(r)}\right|^{2} - \log\left|\check{\boldsymbol{\Sigma}}^{(jm)}\right| \right) \right\} \tag{2.43}$$

where $j, m \in r$ denotes that component m of state j is assigned to projection r. As the Jacobian normalization terms are different across projections, they can no longer be ignored during likelihood calculation. For component m of state j, this is given by

$$p(\boldsymbol{o}_{\tau}|\boldsymbol{\psi}_{\tau} = \mathcal{S}_{j,m}, \lambda) = \left|\boldsymbol{A}^{(r)}\right| \mathcal{N}\left(\boldsymbol{A}^{(r)}\boldsymbol{o}_{\tau}; \check{\boldsymbol{\mu}}^{(jm)}, \check{\boldsymbol{\Sigma}}^{(jm)}\right) \tag{2.44}$$

where $\check{\boldsymbol{\mu}}^{(jm)}$ is transformed Gaussian means in the complete feature space of $\boldsymbol{A}^{(r)}$. The number of Gaussian parameters in an multiple HLDA system may be computed as $\sum_{r}(n^{2}+2N_{r}p_{r}+2(n-p_{r}))$, where N_{r} denotes the number of Gaussians assigned to projection r, and p_{r} the number of useful dimensions for projection r.

The same EM based iterative optimization scheme for standard HLDA may also be used to estimate multiple HLDA transforms on a projection by projection basis. Multiple HLDA also has a structural flexibility as the useful subspace dimensionality may be varied locally across projections. Again the number of useful dimensions for each projection significantly affects the overall system complexity. This important issue must be resolved by a appropriate model complexity control scheme. In addition, it may be argued that multiple HLDA is not a “true” projection scheme as the nuisance subspace parameters are still needed for likelihood calculation.

In contrast, multiple LDA is a “true” multiple projection scheme. Its difference from multiple HLDA is that there is only one globally tied nuisance subspace, despite multiple projections are used. For multiple LDA, the transformed feature vector, $\check{\boldsymbol{o}}_{\tau}^{(r)}$, of the r the projection is given by

$$\check{\boldsymbol{o}}_{\tau}^{(r)} = \begin{bmatrix} \boldsymbol{A}_{[p]}^{(r)}\boldsymbol{o}_{\tau} \\ \boldsymbol{A}_{[n-p]}\boldsymbol{o}_{\tau} \end{bmatrix} \tag{2.45}$$

where model parameters in the single nuisance subspace, $\boldsymbol{A}_{[n-p]}$, are globally tied.

An example of multiple LDA is shown in figure 2.7. In the figure there are two LDA projection. Only one global nuisance subspace is available and is shared between the two projections.

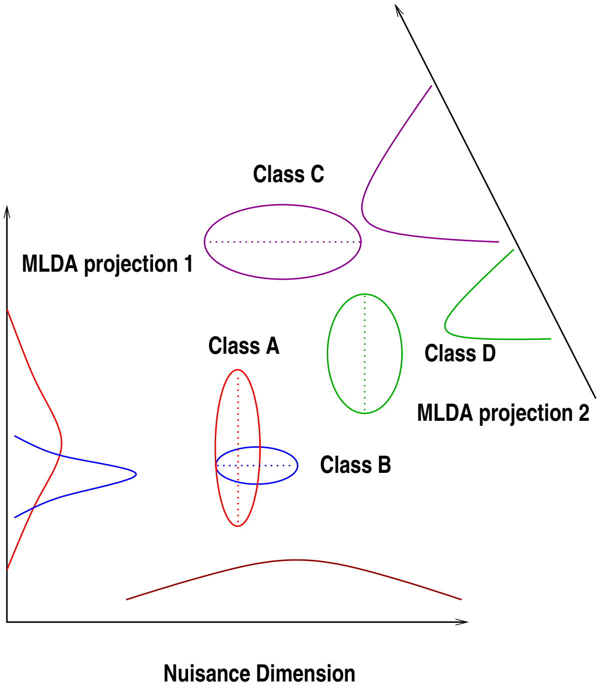

Figure 2.7 *multiple LDA projections*

For multiple LDA systems, the likelihood calculation in the nuisance subspace may be ignored, as it remains constant for all Gaussian components.

Unfortunately, there no efficient optimization scheme for multiple LDA due to the fact that model parameters are globally tied in the nuisance subspace. Numerical methods may be used to optimize the projections. Furthermore, multiple LDA does not have the flexibility of multiple HLDA in locally varying the useful subspace dimensionality. It was reported that multiple LDA was outperformed by both multiple HLDA and STC using ML training on an LVCSR task [34].

One important issue when using multiple projections is the appropriate tying of transform parameters. They may be loosely tied on state level as in [31, 34] or on HMM model level using phonetic expert knowledge. Alternatively, data driven methods may be used to cluster Gaussians into groups for each projection, based on distance measuring of Gaussian components in the acoustic space. By using this assignment scheme significant WER reduction over a single projection was reported on an LVCSR task in [70]. This distance measuring based assignment scheme was originally proposed for linear transformation based speaker adaptation techniques [68, 30] and is further discussed in the next section.

2.5 Speaker Adaptation

Characteristics of speech signals vary substantially depending on the speaker and acoustic environment. Models trained on speaker specific data outperform those trained on speaker independent data. Speaker independent (SI) systems may be adapted to the characteristics of a target speaker or environment. This approach is referred to as *speaker adaptation* and is widely used in state-of-the-art LVCSR systems [125, 51].

One approach for building speaker dependent (SD) models is *maximum a posteriori* (MAP) [36]. This technique allows prior knowledge about HMM parameters to be incorporated into parameter estimation. SI model parameters, for example, may be used as the parameter priors. Model parameters are gradually updated using speaker dependent data toward the target speaker. MAP training may be viewed as a parameter smoothing scheme where the parameter posterior is a combination of the prior and the ML estimates. In case of insufficient data the posterior distribution is close to the prior. The MAP estimates tend to the ML estimates as the amount of training data is increased. One limitation with MAP training is that a large quantity of speaker or environment specific data is required to adapt all the parameters in the system.

Maximum likelihood linear regression (MLLR) is another model based adaptation scheme [68, 30]. The speaker specific information is represented by one or more linear transformations that are applied to the model parameters. The advantage of this method over MAP is that rapid adaptation may be performed using a small amount of speaker specific enrollment data. For instance, the adapted Gaussian mean, $\check{\boldsymbol{\mu}}^{(jm)}$, of component m and state j, may be expressed as

$$\check{\boldsymbol{\mu}}^{(jm)} \quad = \quad \boldsymbol{W}^{(r_{jm})}\boldsymbol{\zeta}^{(jm)} \tag{2.46}$$

where $\boldsymbol{W}^{(r_{jm})}$ is a $n \times (n+1)$ linear transform assigned to component m of state j, and $\boldsymbol{\zeta}^{(jm)} = \begin{bmatrix}\boldsymbol{\mu}^{(jm)\top} & 1\end{bmatrix}^\top$ is the extended mean vector. The transform parameters are optimized using the EM algorithm with adaptation data from the target speaker. The ith row of the extended transform matrix, $\boldsymbol{w}_i^{r_{jm}}$, can be estimated as [68],

$$\hat{\boldsymbol{w}}_i^{(r_{jm})} \quad = \quad \boldsymbol{G}^{(r_{jm},i)-1}\boldsymbol{k}^{(r_{jm},i)} \tag{2.47}$$

and the sufficient statistics $\boldsymbol{G}^{(r_{jm},i)}$ and $\boldsymbol{k}^{(r_{jm},i)}$ are accumulated on a row by row basis,

$$\begin{aligned}\boldsymbol{G}^{(r_{jm},i)} \quad &= \quad \sum_{j,m\in r_{jm},\tau} \gamma_{jm}(\tau)\frac{\boldsymbol{\zeta}^{(jm)}\boldsymbol{\zeta}^{(jm)\top}}{\sigma_i^{(jm)2}} \\ \boldsymbol{k}^{(r_{jm},i)} \quad &= \quad \sum_{j,m\in r_{jm},\tau} \gamma_{jm}(\tau)\frac{o_{\tau i}\boldsymbol{\zeta}^{(jm)}}{\sigma_i^{(jm)2}}\end{aligned} \tag{2.48}$$

where $\sigma_i^{(jm)2}$ is the ith dimensional diagonal variance element of the mth component and the jth state. The above estimation formulas are only valid for systems using diagonal covariances. For systems using full covariance Gaussians, the transform estimation requires inverting an $(n^2 + n) \times (n^2 + n)$ matrix and hence is computationally expensive. A detailed derivation of transform estimation for this case was proposed in [68].

A globally tied MLLR transform may be applied to all the components in the system. To further improve the performance, the number of MLLR transforms may be increased as long as enough adaptation data is available. To determine the number of transforms and assign the components to these transform classes, a regression class tree is often used [28]. A binary regression class tree is constructed to cluster Gaussian components that are close in the acoustic space. This clustering method may also be used for the assignment of Gaussian components for multiple projections discussed in section 2.4.3. The number of transforms, or equivalently the tree cut, is determined by a manually tuned threshold of occupancy counts for each tree node. The regression tree is constructed using a top-down procedure. Creation of a children tree node is considered if the occupancy count assigned to it is above a pre-defined threshold. The tree construction is complete when there is no children tree node to be created. A simple example of a regression tree is shown in figure 2.8. The root node corresponds to all the components being assigned to a global MLLR transform. In the figure, nodes 6 and 7, for instance, do not have sufficient data, and the transform estimation is backed-off to the statistics of parental node 3. In contrast, there is sufficient data available for leaf node 4, and a distinct MLLR transform will be generated. The final number of transform classes in this example is three.

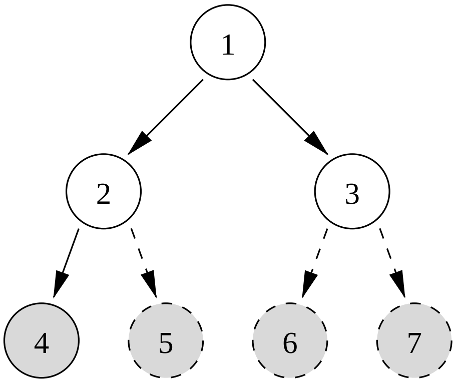

Figure 2.8 *Example of a binary regression tree for MLLR speaker adaptation.*

2.6 Adapting Multiple HLDA Systems

As discussed in section 2.5, in order to compensate for the speaker and environment variation, standard adaptation techniques like MLLR may be used. However, for the systems using multiple linear projection schemes, such as multiple HLDA discussed in section 2.4.3, there is one issue with using MLLR. This is due to the presence of multiple feature subspaces. In earlier research adapting Gaussian parameters within individual subspaces, referred to as normalized domain MLLR in [29], was found to yield poor recognition performance. To overcome this problem, the approach adopted in this work is to estimate the MLLR mean transforms in the original acoustic space. Hence the sufficient statistics for transform estimation, given in equation 2.48, will be accumulated in the standard feature space prior to linear projections. A matrix inverse operation is required to yield the un-projected component parameters from individual subspaces. For efficiency when estimating the MLLR transforms, a diagonal approximation to the covariance in the original feature space is also used. The Gaussian means and covariances in the original space are computed as below.

$$\begin{aligned}\boldsymbol{\mu}^{(jm)} &= \boldsymbol{A}^{(r_{jm})-1}\check{\boldsymbol{\mu}}^{(jm)} \\ \boldsymbol{\Sigma}^{(jm)} &\approx \texttt{diag}\left(\boldsymbol{A}^{(r_{jm})-1}\check{\boldsymbol{\Sigma}}^{(jm)}\boldsymbol{A}^{(r_{jm})-1\top}\right)\end{aligned} \tag{2.49}$$

After the MLLR transforms are estimated, the adapted component means in the original space are then projected back into individual subspaces for each projection. This allows systems using multiple linear projections to be efficiently adapted.

2.7 Summary

The statistical framework for automatic speech recognition systems was outlined in this chapter. First, hidden Markov models were discussed as acoustic models. The optimization of HMM parameters was presented using ML training. Then the standard feature extraction schemes were briefly reviewed for HMM based speech recognition systems. The selection of recognition units and parameter tying were also discussed. Language modeling and pronunciation modeling approaches were outlined, along together with the basic search algorithm used in a state-of-the-art large vocabulary decoder. This was followed by a brief review of linear projection schemes under the framework of linear discriminant analysis. Finally, popular speaker adaptation techniques were briefly described.

3

Model Complexity Control

A standard problem in LVCSR training, and machine learning in general, is how to select a model structure that generalizes well to unseen data. Model structures which are too simple lack the power to fully represent the observed data. On the other hand, structures that are too complex do not generalize well and yield poor performance on unseen data. This chapter presents a survey of techniques to control model complexity. First, word error rate (WER) is introduced as a "golden" complexity control criterion for most ASR tasks. Then existing complexity control techniques are presented. These schemes are classified into two broad categories: Bayesian learning techniques and information theory methods. A survey of previous applications of these techniques to speech recognition is also given. Finally, the limitations of likelihood based complexity control schemes is discussed.

3.1 WER - A Zero Risk Criterion

The aim of model complexity control is to select the optimal number of parameters to train to achieve good generalization to unseen data. For speech recognition the generalization to the unseen test data, $\mathcal{D}$, is commonly measured by the word error rate (WER). Hence, for the majority of speech recognition tasks the aim of model complexity control is to achieve a minimum WER on the unseen data. A good complexity control technique should predict the correct WER performance ranking for all systems with a range of configurations. Therefore WER is a "golden" complexity control criterion with zero ranking risk, since the ordering according to WER is the correct ranking. For speech recognition the task is to select of an optimal structural configuration, $\hat{\mathcal{M}}$, with a minimum WER on unseen data, from a set of candidate models, $\{\mathcal{M}\}$, given a $\mathcal{T}$ length training data set, $\mathcal{O} = \{\boldsymbol{o}_1, ..., \boldsymbol{o}_\mathcal{T}\}$, and the reference transcription $\mathcal{W}$ [122, 121]. However, WER is difficult to directly measure for highly complex state-of-the-art LVCSR systems. A wide range of techniques are currently used which alter the system complexity and WER. Examples of these techniques include the use of mixtures of Gaussians as state distributions, dimensionality reduction schemes, decision tree based state tying and linear transform based speaker adaptation. For current LVCSR systems, explicitly building and evaluating systems with various structural configurations to access WER is infeasible. Therefore, automatic complexity control techniques are needed so that individual systems are not required to be built and evaluated.

3.2 Likelihood Based Model Complexity Control

Standard complexity control schemes do not require direct measurement of the WER for each candidate structure. Instead an inherent model correctness assumption is made. All candidate model structures are assumed to be "close" to the correct model for speech signals. Thus increasing the likelihood on the unseen data will decrease the systems' WER. Under this assumption, likelihood validation test may be used as an alternative to directly accessing WER [122, 121]. When performing likelihood validation test, the optimal model parameters are normally estimated using either the *maximum likelihood* (ML) or *maximum a posteriori* (MAP) criterion. Discriminative training criteria, such as the *maximum mutual information* (MMI) criterion [3], may also be used. However, in most statistical inference literature, the "optimal" model parameters are trained using the ML or MAP criterion. This is the case considered in this chapter. Using the likelihood held-out data set, $\mathcal{D}$, the model selection is based on the following:

$$\hat{\mathcal{M}} \quad = \quad \arg\max_{\mathcal{M}} \left\{ p(\mathcal{D}|\hat{\lambda}, \mathcal{W}, \mathcal{M}) p(\hat{\lambda}|\mathcal{M}) P(\mathcal{M}) \right\} \tag{3.1}$$

where $\hat{\lambda}$ denotes the optimal parameter estimates. One issue with this method is that the training of individual systems is still required. State-of-the-art LVCSR systems are highly complex. Hence explicitly building all possible systems for held-out likelihood test is infeasible. Another issue is how to appropriately determine the the size of the held-out data set. In the statistical inference literature, the power of a likelihood validation test is increased as the held-out data size grows, when measured with a fixed level of statistical significance [122]. However, the computational cost for validation test also increases as the amount of held-out data is increased. Using a large held-out data set will reduce the amount of training data available. Furthermore, it is a non-trivial problem to evaluate the reliability of the selected held-out data.

To overcome this problem many complexity control techniques make use of only the training data. It is assumed that there is a strong correlation between the unseen data likelihood and the training data *marginal* likelihood, given a particular model structure. These schemes may be further classified into two major categories. In *Bayesian learning* techniques, model parameters are treated as random variables and integrated out in the parametric space. In *information theory* approaches, the complexity control problem is viewed as finding an appropriate code length [6]. These two approaches are closely related to each other. Both can be explicitly expressed as the training data marginal likelihood given a model structure and asymptotically tend to the Bayesian Information Criterion (BIC) approximation [104]. In the following sections these two categories of complexity control schemes are presented. Some inherent assumptions made by these schemes and their limitations are also discussed.

3.3 Bayesian Techniques

In Bayesian complexity control techniques, it is assumed that the the training data marginal likelihood over model parameters is strongly correlated with the unseen data likelihood. A Bayesian model selection is based on

$$\begin{aligned}\hat{\mathcal{M}} &= \arg\max_{\mathcal{M}} \left\{ P(\mathcal{M}) \int p(\mathcal{O}|\lambda, \mathcal{W}, \mathcal{M}) p(\lambda|\mathcal{M}) d\lambda \right\} \\ &= \arg\max_{\mathcal{M}} \left\{ P(\mathcal{M}) p(\mathcal{O}|\mathcal{W}, \mathcal{M}) \right\} \qquad (3.2)\end{aligned}$$

where λ denotes a parameterization of $\mathcal{M}$, and $p(\mathcal{O}|\mathcal{W}, \mathcal{M})$ is referred to as the Bayesian *evidence* in the literature.

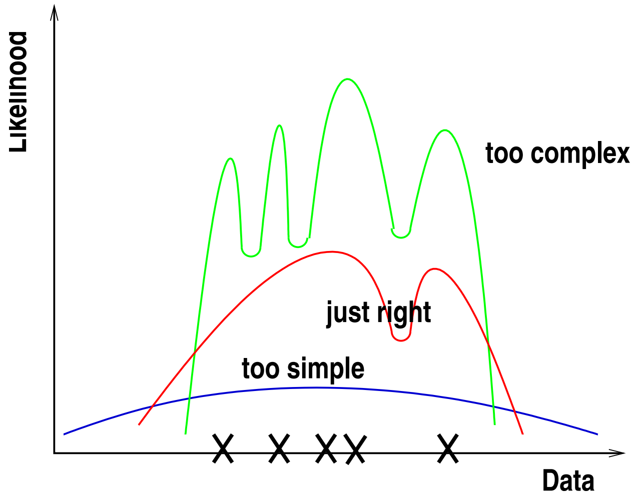

Figure 3.1 *Three model structures with different complexity*

In equation 3.2, $p(\lambda|\mathcal{M})$ and $P(\mathcal{M})$ are the prior distribution of a set of model parameters, λ, and the prior distribution of a particular model structure $\mathcal{M}$. For Bayesian evidence, the selection of a good form of the parameter prior distribution, $p(\lambda|\mathcal{M})$, is a subjective process. Often simplifying assumptions are made about this distribution, which typically constrain it to be a conjugate prior distribution for $p(\mathcal{O}|\lambda, \mathcal{W}, \mathcal{M})$. Under these assumptions the evidence integration may be more tractable [39, 59, 118]. Commonly used forms are the exponential family, such as Gaussian, Gamma and Dirichlet distributions. However, due to the lack of knowledge about the underlying distribution and number of parameters, the parameter prior, $p(\lambda|\mathcal{M})$, is assumed to be uninformative in this work.

If also assuming there is no prior information given by $P(\mathcal{M})$, the optimal model is selected by evaluating the evidence integral for each candidate structure. The model parameters are treated as unknown random variables to be integrated out in the parametric space. By marginalizing over the parameters the model complexity may be controlled. Over-simple model structures are not powerful enough to model the observed data. On the other hand, over-complex model structures are penalized for allowing too much freedom in the parametric space. They are over-fitted to the observed data, which leads to bad generalization performance, despite modeling the training data well. This is shown in figure 3.1. The five observed data samples are represented using crosses along the horizontal axis. A Gaussian distribution, a two component and a 4 component GMMs are used as examples. The GMM with four component marked as "too complex" has been over-fitted to the data. It has more power in modeling the observed data, but generalizes poorly. In contrast the single Gaussian distribution marked as " too simple" has insufficient power to model the observed data. The two component GMM marked as "just right" has the optimal complexity among the three. It is capable of modeling a certain range of *interesting* observed or unseen data sets. It will give a high Bayesian evidence for that range of data sets but little for others. In a word, the simplest model structure that can sufficiently describe the observed data should be selected. This property of Bayesian evidence is often referred to as *Ockham's Razor* [122, 41].

Having simplified the forms of prior distributions, the evidence must be computed for model selection. For HMM based speech recognition systems, it is often computationally intractable to directly integrate out the marginal likelihood in equation 3.2. Appropriate approximation schemes are required to practically evaluate the Bayesian evidence. In the following sections four approximation schemes are discussed. These are a first order expansion using Bayesian information criterion (BIC), a second order Laplace's approximation, a lower bound approximation using EM or variational method and Markov chain Monte Carlo (MCMC) style sampling schemes.

3.3.1 Bayesian Information Criterion (BIC)

The Bayesian evidence integration in equation 3.2, may be asymptotically approximated via a Taylor series expansion around the parameter optimum $\hat{\lambda}$. When the number of training samples $\mathcal{T}$ becomes infinitely large, this gives the Bayesian Information Criterion (BIC) [104]. BIC is the most widely used approximation scheme for the evidence integral. This criterion can be simply expressed in terms of a penalized log likelihood evaluated at the ML or MAP estimate of model parameters $\hat{\lambda}$. The model selection is based on the following approximation,

$$\log p(\mathcal{O}|\mathcal{W},\mathcal{M}) \quad \approx \quad \log p(\mathcal{O}|\hat{\lambda},\mathcal{W},\mathcal{M}) - \rho \cdot \frac{k}{2} \log \mathcal{T} \tag{3.3}$$

where k denotes the number of free parameters in $\mathcal{M}$ and ρ is a penalization coefficient which may be tuned to specific tasks [15]. Schwartz originally proved that when $\rho = 1$, BIC is a first order asymptotic expansion of the log of the evidence integral in equation 3.2, under certain regular assumptions upon the density $p(\mathcal{O}|\lambda,\mathcal{W},\mathcal{M})$ [104]. In [15] it was suggested that the tuning of ρ may compensate for the higher order terms unaccounted for in the BIC expansion, and the temporal correlation of speech signals ignored by HMMs.

There are two issues to consider when BIC is used to approximate the Bayesian evidence. First, BIC is only a first order approximation to the Bayesian evidence. Under the large number assumption, higher order terms from the Taylor series expansion are ignored. However, when the amount of training data is "small" the BIC approximation to the Bayesian evidence becomes increasingly poor. In this case the higher order terms that have been ignored may actually contain important information about model complexity. Hence it would be preferable to have an approximation scheme that can incorporate more information from the higher order terms. The second issue with this method is that the complexity penalization term in equation 3.3, $\rho \cdot \frac{k}{2} \log \mathcal{T}$, does not account for the difference in terms of the form of model parameters. k represents only the total number of free parameters, regardless of their individual nature. In recent research this was found to be a limitation of the BIC metric when optimizing multiple complexity attributes of different forms [71]. This limitation was investigated on an LVCSR task in [71], in which both the number of Gaussian components per state and number of useful dimensions of an HLDA system were optimized. The BIC metric failed to select the appropriate model complexity.

3.3.2 Akaike Information Criterion (AIC)

Another approximation scheme, which is closely related to BIC, is the Akaike Information Criterion (AIC) [1]. AIC was originally developed from research work on hypothesis and significance test. Akaike also gave a Bayesian interpretation to AIC using a likelihood ratio test [1]. The criterion itself is a simple trade-off between the fitness to the observed data, and the number of free parameters in the system. The fitness to the observed data is again expressed in terms of log-likelihood, evaluated at the optimal parameter estimates. Model selection using the AIC criterion is based on the following:

$$\hat{\mathcal{M}} = \arg\max_{\mathcal{M}} \left\{ \log p(\mathcal{O}|\hat{\lambda}, \mathcal{W}, \mathcal{M}) - k \right\}. \tag{3.4}$$

For AIC the complexity penalization term is only associated with the total number of free parameters, k. Compared with BIC, AIC is a simpler complexity control criterion. No information about the size of the training data, $\mathcal{T}$, is accounted for in the AIC penalization term, k. In contrast to the complexity term of BIC in equation 3.3, $\frac{k}{2} \log \mathcal{T}$, AIC lacks of power in penalizing over-complex systems when the amount of training data is increased. Thus, for larger data sets AIC may favor more complex systems than BIC.

3.3.3 Laplace Approximation

To incorporate more information from the higher terms ignored in BIC , a second order Taylor series expansion for the Bayesian evidence may be used. This leads to the Laplace's approximation [122, 78]. The basic idea is to make a local Gaussian approximation of the likelihood curvature in the parametric space. The Gaussian mean is set to the optimum of the model parameters. These parameters are normally estimated using either ML or MAP criterion. The covariance matrix is set to the Hessian evaluated at the optimum of model parameters. The Hessian is also referred to as the *Fisher information* matrix in the statistical inference literature. The volume under that Gaussian is computed as an approximation to the evidence. The Bayesian evidence in equation 3.2 is then approximated as the following:

$$\log p(\mathcal{O}|\mathcal{W},\mathcal{M}) \approx \log p(\mathcal{O}|\hat{\lambda},\mathcal{W},\mathcal{M}) - \frac{1}{2}\log\left|-\nabla^2_{\lambda=\hat{\lambda}}\log p(\mathcal{O}|\lambda,\mathcal{W},\mathcal{M})\right| + \frac{k}{2}\log 2\pi. \quad (3.5)$$

Using this approximation, difference among forms of model parameters can be accounted for in the Hessian, $\nabla^2_{\lambda=\hat{\lambda}}\log p(\mathcal{O}|\lambda,\mathcal{W},\mathcal{M})$, of equation 3.5. A general example of the Laplace's approximation is shown in figure 3.2. A simple case is illustrated in the figure where the variable x only has one single dimension. A Gaussian distribution is fitted to an arbitary function curve, $f(x)$. The Gaussian mean is at the optimal estimate, $\hat{x}$, estimated using either ML or MAP criterion. Its variance is the second order derivative with respect to x, $-\nabla^2_{x=\hat{x}}\log f(x)$, which is also computed at the parameter optimum $\hat{x}$.

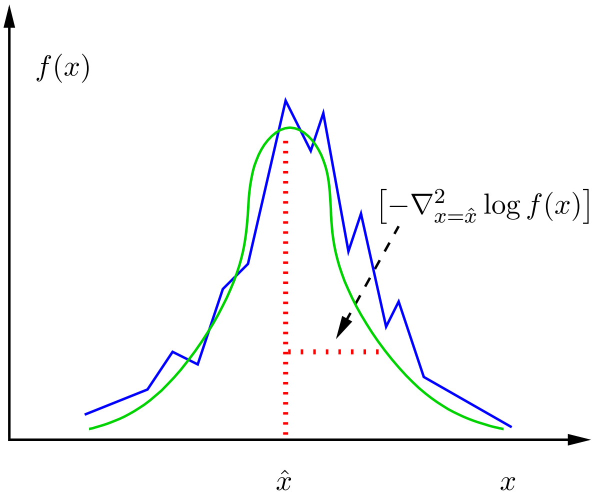

Figure 3.2 *Laplace Approximation*

For many practical situations it is infeasible to compute and store the Hessian as a full matrix when the number of parameters in the system, k, is far too large. In current LVCSR systems the number of model parameters can be in the millions. As the Hessian contain $\mathcal{O}(k^2)$ parameters, storing it as a complete matrix rapidly becomes infeasible as k increases. Therefore, for these systems a memory efficient approximation is required. One practical solution is to use a block diagonal approximation. It is assumed that model parameters belonging to different parts of the system, such as individual Gaussian components, are independent of one other. This may be expressed in equation 3.6,

$$\nabla^2_{\lambda=\hat{\lambda}}\log p(\mathcal{O}|\lambda,\mathcal{W},\mathcal{M}) = \begin{bmatrix} \ddots & & 0 \\ & \nabla^2_{\lambda^{(j)}=\hat{\lambda}^{(j)}}\log p(\mathcal{O}|\lambda,\mathcal{W},\mathcal{M}) & \\ 0 & & \ddots \end{bmatrix} \quad (3.6)$$

where $\lambda^{(j)}$ denotes the parameters of some Gaussian component j [1]. This is the approach adopted in this work and is addressed with more detail in later chapters. It should also be noted

[1] Gaussian components are treated as "hidden states" of HMMs in this work. For clarity in the rest of the thesis,

that under a large number assumption, when the number of training data samples $\mathcal{T}$ is infinitely large, Laplace's approximation tends to the same asymptotic expansion as BIC.

3.3.4 EM Method

One issue with both of the previous two approximation schemes is that the log-likelihood and optimal parameters for each model structure are required. For LVCSR tasks explicitly building all possible systems to obtain the log-likelihood is infeasible. One method to address this problem is to derive an appropriate lower bound for the ML criterion. Such a lower bound should be in a tractable form and marginalized over for complexity control, assuming it yields the same ranking as using the log-likelihood. Let $\tilde{\lambda}$ denote the *current* parameterization for $\mathcal{M}$ and $\{\psi\}$ the set of hidden state sequences allowed by the reference transcription $\mathcal{W}$. Using an expectation maximization (EM) approach [19], as described in section 2.2.1, a lower bound to the training data log-likelihood may be expressed as

$$\begin{aligned}\log p(\mathcal{O}|\lambda,\mathcal{W},\mathcal{M}) &\geq \log p(\mathcal{O}|\tilde{\lambda},\mathcal{W},\mathcal{M}) + \mathcal{Q}_{\mathtt{ml}}(\lambda,\tilde{\lambda}) - \mathcal{Q}_{\mathtt{ml}}(\tilde{\lambda},\tilde{\lambda}) \\ &= \mathcal{L}_{\mathtt{ml}}(\lambda,\tilde{\lambda}) \qquad (3.7)\end{aligned}$$

where the standard EM auxiliary function for HMMs is given by

$$\mathcal{Q}_{\mathtt{ml}}(\lambda,\tilde{\lambda}) = \sum_{j,\tau}\gamma_j(\tau)\log p(\boldsymbol{o}_\tau|\boldsymbol{\psi}_\tau=\mathcal{S}_j,\lambda,\mathcal{M}). \qquad (3.8)$$

$\boldsymbol{\psi}_\tau=\mathcal{S}_j$ indicates that an acoustic feature vector $\boldsymbol{o}_\tau$ was generated by state j at time instance τ, and the hidden state posterior

$$\gamma_j(\tau) = P(\boldsymbol{\psi}_\tau=\mathcal{S}_j|\mathcal{O},\mathcal{W},\tilde{\lambda},\mathcal{M}). \qquad (3.9)$$

To compute the above auxiliary function, the first and second order moments,

$$\begin{aligned}\sum_\tau\gamma_j(\tau)\boldsymbol{o}_\tau &= \sum_\tau P(\boldsymbol{\psi}_\tau=\mathcal{S}_j|\mathcal{O},\mathcal{W},\tilde{\lambda},\mathcal{M})\boldsymbol{o}_\tau \\ \sum_\tau\gamma_j(\tau)\boldsymbol{o}_\tau\boldsymbol{o}_\tau^\top &= \sum_\tau P(\boldsymbol{\psi}_\tau=\mathcal{S}_j|\mathcal{O},\mathcal{W},\tilde{\lambda},\mathcal{M})\boldsymbol{o}_\tau\boldsymbol{o}_\tau^\top, \qquad (3.10)\end{aligned}$$

are also required. Compared with the training data log-likelihood, the dependency upon latent variable sequences has been removed in $\mathcal{L}_{\mathtt{ml}}(\lambda,\tilde{\lambda})$. Thus the above lower bound has a more tractable form.

For LVCSR training the majority of the time is spent accumulating sufficient statistics to estimate the model parameters. Thus, accumulating these statistics for all possible systems is infeasible. To handle this problem, a range of model structures may be required to use information derived from the same set of statistics generated using a single system. For example when determining the number of components, statistics for systems with fewer components per

the notation j is used to denote a component. However, this should not be confused with those notations used earlier in chapter 2.

state may be derived by merging statistics defined in equations 3.9 and 3.10 together from a more complex system. For example, when merging Gaussian components l and k to form a new component j, the statistics given in equation 3.9 may be merged as $\gamma_j(\tau) = \gamma_l(\tau) + \gamma_k(\tau)$. This allows the lower bound in equation 3.7 to be efficiently computed. In fact this approach is also used in decision tree based state clustering, as discussed in section 2.3.2. When tree nodes are merged, the same statistics merging is performed among states with a single Gaussian. This efficient component merging process will be discussed in more details in chapter 5. The following lower bound for the evidence may then be used for model selection:

$$\log p(\mathcal{O}|\mathcal{W},\mathcal{M}) \quad \geq \quad \log \int \exp\left(\mathcal{L}_{\mathtt{ml}}(\lambda,\tilde{\lambda})\right) p(\lambda|\mathcal{M}) d\lambda. \tag{3.11}$$

Though the right hand side of inequality 3.11 may have a closed form solution, in many situations it is still impossible to compute. To further reduce the computational cost, the right hand side of the inequality in equation 3.11 may be efficiently approximated using numerical approximation schemes, such as Laplace's approximation.

One important feature of the lower bound marginalization in equation 3.11 is that it may be related to the integration of the ML auxiliary function in equation 3.8. The only term in the lower bound which is dependent on the model parameters, λ, is the auxiliary function $\mathcal{Q}_{\mathtt{ml}}(\lambda,\tilde{\lambda})$. When multiple model structures use the same set of statistics, $\{\gamma_j(\tau)\}$, the rank ordering derived from the marginalization of $\mathcal{L}_{\mathtt{ml}}(\lambda,\tilde{\lambda})$ is equivalent to the ranking of the integral over $\mathcal{Q}_{\mathtt{ml}}(\lambda,\tilde{\lambda})$. However, when multiple sets of statistics are used, the other terms in the lower bound, $\log p(\mathcal{O}|\tilde{\lambda},\mathcal{W},\mathcal{M})$ and $\mathcal{Q}_{\mathtt{ml}}(\tilde{\lambda},\tilde{\lambda})$, may vary. In this case they can longer be ignored and must be computed. Directly comparing of the marginalization of $\mathcal{Q}_{\mathtt{ml}}(\lambda,\tilde{\lambda})$ between model structures is not meaningful, unless they share the same set of statistics.

One basic assumption is made in the lower bound based approximation in equation 3.11. It is assumed that the ordering of the Bayesian evidence is the same as that of its lower bound. The looser the bound is, the poorer the approximation may become. For the EM lower bound given in equation 3.7, this means that aggressively sharing statistics among very different model structures may lead to a poor evidence approximation. Hence, when sharing statistics the complexity variation among model structures must be constrained to ensure the reliability of statistics, and the bound. This issue will be further discussed in detail in chapter 5.

3.3.5 Variational Method

The ML bound in equation 3.7 requires the the hidden state posterior, $P(\psi_\tau = \mathcal{S}_j|\mathcal{O},\mathcal{W},\tilde{\lambda},\mathcal{M})$. However, in many practical situations when more complicated forms of acoustic models are used this distribution is intractable. To handle this problem, another related approximation scheme, *variational approximation* [2, 38, 39], may be used. In a similar formula to the EM algorithm, Jensen's inequality is applied to derive an evidence lower bound. If the joint posterior distribution over both model parameters and hidden states, $P(\psi_\tau = \mathcal{S}_j, \lambda|\mathcal{O},\mathcal{W},\mathcal{M})$, is intractable, a variational approximation may be made. A computationally tractable variational distribution

$\mathcal{P}(\psi, \lambda)$ will be used in the modified E step instead of the original joint posterior. The evidence lower bound derived using a variational approximation may be written as

$$\log p(\mathcal{O}|\mathcal{W},\mathcal{M}) \;\; \geq \;\; \int \sum_{\boldsymbol{\psi}} \mathcal{P}(\boldsymbol{\psi},\lambda) \log \frac{p(\mathcal{O},\boldsymbol{\psi},\lambda|\mathcal{M})}{\mathcal{P}(\boldsymbol{\psi},\lambda)} d\lambda. \tag{3.12}$$

Maximizing the lower bound in the above equation is equivalent to minimizing the Kullback-Leibler (KL) divergence between the variational distribution, $\mathcal{P}(\boldsymbol{\psi}, \lambda)$, and the true joint posterior, $P(\boldsymbol{\psi}_{\tau} = \mathcal{S}_j, \lambda|\mathcal{O},\mathcal{W},\mathcal{M})$. Variational methods provide an alternative form of evidence lower bound. It is sometimes referred to as Variational Bayesian learning in the literature. The key issue with this approach is how to select an appropriate form of variational distribution. Such a selection is always subjective. One commonly used form assumes the statistical independence between model parameters, λ, and hidden states, $\mathcal{S}_j$, so the variational distribution is in a simplified factorial form, $\mathcal{P}(\boldsymbol{\psi}, \lambda) = \mathcal{P}(\boldsymbol{\psi})\mathcal{P}(\lambda)$, for example, in [118, 117].

The same assumption of the EM lower bound in equation 3.11 is made in variational methods. It is assumed that the ordering of the Bayesian evidence is the same as that of the variational lower bound. Similar to the log-likelihood lower bound derived using EM, the looser the variational lower bound is, the poorer the evidence approximation may be. Hence, the selection of the variational distribution $\mathcal{P}(\boldsymbol{\psi}, \lambda)$ should tighten the bound as much as possible.

3.3.6 Markov Chain Monte Carlo (MCMC) Sampling

Another family of approximation methods for the Bayesian evidence are Markov chain Monte Carlo (MCMC) sampling schemes [79, 97, 82]. The simplest MCMC sampling based approximation is to average out a finite number of random samples drawn in the parametric space. This is given in the following equation and is often referred to as the simple Monte Carlo:

$$\begin{aligned} p(\mathcal{O}|\mathcal{W},\mathcal{M}) &= \int p(\mathcal{O}|\lambda,\mathcal{W},\mathcal{M})p(\lambda|\mathcal{M})d\lambda \\ &\approx \frac{1}{N_{\mathtt{mc}}} \sum_i p(\mathcal{O}|\lambda_i,\mathcal{W},\mathcal{M}) \end{aligned} \tag{3.13}$$

where λ_i is the ith sample of the model parameters and $N_{\mathtt{mc}}$ is the total number of samples drawn. It is assumed that the drawn samples are statistically independent against one another. However, in many practical situations it may difficult to obtain such samples from $p(\lambda|\mathcal{M})$.

To overcome this problem other forms of sampling schemes may be used. In *rejection sampling*, a *proposal distribution*, $q(\lambda)$, and a constant, $c < \infty$, are introduced such that $\forall\lambda, p(\lambda|\mathcal{M}) \leq cq(\lambda)$. Samples that are drawn from the proposal distribution $q(\lambda)$ with a probability $p(\lambda|\mathcal{M})/cq(\lambda)$ are accepted and used for the simple Monte Carlo in equation 3.13 [82, 79]. One issue with rejection sampling is that the scheme only works well if the proposal distribution $q(\lambda)$ is a good approximation to the parameter prior $p(\lambda|\mathcal{M})$. It may be difficult to find $cq(\lambda)$ with a small c which is easy to sample from.

Another closely related sampling scheme is *importance sampling*. Using this method the Bayesian evidence is approximated as the following [82, 79]:

$$p(\mathcal{O}|\mathcal{W},\mathcal{M}) \approx \frac{1}{N_{\mathtt{mc}}}\sum_{i} p(\mathcal{O}|\lambda_i,\mathcal{W},\mathcal{M})\frac{p(\lambda|\mathcal{M})}{q(\lambda)} \tag{3.14}$$

where the proposal distribution $q(\lambda)$ is required to be non-zero when $p(\lambda|\mathcal{M})$ is. Similar to rejection sampling, the issue with this approach is also how to select a suitable form of the proposal distribution $q(\lambda)$ as a good approximation to $p(\lambda|\mathcal{M})$. Another issue with both rejection sampling and importance sampling is that an improper weighting or rejection of samples can cause the Monte Carlo average to be dominated by a few samples. This may lead to a poor approximation of the Bayesian evidence.

In many situations when $p(\lambda|\mathcal{M})$ is a high dimensional distribution, it may be difficult to find a good form of proposal distribution $q(\lambda)$ as an approximation. In this case more complicated sampling schemes, such as *Gibbs sampling* may be used [97]. In Gibbs sampling, it is assumed that $p(\lambda|\mathcal{M})$ is too complex to draw samples from directly. Instead, its conditional distribution, $p\left(\lambda_i^{(n)}|\lambda_1^{(n)},...,\lambda_{i-1}^{(n)},\lambda_{i+1}^{(n-1)},...,\lambda_{N_{\mathtt{mc}}}^{(n-1)}\right)$, may be used as the proposal distribution. The superscript refers to the nth sampling iteration. The algorithm iteratively picks up a model parameter sample, either in turn or randomly, which is then replaced by a sample selected using the proposal distribution. This form of proposal distribution accounts for the statistical dependence between samples.

Unfortunately, MCMC sampling schemes are impractical to use on current LVCSR systems for Bayesian evidence approximation. A state-of-the-art recognition system may contain millions of free parameters. This leads to a very high-dimensional parameter space from which to draw samples. For this reason MCMC based sampling schemes are computationally less feasible than other approximation schemes. They are not considered in this thesis for the approximation of Bayesian evidence.

3.4 Information Theory Methods

The second category of complexity control techniques are based on *information theory*. These approaches treat the complexity control problem as finding an appropriate code length [6] for a data transmission process. Probabilistic distributions may be viewed as code generators. Assume that both the sender and the receiver know from which distribution, $p(\mathcal{O}|\mathcal{W},\mathcal{M})$, a code $\mathcal{O}$ is generated from. Then according to Shannon's Source Coding Theorem, the fitness to the data, $-\log p(\mathcal{O}|\hat{\lambda},\mathcal{W},\mathcal{M})$, penalized by a channel cost, $\mathcal{C}(\mathcal{O},\mathcal{M})$, forms a two-part code description length [16, 6, 96, 54],

$$\hat{\mathcal{M}} = \arg\min_{\mathcal{M}}\left\{-\log p(\mathcal{O}|\hat{\lambda},\mathcal{W},\mathcal{M}) + \mathcal{C}(\mathcal{O},\mathcal{M})\right\}. \tag{3.15}$$

The channel cost may be interpreted as the part of description length which corresponds to the complexity of the code generator. In this section two complexity control criteria within the information theory framework are presented.

3.4.1 Minimum Description Length (MDL)

One commonly used information theory approach is the minimum description length (MDL) criterion. The MDL principle selects the optimal model structure with the shortest two-part code length. For the two-part code given in equation 3.15, the complexity term, $\mathcal{C}(\mathcal{O}, \mathcal{M})$, needs to be explicitly given. Hence the two-part code based MDL criterion in equation 3.15 may not be directly used for complexity control unless the penalization term, $\mathcal{C}(\mathcal{O}, \mathcal{M})$, is explicitly known. The MDL code length may be expressed in multiple forms [95, 47]. A two-part code is only one of these forms. There are other forms of description length that do not require knowing the exact form of the complexity penalization term. One example is the normalized maximum likelihood (NML) proposed in [95]. The standard form is the mixture code length [16, 6, 54].

$$\hat{\mathcal{M}} = \arg\max_{\mathcal{M}} \left\{ P(\mathcal{M}) \int p(\mathcal{O}|\lambda, \mathcal{W}, \mathcal{M}) p(\lambda|\mathcal{M}) d\lambda \right\} \tag{3.16}$$

It is in the same form as the Bayesian evidence integral in equation 3.2. However, it is derived as a form of code description from an information theoretic perspective. In common with Bayesian evidence, this form of code length may be approximated via a first order asymptotic expansion equivalent to BIC, or a second order Laplace's approximation.

3.4.2 Minimum Message Length (MML)

Another information theory approach is the Minimum Message Length (MML) principle [47]. The basic idea of MML is to find a two-part code generator to minimize the expected message length (number of bits needed to encode the data) of the observed data. The MML principle is closely related to MDL. The MML code length has the same form of definition as the mixture MDL given in equation 3.16. However there are some differences between the two schemes. First, the MML code length can only be expressed as a mixture distribution, while MDL may have multiple forms of code length. A mixture distribution is only one of them. Second, the MML principle is more closely related to Bayesian approaches than MDL. A prior distribution over model parameters is always required as in equation 3.16. In contrast, such a prior distribution is not required by the MDL principle when a two-part code length is used. Like the mixture code length of MDL in equation 3.16, the MML code length may be approximated via a BIC style first order, or Laplace's second order approximation.

3.5 Previous Application to Speech Recognition

As discussed in section 3.1, state-of-the-art LVCSR systems are highly complex and many techniques are used to enhance the recognition performance and also alter the systems' complexity. When these techniques are used it is desirable to optimize the model complexity to achieve the optimal WER. However the application of complexity control techniques for speech recognition has been limited, especially for LVCSR tasks. In this section a survey of previous applications of model selection techniques is presented.

BIC is a most commonly used complexity control technique for speech recognition [12, 13, 15, 130]. For example, this method was used in [12, 15] for HMM state tying on LVCSR tasks. As described in section 2.3.2, in decision tree based state clustering the threshold for likelihood gain must be manually tuned. Such a threshold acts to control the depth of the tree, or equivalently the total number of distinct states after tying [132, 133, 131]. This means the system's complexity can not be automatically determined. In contrast, the optimal tree cut was automatically determined using BIC and "penalized" BIC ($\rho = 2.0$ in equation 3.3) in [15]. The EM lower bound of log-likelihood discussed in section 3.3.4 was used to efficiently compute the BIC criterion during clustering. It was reported that compared with a standard likelihood based approach a more compact HMM system with the same WER was obtained. It was also reported that the standard BIC criterion lacked penalization power to prune over-grown trees. In [118, 117] on a Japanese LVCSR task it was also found that BIC yielded a poor approximation to the evidence integral when the training data is limited. The problem may be caused the large number assumption made in BIC, as discussed in section 3.3.1, which may be too strong for small data sets. So as the amount of training data is reduced, the BIC approximation is increasingly poor.

As an alternative to BIC, the variational method has also been used to approximate the evidence integral for complexity control. The large number assumption of BIC is no longer required. The scheme is often referred to as the variational Bayesian method [2, 119, 120, 117, 59]. In [117] this approach was used for decision tree based state clustering. The approximated Bayesian evidence was used instead of likelihood as in a standard approach. Performance improvements were reported with a small vocabulary English name entity recognition task. In [59] on experiments of a small vocabulary Japanese recognition task, the variational Bayesian approach was also found to select a more compact decision tree cut than the standard maximum likelihood method. Performance gains were also obtained over an MDL (equivalent to BIC) based clustering proposed in [105, 107]. As described in section 3.4, when using the MDL principle a certain form of code length is required. In [105, 107], the mixture code length in equation 3.16 was used. A first order approximation to it is equivalent to the BIC metric.

In addition to HMM state tying, another area which complexity control techniques have been applied to is speaker adaption. For these tasks the amount of enrollment data is often sparse. It is therefore important to determine the optimal number of parameters to be robustly estimated when building speaker specific models. For linear transformation schemes, such as MLLR, this corresponds to the number of transforms. As previously discussed in section 2.5, a standard approach uses the training data associated with each regression tree node [128, 28]. If the amount of data assigned to a tree node exceeds a given threshold, an MLLR transform will be generated. Otherwise, the transform estimation will back-off to the parental node's statistics. The occupancy threshold requires empirical tuning. Essentially, this is a simply "more data more parameters" approach. In [106] the MDL principle was used to determine the optimal cut of a regression class tree. The form of code length used was the mixture distribution given in equation 3.16, approximated via a first order expansion (equivalent to BIC). Each MLLR transform was restricted to be a simple bias vector. Experimental results on a medium vocabulary Japanese recognition task showed that marginal WER improvement was obtained over the standard approach.

3.6 Limitations of the Likelihood Paradigm

There is an inherent assumption made in standard evidence based complexity control techniques: there is a strong correlation between WER and likelihood on unseen data. Thus increasing the likelihood on the unseen data should decrease the WER. However, for a speech recognition system using HMMs, such an assumption is not true. As previously discussed in section 2.1.1, when using HMMs two assumptions are made about the nature of the speech signals: the quasi-stationary assumption and the observation independence assumption. Neither assumption is actually true for speech signals. Speech production is a non-stationary process even within minute time intervals. Furthermore, the dynamics of articulators and the use of overlapping frames in speech parameterization, as discussed in section 2.3.1, result in correlation between frames. Hence HMMs are not the correct models for speech signals. Consequently, in recent research the correlation between WER and likelihood has been found fairly weak for current speech recognition systems. In this case, using held-out data likelihood, or equivalently marginalizing the ML criterion as in Bayesian learning and Information theory, may be inappropriate for complexity control. It leads to an incorrect WER ranking and a poor selection of model complexity. For this reason it would be preferable to marginalize a criterion that is more closely related to the recognition error, rather than likelihood.

3.7 Summary

In this chapter standard complexity control techniques were presented. These schemes were developed within a maximum likelihood paradigm and may be classified into two major categories. In Bayesian learning techniques model selection is based on the evidence, or the marginal likelihood of training data. In information theory approaches, a complexity control problem is viewed as finding the optimal code length for a data transmission process. The code length is often expressed as the penalized log likelihood. For both types of techniques numerical approximation is often required to practically compute the Bayesian evidence or the mixture code length.

For these techniques to work well, a strong correlation between the WER and likelihood on unseen data must exist. However, for current speech recognition systems using HMMs this correlation may be fairly weak, as the models used are far from the “ideal” ones. Thus these standard likelihood based approaches may be inappropriate for model complexity control on current ASR tasks. It would be preferable to employ a complexity control criterion that is more directly related to WER. In chapter 5 a novel discriminative method for model selection is presented.

4

Discriminative Training

This chapter presents discriminative training techniques for speech recognition. First the limitations of maximum likelihood training is discussed. Then several commonly used discriminative criteria are presented. This is followed by a survey of the optimization schemes for discriminative training criteria. In particular, the extended Baum-Welch (EBW) algorithm, and a recently introduced weak-sense auxiliary function based approach are discussed.

4.1 Limitations of ML Training

In maximum likelihood training it is assumed that HMMs are the “correct” models for speech signals. It is further assumed that given infinite amount of training data, the global ML estimates tend to the optimum of model parameters. However, for current speech recognition systems neither assumption is true.

First, HMMs are not the “correct” models for speech signals. As discussed in section 2.1.1, two assumptions were made about the nature of speech signals when using HMMs: the quasi-stationary assumption and the observation independence assumption. As discussed in section 3.6, neither is true. Since current HMM based ASR systems are not the correct models for speech signals, the correlation between the WER and likelihood may be weak. Merely increasing the likelihood on the observed or unseen data as in ML training may not necessarily improve the recognition performance.

Second, the training data quantity is limited in practical situations. A large collection of audio data with detailed transcription is highly expensive. The majority of state-of-the-art LVCSR systems are trained using no more than five thousand hours of audio data [11, 24, 65]. To produce accurate manual transcriptions for these large collections of acoustic training data is very expensive.

Third, the EM algorithm used in ML training is only guaranteed to find a local optimum for the model parameters. Even if the above two conditions are met, an EM based optimization still cannot guarantee to yield a global optimal estimate for the model parameters during ML training.

For these reasons, ML training does not guarantee the optimal recognition performance for current speech recognition systems. Hence it is preferable to employ training schemes that explicitly aim at improving the recognition accuracy. One obvious form is to use the recognition error rate. However, the recognition error rate is not in a continuous form and may not be directly used for training based on standard optimization schemes, such as gradient descent. In contrast, discriminative training criteria, such as *maximum mutual information* (MMI), are continuous approximations to the error rate. These criteria do not make the model correctness assumption as in ML training. They are explicitly aimed at reducing the approximated recognition error rate on either a sentence or word level.

4.2 Discriminative Training Criteria

Discriminative criteria have been successfully applied to LVCSR training [124, 93, 90]. In this section three commonly used discriminative training criteria, maximum mutual information (MMI), minimum phone error (MPE) and minimum classification error (MCE), are presented in detail.

4.2.1 Maximum Mutual Information (MMI)

One of the most widely used discriminative criteria is the *maximum mutual information* (MMI) criterion [3]. This is equivalent to maximizing the *a posteriori* probability of the correct transcription, $\mathcal{W}$, for the given training data and model. The MMI criterion may be expressed as

$$\begin{aligned}\mathcal{F}_{\texttt{mmi}}(\lambda,\mathcal{M}) &= \frac{p(\mathcal{O},\mathcal{W}|\lambda,\mathcal{M})}{p(\mathcal{O}|\lambda,\mathcal{M})} \\ &= P(\mathcal{W}|\mathcal{O},\lambda,\mathcal{M}) \qquad (4.1)\end{aligned}$$

When the language model parameters, $P(\mathcal{W})$, are fixed during training, the MMI criterion is equivalent to *conditional maximum likelihood* (CML) criterion [81]. In addition to optimizing the ML criterion, $p(\mathcal{O},\mathcal{W}|\lambda,\mathcal{M})$, the likelihood of a "composite" model $p(\mathcal{O}|\lambda,\mathcal{M})$ is decreased. The composite model, $p(\mathcal{O}|\lambda,\mathcal{M})$, is obtained by summing over all possible hypotheses, $\{\tilde{\mathcal{W}}\}$.

$$p(\mathcal{O}|\lambda,\mathcal{M}) = \sum_{\tilde{\mathcal{W}}} p(\mathcal{O}|\lambda,\tilde{\mathcal{W}},\mathcal{M})P(\tilde{\mathcal{W}}) \qquad (4.2)$$

In the literature these two parts of the MMI criterion are usually referred to as the *numerator* and *denominator* terms respectively [84, 124]. For LVCSR systems it is infeasible to store all possible hypotheses to obtain the composite model, $p(\mathcal{O}|\lambda,\mathcal{M})$. In practice a finite number of confusable word sequences are stored either in an N-best list or lattice. These are used as a compact format to represent the model confusions over the training data [112]. As discussed in section 2.3.5, the dynamic range of likelihood may be very different between the acoustic model and the language model. To overcome this problem, the language model probability is scaled by a constant $\kappa > 0$ to compensate for the difference in dynamic range [124]. The acoustic likelihood is also deweighted using the inverse of the language model probability scale. This broadens the posterior distribution of different word paths in a lattice. Such an increase in confusable data can improve generalization performance [124]. By doing so the likelihood given the composite model may be expressed as

$$p(\mathcal{O}|\lambda,\mathcal{M}) \;=\; \sum_{\tilde{\mathcal{W}}} p(\mathcal{O}|\lambda,\tilde{\mathcal{W}},\mathcal{M})^{\frac{1}{\kappa}} P(\tilde{\mathcal{W}})^{\kappa}.$$

MMI based discriminative training has been extensively used in state-of-the-art LVCSR systems. Significant improvements over ML trained models have been reported [124, 129, 126]. However, it has been found that the MMI criterion can give undue weights to outliers that have very low posterior probability over the correct transcription [113]. Consider the case when the observed data $\mathcal{O}$ is segmented into individual segments for training, $\{\mathcal{O}_1, ..., \mathcal{O}_r, ..., \mathcal{O}_R\}$, where $\mathcal{O}_r$ denotes the rth utterance. The following MMI criterion calculation will be heavily dominated by utterances with very low posteriors.

$$\mathcal{F}_{\texttt{mmi}}(\lambda,\mathcal{M}) \;=\; \sum_{r} \log \frac{p(\mathcal{O}_r,\mathcal{W}|\lambda,\mathcal{M})}{p(\mathcal{O}_r|\lambda,\mathcal{M})} \tag{4.3}$$

4.2.2 Minimum Classification Error (MCE)

Another discriminative criterion closely related to MMI is the *minimum classification error* (MCE) criterion [14, 60]. The MCE criterion was originally proposed for isolated word recognition [14]. In [110, 109] a form of MCE criterion was modified for continuous speech recognition tasks. As with the MMI criterion, word lattices or N-best lists may be used to represent the model's confusion over the training data. The MCE criterion is given by

$$\mathcal{F}_{\texttt{mce}}(\lambda,\mathcal{M}) \;=\; f\left(\log \frac{p(\mathcal{O},\mathcal{W}|\lambda,\mathcal{M})}{\sum_{\tilde{\mathcal{W}}\neq\mathcal{W}} p(\mathcal{O}|\lambda,\tilde{\mathcal{W}},\mathcal{M})P(\tilde{\mathcal{W}})}\right) \tag{4.4}$$

where $f(\cdot)$ is the smoothing function. Commonly used forms of $f(\cdot)$ are either an identity, $f(x) = x$, or a Sigmoid function given by

$$f(x) \;=\; \frac{1}{1+e^{-ax}} \tag{4.5}$$

where a is a tunable parameter. Note that the denominator term in equation 4.4 only contains incorrect word sequences for the MCE criterion, rather than all the possible word sequences as in the MMI criterion. This is a difference between the MCE and MMI criteria. A unified view of both the MMI and MCE criteria was given in [110, 109]. It was shown that both MMI and MCE criteria provide an upper bound to the sentence error rate from a Bayesian perspective. However, compared with MMI training MCE training is less commonly used in state-of-the-art LVCSR systems.

4.2.3 Minimum Phone Error (MPE)

Both the MMI and MCE criteria provide an approximation to the recognition error rate on a sentence level. However, in speech recognition the most commonly used performance measurement is the WER. Therefore, it would be preferable to have a training criterion that is directly related to the WER rather than the sentence error rate. The Overall Risk Criterion (ORC), or equivalently the minimum word error (MWE), is one such criterion. It has a continuous form of WER approximation and may be used for training speech recognition systems [62, 42].

A closely related criterion is the minimum phone error (MPE) criterion. Instead of evaluating recognition accuracy on a word level, a phone level accuracy is computed under the constraint of the reference word transcription [90, 93]. The MPE criterion is expressed as the average accuracy of all possible word sequences $\{\tilde{\mathcal{W}}\}$, measured against the reference transcription in terms of WER. The accuracy contribution from each hypothesis is simply weighted by its posterior probability. The MPE criterion is given by

$$\begin{aligned}\mathcal{F}_{\texttt{mpe}}(\lambda, \mathcal{M}) &= \sum_{\tilde{\mathcal{W}}} P(\tilde{\mathcal{W}}|\mathcal{O}, \lambda, \mathcal{M})\mathcal{A}(\tilde{\mathcal{W}}, \mathcal{W}) \\ &= \sum_{\tilde{\mathcal{W}}} \frac{p(\mathcal{O}, \tilde{\mathcal{W}}|\lambda, \mathcal{M})\mathcal{A}(\tilde{\mathcal{W}}, \mathcal{W})}{p(\mathcal{O}|\lambda, \mathcal{M})} \end{aligned} \tag{4.6}$$

where $\mathcal{A}(\tilde{\mathcal{W}}, \mathcal{W})$ is the phone level accuracy of a word sequence, $\tilde{\mathcal{W}}$, against the reference transcription, $\mathcal{W}$. The computation of $\mathcal{A}(\tilde{\mathcal{W}}, \mathcal{W})$ normally requires a dynamic programming procedure. An efficient approximation of phone accuracy in a lattice context was proposed in [90, 93]. The algorithm first computes the phone level accuracy for each arc in the lattice against the reference transcript. Recognition errors caused by either substitution, deletion or insertion will be accounted for. Then the accuracy measuring of each arc is further smoothed using a forward-backward algorithm like procedure. This acts to de-weight the accuracy of lattice arcs that have very low "combined" accuracy for all the hypotheses that pass through it, and scale up that of those which are more correct. A more detailed description of the algorithms was given in [93].

MPE training has consistently outperformed MMI training on a range of LVCSR tasks [93]. Many state-of-the-art LVCSR systems are trained using the MPE criterion [51, 127, 64, 65, 23, 24]. No significant difference was found between MPE and MWE training in terms of recognition performance, although MWE training was found to be more powerful to fit the training data.

4.3 Optimization of Discriminative Criteria

The optimization of discriminative criteria is non-trivial. The EM algorithm for ML training can not be directly used for these criteria. In this section optimization schemes for discriminative training criteria are presented. First, the extended Baum-Welch (EBW) algorithm and a weak-sense auxiliary function based approach are discussed. Both approaches yield similar parameter updates. Then gradient descent based numerical techniques are discussed for the optimization of discriminative criteria.

4.3.1 Extended Baum-Welch Algorithm

The extended Baum-Welch (EBW) algorithm is the most commonly used method for the optimization of discriminative criteria [43, 44, 84, 113, 110]. The algorithm was originally proposed for discrete density HMMs and then extended to the continuous case, but in this section the BW algorithm for ML training is revisited first. Then the EBW update formula for discrete density HMMs are presented. The relationship between the derivation of the EBW and the BW algorithm is also discussed. Then the extension to the EBW algorithm for continuous density HMMs is presented for both the MMI and MPE criteria. Finally, a recently introduced I-smoothing technique for the EBW algorithm is discussed.

4.3.1.1 Baum-Welch Algorithm

The Baum-Welch (BW) algorithm provides a way to iteratively maximize polynomials which satisfy the following two conditions [7]:

- All coefficients in the polynomial are non-negative.
- All variables in the polynomial are non-negative and subject to a sum-to-one constraint.

This is exactly the case encountered in the parameter optimization of discrete density HMMs during ML training. These discrete parameters may include the transition probabilities and hidden state densities. Let λ_{ij} denote the jth free parameter of the ith distribution of the model, the Baum-Welch (BW) re-estimation formula is given by

$$\lambda_{ij} = \frac{\tilde{\lambda}_{ij} \left. \frac{\partial \mathcal{F}(\lambda,\mathcal{M})}{\partial \lambda_{ij}} \right|_{\lambda=\tilde{\lambda}}}{\sum_j \tilde{\lambda}_{ij} \left. \frac{\partial \mathcal{F}(\lambda,\mathcal{M})}{\partial \lambda_{ij}} \right|_{\lambda=\tilde{\lambda}}} \tag{4.7}$$

where again $\tilde{\lambda}$ is the current parameter estimate. During ML training the derivatives with respect to model parameters in equation 4.7 are equivalent to the hidden state posterior occupancies for an HMM system. These statistics may be efficiently computed using a forward-backward approach, as described in section 2.2.2. However, the BW algorithm cannot be used for the optimization of discriminative criteria, such as MMI in equation 4.1. This is because these criteria cannot be expressed as valid polynomials that satisfy the above two conditions required by the BW algorithm.

4.3.1.2 EBW for Discrete Density HMMs

To overcome the limitation of the Baum-Welch algorithm, the extended Baum-Welch (EBW) algorithm was introduced for the discriminative training of discrete density HMMs [43, 44]. The EBW algorithm can be shown to converge to a local optimum for discriminative training criteria that may be classified as a certain family of rational objective functions. The type of rational objective function considered by the algorithm is expressed as a ratio of two polynomials,

$$\mathcal{F}(\lambda,\mathcal{M}) = \frac{\mathcal{F}_{\texttt{num}}(\lambda,\mathcal{M})}{\mathcal{F}_{\texttt{den}}(\lambda,\mathcal{M})} \tag{4.8}$$

where the numerator $\mathcal{F}_{\texttt{num}}(\lambda,\mathcal{M})$ and denominator $\mathcal{F}_{\texttt{den}}(\lambda,\mathcal{M})$ are rational polynomials with non-negative coefficients, and variables that are non-negative and subject to a sum-to-one constraint. Hence both the numerator and denominator polynomials satisfy the two conditions required by the BW algorithm, as explained in section 4.3.1.1. Again let λ_{ij} denote the jth free parameter of the ith distribution of the model, the EBW re-estimation formula is given by

$$\lambda_{ij} \quad = \quad \frac{\tilde{\lambda}_{ij}\left(\left.\frac{\partial \mathcal{F}(\lambda,\mathcal{M})}{\partial \lambda_{ij}}\right|_{\lambda=\tilde{\lambda}} + D\right)}{\sum_j \tilde{\lambda}_{ij}\left(\left.\frac{\partial \mathcal{F}(\lambda,\mathcal{M})}{\partial \lambda_{ij}}\right|_{\lambda=\tilde{\lambda}} + D\right)} \tag{4.9}$$

where D is a regularization constant. The convergence of the algorithm is only guaranteed given a sufficiently large D.

Although the BW and EBW algorithms are used to optimize training criterion in very different forms, the derivation of the EBW update in equation 4.9 may be related to the BW algorithm. A direct maximization of discriminative criteria, expressed in the form of equation 4.8, can be difficult. The approach adopted in [43, 44] is to convert the original criterion to a related polynomial, $\mathcal{R}(\lambda, \mathcal{M})$, which may then be optimized using the BW algorithm. First, the two conditions required by the BW algorithm given in section 4.3.1.1 must be met by a valid polynomial, $\mathcal{R}(\lambda, \mathcal{M})$. Second, maximizing a valid polynomial, $\mathcal{R}(\lambda, \mathcal{M})$, should be equivalent to that of the original criterion, $\mathcal{F}(\lambda, \mathcal{M})$. This last condition is essential as to guarantee that the original objective function will never be decreased. Let $C > 0$ denote a regularization constant. The form of the related polynomial proposed in [43, 44] is given by

$$\mathcal{R}(\lambda, \mathcal{M}) \quad = \quad \mathcal{F}_{\texttt{den}}(\lambda, \mathcal{M}) \left[\mathcal{F}(\lambda, \mathcal{M}) - \mathcal{F}(\tilde{\lambda}, \mathcal{M})\right] + C \prod_i \sum_j \lambda_{ij} \tag{4.10}$$

where $\tilde{\lambda}$ is the current parameter estimate.

It can be shown that the polynomial in equation 4.10 satisfies the following three conditions:

- $\mathcal{R}(\lambda, \mathcal{M})$ is a polynomial of discrete probabilities $\{\lambda_{ij}\}$ that are non-negative and subject to a sum-to-one constraint $\sum_j \lambda_{ij} = 1$.

- As long as the regularization constant C is big enough, all the coefficients in $\mathcal{R}(\lambda, \mathcal{M})$ can be non-negative.

- Around the current parameter estimates $\tilde{\lambda}$, maximizing $\mathcal{R}(\lambda, \mathcal{M})$ is equivalent to maximize $\mathcal{F}(\lambda, \mathcal{M})$. This is because $\mathcal{F}_{\texttt{den}}(\lambda, \mathcal{M}) > 0$ holds for any valid λ, and the third regularization term in equation 4.10, $C \prod_i \sum_j \lambda_{ij}$ is invariant of λ, under the sum-to-one constraint $\sum_j \lambda_{ij} = 1$. Hence one may write

$$\mathcal{R}(\lambda, \mathcal{M}) > \mathcal{R}(\tilde{\lambda}, \mathcal{M}) \quad \Rightarrow \quad \mathcal{F}(\lambda, \mathcal{M}) > \mathcal{F}(\tilde{\lambda}, \mathcal{M}).$$

Under these three conditions, a direct maximization of the rational objective function in the form of equation 4.8 may be converted to the maximization of the polynomial, $\mathcal{R}(\lambda, \mathcal{M})$, using the BW algorithm in equation 4.7. Thus one may write the update formula for λ_{ij}, the jth free parameter of the ith distribution of the model.

$$\lambda_{ij} \quad = \quad \frac{\tilde{\lambda}_{ij} \left.\frac{\partial \mathcal{R}(\lambda,\mathcal{M})}{\partial \lambda_{ij}}\right|_{\lambda=\tilde{\lambda}}}{\sum_j \tilde{\lambda}_{ij} \left.\frac{\partial \mathcal{R}(\lambda,\mathcal{M})}{\partial \lambda_{ij}}\right|_{\lambda=\tilde{\lambda}}} \tag{4.11}$$

In order to prove that the above update is equivalent to the EBW update in equation 4.9, the gradients of the polynomial, $\mathcal{R}(\lambda, \mathcal{M})$, and the original criterion, $\mathcal{F}(\lambda, \mathcal{M})$, need to be examined. The gradient of the polynomial $\mathcal{R}(\lambda, \mathcal{M})$ in equation 4.10 around the current parameter estimates, $\tilde{\lambda}$, is given by

$$\left.\frac{\partial \mathcal{R}(\lambda, \mathcal{M})}{\partial \lambda_{ij}}\right|_{\lambda=\tilde{\lambda}} = \left.\frac{\partial \mathcal{F}_{\mathtt{num}}(\lambda, \mathcal{M})}{\partial \lambda_{ij}}\right|_{\lambda=\tilde{\lambda}} - \mathcal{F}(\tilde{\lambda}, \mathcal{M}) \left.\frac{\partial \mathcal{F}_{\mathtt{den}}(\lambda, \mathcal{M})}{\partial \lambda_{ij}}\right|_{\lambda=\tilde{\lambda}} + C \tag{4.12}$$

Furthermore, the gradient of the original criterion in equation 4.8 around $\tilde{\lambda}$ is given by,

$$\left.\frac{\partial \mathcal{F}(\lambda, \mathcal{M})}{\partial \lambda_{ij}}\right|_{\lambda=\tilde{\lambda}} = \frac{1}{\mathcal{F}_{\mathtt{den}}(\tilde{\lambda}, \mathcal{M})} \left[\left.\frac{\partial \mathcal{F}_{\mathtt{num}}(\lambda, \mathcal{M})}{\partial \lambda_{ij}}\right|_{\lambda=\tilde{\lambda}} - \mathcal{F}(\tilde{\lambda}, \mathcal{M}) \left.\frac{\partial \mathcal{F}_{\mathtt{den}}(\lambda, \mathcal{M})}{\partial \lambda_{ij}}\right|_{\lambda=\tilde{\lambda}} \right] \tag{4.13}$$

Combining the gradient of the criterion, $\mathcal{F}(\lambda, \mathcal{M})$, in equation 4.13, and the gradient of the polynomial, $\mathcal{R}(\lambda, \mathcal{M})$, in equation 4.12, yields the following.

$$\left.\frac{\partial \mathcal{R}(\lambda, \mathcal{M})}{\partial \lambda_{ij}}\right|_{\lambda=\tilde{\lambda}} = \mathcal{F}_{\mathtt{den}}(\tilde{\lambda}, \mathcal{M}) \left[\left.\frac{\partial \mathcal{F}(\lambda, \mathcal{M})}{\partial \lambda_{ij}}\right|_{\lambda=\tilde{\lambda}} + C/\mathcal{F}_{\mathtt{den}}(\tilde{\lambda}, \mathcal{M}) \right] \tag{4.14}$$

Substituting the polynomial's gradient above in equation 4.14 into the update formula of equation 4.11 yields

$$\lambda_{ij} = \frac{\tilde{\lambda}_{ij} \left(\left.\frac{\partial \mathcal{F}(\lambda, \mathcal{M})}{\partial \lambda_{ij}}\right|_{\lambda=\tilde{\lambda}} + C/\mathcal{F}_{\mathtt{den}}(\tilde{\lambda}, \mathcal{M}) \right)}{\sum_j \tilde{\lambda}_{ij} \left(\left.\frac{\partial \mathcal{F}(\lambda, \mathcal{M})}{\partial \lambda_{ij}}\right|_{\lambda=\tilde{\lambda}} + C/\mathcal{F}_{\mathtt{den}}(\tilde{\lambda}, \mathcal{M}) \right)} \tag{4.15}$$

which is equivalent to the EBW algorithm in equation 4.9 if we let $D = C/\mathcal{F}_{\mathtt{den}}(\tilde{\lambda}, \mathcal{M})$.

A variety of discriminative training criteria may be optimized using this iterative EM-like scheme. These include all three discriminative training criteria presented in section 4.2. The EBW re-estimation formula was originally shown to be valid only for discrete density HMMs. Hence it can not be directly used for parameters of HMMs with continuous densities, such as Gaussian means and covariances [1]. State-of-the-art speech recognition systems normally use continuous density HMM models. In the next section the extension of the EBW update to continuous density HMMs is presented.

4.3.1.3 EBW for Continuous Density HMMs

The extension of the EBW update formula in equation 4.9 to continuous density HMMs is a non-trivial problem. The approach adopted in [84] was to use a simple discrete Gaussian approximation. The number of codebook entries for each discrete distribution in the HMM set was raised to infinity. This gives the following the re-estimation formula for Gaussian means and covariances

$$\begin{aligned}
\boldsymbol{\mu}^{(j)} &= \frac{\boldsymbol{\chi}_j^{\mathrm{num}}(\mathcal{O}) - \boldsymbol{\chi}_j^{\mathrm{den}}(\mathcal{O}) + D_j \tilde{\boldsymbol{\mu}}^{(j)}}{\boldsymbol{\chi}_j^{\mathrm{num}} - \boldsymbol{\chi}_j^{\mathrm{den}} + D_j} \\
\boldsymbol{\Sigma}^{(j)} &= \frac{\boldsymbol{\chi}_j^{\mathrm{num}}(\mathcal{O}^2) - \boldsymbol{\chi}_j^{\mathrm{den}}(\mathcal{O}^2) + D_j \left(\tilde{\boldsymbol{\mu}}^{(j)} \tilde{\boldsymbol{\mu}}^{(j)\top} + \tilde{\boldsymbol{\Sigma}}^{(j)} \right)}{\boldsymbol{\chi}_j^{\mathrm{num}} - \boldsymbol{\chi}_j^{\mathrm{den}} + D_j} - \boldsymbol{\mu}^{(j)} \boldsymbol{\mu}^{(j)\top}
\end{aligned} \tag{4.16}$$

[1] In practice the update rule in equation 4.9 is not used for estimating component priors and state transitions in LVCSR training, due to the algorithm's high sensitivity to small-valued parameters. Instead a more robust update is proposed in [124, 93] by maximizing a different objective function.

where the *numerator* statistics are given by

$$\begin{aligned} \boldsymbol{\chi}_j^{\texttt{num}} &= \sum_{\tau} \gamma_j^{\texttt{num}}(\tau) \\ \boldsymbol{\chi}_j^{\texttt{num}}(\mathcal{O}) &= \sum_{\tau} \gamma_j^{\texttt{num}}(\tau)\boldsymbol{o}_\tau \\ \boldsymbol{\chi}_j^{\texttt{num}}(\mathcal{O}^2) &= \sum_{\tau} \gamma_j^{\texttt{num}}(\tau)\boldsymbol{o}_\tau\boldsymbol{o}_\tau^\top \end{aligned} \tag{4.17}$$

and the *denominator* statistics are

$$\begin{aligned} \boldsymbol{\chi}_j^{\texttt{den}} &= \sum_{\tau} \gamma_j^{\texttt{den}}(\tau) \\ \boldsymbol{\chi}_j^{\texttt{den}}(\mathcal{O}) &= \sum_{\tau} \gamma_j^{\texttt{den}}(\tau)\boldsymbol{o}_\tau \\ \boldsymbol{\chi}_j^{\texttt{den}}(\mathcal{O}^2) &= \sum_{\tau} \gamma_j^{\texttt{den}}(\tau)\boldsymbol{o}_\tau\boldsymbol{o}_\tau^\top. \end{aligned} \tag{4.18}$$

$\gamma_j^{\texttt{num}}(\tau)$ and $\gamma_j^{\texttt{den}}(\tau)$ are the *numerator* and *denominator* Gaussian posterior occupancies respectively. Rather than using a global setting for D, a Gaussian specific smoothing constant, D_j, is used in equation 4.16. It was found that by using a Gaussian specific smoothing constant, a faster and more stable criterion convergence may be achieved than a global setting [124, 93]. The exact form of $\gamma_j^{\texttt{num}}(\tau)$ and $\gamma_j^{\texttt{den}}(\tau)$ depends on the underlying criterion being optimized.

In the case of MMI training, the numerator occupancy $\gamma_j^{\texttt{num}}(\tau)$ is equivalent to the ML Gaussian posterior probability given the correct transcription. The denominator $\gamma_j^{\texttt{den}}(\tau)$ is computed from all possible word sequences [124]. The MMI numerator and denominator Gaussian occupancies are given by

$$\begin{aligned} \gamma_j^{\texttt{num}}(\tau) &= P(\boldsymbol{\psi}_\tau = \mathcal{S}_j|\mathcal{O}, \mathcal{W}, \tilde{\lambda}, \mathcal{M}) \\ \gamma_j^{\texttt{den}}(\tau) &= P(\boldsymbol{\psi}_\tau = \mathcal{S}_j|\mathcal{O}, \tilde{\lambda}, \mathcal{M}) \end{aligned} \tag{4.19}$$

where again $\boldsymbol{\psi}_\tau = \mathcal{S}_j$ indicates that acoustic observation $\boldsymbol{o}_\tau$ was generated by hidden state j at time instance τ.

For MPE and MWE training, both the numerator and denominator occupancies must be computed from the recognition lattices. These lattices contain both the correct and incorrect word sequences. It has been found that applying a binary decision on lattices paths (or equivalently on word arcs), based on whether the the accuracy of the current path is below the average of the whole lattice, yields an efficient MPE criterion optimization [93, 90]. The numerator and denominator occupancies for MPE training may be written as below,

$$\begin{aligned} \gamma_j^{\texttt{num}}(\tau) &= \sum_{\tilde{\mathcal{W}}} P(\boldsymbol{\psi}_\tau = \mathcal{S}_j|\mathcal{O}, \tilde{\lambda}, \tilde{\mathcal{W}}, \mathcal{M})\gamma_{\tilde{\mathcal{W}}}^{\texttt{mpe}} \quad (\gamma_{\tilde{\mathcal{W}}}^{\texttt{mpe}} \geq 0) \\ \gamma_j^{\texttt{den}}(\tau) &= -\sum_{\tilde{\mathcal{W}}} P(\boldsymbol{\psi}_\tau = \mathcal{S}_j|\mathcal{O}, \tilde{\lambda}, \tilde{\mathcal{W}}, \mathcal{M})\gamma_{\tilde{\mathcal{W}}}^{\texttt{mpe}} \quad (\gamma_{\tilde{\mathcal{W}}}^{\texttt{mpe}} < 0) \end{aligned} \tag{4.20}$$

where the MPE path occupancy $\gamma_{\tilde{\mathcal{W}}}^{\texttt{mpe}}$ is the gradient of the MPE criterion against the log likelihood of a word sequence $\tilde{\mathcal{W}}$,

$$\gamma_{\tilde{\mathcal{W}}}^{\texttt{mpe}} = \left.\frac{\partial \mathcal{F}_{\texttt{mpe}}(\lambda, \mathcal{M})}{\partial \log p(\mathcal{O}, \tilde{\mathcal{W}}|\lambda, \mathcal{M})}\right|_{\lambda=\tilde{\lambda}} . \tag{4.21}$$

Following equation 4.6, the above may be re-written as [93],

$$\gamma_{\tilde{\mathcal{W}}}^{\texttt{mpe}} = P(\tilde{\mathcal{W}}|\mathcal{O}, \tilde{\lambda}, \mathcal{M}) \left[\mathcal{A}(\tilde{\mathcal{W}}, \mathcal{W}) - \mathcal{F}_{\texttt{mpe}}(\tilde{\lambda}, \mathcal{M})\right] \tag{4.22}$$

where $\mathcal{A}(\tilde{\mathcal{W}}, \mathcal{W})$ is the phone level accuracy of $\tilde{\mathcal{W}}$ against the reference transcription $\mathcal{W}$, as discussed in section 4.2.3.

4.3.1.4 Setting of Smoothing Constant for EBW

An important issue for the EBW algorithm is the value of the smoothing constant D in equation 4.9 for discrete density HMMs, or the component specific D_j in equation 4.16 for continuous cases. This constant controls the convergence of the underlying criterion. Hence setting its value is important for discriminative training. In the original EBW update given in equation 4.9, a global D value is set so that all derivatives are positive. This may be achieved using

$$D = \max\left\{\max_{i,j}\left\{\left.\frac{\partial \mathcal{F}(\lambda, \mathcal{M})}{\partial \lambda_{ij}}\right|_{\lambda=\tilde{\lambda}}\right\}, 0\right\} + \epsilon \tag{4.23}$$

where ϵ is small positive constant [43, 44]. As discussed in section 4.3.1.2, D must be sufficiently large to guarantee the criterion convergence. However, it appears that no proof has been published to show the convergence is also guaranteed when using a finite valued D. Hence it was argued in [110] that the above form of D may no longer guarantee the convergence of the algorithm.

For the EBW update of continuous density HMMs, various ways of setting D_j were investigated for MMI training in [113, 124, 93]. It was reported that the following form of D_j outperformed other alternatives,

$$D_j = E \sum_{\tau} \gamma_j^{\texttt{den}}(\tau) \tag{4.24}$$

where $E > 0$, and is typically set to 1 or 2. However using this form of setting for D_j, the Gaussian variances may not necessarily be positive. To overcome this problem, an even bigger D_j may be used. Such D_j should be twice the value that ensures the re-estimated variance elements are positive [124, 93]. As with the original EBW algorithm there has been no published proof showing this form of finite valued setting of D_j still guarantees the convergence of the algorithm.

4.3.2 Weak-sense and Strong-sense Auxiliary Functions

The EBW update formula in equation 4.16 has been successfully applied for LVCSR training of continuous density HMM models. However its extension from discrete to continuous density

HMMs was based on a discrete Gaussian approximation, as discussed in section 4.3.1.3. Recently a *weak-sense* auxiliary function based approach was proposed as an alternative flexible and intuitive derivation of the EBW update for continuous HMMs [91, 89]. The concept of weak-sense auxiliary functions is the opposite to that of *strong-sense* auxiliary functions. A strong-sense auxiliary function is closely related to the original criterion because of two constraints. First, it shares the same gradient information with the criterion, around the current parameter estimate. Second, increasing a strong-sense auxiliary function guarantees not to decrease the original criterion. The auxiliary function used for ML training described in section 2.2.1, for instance, may be referred to as a strong-sense auxiliary function. In contrast, the relationship between a weak-sense auxiliary function and the criterion is looser. The only constraint imposed is that the criterion and its weak-sense auxiliary function share the same gradient around the current parameter estimates. Increasing the weak-sense auxiliary function may not guarantee not to decrease the original criterion. An example of a strong-sense and weak-sense auxiliary function is shown in figure 4.1. In the left figure, the criterion, $\mathcal{F}(\lambda, \mathcal{M})$, and its strong-sense auxiliary function, $\mathcal{Q}(\lambda, \tilde{\lambda})$, share the same gradient around the current parameter estimate $\tilde{\lambda}$. Furthermore, the strong-sense auxiliary function, $\mathcal{Q}(\lambda, \tilde{\lambda})$, and the original criterion, $\mathcal{F}(\lambda, \mathcal{M})$, reach their maximum at $\hat{\lambda}_{\mathcal{Q}}$ and $\hat{\lambda}_{\mathcal{F}}$ respectively. The maximization of $\mathcal{Q}(\lambda, \tilde{\lambda})$ guarantees not to decrease $\mathcal{F}(\lambda, \mathcal{M})$. In the right figure which shows an example of weak-sense auxiliary functions, the criterion and the weak-sense auxiliary function share the same gradient around the current parameter estimate. However, in the interval between $\hat{\lambda}_{\mathcal{F}}$ and $\hat{\lambda}_{\mathcal{Q}}$, maximizing $\mathcal{Q}(\lambda, \tilde{\lambda})$ actually decreases $\mathcal{F}(\lambda, \mathcal{M})$.

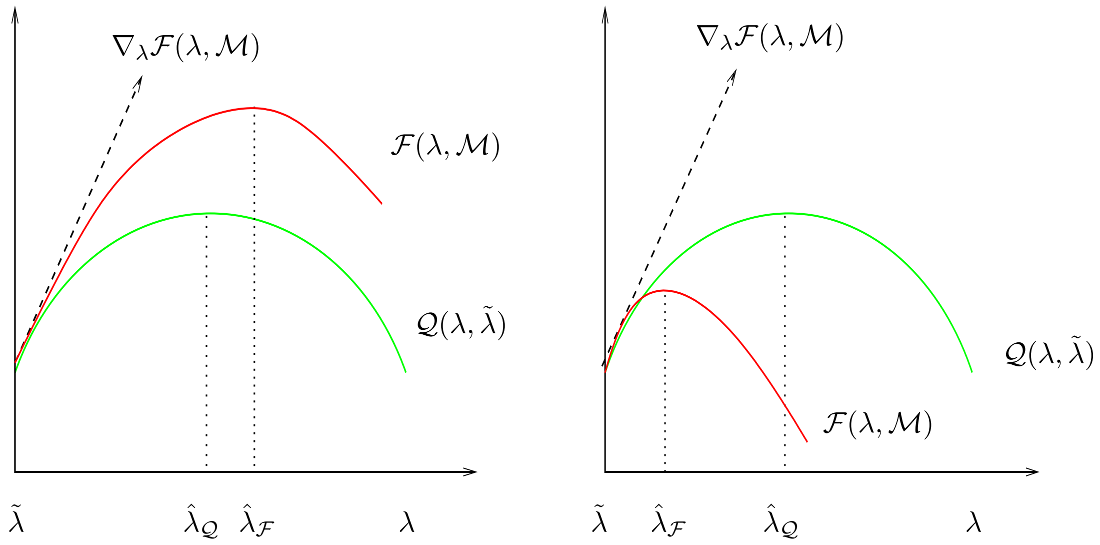

Figure 4.1 *Strong-sense (left) and weak-sense (right) auxiliary functions*

In [91, 89] a weak-sense auxiliary function is formulated as:

$$\mathcal{Q}(\lambda, \tilde{\lambda}) \quad = \quad \mathcal{Q}^{\texttt{num}}(\lambda, \tilde{\lambda}) - \mathcal{Q}^{\texttt{den}}(\lambda, \tilde{\lambda}) + \mathcal{Q}^{\texttt{sm}}(\lambda, \tilde{\lambda}). \tag{4.25}$$

where

$$\begin{aligned}
\mathcal{Q}^{\texttt{num}}(\lambda,\tilde{\lambda}) &= \sum_{j,\tau}\gamma_j^{\texttt{num}}(\tau)\log p(\boldsymbol{o}_\tau|\boldsymbol{\psi}_\tau=\mathcal{S}_j,\lambda,\mathcal{M}) \\
\mathcal{Q}^{\texttt{den}}(\lambda,\tilde{\lambda}) &= \sum_{j,\tau}\gamma_j^{\texttt{den}}(\tau)\log p(\boldsymbol{o}_\tau|\boldsymbol{\psi}_\tau=\mathcal{S}_j,\lambda,\mathcal{M}) \qquad (4.26)
\end{aligned}$$

and again $\gamma_j^{\texttt{num}}(\tau)$ and $\gamma_j^{\texttt{den}}(\tau)$ are the *numerator* and *denominator* Gaussian posterior occupancies respectively. The third term in equation 4.25, $\mathcal{Q}^{\texttt{sm}}(\lambda,\tilde{\lambda})$, is closely associated with the smoothing term of the EBW update formula in equation 4.16. This term must satisfy the following constraint.

$$\left.\frac{\partial\mathcal{Q}^{\texttt{sm}}(\lambda,\tilde{\lambda})}{\partial\lambda}\right|_{\lambda=\tilde{\lambda}} = 0. \qquad (4.27)$$

The common used $\mathcal{Q}^{\texttt{sm}}(\lambda,\tilde{\lambda})$ that satisfies this constraint may be expressed in the following general form,

$$\mathcal{Q}^{\texttt{sm}}(\lambda,\tilde{\lambda}) = \sum_j D_j\int p(\boldsymbol{o}|\boldsymbol{\psi}_{\boldsymbol{O}}=\mathcal{S}_j,\tilde{\lambda},\mathcal{M})\log p(\boldsymbol{o}|\boldsymbol{\psi}_{\boldsymbol{O}}=\mathcal{S}_j,\lambda,\mathcal{M})d\boldsymbol{o} \qquad (4.28)$$

where slightly different from previously used notations, $\boldsymbol{\psi}_{\boldsymbol{O}}=\mathcal{S}_j$, indicates acoustic observation $\boldsymbol{o}$ is generated by a hidden state j. Note that the integral in equation 4.28 is over the entire observation space. Hence, the discrete time instances have to be omitted. The above form of smoothing term was originally proposed in [84], but was only employed to interpret the discrete Gaussian approximation used to derive the EBW algorithm in 4.16 for means and covariances. However, it should be noted that the smoothing term in equation 4.28 may be applied to a variety of forms of model parameters, as no assumption about the underlying structure of hidden state distribution $p(\boldsymbol{o}|\boldsymbol{\psi}_{\boldsymbol{O}}=\mathcal{S}_j,\lambda,\mathcal{M})$ is made.

When using the above form of weak-sense auxiliary function to derive the EBW algorithm for Gaussian densities, the exact form of the smoothing term, $\mathcal{Q}^{\texttt{sm}}(\lambda,\tilde{\lambda})$, needs to be explicitly given. For example, in case of using diagonal covariances, the appropriate form of the smoothing term is given by

$$\mathcal{Q}^{\texttt{sm}}(\lambda,\tilde{\lambda}) = -\frac{1}{2}\sum_{j,i}D_j\left[\log 2\pi + \log\sigma_i^{(j)2} + \sigma_i^{(j)-2}\left(\mu_i^{(j)2} - 2\tilde{\mu}_i^{(j)}\mu_i^{(j)} + \tilde{\mu}_i^{(j)2} + \tilde{\sigma}_i^{(j)2}\right)\right] \qquad (4.29)$$

where i is the index of the feature dimensions, and $\sigma_i^{(j)2}$ is the ith dimensional variance element of component j. Using the above form of smoothing term, the EBW update formula for Gaussian means and diagonal covariances in equation 4.16 may be derived. Weak-sense auxiliary functions provide a heuristic and flexible derivation of the EBW algorithm.

4.3.3 I-Smoothing

For MPE and MWE training, both the lattice arc accuracy, $\mathcal{A}(\tilde{\mathcal{W}},\mathcal{W})$, and the criterion, $\mathcal{F}_{\texttt{mpe}}(\lambda,\mathcal{M})$, are positive numbers between 0 and 1. In this case the MPE, or MWE lattice arc occupancies in equation 4.22 may be very small. So also are the numerator and denominator occupancies given in equation 5.11. This may lead to un-reliable estimation of model parameters. To overcome this problem, it has been found important to add a portion of standard ML or MMI statistics to the numerator. This is referred to as I-smoothing [90, 93, 24]. This technique is closely related to the use of parameter priors in *maximum a posteriori* (MAP) estimation [36]. From a MAP training perspective, I-smoothing introduces an ML or MMI statistics based prior over Gaussian parameters. Using a weak-sense auxiliary function, this may be expressed as

$$\mathcal{Q}(\lambda,\tilde{\lambda}) \;=\; \mathcal{Q}^{\texttt{num}}(\lambda,\tilde{\lambda}) - \mathcal{Q}^{\texttt{den}}(\lambda,\tilde{\lambda}) + \mathcal{Q}^{\texttt{sm}}(\lambda,\tilde{\lambda}) + \log P(\lambda). \tag{4.30}$$

where $P(\lambda)$ is prior distribution over model parameters, λ. In case of using an ML statistics based $P(\lambda)$, the smoothed numerator statistics are given by

$$\begin{aligned}
\boldsymbol{\chi}_j^{\texttt{num}\prime} &= \boldsymbol{\chi}_j^{\texttt{num}} + \tau^I \\
\boldsymbol{\chi}_j^{\texttt{num}\prime}(\mathcal{O}) &= \boldsymbol{\chi}_j^{\texttt{num}}(\mathcal{O}) + \tau^I \frac{\boldsymbol{\chi}_j^{\texttt{ml}}(\mathcal{O})}{\boldsymbol{\chi}_j^{\texttt{ml}}} \\
\boldsymbol{\chi}_j^{\texttt{num}\prime}(\mathcal{O}^2) &= \boldsymbol{\chi}_j^{\texttt{num}}(\mathcal{O}^2) + \tau^I \frac{\boldsymbol{\chi}_j^{\texttt{ml}}(\mathcal{O}^2)}{\boldsymbol{\chi}_j^{\texttt{ml}}}
\end{aligned} \tag{4.31}$$

where the I-smoothing prior $\tau^I > 0$. In practice τ^I may be tuned for specific tasks [93, 24]. The ML smoothing statistics are given by

$$\begin{aligned}
\boldsymbol{\chi}_j^{\texttt{ml}} &= \sum_{\tau} \gamma_j(\tau) \\
\boldsymbol{\chi}_j^{\texttt{ml}}(\mathcal{O}) &= \sum_{\tau} \gamma_j(\tau)\boldsymbol{o}_\tau \\
\boldsymbol{\chi}_j^{\texttt{ml}}(\mathcal{O}^2) &= \sum_{\tau} \gamma_j(\tau)\boldsymbol{o}_\tau\boldsymbol{o}_\tau^\top
\end{aligned} \tag{4.32}$$

where $\gamma_j(\tau) = P(\boldsymbol{\psi}_\tau = \mathcal{S}_j|\mathcal{O},\mathcal{W},\tilde{\lambda},\mathcal{M})$ is the frame Gaussian posterior probability used in ML training. Recently it has been found that using MMI statistics for I-smoothing outperformed using the ML statistics [24] for LVCSR tasks. This is due to the nature of the I-smoothing statistics being used. The MMI smoothing statistics correspond to a parameter prior which typically outperforms the ML prior in terms of recognition performance.

4.3.4 Gradient Descent Based Optimization

Like many other forms of objective functions, discriminative training criteria may also be optimized using gradient descent style numerical methods. The steepest descent algorithm is a simple numerical scheme for optimizing multivariate functions. At each iteration the parameters to be optimized are modified in the direction of the the objective function's gradient. The gradient of the objective function is evaluated at the current parameter estimates. The magnitude of change to the parameters is a constant portion of the gradient. The proportion is commonly referred to as the learning rate, or step size. The update formula is given by

$$\lambda^{(n+1)} \;=\; \lambda^{(n)} - \eta\nabla_{\lambda=\lambda^{(n)}}\mathcal{F}(\lambda,\mathcal{M}) \tag{4.33}$$

where $\lambda^{(n)}$ is the current parameter estimated at iteration n, and η is the learning rate. If a more complex Newton search is used, the Hessian, or the second order derivative, is also required. At each iteration the gradient information is required for the update.

$$\lambda^{(n+1)} \;=\; \lambda^{(n)} - \eta \left[\nabla^2_{\lambda=\lambda^{(n)}} \mathcal{F}(\lambda, \mathcal{M})\right]^{-1} \nabla_{\lambda=\lambda^{(n)}} \mathcal{F}(\lambda, \mathcal{M}) \tag{4.34}$$

It may be shown that for discriminative training criteria, such as MMI and MPE, the gradient with respect to model parameters is closely related to the numerator and denominator occupancies used for the EBW update of equation 4.16. First, the following useful derivations are given, before examining the gradient information for individual criteria. The gradient of the log likelihood given the word sequence, $\tilde{\mathcal{W}}$, against Gaussian means, $\boldsymbol{\mu}^{(j)}$, and covariances, $\boldsymbol{\Sigma}^{(j)}$, for HMMs are given below [113].

$$\begin{aligned}
\frac{\partial \log p(\mathcal{O}|\lambda, \tilde{\mathcal{W}}, \mathcal{M})}{\partial \boldsymbol{\mu}^{(j)}} &= \sum_{\tau} P(\boldsymbol{\psi}_\tau = \mathcal{S}_j | \mathcal{O}, \lambda, \tilde{\mathcal{W}}, \mathcal{M}) \frac{\partial \log p(\boldsymbol{o}_\tau | \boldsymbol{\psi}_\tau = \mathcal{S}_j, \lambda, \mathcal{M})}{\partial \boldsymbol{\mu}^{(j)}} \\
\frac{\partial \log p(\mathcal{O}|\lambda, \tilde{\mathcal{W}}, \mathcal{M})}{\partial \boldsymbol{\Sigma}^{(j)}} &= \sum_{\tau} P(\boldsymbol{\psi}_\tau = \mathcal{S}_j | \mathcal{O}, \lambda, \tilde{\mathcal{W}}, \mathcal{M}) \frac{\partial \log p(\boldsymbol{o}_\tau | \boldsymbol{\psi}_\tau = \mathcal{S}_j, \lambda, \mathcal{M})}{\partial \boldsymbol{\Sigma}^{(j)}}
\end{aligned} \tag{4.35}$$

where

$$\begin{aligned}
\frac{\partial \log p(\boldsymbol{o}_\tau | \boldsymbol{\psi}_\tau = \mathcal{S}_j, \lambda, \mathcal{M})}{\partial \boldsymbol{\mu}^{(j)}} &= \boldsymbol{\Sigma}^{(j)-1} \left(\boldsymbol{o}_\tau - \boldsymbol{\mu}^{(j)}\right) \\
\frac{\partial \log p(\boldsymbol{o}_\tau | \boldsymbol{\psi}_\tau = \mathcal{S}_j, \lambda, \mathcal{M})}{\partial \boldsymbol{\Sigma}^{(j)}} &= \frac{1}{2} \boldsymbol{\Sigma}^{(j)-1} \left(\boldsymbol{o}_\tau - \boldsymbol{\mu}^{(j)}\right) \left(\boldsymbol{o}_\tau - \boldsymbol{\mu}^{(j)}\right)^{\top} \boldsymbol{\Sigma}^{(j)-1} - \frac{1}{2} \boldsymbol{\Sigma}^{(j)-1}
\end{aligned} \tag{4.36}$$

For the MMI criterion, given in equation 4.1, the gradient information may be written as the following:

$$\frac{\partial \log \mathcal{F}_{\texttt{mmi}}(\lambda, \mathcal{M})}{\partial \lambda} = \frac{\partial \log p(\mathcal{O}|\lambda, \mathcal{W}, \mathcal{M})}{\partial \lambda} - \frac{\partial \log p(\mathcal{O}|\lambda, \mathcal{M})}{\partial \lambda}. \tag{4.37}$$

Now, using the gradient information in equation 4.35 and 4.36 and the numerator and denominator occupancies defined in equation 4.19, the MMI gradient at the current estimates for Gaussian means and covariances are given by

$$\begin{aligned}
\left.\frac{\partial \log \mathcal{F}_{\texttt{mmi}}(\lambda, \mathcal{M})}{\partial \boldsymbol{\mu}^{(j)}}\right|_{\lambda=\tilde{\lambda}} &= \sum_{\tau} \left(\gamma_j^{\texttt{num}}(\tau) - \gamma_j^{\texttt{den}}(\tau)\right) \tilde{\boldsymbol{\Sigma}}^{(j)-1} \left(\boldsymbol{o}_\tau - \tilde{\boldsymbol{\mu}}^{(j)}\right) \\
\left.\frac{\partial \log \mathcal{F}_{\texttt{mmi}}(\lambda, \mathcal{M})}{\partial \boldsymbol{\Sigma}^{(j)}}\right|_{\lambda=\tilde{\lambda}} &= \sum_{\tau} \left(\gamma_j^{\texttt{num}}(\tau) - \gamma_j^{\texttt{den}}(\tau)\right) \left[\frac{1}{2} \left(\boldsymbol{o}_\tau - \tilde{\boldsymbol{\mu}}^{(j)}\right)\right. \\
&\quad \left. \times \left(\boldsymbol{o}_\tau - \tilde{\boldsymbol{\mu}}^{(j)}\right)^{\top} \tilde{\boldsymbol{\Sigma}}^{(j)-2} - \frac{1}{2} \tilde{\boldsymbol{\Sigma}}^{(j)-1}\right].
\end{aligned} \tag{4.38}$$

In the above equation the MMI criterion's gradient information at the current parameter estimates is closely related to the numerator and denominator statistics required by the EBW algorithm in equation 4.16.

For MPE a close relationship to the criterion also exists. Following the MPE criterion in equation 4.6, applying the chain rule for derivatives, and using the statistics defined in equation 4.21

and 4.22, one may write the following.

$$
\begin{aligned}
\frac{\partial \mathcal{F}_{\texttt{mpe}}(\lambda,\mathcal{M})}{\partial \lambda} &\propto \sum_{\tilde{\mathcal{W}}} \frac{\partial \mathcal{F}_{\texttt{mpe}}(\lambda,\mathcal{M})}{\partial \lambda} \\
&= \sum_{\tilde{\mathcal{W}}} \frac{\partial \mathcal{F}_{\texttt{mpe}}(\lambda,\mathcal{M})}{\partial \log p(\mathcal{O}|\lambda,\tilde{\mathcal{W}},\mathcal{M})} \frac{\partial \log p(\mathcal{O}|\lambda,\tilde{\mathcal{W}},\mathcal{M})}{\partial \lambda} \\
&= \sum_{\tilde{\mathcal{W}}} P(\tilde{\mathcal{W}}|\mathcal{O},\lambda,\mathcal{M}) \left[\mathcal{A}(\tilde{\mathcal{W}},\mathcal{W}) - \mathcal{F}_{\texttt{mpe}}(\lambda,\mathcal{M})\right] \frac{\partial \log p(\mathcal{O}|\lambda,\tilde{\mathcal{W}},\mathcal{M})}{\partial \lambda} \qquad (4.39)
\end{aligned}
$$

Combing the gradient information in equation 4.35, 4.36 and the MPE numerator and denominator occupancies in equation 4.20, the gradient direction of the MPE criterion against Gaussian means and covariances may be expressed as

$$
\begin{aligned}
\left.\frac{\partial \mathcal{F}_{\texttt{mpe}}(\lambda,\mathcal{M})}{\partial \boldsymbol{\mu}^{(j)}}\right|_{\lambda=\tilde{\lambda}} &\propto \sum_{\tau} \left(\gamma_j^{\texttt{num}}(\tau) - \gamma_j^{\texttt{den}}(\tau)\right) \tilde{\boldsymbol{\Sigma}}^{(j)-1} \left(\boldsymbol{o}_\tau - \tilde{\boldsymbol{\mu}}^{(j)}\right) \\
\left.\frac{\partial \mathcal{F}_{\texttt{mpe}}(\lambda,\mathcal{M})}{\partial \boldsymbol{\Sigma}^{(j)}}\right|_{\lambda=\tilde{\lambda}} &\propto \sum_{\tau} \left(\gamma_j^{\texttt{num}}(\tau) - \gamma_j^{\texttt{den}}(\tau)\right) \left[\frac{1}{2}\left(\boldsymbol{o}_\tau - \tilde{\boldsymbol{\mu}}^{(j)}\right)\right. \\
&\quad \times \left.\left(\boldsymbol{o}_\tau - \tilde{\boldsymbol{\mu}}^{(j)}\right)^{\top} \tilde{\boldsymbol{\Sigma}}^{(j)-2} - \frac{1}{2}\tilde{\boldsymbol{\Sigma}}^{(j)-1}\right]. \qquad (4.40)
\end{aligned}
$$

Hence the gradient information of the MPE criterion, given in equation 4.40, may also be related to the MPE statistics required by the EBW update.

Though gradient descent style numerical schemes may be used for optimizing discriminative training criteria, in practice these techniques are slow and have difficulty guaranteeing convergence. In early research it was reported that the EBW algorithm is a more efficient optimization scheme for discriminative objection functions than numerical methods [44]. The majority of state-of-the-art LVCSR systems employ the EBW algorithm for discriminative training [124, 126, 127, 23, 64].

4.4 Summary

In this chapter several commonly used discriminative training criteria and the associated optimization schemes were presented. The model correctness assumption made in ML training may be too strong for current speech recognition systems using HMMs. As is discussed earlier in section 3.6, this is also an issue for standard complexity control techniques under the maximum likelihood paradigm. It would therefore be preferable to use discriminative methods that are more explicitly related to classification error for complexity control and parameter estimation. The model correctness assumption of ML learning may then be removed from both the structural and parametric optimization. In the following chapter a novel complexity control approach is proposed using the marginalization of a discriminative measure. Then it is followed by an investigation of discriminative training of linear projection schemes discussed earlier in section 2.4.

5

Discriminative Model Complexity Control

In this chapter a novel model complexity control technique using a discriminative measure is presented. First, some previous work related to discriminative complexity control is briefly reviewed. Then issues with a direct marginalization of discriminative criteria for complexity control will be discussed. Due to the sensitivity to outliers, discriminative training criteria, such as MMI, cannot be directly integrated over for complexity control. This motivates the use of a closely related discriminative growth function, rather than the original criterion itself. This growth function maintains some of the attributes of the original discriminative criterion, but is less sensitive to outliers. The marginalization of the growth function is used to determine the appropriate model complexity. Two forms of growth functions for the MMI and MPE criteria are presented. Finally, some important implementation issues that arise when using marginalized discriminative growth functions for complexity control are discussed, in particular for the HLDA systems discussed in section 2.4.

5.1 Toward Discriminative Complexity Control

As discussed in chapter 3 the majority of complexity control research for speech recognition has focused on methods within the maximum likelihood paradigm. Under this likelihood based framework, HMMs are implicitly assumed to be the “correct” models for speech signals. Unfortunately the assumptions about the nature of speech signals when using HMMs are not valid, as discussed in section 3.6. Hence the model correctness assumption of existing techniques may be too strong for current ASR systems, and it is preferable to employ discriminative criteria for complexity control. They are more directly related to the recognition error, rather than likelihood.

A discriminative measure has previously been used in [4, 88], as a method of incrementally splitting Gaussian mixture components in an HMM based speech recognition system. The method proposed may be described in two steps. First, the state level alignment is obtained for both the correct and incorrect word sequences. In the second step, these alignments are kept fixed during the splitting of Gaussian components. For each state a splitting operation is considered if it increases the posterior probability of the correct state label. Using this method WER improvements were reported on a Wall Street Journal task. The main issue with this approach is that the complexity of the underlying model structure is not considered during the splitting process. No penalty is given to penalize over-complex structures. Strictly this method cannot be regarded as a control of model complexity, because no stopping criterion is provided. Instead it is more appropriate to view it as a discriminative increase of model complexity.

A similar approach using the MMI statistics given in equation 4.19 was also proposed in [85] to split Gaussian components in a discriminative fashion. The numerator and denominator statistics, given in equations 4.17 and 4.18, are accumulated on a component level, using a standard forward-backward procedure for both the reference transcription and confusable word sequences. For Gaussian component j, if the difference between the numerator and denominator occupancies, $\chi_j^{\tt num} - \chi_j^{\tt den}$ had a high ranking, for instance in the top 20% among all components, then the component is selected for splitting. Error rate reduction on a digit recognition task was reported using this component splitting method. Again, the same issue discussed above also applies to this approach. No penalty is assigned to model structures that are over-complex, and the splitting process cannot be terminated automatically.

Complexity control using a discriminative measure has also been investigated for speech recognition systems using more complicated acoustic models rather than HMMs. In [9] the MMI criterion was used to determine the appropriate complexity for a graph model. The system complexity considered was the conditional dependencies between random variables, which are denoted by nodes and edges in a graph model. The aim was to increase the model's discriminative power and reduce the recognition error rate, in common with the complexity control problem for HMMs. Unfortunately the issue with this method, in the same fashion as the above two approaches, is that over-complex model structures are not penalized. Hence the over-fitting problem cannot be prevented.

5.2 Marginalizing Discriminative Training Criteria

So far the major issue with the existing discriminative approaches for model selection is the lack of a complexity penalty term. As described in section 3.3, the marginalization of the conventional ML criterion in the parametric space may automatically penalize over-complex models. Hence one natural form of discriminative model complexity control is to marginalize a discriminative measure instead. This ensures the generalization of discriminative measures to unseen data. Replacing the ML criterion in the evidence integral of equation 3.2 by a discriminative criterion yields a "discriminative evidence". This should be more closely related to recognition error than likelihood based schemes. If the MMI criterion is used and the model prior, $P(\mathcal{M})$, is assumed uninformative, this yields

$$\hat{\mathcal{M}} \;=\; \arg\max_{\mathcal{M}} \int \mathcal{F}_{\tt mmi}(\lambda, \mathcal{M}) p(\lambda|\mathcal{M}) d\lambda \tag{5.1}$$

A similar form of integral of MPE criterion may also be considered for complexity control. However, for both criteria, such a direct marginalization may be inappropriate. The primary reason is that undue weights are given to outliers. State-of-the-art large vocabulary speech recognition systems are trained with hundreds of hours of data. Outliers, which are far from the decision boundary, are likely to exist in the training data. They are often utterances with very low likelihood, or explicitly associated with high recognition error rate. In may situations these may be caused by problems associated with the collection of the data, for instance, the corruption of the audio recording or human errors when producing the reference transcriptions. The sensitivity to outliers is a well known feature of the MMI criterion [56, 113]. Sentences with very low posteriors are heavily weighted. The performance ranking prediction will be distorted due to the presence of these outliers. The same issue exists with the MPE criterion for sentences with very high recognition error rate.

5.3 Discriminative Growth Functions

One approach to compensate for the sensitivity to outliers is to explicitly de-weight the outliers utterances. The use of a sigmoid function for the smoothing of the MMI criterion was studied in [113]. Unfortunately, using this method the smoothed MMI criterion is in a complicated form and difficult to integrate over. To handle this problem, the approach proposed in this thesis is to transform the original discriminative criterion into a closely related polynomial that has a more tractable form. This method is similar to the use of the polynomial $\mathcal{R}(\lambda, \mathcal{M})$ in section 4.3.1.2 to derive the EBW algorithm for discrete HMMs. For complexity control the proposed polynomial should maintain certain attributes of the original discriminative criterion, but must also be less sensitive to outliers. Note that the removal the sensitivity to outliers does not imply ignoring any difficult data during complexity control. Once again it should be made clear that only those are far from the decision boundary, typically with very low likelihood, or very high error rate, are considered as outliers. The marginalization of this polynomial function is then used to determine the appropriate model complexity. To efficiently compute this "discriminative evidence", similar approximations to those used for the standard Bayesian evidence, as discussed in chapter 3, may be used. In this section a general form of polynomial function for a certain family of discriminative criteria is introduced.

The form of polynomial function considered here is applicable to any discriminative criterion which may be expressed as a ratio between two polynomials with positive coefficients and variables. The MMI and MPE criteria are in this category. Consider a discriminative training criterion expressed in the following form (the model structure $\mathcal{M}$ is omitted for clarity).

$$\mathcal{F}(\lambda) = \frac{\mathcal{F}_{\tt num}(\lambda)}{\mathcal{F}_{\tt den}(\lambda)} \tag{5.2}$$

The general form of a polynomial function proposed here may be expressed as,

$$\mathcal{G}(\lambda) = \mathcal{F}_{\tt den}(\lambda)\left[\mathcal{F}(\lambda) - \mathcal{F}(\tilde{\lambda}) + C\mathcal{F}_{\tt sm}(\lambda, \tilde{\lambda})\right] \tag{5.3}$$

where $\tilde{\lambda}$ is the *current* parameter estimate. The first two terms in the bracket give information about the curvature of the criterion surface in the parametric space. Since they describe the variation, or *growth*, of the underlying criterion value between different parameter estimates, the polynomial in equation 5.3 will be renamed as a *discriminative growth function* in the rest of this thesis. The third term in the bracket is a smoothing term, scaled by a positive constant, C. To reduce the growth function's sensitivity to outliers, the smoothing criterion should be selected to compensate for the low likelihood, or high error rate, contribution from these outliers. Thus the smoothing term may be associated with the likelihood or WER. The constant C in equation 5.3 determines the effect from this smoothing criterion. The exact form of $\mathcal{F}_{\texttt{sm}}(\lambda, \tilde{\lambda})$ depends on the underlying discriminative criterion being considered and is further discussed in the following section for the MPE and MMI criteria. In addition, the denominator term, $\mathcal{F}_{\texttt{den}}(\lambda)$, outside the bracket in equation 5.3 may also help to reduce the sensitivity to outliers. This is the case for both the MMI and MPE criteria where the smoothing term is associated with the likelihood of a sentence, $\mathcal{F}_{\texttt{den}}(\lambda) = p(\mathcal{O}|\lambda)$. Thus highly unlikely word sequences will have a smaller effect on the growth function. However, it should be noted that the smoothing criterion, $\mathcal{F}_{\texttt{sm}}(\lambda, \tilde{\lambda})$, plays a more explicit, and flexible, role in reducing the sensitivity to outliers than $\mathcal{F}_{\texttt{den}}(\lambda)$. This especially the case when the original criterion, $\mathcal{F}(\lambda)$, is an approximation to recognition error rate, rather than likelihood.

The gradient of the growth function, $\mathcal{G}(\lambda)$, may be expressed as

$$\begin{aligned} \frac{\partial \mathcal{G}(\lambda)}{\partial \lambda} &= \left[\mathcal{F}(\lambda) - \mathcal{F}(\tilde{\lambda}) + C\mathcal{F}_{\texttt{sm}}(\lambda, \tilde{\lambda})\right] \frac{\partial \mathcal{F}_{\texttt{den}}(\lambda)}{\partial \lambda} \\ &\quad + \mathcal{F}_{\texttt{den}}(\lambda) \left[\frac{\partial \mathcal{F}(\lambda)}{\partial \lambda} + C\frac{\partial \mathcal{F}_{\texttt{sm}}(\lambda, \tilde{\lambda})}{\partial \lambda}\right]. \end{aligned} \tag{5.4}$$

When C approaches zero, around the current parameter estimate, $\tilde{\lambda}$, a turning point of the original criterion is also a turning point of the growth function. This may be expressed as

$$\lim_{C \to 0} \left.\frac{\partial \mathcal{G}(\lambda)}{\partial \lambda}\right|_{\lambda = \tilde{\lambda}} = \mathcal{F}_{\texttt{den}}(\tilde{\lambda}) \left.\frac{\partial \mathcal{F}(\lambda)}{\partial \lambda}\right|_{\lambda = \tilde{\lambda}}. \tag{5.5}$$

This constrains the attributes of the growth function to be related to those of the original criterion.

The proposed discriminative growth function in equation 5.3 is in a similar form to the polynomial $\mathcal{R}(\lambda, \mathcal{M})$ of equation 4.10. However, it is more appropriate to use the growth function in equation 5.3 for complexity control due to two reasons. First, the use of the smoothing term $\mathcal{F}_{\texttt{sm}}(\lambda, \tilde{\lambda})$ may explicitly reduce the sensitivity to outliers. In contrast, the third term of $\mathcal{R}(\lambda, \mathcal{M})$ in equation 4.10 does not have such properties. As discussed above, the reduction of sensitivity to outliers is very important when using discriminative criteria for complexity control. Second, the proposed growth function in equation 5.3 has a more general form, and is not restricted to models with discrete densities. This is also a preferable feature as models with continuous densities are widely used in current ASR systems. However, one disadvantage is that increasing the growth function in equation 5.3 does not guarantee not to decrease the original criterion, because the smoothing term, $\mathcal{F}_{\texttt{sm}}(\lambda, \tilde{\lambda})$, is also dependent on the model parameters.

As discussed in chapter 4, the majority of state-of-the-art LVCSR systems are trained using either the MMI or MPE criterion. Therefore in the following sections two forms of discriminative growth functions are proposed for the MMI and MPE criteria respectively. As the MPE criterion provides a closer approximation to WER than MMI, a growth function based on the MPE criterion is introduced first.

5.4 MPE Growth Function

The MPE growth function considered in this thesis is

$$\mathcal{G}(\lambda) \;=\; p(\mathcal{O}|\lambda)\left[\mathcal{F}_{\tt mpe}(\lambda) - \mathcal{F}_{\tt mpe}(\tilde{\lambda}) + C\mathcal{F}_{\tt sm}(\lambda,\tilde{\lambda})\right] \tag{5.6}$$

where the smoothing term is given by

$$\begin{aligned}\mathcal{F}_{\tt sm}(\lambda,\tilde{\lambda}) \;&=\; -\sum_{\substack{\tilde{\mathcal{W}} \\ \mathcal{A}(\tilde{\mathcal{W}},\mathcal{W})<\mathcal{F}_{\tt mpe}(\tilde{\lambda})}} P(\tilde{\mathcal{W}}|\mathcal{O},\lambda)\left[\mathcal{A}(\tilde{\mathcal{W}},\mathcal{W}) - \mathcal{F}_{\tt mpe}(\tilde{\lambda})\right]. \\ &=\; -\sum_{\tilde{\mathcal{W}},\gamma^{\tt mpe}_{\tilde{\mathcal{W}}}<0} \gamma^{\tt mpe}_{\tilde{\mathcal{W}}}\end{aligned} \tag{5.7}$$

where the MPE word sequence occupancy is in the same form as in equation 4.22,

$$\gamma^{\tt mpe}_{\tilde{\mathcal{W}}} \;=\; P(\tilde{\mathcal{W}}|\mathcal{O},\tilde{\lambda})\left[\mathcal{A}(\tilde{\mathcal{W}},\mathcal{W}) - \mathcal{F}_{\tt mpe}(\tilde{\lambda})\right] \tag{5.8}$$

and $\mathcal{A}(\tilde{\mathcal{W}},\mathcal{W})$, as discussed in section 4.2.3, is the the phone level accuracy of a word sequence, $\tilde{\mathcal{W}}$, against the reference transcription, $\mathcal{W}$. This smoothing criterion has the attributes discussed in section 5.3, as the effect of word sequences whose accuracy are below the average level is reduced. However, it should be noted that using this form of smoothing criterion, no data will be removed. Instead, only the accuracy contribution from highly erroneous recognition hypotheses will be reduced. In addition the term outside the bracket in the MPE growth function, $p(\mathcal{O}|\lambda)$, is associated with the likelihood of a sentence and will further reduce the sensitivity to outliers.

Direct marginalization of the growth function in equation 5.6 may be difficult for HMM based speech recognition systems, due to the dependency upon latent variables making it highly inefficient for complexity control. An approach similar to that discussed in section 3.3.4 is therefore used. The following lower bound for the MPE growth function may be derived using an EM-like approach. A detailed proof can be found in appendix A.

$$\mathcal{L}_{\tt mpe}(\lambda,\tilde{\lambda}) \;=\; \log\mathcal{G}(\tilde{\lambda}) + \frac{\mathcal{Q}_{\tt mpe}(\lambda,\tilde{\lambda}) - \mathcal{Q}_{\tt mpe}(\tilde{\lambda},\tilde{\lambda})}{\sum_{j,\tau}\gamma^{\tt mpe}_j(\tau)} \tag{5.9}$$

where the MPE "auxiliary function" is given by [1]

$$\mathcal{Q}_{\tt mpe}(\lambda,\tilde{\lambda}) \;=\; \sum_{j,\tau}\gamma^{\tt mpe}_j(\tau)\log p(\boldsymbol{o}_\tau|\boldsymbol{\psi}_\tau = \mathcal{S}_j,\lambda) \tag{5.10}$$

[1] Only the optimization of Gaussian means and variances are considered.

and $\gamma_j^{\texttt{mpe}}(\tau)$ is the MPE hidden state occupancy.

The calculation of the growth function lower bound requires the MPE occupancy statistics $\{\gamma_j^{\texttt{mpe}}(\tau)\}$. For the MPE growth function, the hidden state occupancy $\gamma_j^{\texttt{mpe}}(\tau)$ in equation 5.10 is given by [75]

$$\begin{aligned}\gamma_j^{\texttt{mpe}}(\tau) &= \gamma_j^{\texttt{num}}(\tau) - \gamma_j^{\texttt{den}}(\tau)\\ &\quad -C \sum_{\tilde{\mathcal{W}}, \gamma_{\tilde{\mathcal{W}}}^{\texttt{mpe}} < 0} P(\psi_\tau = \mathcal{S}_j | \mathcal{O}, \tilde{\mathcal{W}}, \tilde{\lambda}) \gamma_{\tilde{\mathcal{W}}}^{\texttt{mpe}}\end{aligned} \tag{5.11}$$

The numerator and denominator occupancies are given by

$$\begin{aligned}\gamma_j^{\texttt{num}}(\tau) &= \sum_{\tilde{\mathcal{W}}} P(\psi_\tau = \mathcal{S}_j | \mathcal{O}, \tilde{\mathcal{W}}, \tilde{\lambda}) \gamma_{\tilde{\mathcal{W}}}^{\texttt{mpe}} \quad (\gamma_{\tilde{\mathcal{W}}}^{\texttt{mpe}} \geq 0)\\ \gamma_j^{\texttt{den}}(\tau) &= -\sum_{\tilde{\mathcal{W}}} P(\psi_\tau = \mathcal{S}_j | \mathcal{O}, \tilde{\mathcal{W}}, \tilde{\lambda}) \gamma_{\tilde{\mathcal{W}}}^{\texttt{mpe}} \; (\gamma_{\tilde{\mathcal{W}}}^{\texttt{mpe}} < 0).\end{aligned} \tag{5.12}$$

A detailed derivation of the above statistics may be found in appendix A. It is interesting to compare the MPE occupancy derived from the growth function, given in equation 5.11, with the standard form used in LVCSR MPE training [93] given in equation 4.20 and the smoothing term in equation 4.24. Combining these two gives

$$\begin{aligned}\gamma_j^{\texttt{mpe}}(\tau) &= \gamma_j^{\texttt{num}}(\tau) - \gamma_j^{\texttt{den}}(\tau)\\ &\quad -E \sum_{\tilde{\mathcal{W}}, \gamma_{\tilde{\mathcal{W}}}^{\texttt{mpe}} < 0} P(\psi_\tau = \mathcal{S}_j | \mathcal{O}, \tilde{\mathcal{W}}, \tilde{\lambda}) \gamma_{\tilde{\mathcal{W}}}^{\texttt{mpe}}\end{aligned} \tag{5.13}$$

where a constant $E > 0$ is empirically tuned. These two forms of MPE occupancy are equivalent to one another when $E = C$. However, the two smoothing terms serve very different purposes. The smoothing term in the standard MPE occupancy, in equation 5.13, ensures a stable convergence during training, whereas the smoothing term derived from the growth function helps reduce the sensitivity to outliers sentences with high error rates.

The following lower bound marginalization is then used for complexity control.

$$\hat{\mathcal{M}} = \arg\max_{\mathcal{M}} \int \exp\left(\mathcal{L}_{\texttt{mpe}}(\lambda, \tilde{\lambda})\right) p(\lambda|\mathcal{M}) d\lambda \tag{5.14}$$

Although the dependency upon latent variables has been removed for the growth function lower bound, the marginalization in equation 5.14 is still non-trivial. To solve this problem, the integral in equation 5.14 may be computed using approximation schemes for Bayesian evidence as discussed in chapter 3. As the BIC based first order approximation can not count for different forms of model parameters, the second order Laplace's approximation is used to compute the growth function marginalization.

The growth function lower bound in equation 5.9 has a similar form to the log-likelihood bound in equation 3.7. Both may be expressed as the value of the underlying objective function at the current parameter estimate, $\tilde{\lambda}$, plus a second term that is related to the difference in auxiliary functions. In the same fashion as the log-likelihood bound, for efficiency multiple complexity configurations may make use of a single set of statistics. In this case, the only term that will determine the rank-ordering of the systems will be the MPE auxiliary function, $\mathcal{Q}_{\tt mpe}(\lambda, \tilde{\lambda})$. When determining the number of components, for example, sufficient statistics for systems with fewer components per state may be obtained by merging appropriate statistics together from a more complex system. The form of statistic merging used in this work is discussed in more detail in the later sections. One important aspect for both this discriminative bound and the log-likelihood bound is the accuracy of the derived statistics. As the differences between the model used to derive the statistics and the model being considered increases, the bound may become increasingly loose and the performance ranking increasingly poor. To reduce this effect an upper limit on the level of structural mutation, or change of model complexity, allowed from the system used to derive the statistics may be enforced. This is discussed in more detail in the following sections.

Another issue with using growth functions for complexity control is the setting of the regularization constant C. The setting of this constant has two effects. First, it controls the contribution from the smoothing term of the MPE occupancy, given in equation 5.11, to reduce the sensitivity to outliers. Second, the setting of C may affect the selection of the optimal configuration, and the speed of structural mutation from the current model. In a similar fashion as in standard MPE training, in order to ensure the stability during model complexity optimization, this constant needs to be appropriately set. For all the experiments in this paper the value of C was set to 2.0 and not altered. This is also a standard value used for MPE training [93].

5.5 MMI Growth Function

Although the MMI criterion is an approximation to the classification error on a sentence level, it is still interesting to find an appropriate form of MMI growth function for complexity control. The MMI growth function considered here is given by

$$\mathcal{G}(\lambda) \quad = \quad p(\mathcal{O}|\lambda)\left[\mathcal{F}_{\tt mmi}(\lambda) - \mathcal{F}_{\tt mmi}(\tilde{\lambda}) + C\mathcal{F}_{\tt sm}(\lambda, \tilde{\lambda})\right] \qquad (5.15)$$

where the smoothing criterion $\mathcal{F}_{\tt sm}(\lambda, \tilde{\lambda})$ is given by

$$\mathcal{F}_{\tt sm}(\lambda, \tilde{\lambda}) \quad = \quad P(\mathcal{W}|\mathcal{O}, \tilde{\lambda}) \qquad (5.16)$$

This smoothing function is equivalent to the MMI criterion evaluated at the current parameter estimates $\tilde{\lambda}$. Similar to the smoothing criterion for the MPE growth function in equation 5.7, this form of $\mathcal{F}_{\tt sm}(\lambda, \tilde{\lambda})$ also has the attributes discussed in section 5.3. As discussed in section 4.2.1, the MMI criterion, or the posterior probability of the reference transcription, is an approximation to the sentence error rate. Hence utterances with higher error rates on a sentence level may be penalized using this form of smoothing criterion. Furthermore, the term outside the bracket in the MMI growth function, $p(\mathcal{O}|\lambda)$, is associated with the likelihood of a sentence and may further reduce such sensitivity.

Like the MPE growth function in section 5.4, a direct marginalization of the growth function in equation 5.15 may be difficult for HMMs, due to the dependency upon latent variables. Again for efficiency a lower bound based approach similar to that discussed in section 3.3.4 is used. Using an EM-like approach, a lower bound for the MMI growth function may be given by

$$\mathcal{L}_{\texttt{mmi}}(\lambda, \tilde{\lambda}) \quad = \quad \log \mathcal{G}(\tilde{\lambda}) + \frac{\mathcal{Q}_{\texttt{mmi}}(\lambda, \tilde{\lambda}) - \mathcal{Q}_{\texttt{mmi}}(\tilde{\lambda}, \tilde{\lambda})}{\sum_{j,\tau} \gamma_j^{\texttt{mmi}}(\tau)} \tag{5.17}$$

where the MMI "auxiliary" function is given by [2]

$$\mathcal{Q}_{\texttt{mmi}}(\lambda, \tilde{\lambda}) \quad = \quad \sum_{j,\tau} \gamma_j^{\texttt{mmi}}(\tau) \log p(\boldsymbol{o}_\tau | \boldsymbol{\psi}_\tau = \mathcal{S}_j, \lambda) \tag{5.18}$$

and $\gamma_j^{\texttt{mmi}}(\tau)$ is the MMI hidden state occupancy. A detailed proof may be found in appendix B.

It is interesting that the MMI growth function bound has some similar features to those of the MPE growth function bound discussed in section 5.4. First, the MMI statistics, $\{\gamma_j^{\texttt{mmi}}(\tau)\}$, required to compute the growth function lower bound in equation 5.17, are closely related to the standard form of statistics used for MMI training. For the MMI growth function, the statistics $\gamma_j^{\texttt{mmi}}(\tau)$ in equation 5.10 is given by [75]

$$\gamma_j^{\texttt{mmi}}(\tau) \quad = \quad \gamma_j^{\texttt{num}}(\tau) - \gamma_j^{\texttt{den}}(\tau) + CP(\boldsymbol{\psi}_\tau = \mathcal{S}_j | \mathcal{O}, \tilde{\lambda}) \tag{5.19}$$

where the numerator and denominator occupancies are in the same form as in equation 4.19.

$$\begin{aligned} \gamma_j^{\texttt{num}}(\tau) \quad &= \quad P(\boldsymbol{\psi}_\tau = \mathcal{S}_j | \mathcal{O}, \mathcal{W}, \tilde{\lambda}) \\ \gamma_j^{\texttt{den}}(\tau) \quad &= \quad P(\boldsymbol{\psi}_\tau = \mathcal{S}_j | \mathcal{O}, \tilde{\lambda}) \end{aligned} \tag{5.20}$$

A detailed derivation of the above statistics may be found in appendix B. The standard form of MMI statistics [124, 93] for discriminative training is given in equation 4.20 and the smoothing term in equation 4.24. This may be written as

$$\gamma_j^{\texttt{mmi}}(\tau) \quad = \quad \gamma_j^{\texttt{num}}(\tau) - \gamma_j^{\texttt{den}}(\tau) + EP(\boldsymbol{\psi}_\tau = \mathcal{S}_j | \mathcal{O}, \tilde{\lambda}) \tag{5.21}$$

where the smoothing constant $E > 0$ is empirically tuned. These two forms of MMI occupancies are equivalent to one another when $E = C$. The second similarity between the MMI and MPE growth function lower bounds is that multiple configurations may make use of a single set of statistics for greater efficiency. In this case, the only term that affects model selection will be the MMI auxiliary function, $\mathcal{Q}_{\texttt{mmi}}(\lambda, \tilde{\lambda})$. In order to obtain a good performance ranking, it is important to tighten the bound by using reliable statistics. Third, the setting of the smoothing constant C is also an issue for the MMI growth function. The setting of C has the same two effects as discussed in section 5.4 for the MPE growth function. Again in common with the standard C setting used for MMI training, the value of C was always set to 2.0 for MMI growth functions in the experiments.

[2] Here only Gaussian means and variances are considered.

The following lower bound marginalization is then used for complexity control.

$$\hat{\mathcal{M}} = \arg\max_{\mathcal{M}} \int \exp\left(\mathcal{L}_{\tt mmi}(\lambda, \tilde{\lambda})\right) p(\lambda|\mathcal{M}) d\lambda \tag{5.22}$$

The marginalization in equation 5.22 may be difficult for HMMs in many practical situations, though the dependency upon latent variables has been removed. In order to compute the integral more efficiently, Laplace's approximation may be used, as with the marginalization of the ML bound in equation 3.11, and the MPE growth function bound in equation 5.14.

5.6 Implementation Issues

In this section several implementation issues when using marginalized discriminative growth functions for model complexity control are discussed. These issues are important and may affect the performances of complexity controlled systems.

5.6.1 Sharing Statistics among Model Structures

For LVCSR systems exhaustively accumulating the sufficient statistics for each possible system is highly inefficient. When determining the number of Gaussian components in a state, it is impractical to obtain new statistics for each number of components, even if the state alignments are fixed. To handle this problem, as discussed in sections 3.3.4, 5.4 and 5.5, the same set of statistics may be used for a range of model structures. As it is only possible to merge statistics, the number of components, or other complexity control attributes, can only be reduced. For this merging process, the statistics from a pair of Gaussians must be combined to form a single Gaussian. This is a standard problem and is solved by simply combining the appropriate first, or second, order statistics and the occupancy counts. For example, when joining component j and k to yield l, the MPE statistics are merged as

$$\gamma_l^{\tt mpe}(\tau) = \gamma_j^{\tt mpe}(\tau) + \gamma_k^{\tt mpe}(\tau). \tag{5.23}$$

This same holds for the first and second order statistics.

$$\begin{aligned} \sum_\tau \gamma_l^{\tt mpe}(\tau)\boldsymbol{o}_\tau &= \sum_\tau \gamma_j^{\tt mpe}(\tau)\boldsymbol{o}_\tau + \sum_\tau \gamma_k^{\tt mpe}(\tau)\boldsymbol{o}_\tau \\ \sum_\tau \gamma_l^{\tt mpe}(\tau)\boldsymbol{o}_\tau\boldsymbol{o}_\tau^\top &= \sum_\tau \gamma_j^{\tt mpe}(\tau)\boldsymbol{o}_\tau\boldsymbol{o}_\tau^\top + \sum_\tau \gamma_k^{\tt mpe}(\tau)\boldsymbol{o}_\tau\boldsymbol{o}_\tau^\top. \end{aligned} \tag{5.24}$$

Similar merging will also be performed for the ML statistics of the log-likelihood lower-bound discussed in section 3.3.4, and the MMI statistics for the MMI growth function in section 5.5. In the majority of the cases considered in this work, the mean and covariance of merged component l are estimated in an ML fashion using the merged ML statistics. However, if the mean and covariance of component l are discriminatively updated, the merged sufficient MPE or MMI statistics may be required. This is an interesting scenario where a consistently discriminative optimization of both model complexity and parameters is performed. This case will be further discussed and investigated in the experiments of following chapters. All possible pairs of component merging are considered. The pair with the largest increase in the objective function is selected.

5.6.2 Constrained Maximum Structural Mutation

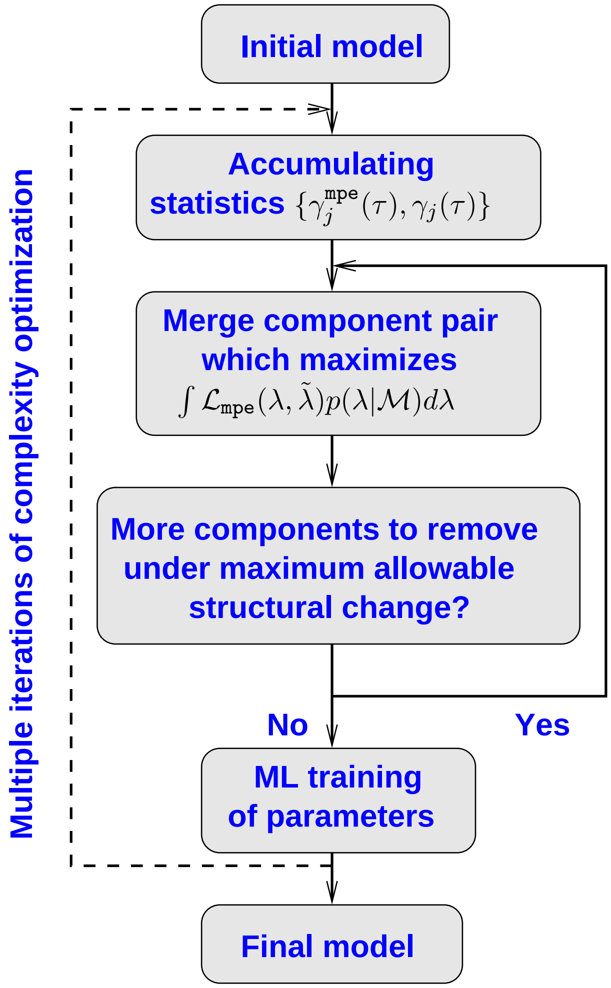
Initial model
Accumulating statistics $\{\gamma_j^{\mathtt{mpe}}(\tau), \gamma_j(\tau)\}$
Merge component pair which maximizes $\int \mathcal{L}_{\mathtt{mpe}}(\lambda, \tilde{\lambda}) p(\lambda|\mathcal{M}) d\lambda$
More components to remove under maximum allowable structural change?
No
Yes
ML training of parameters
Final model
Multiple iterations of complexity optimization

Figure 5.1 *Selecting the number of Gaussian components per state using marginalized MPE growth functions via component merging*

For efficiency, the lower bound of a discriminative growth function, or log-likelihood, is derived from the statistics of a single system as discussed in sections 3.3.4, 5.4 and 5.5. As discussed in section 5.4, when the magnitude of the structural mutation from the current model increases, the reliability of the fixed statistics decreases, and looser the bound. This may lead to a poor selection of model complexity. To overcome this problem, the whole structural optimization process can be performed in an iterative mode. An overview of the algorithm, when using marginalized MPE growth functions to select the number of Gaussians per state, is shown in figure 5.1. A maximum mutation limit in the model complexity is imposed. For instance, the maximum number of Gaussians that may be removed from any state per iteration is constrained. In this work the maximum mutation was set to be 2 for all experiments. Between iterations of structural optimization, model parameters were re-estimated using ML training to obtain improved statistics. Slightly modifying the procedure illustrated in figure 5.1, it may also be applied to BIC. This requires that the growth function integral in the third box to be replaced with the BIC metric in equation 3.3, and the lower bound in equation 3.7 is used to approximate the log-likelihood. In all experiments a total of four iterations of complexity control were performed for both BIC and marginalized growth function systems. For multiple HLDA systems, varying the number of useful dimensions per Gaussian will have a far less impact on component alignments, compared with varying the number Gaussians per state. Thus the sufficient statistics may be assumed to be the same for all possible number of retained dimensions and no constraint on the complexity variation is required.

5.6.3 Hessian Approximation for HLDA Systems

The lower bound marginalization for discriminative growth functions in equation 5.9 and 5.17, and the log-likelihood in equation 3.11 may be approximated via Laplace's approximation. This approximation requires the storage of a Hessian matrix with respect to all the model parameters. However, because the number of model parameters in an LVCSR system can be in the millions, the storage and calculation of the Hessian as a full matrix is impractical. To solve this problem, assumptions can be made about the structure of the Hessian. In particular, by assuming that the Hessian has a block diagonal structure [71, 70, 75] the problem becomes tractable. This form of Hessian approximation can be used for both the discriminative and log-likelihood lower bounds. The exact form of the approximated Hessian depends on that of the lower bound being considered. Let

$$\check{\boldsymbol{o}}_\tau^{(r_j)} \quad = \quad \boldsymbol{A}^{(r_j)}\boldsymbol{o}_\tau \tag{5.25}$$

denote the projected feature after the HLDA transform, $\boldsymbol{A}^{(r_j)}$, to which component j is assigned. Let $\check{\boldsymbol{\mu}}^{(j)}$, $\check{\boldsymbol{\Sigma}}^{(j)}$ denote the component means and covariances in the transformed space. Take the MPE lower bound in equation 5.9 as an example. The MPE auxiliary function in equation 5.10 may be expressed as

$$\begin{aligned}\mathcal{Q}_{\mathtt{mpe}}(\lambda,\tilde{\lambda}) \quad &= \quad \frac{1}{2}\sum_{j,\tau}\gamma_j^{\mathtt{mpe}}(\tau)\left\{\log\left|\boldsymbol{A}^{(r_j)}\right|^2 - \log\left|\check{\boldsymbol{\Sigma}}^{(j)}\right|\right.\\ &\qquad \left. - \left(\check{\boldsymbol{o}}_\tau^{(r_j)} - \check{\boldsymbol{\mu}}^{(j)}\right)^\top \check{\boldsymbol{\Sigma}}^{(j)-1}\left(\check{\boldsymbol{o}}_\tau^{(r_j)} - \check{\boldsymbol{\mu}}^{(j)}\right)\right\}.\end{aligned} \tag{5.26}$$

Each Gaussian component is assumed to be independent of all others. Furthermore, within each Gaussian component, the mean, variance and each row of the HLDA transforms are also assumed independent of each other. For the integral over the growth function's lower bound in equation 5.14, the log-determinant of the Hessian matrix may be approximated as

$$\begin{aligned}\log\left|-\nabla^2_{\lambda}\mathcal{Q}_{\texttt{mpe}}(\lambda,\tilde{\lambda})\right| \quad &\approx \quad \sum_{r,i}\log\left|-\frac{\partial^2\mathcal{Q}_{\texttt{mpe}}(\lambda,\tilde{\lambda})}{\partial^2\boldsymbol{a}_i^{(r)}}\right| + \sum_{j}\log\left|-\frac{\partial^2\mathcal{Q}_{\texttt{mpe}}(\lambda,\tilde{\lambda})}{\partial^2\check{\boldsymbol{\mu}}^{(j)}}\right| \\ &\quad + \sum_{j}\log\left|-\frac{\partial^2\mathcal{Q}_{\texttt{mpe}}(\lambda,\tilde{\lambda})}{\partial^2\check{\boldsymbol{\Sigma}}^{(j)}}\right|. \end{aligned} \tag{5.27}$$

The second order differentials are derived from equation 5.26 and yield [75].

$$\begin{aligned}\frac{\partial^2\mathcal{Q}_{\texttt{mpe}}(\lambda,\tilde{\lambda})}{\partial^2\check{\boldsymbol{\mu}}^{(j)}} \quad &= \quad -\frac{1}{2}\sum_{\tau}\gamma_j^{\texttt{mpe}}(\tau)\check{\boldsymbol{\Sigma}}^{(j)-1} \\ \frac{\partial^2\mathcal{Q}_{\texttt{mpe}}(\lambda,\tilde{\lambda})}{\partial^2\check{\boldsymbol{\Sigma}}^{(j)}} \quad &= \quad -\frac{1}{2}\sum_{\tau}\gamma_j^{\texttt{mpe}}(\tau)\left[2\texttt{diag}\left(\left(\check{\boldsymbol{o}}_{\tau}^{(r_j)}-\check{\boldsymbol{\mu}}^{(j)}\right)\left(\check{\boldsymbol{o}}_{\tau}^{(r_j)}-\check{\boldsymbol{\mu}}^{(j)}\right)^{\top}\right)\check{\boldsymbol{\Sigma}}^{(j)-3}-\check{\boldsymbol{\Sigma}}^{(j)-2}\right] \\ \frac{\partial^2\mathcal{Q}_{\texttt{mpe}}(\lambda,\tilde{\lambda})}{\partial^2\boldsymbol{a}_i^{(r)}} \quad &= \quad -\frac{\boldsymbol{c}_i^{(r)}\boldsymbol{c}_i^{(r)\top}}{\left|\boldsymbol{A}^{(r)}\right|^2}\sum_{j\in r,\tau}\gamma_j^{\texttt{mpe}}(\tau)-\boldsymbol{G}^{(r,i)}\end{aligned} \tag{5.28}$$

where $\boldsymbol{c}_i^{(r)}$ denotes the cofactor vector associated with row $\boldsymbol{a}_i^{(r)}$ and the transform specific statistics $\left\{\boldsymbol{G}^{(r,i)}\right\}$ are accumulated on a row by row basis. Take the useful dimensions for example, this gives

$$\boldsymbol{G}^{(r,i)} \quad = \quad \sum_{j\in r,\tau}\frac{\gamma_j^{\texttt{mpe}}(\tau)}{\check{\sigma}_i^{(j)2}}\left(\boldsymbol{o}_{\tau}-\boldsymbol{\mu}^{(j)}\right)\left(\boldsymbol{o}_{\tau}-\boldsymbol{\mu}^{(j)}\right)^{\top} \tag{5.29}$$

where again the transformed component covariance, $\check{\boldsymbol{\Sigma}}^{(j)}$, is constrained to be diagonal as in section 2.4.3, and $\check{\sigma}_i^{(j)2}$ is the ith dimensional variance element in the transformed space given by $\boldsymbol{A}^{(r_j)}$. As $\boldsymbol{G}^{(r,i)}$ is accumulated using statistics from the original feature-space, there is no need to perform statistic merging as described in section 5.6.1 for multiple Gaussian components. The same statistics can be used to generate a range of sizes of useful dimension. Note that this assumes that the assignment of component to transform is fixed, which is the situation considered in this work.

5.7 Summary

The majority of current complexity control schemes can be described within the maximum likelihood paradigm. Unfortunately, the model correctness assumption made in these standard techniques may be too strong for current speech recognition systems using HMMs. Hence it is preferable to employ discriminative criteria for complexity control. These criteria are more directly related to the recognition error, rather than to the likelihood. In this chapter a novel model complexity control technique has been proposed, using the marginalization of a discriminative growth function. The discriminative growth functions investigated were closely related to the MPE and MMI criteria, but have a reduced sensitivity to outliers utterances. For efficiency an EM-like approach was used to derive tractable lower bounds of the growth functions, with the dependency on latent variables removed. This lower bound was then marginalized efficiently using Laplace's approximation for complexity control.

6

Discriminative Training of Linear Projections

In chapter 5, a discriminative model selection technique based on the marginalization of a growth function was presented. Using this method the complexity control problem for systems using linear projections such as HLDA was discussed. In this chapter, the discriminative training of linear projection schemes is presented. First, the motivation for developing discriminative training algorithms for linear projections is discussed. Second, previous research on discriminative training of linear transformation schemes for speech recognition is briefly reviewed. As the EBW algorithm may be only used to optimize standard forms of HMM parameters, a more general form of discriminative criteria optimization is preferred. The proposed method is based on the optimization of a weak-sense auxiliary function. Using this method the discriminative training of linear projection schemes is investigated. Finally some implementations issues when estimating linear projections discriminatively are also discussed.

6.1 Introduction and Motivation

For any pattern recognition task an important aspect of the problem is the derivation of a good and compact feature representation. This representation should contain sufficient discriminant information to minimize the classification error. One family of techniques that may be used for this purpose is the linear projection schemes discussed in section 2.4. However, one limitation with these techniques is that that projections are normally trained using the ML criterion. As discussed in chapter 4, an inherent model correctness assumption is made in ML training of current ASR systems based on HMMs. HMMs are assumed to be the "correct" models for speech signals. This is untrue for current speech recognition systems using HMMs, as explained in section 4.1. When the correlation between the WER and likelihood is weak, merely increasing the likelihood on the observed, or unseen data, does not necessarily improve the recognition performance. Hence it is preferable to employ discriminative criteria, which are more explicitly related to the recognition error, to estimate linear projections. The ultimate aim of linear projection schemes for speech recognition is to obtain a good feature representation that minimizes the WER.

Most state-of-the-art LVCSR systems are built using discriminative training techniques [124, 51, 23, 64]. As discussed in section 4.3.4, gradient descent base numerical techniques are expensive for LVCSR training and have difficulty guaranteeing convergence. The commonly used EBW algorithm provides an iterative, efficient, EM-like optimization for discriminative training criteria. However, using the EBW algorithm only standard forms of HMM parameters may be optimized [84, 112, 124, 93]. These include state transitions, Gaussian component priors, means and covariances. Since the EBW algorithm may not be directly used to estimate linear projections, it is useful to have a more general approach, to discriminatively optimize a variety of forms of model parameters including linear projections. The weak-sense auxiliary function described in section 4.3.2 is one such approach. It provides a flexible and heuristic derivation of the EBW algorithm, and may be generalized to a variety of forms of parameters [115, 108]. Hence, rather than using gradient descent techniques as proposed in [134], weak-sense auxiliary functions are used in this chapter for the discriminative estimation of linear projections.

6.2 Previous Work for Speech Recognition

In recent years there has been active research on discriminative training of linear transformation schemes for speech recognition. In particular the discriminative training of linear transformations have been studied for a feature projection and diagonalizing purpose. Using a discriminative criterion, multiple feature space transformations were investigated in [94]. However the training of these linear transformations and other HMM parameters was not integrated into a consistent discriminative framework. After the estimation of the transforms, the other HMM parameters were still trained using the ML criterion. More importantly the likelihood computation across different subspaces associated with each linear transformation was not directly comparable in [94]. This was because the Jacobian normalization term was ignored for each transform. Recently a novel linear feature projection, called *fMPE*, was proposed in [92]. The fMPE transform operates by projecting from a very high dimensional, sparse feature space derived from Gaussian posteriors to the normal feature space and adding the projected posteriors to the standard features. A global non-square matrix is trained to maximize the MPE criterion via gradient descent based numerical methods. Significant WER improvement have been reported on LVCSR tasks.

Another related area has been focused on the discriminative training of linear transformations for speaker normalization and adaptation [48, 80, 115, 116, 20]. Although these techniques are used for a very different purpose from the projection schemes considered here, some of them may be expressed as feature space linear transformations. The optimization of them may be closely related to those of linear projections [30, 31, 34]. This area of research considers the estimation of MLLR transforms using a discriminative criterion, instead of ML training as described in section 2.5. These transforms may then be used for speaker adaptive training (SAT). During the discriminative training of a SAT system, the common adopted approach is a "hybrid" procedure. This idea is to use the EBW algorithm to discriminatively update standard HMM parameters, whilst the previously ML estimated MLLR transforms are fixed [51, 23]. In contrast, when using discriminative criteria to estimate MLLR transforms, the entire training is in a consistent discriminative framework. In [20, 115] using a consistent optimization of both the MLLR transforms and HMM parameters of SAT systems, WER improvements were obtained over the "hybrid" approach on LVCSR tasks.

6.3 Discriminative Training of Projection Schemes

In this section the estimation of linear projections are presented using weak-sense auxiliary functions presented in section 4.3.2. As most state-of-the-art LVCSR systems are trained using the MPE criterion [51, 23, 64], the MPE training of linear projection schemes is the focus of this section.

6.3.1 MPE Training of multiple HLDA

The relationship between HLDA and multiple HLDA was discussed in section 2.4.3. HLDA is subsumed by multiple HLDA as a special case when a global subspace is used. Hence the MPE training of multiple HLDA projections are be considered here. The approach adopted here is to examine the weak-sense auxiliary function's gradient against parameters of HLDA projections.

Using the general form of smoothing term in equation 4.28, the weak-sense auxiliary function in equation 4.25 may be expressed as

$$\begin{aligned}\mathcal{Q}(\lambda,\tilde{\lambda}) &= \sum_{j,\tau}\left(\gamma_j^{\texttt{num}}(\tau)-\gamma_j^{\texttt{den}}(\tau)\right)\log p(\boldsymbol{o}_\tau|\boldsymbol{\psi}_\tau=\mathcal{S}_j,\lambda)\\ &\quad+\sum_j D_j\int p(\boldsymbol{o}|\boldsymbol{\psi}_{\boldsymbol{O}}=\mathcal{S}_j,\tilde{\lambda})\log p(\boldsymbol{o}|\boldsymbol{\psi}_{\boldsymbol{O}}=\mathcal{S}_j,\lambda)d\boldsymbol{o}. \qquad (6.1)\end{aligned}$$

Let $\boldsymbol{A}^{(r)}$ denote the rth HLDA transform. the gradient of the weak-sense auxiliary function in equation 6.1 around the current parameter estimate, $\tilde{\lambda}$, with respect to $\boldsymbol{a}_i^{(r)}$, the ith row of $\boldsymbol{A}^{(r)}$, is given by

$$\begin{aligned}\left.\frac{\partial\mathcal{Q}(\lambda,\tilde{\lambda})}{\partial\boldsymbol{a}_i^{(r)}}\right|_{\lambda=\tilde{\lambda}} &= \sum_{j\in r,\tau}\left(\gamma_j^{\texttt{num}}(\tau)-\gamma_j^{\texttt{den}}(\tau)\right)\left.\frac{\partial\log p(\boldsymbol{o}_\tau|\boldsymbol{\psi}_\tau=\mathcal{S}_j,\lambda)}{\partial\boldsymbol{a}_i^{(r)}}\right|_{\lambda=\tilde{\lambda}}\\ &\quad+\sum_{j\in r}D_j\int p(\boldsymbol{o}|\boldsymbol{\psi}_{\boldsymbol{O}}=\mathcal{S}_j,\tilde{\lambda})\left.\frac{\partial\log p(\boldsymbol{o}|\boldsymbol{\psi}_{\boldsymbol{O}}=\mathcal{S}_j,\lambda)}{\partial\boldsymbol{a}_i^{(r)}}\right|_{\lambda=\tilde{\lambda}}d\boldsymbol{o}. \qquad (6.2)\end{aligned}$$

Note that the model structure, $\mathcal{M}$, of the weak-sense auxiliary function in equation 4.25 is omitted for clarity, as only the optimization of model parameters is considered.

In order to further simplify the above, the gradient of the frame Gaussian log likelihood, $\log p(\boldsymbol{o}_\tau|\boldsymbol{\psi}_\tau=\mathcal{S}_j,\lambda)$, against rows of HLDA transforms is required. Let $\boldsymbol{c}_i^{(r)}$ denote the cofactor vector of $\boldsymbol{a}_i^{(r)}$ [1], and $\breve{\sigma}_i^{(j)}$ the variance elements of component j in the projected space. For

[1] Assume the HLDA transform rows have been re-ordered so that the nuisance dimensions always correspond to the last $n-p$ rows.

multiple HLDA systems, the log likelihood of an observation, $\boldsymbol{o}_\tau$, given a Gaussian component j that is assigned to projection r, $j \in r$, may be written as [31, 34]

$$\begin{aligned}\log p(\boldsymbol{o}_\tau|\boldsymbol{\psi}_\tau = \mathcal{S}_j, \lambda) \;=\; & \frac{1}{2}\Bigg[\log\left(\boldsymbol{a}_i^{(r)}\boldsymbol{c}_i^{(r)}\right)^2 - n\log 2\pi - \log\left|\check{\boldsymbol{\Sigma}}^{(j)}\right| \\ & - \sum_{i\le p}\left(\boldsymbol{o}_\tau - \boldsymbol{\mu}^{(j)}\right)^\top \boldsymbol{a}_i^{(r)\top}\check{\sigma}_i^{(j)-2}\boldsymbol{a}_i^{(r)}\left(\boldsymbol{o}_\tau - \boldsymbol{\mu}^{(j)}\right) \\ & - \sum_{i>p}\left(\boldsymbol{o}_\tau - \boldsymbol{\mu}^{(g,r)}\right)^\top \boldsymbol{a}_i^{(r)\top}\check{\sigma}_i^{(j)-2}\boldsymbol{a}_i^{(r)}\left(\boldsymbol{o}_\tau - \boldsymbol{\mu}^{(g,r)}\right)\Bigg] \end{aligned} \tag{6.3}$$

where $\boldsymbol{\Sigma}^{(g,r)}$ denotes the global covariance for transforms class r. Differentiating equation 6.3 with respect to $\boldsymbol{a}_i^{(r)}$ yields

$$\begin{aligned}\frac{\partial \log p(\boldsymbol{o}_\tau|\boldsymbol{\psi}_\tau = \mathcal{S}_j, \lambda)}{\partial \boldsymbol{a}_{i,i\le p}^{(r)}} &= \frac{\boldsymbol{c}_i^{(r)\top}}{\boldsymbol{a}_i^{(r)}\boldsymbol{c}_i^{(r)}} - \boldsymbol{a}_i\frac{\left(\boldsymbol{o}_\tau - \boldsymbol{\mu}^{(j)}\right)\left(\boldsymbol{o}_\tau - \boldsymbol{\mu}^{(j)}\right)^\top}{\check{\sigma}_i^{(j)2}} \\ \frac{\partial \log p(\boldsymbol{o}_\tau|\boldsymbol{\psi}_\tau = \mathcal{S}_j, \lambda)}{\partial \boldsymbol{a}_{i,i>p}^{(r)}} &= \frac{\boldsymbol{c}_i^{(r)\top}}{\boldsymbol{a}_i^{(r)}\boldsymbol{c}_i^{(r)}} - \boldsymbol{a}_i\frac{\left(\boldsymbol{o}_\tau - \boldsymbol{\mu}^{(g,r)}\right)\left(\boldsymbol{o}_\tau - \boldsymbol{\mu}^{(g,r)}\right)^\top}{\check{\sigma}_i^{(j)2}}. \end{aligned} \tag{6.4}$$

Substituting the gradient in equation 6.4 into equation 6.2 gives

$$\left.\frac{\partial \mathcal{Q}(\lambda, \tilde{\lambda})}{\partial \boldsymbol{a}_i^{(r)}}\right|_{\lambda=\tilde{\lambda}} = \left[\sum_{j\in r,\tau}(\gamma_j^{\texttt{num}}(\tau) - \gamma_j^{\texttt{den}}(\tau)) + \sum_{j\in r} D_j\right]\frac{\tilde{\boldsymbol{c}}_i^{(r)\top}}{\tilde{\boldsymbol{a}}_i^{(r)}\tilde{\boldsymbol{c}}_i^{(r)}} - \tilde{\boldsymbol{a}}_i^{(r)}\boldsymbol{G}^{(r,i)} \tag{6.5}$$

where the sufficient discriminative statistics, $\boldsymbol{G}^{(r,i)}$, are accumulated for each transform class on a row by row basis

$$\boldsymbol{G}^{(r,i)} = \begin{cases} \sum_{j\in r}\check{\sigma}_i^{(j)-2}\left[\sum_\tau(\gamma_j^{\texttt{num}}(\tau) - \gamma_j^{\texttt{den}}(\tau)) + D_j\right]\boldsymbol{\Sigma}^{(j)} & i \text{ is retained} \\ \sum_{j\in r}\check{\sigma}_i^{(j)-2}\left[\sum_\tau(\gamma_j^{\texttt{num}}(\tau) - \gamma_j^{\texttt{den}}(\tau)) + D_j\right]\boldsymbol{\Sigma}^{(g,r)} & i \text{ is nuisance} \end{cases} \tag{6.6}$$

and $\boldsymbol{\Sigma}^{(j)}$ is the discriminatively updated full covariance using the EBW algorithm in equation 4.16, and $\boldsymbol{\Sigma}^{(g,r)}$ the transform specific global covariance updated using the statistics of all components within class r. A detailed derivation of the above may be found in appendix C.

The aim is to zero the weak sense function's gradient in equation 6.5 to find the optimal estimate for $\boldsymbol{a}_i^{(r)}$. This yields

$$\left[\sum_{j\in r,\tau}(\gamma_j^{\texttt{num}}(\tau) - \gamma_j^{\texttt{den}}(\tau)) + \sum_{j\in r} D_j\right]\frac{\tilde{\boldsymbol{c}}_i^{(r)\top}}{\boldsymbol{a}_i^{(r)}\tilde{\boldsymbol{c}}_i^{(r)}} - \boldsymbol{a}_i^{(r)}\boldsymbol{G}^{(r,i)} \;=\; 0 \tag{6.7}$$

To solve the above equation, the iterative optimization scheme proposed in [31] for the ML optimization of semi-tied covariance (STC) transforms may be used. For the STC system, an equation of the same form is solved, except that the ML statistics are used. This gives an iterative MPE update of HLDA transform on a row by row basis.

$$\boldsymbol{a}_i^{(r)} \;=\; \tilde{\boldsymbol{c}}_i^{(r)\top}\boldsymbol{G}^{(r,i)-1}\sqrt{\frac{\sum_{j\in r,\tau}(\gamma_j^{\texttt{num}}(\tau) - \gamma_j^{\texttt{den}}(\tau)) + \sum_{j\in r} D_j}{\tilde{\boldsymbol{c}}_i^{(r)\top}\boldsymbol{G}^{(r,i)-1}\tilde{\boldsymbol{c}}_i^{(r)}}} \tag{6.8}$$

Like standard forms of HMM parameters, an important issue in discriminative training of HLDA projections is the setting of the smoothing constant D_j. This constant is used both in the iterative update formula in equation 6.8, and the second order statistics, $\boldsymbol{G}^{(r,i)}$, in equation 6.6. During training this constant ensures a stable convergence and should be appropriately set. For all the experiments the standard form of D_j discussed in section 4.3.1.4, $D_j = E\sum_\tau \gamma_j^{\texttt{den}}(\tau)$, was used and E was always set as 2.0. This is also a setting used for MPE training of other HMM parameters [93]. If this form of D_j is not still big enough to ensure the updated full covariances are positive definite, then the minimum E which satisfies this condition will be used instead. Such a E may be efficiently selected by examining if the updated covariance is positive definite via Cholesky decomposition.

6.3.2 MPE Training of multiple LDA

As discussed in section 2.4.3 the only difference between multiple HLDA and multiple LDA is whether the nuisance subspace parameters are tied on a local, or a global level. Based on this, a slightly modified form of the gradient in equation 6.5, may be applicable to multiple LDA. This is given by

$$\begin{aligned}
\left.\frac{\partial \mathcal{Q}(\lambda,\tilde{\lambda})}{\partial \boldsymbol{a}^{(r)}_{i,i\le p}}\right|_{\lambda=\tilde{\lambda}} &= \left[\sum_{j\in r,\tau}(\gamma_j^{\texttt{num}}(\tau)-\gamma_j^{\texttt{den}}(\tau))+\sum_{j\in r}D_j\right]\frac{\tilde{\boldsymbol{c}}_i^{(r)\top}}{\tilde{\boldsymbol{a}}_i^{(r)}\tilde{\boldsymbol{c}}_i^{(r)}}-\tilde{\boldsymbol{a}}_i^{(r)}\boldsymbol{G}^{(r,i)}\\
\left.\frac{\partial \mathcal{Q}(\lambda,\tilde{\lambda})}{\partial \boldsymbol{a}^{(r)}_{i,i>p}}\right|_{\lambda=\tilde{\lambda}} &= \left[\sum_{j,\tau}(\gamma_j^{\texttt{num}}(\tau)-\gamma_j^{\texttt{den}}(\tau))+\sum_{j}D_j\right]\frac{\tilde{\boldsymbol{c}}_i^{(r)\top}}{\tilde{\boldsymbol{a}}_i^{(r)}\tilde{\boldsymbol{c}}_i^{(r)}}-\tilde{\boldsymbol{a}}_i^{(r)}\boldsymbol{K}^{(i)}
\end{aligned} \tag{6.9}$$

where $\boldsymbol{G}^{(r,i)}$ is the same as equation 6.6 for all useful dimensions. $\boldsymbol{K}^{(i)}$ is accumulated for nuisance dimensions over all Gaussians,

$$\boldsymbol{K}^{(i)} = \sum_j \breve{\sigma}_i^{(j)-2}\left[\sum_\tau(\gamma_j^{\texttt{num}}(\tau)-\gamma_j^{\texttt{den}}(\tau))+D_j\right]\boldsymbol{\Sigma}^{(g)} \tag{6.10}$$

where the global covariance $\boldsymbol{\Sigma}^{(g)}$ is fixed given the training data and does not require an discriminative update.

Unfortunately individual projections can not be independently optimized for multiple LDA, because the transform parameters in the nuisance subspace is globally tied. Hence the efficient row by row optimization, given in equation 6.8, may not be used for multiple LDA. To handle this problem, the approach proposed here is to use a gradient descent based optimization given in equation 4.34. This approach requires the following second order information

$$\begin{aligned}
\left.\frac{\partial^2 \mathcal{Q}(\lambda,\tilde{\lambda})}{\partial^2 \boldsymbol{a}^{(r)}_{i,i\le p}}\right|_{\lambda=\tilde{\lambda}} &= \left[\sum_{j\in r,\tau}(\gamma_j^{\texttt{num}}(\tau)-\gamma_j^{\texttt{den}}(\tau))+\sum_{j\in r}D_j\right]\frac{\tilde{\boldsymbol{c}}_i^{(r)}\tilde{\boldsymbol{c}}_i^{(r)\top}}{\left(\tilde{\boldsymbol{a}}_i^{(r)}\tilde{\boldsymbol{c}}_i^{(r)}\right)^2}-\boldsymbol{G}^{(r,i)}\\
\left.\frac{\partial^2 \mathcal{Q}(\lambda,\tilde{\lambda})}{\partial^2 \boldsymbol{a}^{(r)}_{i,i>p}}\right|_{\lambda=\tilde{\lambda}} &= \left[\sum_{j,\tau}(\gamma_j^{\texttt{num}}(\tau)-\gamma_j^{\texttt{den}}(\tau))+\sum_{j}D_j\right]\frac{\tilde{\boldsymbol{c}}_i^{(r)}\tilde{\boldsymbol{c}}_i^{(r)\top}}{\left(\tilde{\boldsymbol{a}}_i^{(r)}\tilde{\boldsymbol{c}}_i^{(r)}\right)^2}-\boldsymbol{K}^{(i)}.
\end{aligned} \tag{6.11}$$

Using the transform rows of useful dimensions for an example, the update formula is given by

$$\boldsymbol{a}_i^{(r)} \quad = \quad \tilde{\boldsymbol{a}}_i^{(r)} + \eta \left[\left. \frac{\partial^2 \mathcal{Q}(\lambda, \tilde{\lambda})}{\partial^2 \boldsymbol{a}_{i,i \le p}^{(r)}} \right|_{\lambda = \tilde{\lambda}} \right]^{-1} \left. \frac{\partial \mathcal{Q}(\lambda, \tilde{\lambda})}{\partial \boldsymbol{a}_{i,i \le p}^{(r)}} \right|_{\lambda = \tilde{\lambda}} \tag{6.12}$$

where the learning rate η requires empirical tuning to ensure a stable convergence.

Although an MPE update of multiple LDA has been given, HLDA and multiple HLDA systems are the focus of the experiments for two main reasons: First, numerical methods can be expensive and difficult to guarantee convergence in practice, as discussed in section 4.3.4,. Second, multiple HLDA can also provide a more flexible model structural configuration, by locally varying the retained subspace dimensionality, as explained in section 2.4.3. In [70, 72] WER improvements were reported by locally optimizing the number of useful dimensions on LVCSR tasks. Furthermore, in general multiple HLDA was found to outperform multiple LDA in ML training stage for LVCSR in earlier research [34]. Therefore the discriminative training of HLDA and multiple HLDA systems is the focus of this work.

6.4 Implementation Issues

In this section implementation issues for discriminative training of linear projection schemes are discussed. These issues may affect the performance of systems using linear projections, and are therefore important.

6.4.1 Variance Flooring

For speech recognition systems using HMMs, the re-estimated Gaussian covariances are often floored to ensure they are positive and definite. For systems using diagonal covariances with a standard feature front-end or a global feature projection, the variance flooring problem may be straightforward. The simple approach described in [131] may be used, by setting the variance floor to be a small portion of the global covariance, or average state covariance [51]. In this case the flooring is only considered in one global feature space. However the variance flooring for systems using multiple projections, such as multiple HLDA, is more complicated. This is due to the presence of multiple feature subspaces. To handle this problem, the solution adopted in this work is to use a global minimum variance floor, $\boldsymbol{f}$, for all subspaces. The ith dimension of $\boldsymbol{f}$ is given by

$$\boldsymbol{f}_i \quad = \quad \arg\min_r \left\{ \eta \boldsymbol{a}_i^{(r)} \bar{\boldsymbol{\Sigma}} \boldsymbol{a}_i^{(r)\top} \right\} \tag{6.13}$$

where $\bar{\boldsymbol{\Sigma}}$ is the average state covariance in the original feature space and the variance floor scale η is commonly set to 0.01.

6.4.2 Setting of I-smoothing

In order to obtain a more robust parameter estimates during MPE training, I-smoothing of the MPE numerator statistics may be used, as discussed in section 4.3.3. The ML or MMI statistics may be used as priors for Gaussian parameters. A key issue with this approach is the setting of the constant τ^I. From a MAP perspective, this constant controls how much the parameter estimate will back-off to the ML or MMI statistics based prior. In [93] a commonly used setting for HMM systems with diagonal covariances is $\tau^I = 50$. This setting is also used in all experiments for estimating linear projections. Unless otherwise stated the I-smoothing statistics will be ML based in all experiments.

6.4.3 Use of Lattices

In discriminative training, lattices are commonly used to represent the model's confusion over the data. Ideally individual models should be used to generate the matched lattices for training. However for LVCSR systems this is infeasible. The commonly used approach is to use one set of HMMs to generate word lattices by recognizing the training data. Then they will be further marked with phone alignment and kept fixed for training. This is the "exact match" approach described in [124]. One issue with this approach is whether it is appropriate to use the same set of lattices for training systems, which are very different from the one used to generate these lattices. In [124] WER improvements were reported by re-generating triphone model alignment in the intermediate stage of MMI training for an LVCSR task. This issue also exists with systems using multiple projections, because the training lattices are normally generated by a system using the standard front-end, or a global ML trained projection. Optimizing the projections using a discriminative criterion may further enlarge the mismatch between the model set and lattices. In this work, one single set of training lattices are used initially for estimating the projections. Then this issue is investigated by using the matched lattices for the subsequent MPE training of individual systems.

6.4.4 Integrated Structural and Parametric Optimization

In general a machine learning problem may be partitioned into two distinct stages. In the first stage the optimal model structural configuration is selected using a complexity control criterion. In the second stage parameters are estimated using some training criterion after the appropriate complexity is determined. The underlying criteria used for these two different stages may not necessarily be the same one. In chapter 5 model complexity was determined in a discriminative fashion. However in many practical situations, model parameters are considered to be trained using the ML criterion. Hence there may be a mis-match between the criteria used for model selection and parameter estimation. To handle this problem, the model selection and parameter estimation may be integrated into a consistent discriminative learning process. When selecting the number of Gaussian components per state, for instance, component means and variances are considered to be discriminatively estimated for each candidate model structure. Similarly for multiple HLDA systems, when selecting the number of useful dimensions for each projection, the HLDA projections and other model parameters are also considered to be discriminatively updated.

6.5 Summary

In this chapter the discriminative training of linear projection schemes is investigated using a weak-sense auxiliary function. This weak-sense auxiliary function has a general form and may be applied to the discriminative optimization of a variety of forms of model parameters. Using this approach, the discriminative training algorithms for HLDA, multiple HLDA and multiple LDA systems were presented. A number of implementation issues when estimating linear projections using discriminative criteria were also discussed. Experimental results for discriminative training of linear projections are presented later in chapter 8.

7

Experiments on Model Complexity Control

In this chapter experimental results are presented for model complexity control using marginalized discriminative growth functions and standard model selection techniques. In the first part of this chapter, a series of complexity control experiments are conducted on an LVCSR task for conversational telephone speech (CTS) data. Initially complexity control schemes are used to optimize multiple model complexity attributes on a "global" level. This restricts the complexity of different part of the model to be the same, and allows all possible systems to be explicitly trained and evaluated. The correlation with WER and the performance ranking error are then examined for a variety of complexity control techniques. These are followed by optimizing multiple model complexity attributes on a local level. Then the interaction with other techniques is investigated. The generalization to two other LVCSR tasks is also investigated. Finally, the performances of complexity controlled systems are evaluated in a state-of-the-art 10 time real-time LVCSR system for a CTS transcription task.

7.1 Experiments on CTS English

This section presents complexity control experiments for CTS English data. First, the experimental setups and conditions of the experiments is briefly described. Second, experimental results of model complexity control on a "global" level is presented on an LVCSR task. Issues with existing likelihood based complexity control schemes are also discussed. Finally, complexity control on a local level are performed on four different LVCSR setups, where multiple complexity attributes are allowed to vary locally across different parts of the system.

7.1.1 Summary of Experimental Setups

In order to fully investigate the performances of complexity control techniques, five CTS English training configurations were used. The first is a full system using a 297 hour training set h5etrain03, consisting of 4800 Switchboard I, 228 Call Home English (CHE) and 418 Linguistic Data Consortium (LDC) Cellular conversation sides [23]. Three subsets of this were also used:

46 hour minitrain04; 68 hour h5etrain00sub; 76 hour h5etrain03sub; 148 hour meditrain04. All subsets were selected to have the same gender and channel condition distribution of the full set. The total number of training speakers in the full set is approximately 8 times as the 46 hour minitrain04, 4 times the 76 hour h5etrain03sub, and twice the 148 hour subset meditrain04. Note that each subset is hierarchically subsumed by the other larger sets. The baseline feature vector used for all projections was a 52-dimensional PLP feature extracted by appending derivatives up to the third order and then normalized using VTLN, mean and variance normalization on a conversation side basis. For the baseline configuration this 52-dimensional feature vector was projected down to 39 dimensions using one or more HLDA projections. For multiple HLDA systems the silence Gaussians were assigned to one transform class, while the speech Gaussians were split into 64 distinct classes. The component assignment used a top-down splitting procedure, based on distance measure of Gaussian components in the acoustic space. Continuous density, mixture of Gaussians, cross-word triphone, gender independent HMM systems were used. After phonetic decision tree based tying, there are approximately 3k speech states for the 46 hour subset, and 6k states for the other four training sets. Basic features of these five setups are presented in table 7.1.

Corpus	Size	#States
minitrain04	46 hr	3k
h5etrain00sub	68 hr	6k
h5etrain03sub	76 hr	6k
meditrain04	148 hr	6k
h5etrain03	297 hr	6k

Table 7.1 *Training setups used for experiments on CTS English data*

As discussed in chapters 4 and 5, to obtain sufficient statistics for discriminative training, or complexity control, lattices are normally used for LVCSR tasks. The training data lattices used to obtain the statistics for complexity control experiments were generated using the baseline 39-dimensional global HLDA systems. These lattices were further marked with model alignment and kept fixed for complexity control using marginalized discriminative growth functions. This was the “exact match” approach described in [124]. For evaluation a 3 hour dev01sub was used. The test set contains 20 Switchboard I and 20 Switchboard II phase II conversation sides of the NIST LVCSR evaluation data in 2000 and 1998 respectively, and another 19 Linguistic Data Consortium (LDC) Cellular sides. The audio data was manually segmented. The test set was also used as the held-out data in the experiments. The same front-end processing and normalization schemes were also used. Unless otherwise stated ML training was used for all systems. All recognition experiments used a 58k word trigram language model.

7.1.2 Experiments on Global Complexity Control

As discussed in section 3.1, word error rate is the most commonly used performance measurement for speech recognition systems. An ideal complexity control scheme should yield the same ranking as the WER for all systems being considered. Hence, one natural way of evaluating a complexity control criterion is to examine its correlation with the WER. This requires a variety of systems to be explicitly built and evaluated, which is infeasible for highly complex LVCSR systems. However, if the complexity attributes considered are optimized on a global level, the permutation of all possible structural configurations can be far more tractable. This is the case considered in the experiments of this section. Existing complexity control schemes are evaluated on an LVCSR task for CTS English data. Since complexity attributes are optimized on a global level, all possible systems may be explicitly trained and evaluated. This can give an intuitive feel of how strongly the underlying complexity control scheme is correlated with the error rate. The 68 hour CTS English corpus, h5etrain00sub, as described in section 7.1.1, was used as the training set. Two complexity attributes of an HLDA system with a single projection were optimized globally: the number of Gaussian components per state from the set $\{12, 16, 24\}$; and the number of useful dimensions in the range $\{28, ..., 52\}$. The permutation of these two attributes led to a total of 75 different configurations.

7.1.2.1 Correlation Between Criteria and WER

After these 75 systems were explicitly trained and evaluated, the correlation with WER was examined for likelihood on held-out data first. As discussed in section 3.2, the majority of existing complexity control techniques inherently assume a strong correlation between the likelihood on unseen data and WER. This correlation between likelihood and WER for all the 75 systems is shown in figure 7.1.

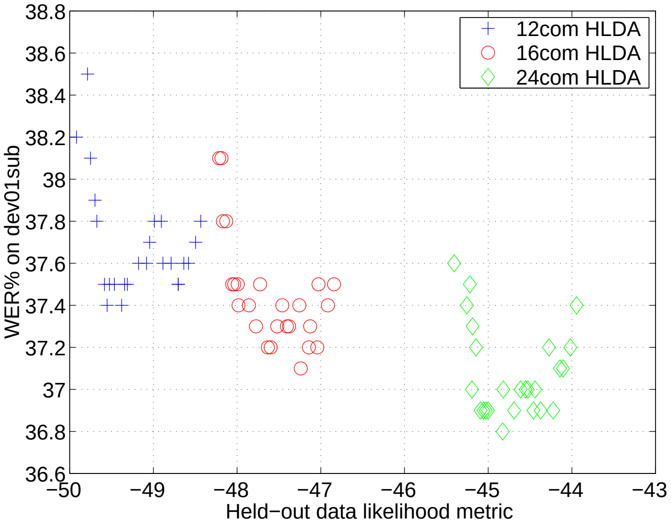

Figure 7.1 *Held out data likelihood vs. WER for* dev01sub *on CTS English 68 hour* h5etrain00sub

Although the figure illustrates a very general trend that error rate decreases as the held-out data likelihood increases, the precise ordering of systems is poor. Noticeably, this scheme favored the most complex system. The best model structure predicted had 24 Gaussians per state and 52 useful dimensions. However, the performance of this system is significantly worse than the actual best system by 0.6% absolute. For these 75 HLDA systems, the correlation between the likelihood on held-out data and WER shown in 7.1 is quite weak. This weakness indicates that the model correctness assumption of standard complexity control schemes within the likelihood based framework may be too strong for current speech recognition systems using HMMs.

Despite this limitation, it is still useful to examine the performances of approximation schemes for the Bayesian evidence integral. These should be closely related to the held-out data likelihood. As discussed in section 3.3.1, using BIC the approximated Bayesian evidence is only a function of the log likelihood and number of model parameters. The training data log-likelihood is expected to monotonically increase as the number of model parameters increases, irrespective of the form of the parameters being considered.

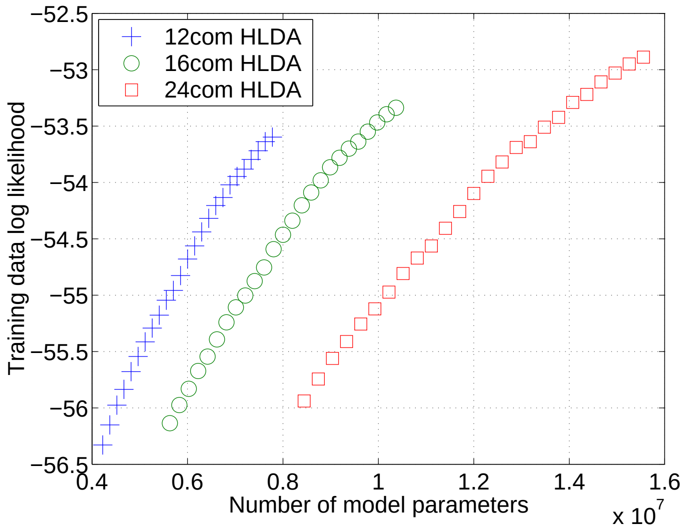

Figure 7.2 *Training data log likelihood vs. the number of parameters on CTS English 68 hour* h5etrain00sub

Unfortunately, in this setup such a relationship does not exist, as is shown in figure 7.2. In the figure there are three distinct lines associated with the 12, 16 and 24 component systems. On each of the three lines the training data log-likelihood increases as the number of useful dimensions is increased. However, across these three lines the log likelihood is not increasing monotonically as the system becomes more complex. In the figure each log likelihood value in the figure may correspond up to three model structures, each with different complexity. The same issue still exists even if the penalization coefficient, ρ, of the BIC criterion in equation 3.3, is finely tuned. This indicates that the log-likelihood contribution from different forms of model parameters, in this case the number of components and dimensions, is not the same. Hence, BIC may have lead to a poor Evidence approximation when multiple complexity attributes were optimized simultaneously.

Compared with the likelihood, discriminative criteria are more closely related to the recog-

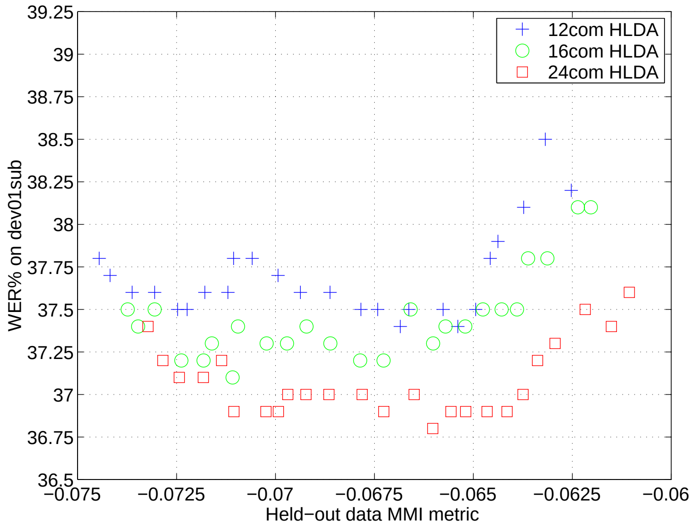

Figure 7.3 *MMI criterion on Held out data vs. WER for* dev01sub *on CTS English 68 hour* h5etrain00sub

nition error. Therefore the correlation between these criteria and WER should be stronger. However discriminative criteria may not be directly used for complexity control. As discussed in section 5.2, this is due to the sensitivity to outliers utterances. Here the MMI criterion was taken as an example. Figure 7.3 shows the MMI criterion values on held-out data against WER. The correlation between the MMI criterion and WER was quite poor. This may have been caused two issues. First, the average segment length may have an impact on the held-out data MMI scores. As the MMI criterion is related to the sentence error rate, short sentences may tend to be penalized more if they contain any wrong words. Second, more importantly, as discussed in section 5.2, the existence of outliers can heavily influence the value of the MMI criterion. These outliers are sentences with very low posteriors. This is the motivation of using discriminative growth functions for complexity control. As discussed in section 5.3, a discriminative growth function should have reduced sensitivities to outliers whilst still retaining some attributes of the original criterion.

Figure 7.4 shows the correlation between the marginalized MMI growth function in equation 5.22 and the WER. As discussed in section 5.6, for efficiency three sets of MMI statistics were generated by standard 39 dimensional systems with 12, 16 or 24 components per state respectively. These were shared among systems that had the same number of Gaussians. The block diagonal Hessian approximation described in section 5.6.3 was used to compute the growth function's lower bound marginalization in equation 5.22. In the experiments the smoothing constant, $C = 2.0$, was fixed. Even under these approximations, in figure 7.4 a strong correlation with the WER is still observed with the WER. The best system selected was only 0.1%-0.2% absolute worse than the actual best one.

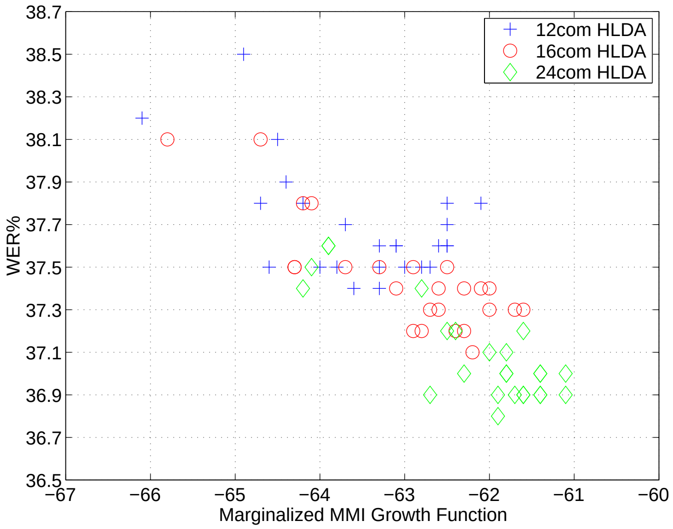

Figure 7.4 *Marginalized MMI growth function vs. WER for* dev01sub *on CTS English 68 hour* h5etrain00sub

7.1.2.2 Recognition Performance Ranking Error

A good complexity control scheme should rank all the systems in a way that matches the ranking of their recognition performances. A measure of the distance between the predicted and correct ranking is required to evaluate various complexity control schemes. In this work, an empirical ranking error metric is proposed as

$$\text{RankErr\%} \quad = \quad \frac{\sum_{i,j} \delta(\boldsymbol{we}_i, \boldsymbol{we}_j) \times |\boldsymbol{we}_i - \boldsymbol{we}_j| \times |i-j|}{N \times \max_{i,j}\{|\boldsymbol{we}_i - \boldsymbol{we}_j|\} \times \max_{i,j}\{|i-j|\}} \tag{7.1}$$

where $\{\boldsymbol{we}_1, ..., \boldsymbol{we}_i, ..., \boldsymbol{we}_N\}$ denotes a WER ranking prediction, for all N possible systems being considered, according to a particular complexity control scheme. Let $\boldsymbol{we}_i$ denote the WER of the system ranked as the ith, $|\boldsymbol{we}_i - \boldsymbol{we}_j|$ the WER difference between system i and j, and $|i-j|$ the position shift between them. The binary function $\delta(\boldsymbol{we}_i, \boldsymbol{we}_j)$ will be true, only if the ranking between $\boldsymbol{we}_i$ and $\boldsymbol{we}_j$ is incorrect and the difference in WER is significant (above a given *WER threshold*). This has a good intuitive feel, as penalizing systems that differ only slightly in error rate seems inappropriate. The WER difference between a pair of systems that falls below the *WER threshold* may be ignored. Hence the ranking error may be related to the total number of position shifts, weighted by WER differences between all mis-ranked pairs of systems if the differences are significant. The normalization term in equation 7.1 guarantees the ranking error will be positive and less than one.

Table 7.2 shows the error of the predicted recognition performance ranking using the metric given in equation 7.1. Three different WER thresholds were also used to determine whether the mistaking between two systems is considered. The first line in the table serves as a baseline. It ranks the systems according to the training data likelihood, which simply yields an ordering on system complexity with no penalization. Using the likelihood on held-out data, the ranking error is still fairly high although much improvements were obtained over using the training data likelihood.

Complexity	WER threshold		
Control	0.0	0.1	0.2
Training Likelihood	22.08	22.08	21.59
Held-out Likelihood	8.94	8.89	8.19
BIC ($\rho = 1.0$)	48.43	48.36	47.35
BIC ($\rho = 2.0$)	55.68	55.68	55.42
Held-out MMI	37.40	37.40	35.91
MMI GFunc	4.74	4.64	3.10

Table 7.2 *Performance ranking prediction error (%) for* dev01sub *on CTS English 68 hour* h5etrain00sub

The table also shows the ranking errors for approximated Bayesian evidence using BIC, which should be closely related to the held-out data likelihood. As previously described there are issues with BIC when controlling multiple complexity attributes. Hence, using both standard BIC and penalized BIC ($\rho = 2.0$), the ranking scores were poor. The final line of the table shows the ranking performance using the held-out MMI criterion. The poor performance of the MMI criterion is clearly shown. The best performance was obtained using the marginalized MMI growth function and the score is related to figure 7.4. As expected, if the WER threshold in the table is increased then the ranking error decreases, though the general ranking of all complexity control schemes remains about the same.

7.1.2.3 Discussion

In this section a few complexity control schemes were evaluated by examining the correlation with WER and the performance ranking error. Two model complexity attributes of an HLDA system, the number of Gaussian components per state, and the number of useful dimensions, were optimized on a global level. This allowed systems with all possible configurations to be explicitly built and evaluated. A few issues associated with existing complexity control techniques are presented below:

- First, in these experiments the correction between the likelihood on held-out data and WER was found to be fairly weak. This is because the model correctness assumption made in standard complexity control techniques may be too strong for current ASR systems using HMMs. As discussed in section 3.6, HMMs are not the correct models for speech signals. Hence, merely increasing the likelihood for the unseen data does not necessarily decrease the error rate.

- Second, a limitation of BIC was found when optimizing multiple complexity attributes simultaneously. As discussed in section 3.3.1, the BIC approximation may become increasingly poorer as the amount of observed data decreases. In addition, the differences in the form of model parameters is not considered by BIC. This probably leads to the non-monotonic increase of log-likelihood against model complexity, as is shown in figure 7.2. In contrast the Laplace's approximation discussed in section 3.3.3 accounts for such differences. The second order information, or Hessian matrix, explicitly describes the likelihood contribution from different forms of model parameters. Hence, it is preferable to use Laplace's approximation to compute the evidence integral.

- Finally, for current speech recognition systems, it is preferable to marginalize discriminative criteria for complexity control. These criteria are more directly related to the recognition error than likelihood. However, discriminative criteria, such as MMI, are prone to be sensitive to outliers as found in the experiments. Hence they may not be directly used for complexity control. This was the motivation for using a marginalized growth function for model selection. As discussed in section 5.3, a discriminative growth function should have reduced sensitivities to outliers whilst still retaining certain attributes of the original criterion. This is further investigated in detail in the following sections.

7.1.3 Experiments on Local Complexity Control

In the previous section two complexity attributes of an HLDA system, the number of Gaussians per state and useful dimensions per Gaussian, were optimized on a global level. Limiting the complexity control on a global level is an unnecessary restriction. When varying the system complexity locally, more flexibility in the model structure may be introduced. Hence it is preferable to optimize complexity attributes on a local level. In this section the performances of complexity control techniques are further investigated by locally optimizing the same two complexity attributes on a standard LVCSR task for CTS English data.

7.1.3.1 Experimental Conditions

Four CTS English training configurations as described in section 7.1.1 were used: the 46 hour minitrain04; 76 hour h5etrain03sub; 148 hour meditrain04; the 297 hour full set h5etrain03. The total number of training speakers is approximately log-linearly increasing across these four sets. Each subset is also hierarchically subsumed by the other larger sets. For each training set, the following forms of complexity control were compared:

- *Fixed*, the baseline approach of using an even number of components per state, or dimensions per Gaussian. This effectively performs no control of the model complexity and the number of parameters is manually tuned.

- *VarMix*, a simple "more data more parameters" approach. The number of components in a state is set to be proportional to the number of frames assigned to that state raised to a power. In all these experiments that power was set as 0.2. The total number of components in the system is fixed so that the average number of Gaussians per state remains the same as the standard, Fixed, system from which it was derived. This is a standard technique used in the CU-HTK LVCSR evaluation systems [23]. However, this is not strictly a complexity control approach since the total number of components is not automatically determined.

- *BIC*, an example of a Bayesian complexity control that was discussed in section 3.3.1, was implemented.
- *MPE GFunc*, the discriminative evidence framework using marginalized MPE growth functions presented in chapter 5 was evaluated. As the MPE criterion is a closer approximation to WER than MMI, the marginalization of MPE growth functions is a focus of the following experiments.

For both BIC and MPE GFunc systems, the efficient implementation discussed in section 5.6 was used. The penalization coefficient of BIC, ρ, in equation 3.3, was manually tuned with three values, 0.5, 1.0 and 2.0, to obtain the best performances. In contrast, for all MPE GFunc systems, the smoothing constant C in equation 5.6 was set to 2.0 and never altered.

The same set of experiments are conducted for each training set to fully investigate model selection using marginalized discriminative growth functions. First, only the number of components associated with each state is determined. Second, a more complex model selection problem is examined. Both the number of Gaussians per state and useful dimensions per projection in a multiple HLDA system are to be optimized. As with the experiments in section 7.1.2, the number of useful dimensions to be considered is in the range from 28 to 52 for each projection.

7.1.3.2 Optimizing the Number of Components

Table 7.3 shows the performances of various global HLDA systems after complexity control. The front-end for these experiments use the standard global HLDA projection to 39 dimensions. In the first section of the table, the performances of the baseline systems are shown with a range of fixed number of components per state from 12 to 20. Two general trends are observed for these Fixed systems. First, increasing the amount of training data while fixing the number of components consistently reduced the WER for all configurations. Note that the WER differences between the 46 hour and 76 hour setups for all Fixed systems were at least 2.0% absolute. These are bigger than the WER differences between other larger sets, for example 0.4%-0.7% between the 76 and 148 hour setups. This is expected as the number of tied states on the 46 hour setup is only 3k, while for the other larger sets 6k states were used, as described in table 7.1. Second, within each training set, increasing the number of components per state gradually lead to saturated WER performances after the number of components reached more than 16. For example, on both the 76 and 148 hour setups, the 18 and 20 component Fixed system gave the same error rates. The best Fixed systems, also with fewest parameters possible, had 20, 16, 18 and 20 components for the four training sets respectively.

The second section of table 7.3 shows the performances of various VarMix systems with the average number of components per state ranging from 12 to 20. Using VarMix to re-arrange the number of components according to state occupancies, a WER reduction of 0.1%-0.4% absolute

Complexity Control		WER%			
		46 hr	76 hr	148 hr	297 hr
Fixed	12	38.3	36.1	35.7	35.1
	14	38.0	36.0	35.4	34.8
	16	37.8	$35.8^{\star}$	35.2	34.9
	18	37.9	35.8	$35.1^{\star}$	34.3
	20	$37.8^{\star}$	35.8	35.1	$34.1^{\star}$
VarMix	12	37.9	36.1	35.2	34.9
	14	37.7	35.8	35.0	34.7
	16	37.6	35.7	35.0	34.3
	18	37.6	35.7	34.8	34.0
	20	37.5	35.6	34.8	33.9
BIC ($\rho = 0.5$)		37.4	35.7	34.5	34.1
(#Gauss)		(19.38)	(15.57)	(17.13)	(19.21)
BIC ($\rho = 1.0$)		37.4	35.8	34.6	34.2
(#Gauss)		(18.45)	(14.68)	(16.34)	(18.68)
BIC ($\rho = 2.0$)		37.5	36.1	34.7	34.2
(#Gauss)		(18.04)	(12.73)	(14.78)	(17.71)
MPE GFunc		37.2	35.7	34.4	33.8
(#Gauss)		(18.34)	(14.52)	(15.43)	(17.54)

Table 7.3 *Optimizing #Gauss for global HLDA systems for* dev01sub *on CTS English 46 hour* minitrain04*, 76 hour* h5etrain03sub*, 148 hour* meditrain04 *and 297 hour* h5etrain03*; $\star$ marks the starting model for component merging of BIC and MPE GFunc systems on each training set.*

was obtained over the baseline Fixed systems for most configurations in the table. This improvement is not surprising as the amount of data associated with each state can vary dramatically. For the 46, 76 and 297 hour sets, the best VarMix result was associated with the most complex configuration using 20 Gaussian components per state. On the 148 hour setup, the 18 and 20 component VarMix systems yielded the same WER performance. Similar to the Fixed systems in the first section of the table, for each training set the gain from having more components was gradually reduced when the number of Gaussians per state is more than 16.

The results using BIC and marginalized MPE growth functions, along with the average number of components per state, are shown in the third and fourth sections of table 7.3. One interesting issue with the iterative complexity control used here for both the BIC and the GFunc systems is the selection of the initial model. As discussed in section 5.6.1, for efficiency a starting model is used to obtained a single set of statistics that may be shared by a range of configurations. This starting model may affect both the complexity and WER of the final system. Its selection may be determined by the following factors:

- First, the starting model should give the lowest WER. This ensures a good initialization for the whole complexity optimization process.
- Second, the starting model should not be too simple. This is because that it is not possible to have a final system that is more complex than the starting model using the component merging approach in section 5.6.1. As expected, if the starting model is under-fitting to the training data, so will the final system.
- Third, the starting model should not be too complex. To ensure the stability of complexity control, the constrained maximum mutation from the current model structure is imposed, as discussed in section 5.6.2. Hence, for a highly complex starting model, a large number of iterations of complexity control may be required to obtain an optimal, compact, structural configuration for the final system.

Starting Model	#Gauss	12	14	16	18	20	24	32	48
	WER%	36.1	36.0	35.8	35.8	35.8	35.7	35.7	36.1
Final Model	#Gauss	11.28	12.95	14.52	16.56	18.71	22.62	30.63	46.82
	WER%	36.3	36.0	35.7	35.7	35.6	35.6	35.7	36.0

Table 7.4 *Varying #Gauss of the starting model when using marginalized MPE growth functions to optimize #Gauss of global HLDA systems for* dev01sub *on CTS English 76 hour* h5etrain03sub.

To further illustrate this, here the 76 hour set is taken as an example and a variety of MPE GFunc systems were built with varying number of components per state in a Fixed system as the starting model. The error rates of the final systems, along with the number of components per state in both the starting and final models, are shown in table 7.4. In the table, when the number of components per state of the starting model reaches 16, the final MPE GFunc systems performances are saturated. In particular using the 32 component Fixed system as the starting model a 30.63 component per state system was selected. This is much more complex than 14.52 component system (derived from the 16 component Fixed system) by having more than twice parameters. Similarly, if using the most complex 48 component system as the starting model, both the starting and final model is clearly over-fitting to the data. In both cases, it is conceivable that many more additional iterations of complexity control may be required to obtain the best performances. Hence, in order to obtain the best WER performance with the fewest model parameters, it is preferable to use a starting system that has the lowest error rate and a relatively compact model structure.

For the reasons explained above, the 20, 16, 18 and 20 component Fixed systems were selected as the starting models for both BIC and MPE GFunc approaches on the four training sets respectively. In table 7.3 these four starting models used to obtain the initial statistics and determine the maximum complexity of the systems are marked with a “$\star$”. They are equivalent to the comparable BIC systems when setting $\rho = 0$. As described in section 5.6.2, a total of four iterations of complexity control were performed for both BIC and marginalized growth function systems. Between iterations ML training was performed to refine the parameter estimates.

In table 7.3 setting the BIC penalization coefficient, $\rho = 0.5$, gave the best WER performances consistently for all training sets. Note that on the 46 hour subset, the standard BIC system ($\rho = 1.0$) gave the same error rate as $\rho = 0.5$ but had fewer parameters. As expected, for each training set the complexity of BIC systems is increasing as the value of ρ decreases. Compared with the best baseline Fixed systems, the gains from the best BIC systems ($\rho = 0.5$) were 0.4%, 0.1% and 0.6% for the 46, 76 and 148 hour subsets respectively. On the 297 hour full set, the best BIC system outperformed the 20 component Fixed system by having fewer parameters. Slight WER reductions of 0.1%-0.3% were also obtained from the BIC systems against the best VarMix systems for most training setups. For example, on the 148 hour set the best BIC system ($\rho = 0.5$), which had 17.13 components per state on average, outperformed the more complex 20 component VarMix system by a marginal 0.3%. On the 76 hour training set, there was a slight WER degradation of 0.1% using the best BIC system ($\rho = 0.5$). The general trend is that the BIC systems were comparable to the best VarMix systems, but with fewer components per state.

The performances using the marginalized MPE growth functions are also shown in table 7.3. In contrast to the VarMix and BIC approaches, there was no tuning of any free parameters. The MPE GFunc systems outperforms all Fixed systems in the table. Compared with the best VarMix systems, there were also WER gains of 0.1%-0.4% on the 46, 148 and 297 hour sets. On the 76 hour subset, the more compact 14.52 MPE GFunc system outperformed the 16 and 18 component VarMix systems. For each training set, the MPE GFunc system outperformed all three BIC systems by having a lower WER and fewer parameters. For example, on the 46 hour set, the GFunc system had 18.34 Gaussians per state on average and gave a WER of 37.2%. It outperformed both the best BIC ($\rho = 1.0$, 18.45 components per state and $\rho = 0.5$, 19.38

components per state) by 0.2%. Similarly on the 297 hour full set, the MPE GFunc system (17.54 component per state) outperformed all three BIC systems by 0.3%-0.4%. On this setup it is also interesting to find that this GFunc system is also more compact than the smallest penalized BIC configuration ($\rho = 2.0$, 17.71 components per state). These results indicate that the MPE GFunc system is able to select configurations that make more efficient use of the number of Gaussian components. Overall, the MPE GFunc approach outperformed, or approximately matched, the best manually tuned system in table 7.3 with a more compact model structure.

7.1.3.3 Optimizing the Number of Components and Dimensions

Complexity Control				WER%			
#Gauss		#Dim		46 hr	76 hr	148 hr	297 hr
VarMix	12	Fixed	39	-	35.8	-	-
	12		52	-	35.3	-	-
	16		39	38.0	35.9	34.9	34.2
	16		52	37.6	$35.6^{\dagger}$	34.6	33.7
	18		39	-	-	34.5	-
	18		52	-	-	$34.3^{\dagger}$	-
	20		39	37.5	-	-	34.0
	20		52	$37.3^{\dagger}$	-	-	$33.6^{\dagger}$
BIC ($\rho = 0.5$)				36.6	34.9	33.9	33.4
(#Gauss)				(19.38)	(15.57)	(17.13)	(19.21)
(#Dim)				(49.89)	(49.36)	(50.17)	(50.91)
BIC ($\rho = 1.0$)				36.9	35.2	33.9	33.4
(#Gauss)				(18.45)	(14.68)	(16.34)	(18.68)
(#Dim)				(44.59)	(42.89)	(47.62)	(49.33)
BIC ($\rho = 2.0$)				37.2	35.2	34.3	33.6
(#Gauss)				(18.04)	(12.73)	(14.78)	(17.71)
(#Dim)				(35.77)	(33.39)	(39.43)	(43.75)
MPE GFunc				36.7	34.6	33.9	33.0
(#Gauss)				(18.34)	(14.52)	(15.43)	(17.54)
(#Dim)				(41.78)	(36.67)	(47.23)	(44.77)

Table 7.5 *Optimizing #Gauss and #Dim of 65 transform HLDA systems for* dev01sub *on CTS English 46 hour* minitrain04*, 76 hour* h5etrain03sub*, 148 hour* meditrain04 *and 297 hour* h5etrain03*; † marks the most complex system for each training set.*

To further investigate marginalized growth functions for model selection, a more complex problem was examined. Both the number of Gaussians per state and useful dimensions per projection in a multiple HLDA system were optimized. Table 7.5 shows the performances of various multiple HLDA systems after complexity control. This table contains three sections. The first section are the baseline systems that used VarMix to tune the number of components per state, and the number of dimensions fixed globally as either 39 or 52 across all projections. In the second section the experimental results of using BIC to control both complexity attributes, along together with the relative complexity (number of components per state and useful dimensions per Gaussian) are shown. As with table 7.3, the values of the penalization coefficient, ρ, was manually tuned to achieve the best performances. The final section of the table shows the comparable results of using marginalized MPE growth functions.

For each training set, four VarMix systems were built. Although not all the possible configurations in the first section were evaluated, a fair comparison may still be made against all the BIC and MPE GFunc systems in the table. On each training setup, a most complex system was built which provided an upper bound of model complexity for all the BIC and MPE GFunc systems. These are marked with a "†" in the table. For example, on the 46 hour subset, the most complex VarMix system had 20 components per state on average and 52 dimensions per Gaussian. This system was larger than any of the comparable BIC or MPE GFunc systems on the same setup. The general trend of these VarMix systems are three-fold. First, compared with the global HLDA VarMix systems in table 7.3, increasing the number of HLDA transforms to 65 while fixing the number of components and dimensionality led to mixed results. Marginal WER reductions were obtained for some systems. For example, on the 148 hour set, the gains from using more HLDA transforms were 0.1% and 0.3% for the 16 and 18 component configurations respectively. In contrast, on the 76 hour setup, increasing the number of transforms to 65 actually degraded the performance of the 16 component VarMix system by 0.2%. This shows that in order to make a better use of multiple HLDA, it is preferable to locally optimize the number of useful dimensions for each projection. Second, for all four subsets increasing the number of components per state while fixing the dimensionality only gave small improvement. For example, on the 297 hour full set, increasing the number of components per state from 16 to 20 reduced the WER marginally by 0.1%-0.2% for both the 39 and 52 dimensional configurations. For the 76 hour set increasing the number of components from 12 to 16 actually degraded the performance of the 52 dimensional configuration by 0.3%. Third, fixing the number of Gaussians per state and increasing the dimensionality from 39 to 52 further reduced the WER for all four training sets by 0.2%-0.5%.

In order to automatically control both the number of components and dimensions, the performances of BIC and MPE GFunc systems were examined. As discussed in section 7.1.2, there are issues for using BIC to optimize multiple complexity attributes simultaneously. Furthermore, when both complexity attributes are controlled locally, the number of possible permutations is intractable. To handle these issues, the two complexity attributes considered were optimized sequentially: the number of Gaussian components first, then the number of useful dimensions after the number of Gaussians is determined. This approach was used for all BIC and MPE GFunc systems in the table. The same starting models in table 7.3, marked with a "$\star$", were also used for all BIC and MPE GFunc systems in table 7.5. As with the results in table 7.3 for global HLDA systems, setting the BIC penalization coefficient, $\rho = 0.5$, gave the lowest error rates consistently for each training set. Compared with the best VarMix baselines with a fixed number of useful dimensions in the table, the gains from the best BIC systems were 0.2%-0.7%. In particular, on the 46 hour set a 0.7% WER reduction was obtained over the comparable best VarMix system. This is expected as it is increasingly important to appropriately control the number of HLDA dimensions when the amount of training data decreases. It should also be pointed out that using the best configuration ($\rho = 0.5$), the complexity of the BIC systems were fairly close to that of the most complex VarMix systems. For instance, on the 46 hour setup a system with 49.89 useful dimensions per Gaussian and 19.38 components per state on average was selected. This is only about 7% smaller than the 20 component 52 dimensional VarMix configuration.

Marginalized MPE growth function was then used to determine both the number of components and dimensions. The bottom section of table 7.5 shows the MPE GFunc systems' WER along with the their sizes. Across all four training sets, significant WER reductions of 0.4%-1.0% absolute were obtained over the VarMix baselines. For example, on the 76 hour setup, a highly compact system with 14.52 components per state and 36.67 dimensions per Gaussian on average was selected. This MPE GFunc system outperformed the most complex 16 component 52 dimensional VarMix baseline by 1.0%. The gain over the best VarMix configuration (12 component per state and 52 dimensions per Gaussian) was 0.7%. Similarly on the 297 hour set, the MPE GFunc system (17.54 components per state and 44.77 dimensions per Gaussian) outperformed the best, and also most complex, VarMix system on the same setup by 0.6% absolute.

Compared with all the BIC systems in the table, the MPE GFunc approach outperformed the best BIC configuration ($\rho = 0.5$) on the 76 hour training set by 0.3%, and a statistical significant 0.4% on the 297 hour corpus. On the 148 hour set, the MPE GFunc system (15.43 components per state and 47.23 dimensions per Gaussian) outperformed the best BIC system ($\rho = 0.5$, 17.13 components per state and 50.17 dimensions per Gaussian) by having fewer parameters. On the 46 hour setup, although the MPE GFunc system (18.34 components per state and 41.78 dimensions per Gaussian) was outperformed by the best BIC system ($\rho = 0.5$, 19.38 components per state and 49.89 dimensions per Gaussian) by a marginal 0.1%, it has approximately 20% fewer parameters. For all training sets, the MPE GFunc system was more compact than the comparable best BIC system. For example, on the 76 hour set the MPE GFunc system (14.52 components per state and 36.67 dimensions per Gaussian) is about 25% smaller than the best BIC configuration ($\rho = 0.5$, 15.57 components per state and 49.36 dimensions per Gaussian). Like the results in table 7.3, it is interesting to find that the MPE GFunc system requires no tuning in terms of the nature of the complexity attributes being optimized. One again the scheme outperformed, or approximately matched, the best manually tuned system with a more compact model structure for each training set. This is a desirable feature of a good complexity control technique.

7.1.3.4 Correlation Between Criteria and WER

In section 7.1.2 the correlation between standard complexity control techniques and WER was investigated when optimizing complexity attributes on a global level. In this section this correlation is further examined. The aim here is to intuitively show that a strong correlation between

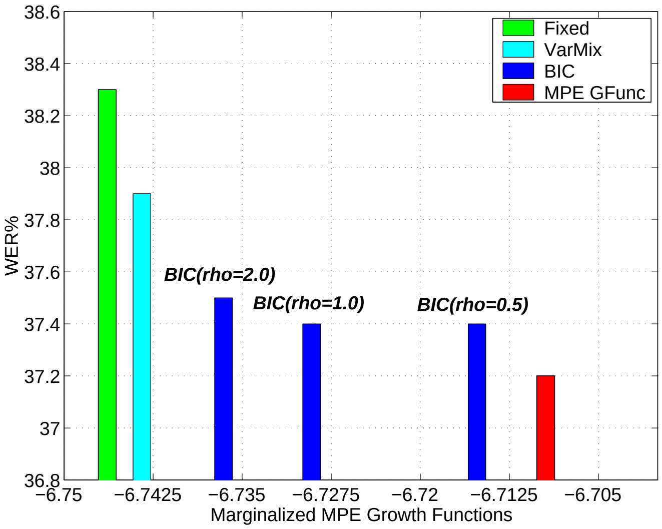

Figure 7.5 *Log Scale Marginalized MPE growth function vs. WER for* dev01sub *on CTS English 46 hour* minitrain04

marginalized discriminative growth functions and WER exists for complexity control on a local level. As the complexity is varied locally, the permutation of all possible configurations is intractable. Hence, the correlation was only investigated for selected systems. Initially several global HLDA systems trained on the 46 hour set minitrain04 in table 7.3 were selected for this purpose: the 12 component Fixed and VarMix systems, all three BIC systems and the MPE GFunc system. For each of these systems, the value of the marginalized MPE growth function was computed on a log scale to compare with the variation of WER. This correlation is shown in figure 7.5. A general trend is observed that increasing the marginalized MPE growth function never increased the WER.

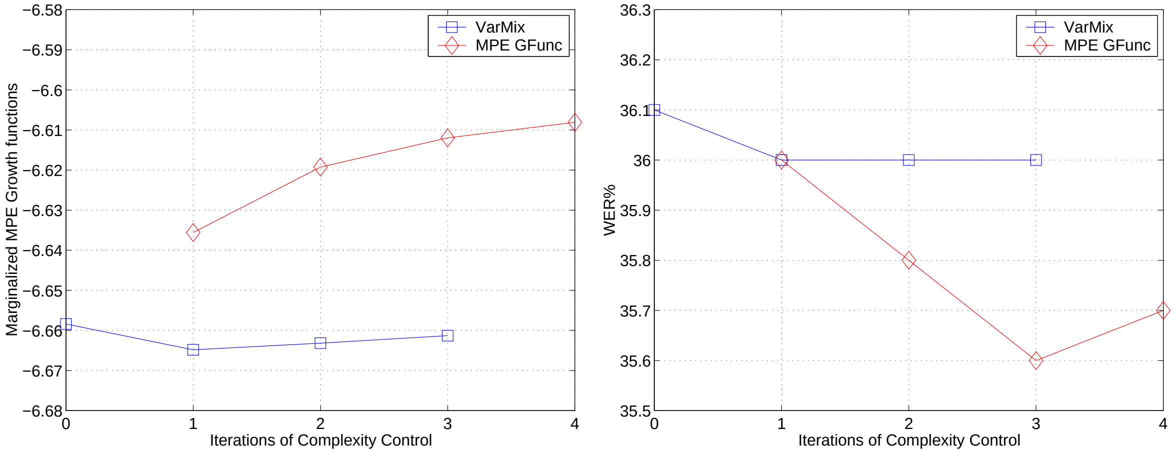

Figure 7.6 *Log Scale Marginalized MPE growth function vs. WER for* dev01sub *on CTS English 76 hour* h5etrain03sub

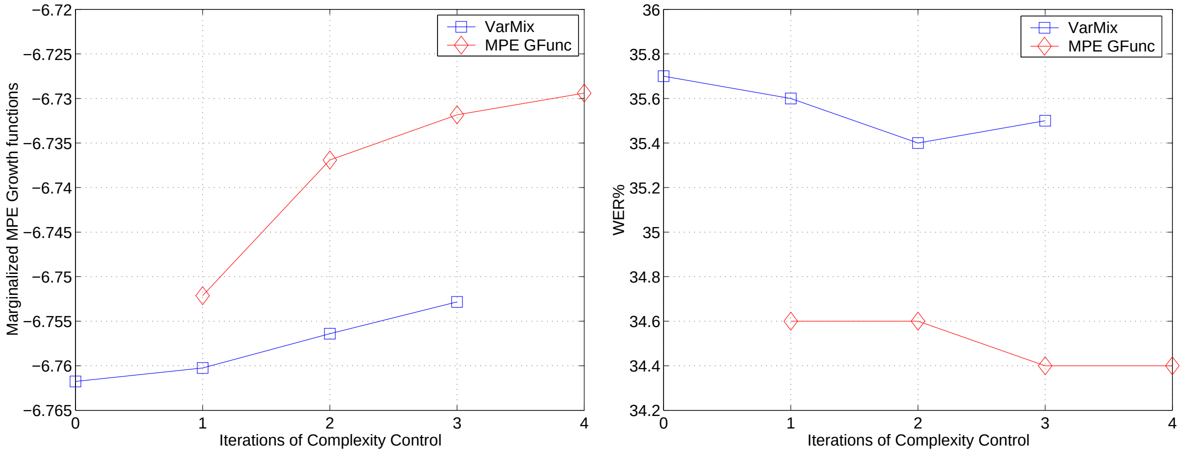

Figure 7.7 *Log Scale Marginalized MPE growth function vs. WER for* dev01sub *on CTS English 148 hour* meditrain04

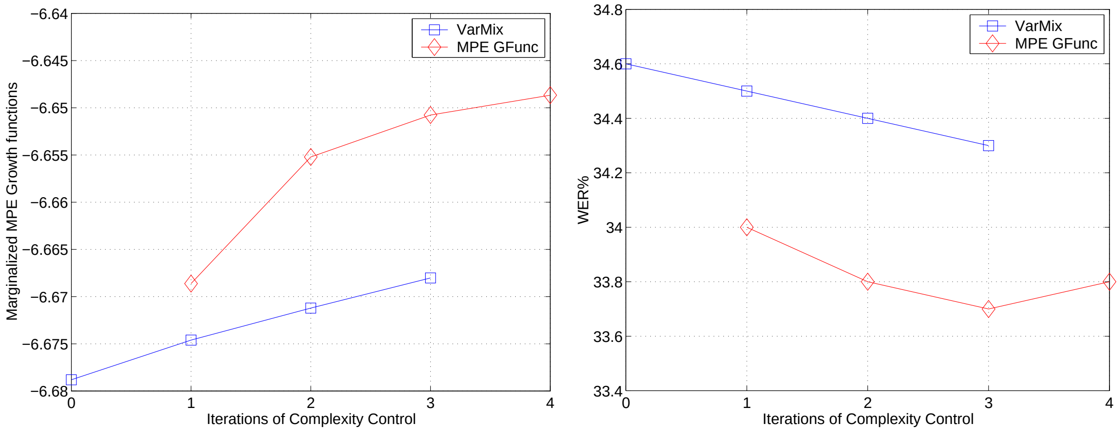

Figure 7.8 *Log Scale Marginalized MPE growth function vs. WER for* dev01sub *on CTS English 297 hour* h5etrain03

The correlation between marginalized MPE growth functions and WER was further investigated on the other three larger training sets using selected systems. For the 76, 148 and 297 hour sets, the 12 component VarMix baselines and the MPE GFunc systems in table 7.3 were selected first. As discussed in section 5.6.2, the optimal model complexity was determined in an iterative mode for the MPE GFunc system in order to obtain reliable statistics. Four iterations of structural optimization were performed for the MPE GFunc systems. Similarly, the VarMix systems were also trained in an iterative fashion. The number of components per state was adjusted three times according to the state occupancies. Hence, it is also interesting to investigate the correlation between WER and marginalized MPE growth functions for systems developed at each iteration of complexity control. This gives a total of 8 systems for each training setup, including the 12 component Fixed systems from which the VarMix systems were derived. The values of marginalized MPE growth functions were computed for each system to compare with the WER. Figures 7.6, 7.7 and 7.8 show the variation of marginalized MPE growth functions and WER performances at different stages of structural optimization for VarMix and MPE GFunc systems on each training setup. In these figure again a general trend was observed that increasing the marginalized MPE growth function's value will decrease the WER. There is also a steady increase of the marginalized MPE growth function between iterations for the MPE GFunc system. In contrast the variation for the VarMix systems was fairly noisy in some cases, for example, in figure 7.6 for the 76 hour set. This difference may be expected as the two schemes are very different model selection criteria.

It is interesting to further examine the differences between the VarMix and MPE GFunc systems in terms of the model complexity determined. The 76 hour set h5etrain03sub was taken as an example. The structural difference between the 16 component VarMix, and the GFunc system in table 7.3 was investigated for this training set. Figure 7.9 illustrates the log scale histogram distribution of the number of components assigned to each state. In the figure the differences these two systems are clearly shown. As expected for the MPE system the number of components in a state is always no larger than 16, because the 16 component Fixed system was used as the starting model for component merging. In contrast, the maximum number of components per state in the VarMix system can be as large as 22. Furthermore, the modes of the two distributions are also fairly apart from one another.

7.1.3.5 MMI Growth Functions vs. MPE Growth Functions

Although the MPE criterion should be more directly related to WER than MMI, it is still interesting to compare the performances of marginalizing the MPE and MMI growth functions for complexity control. This was investigated on the 46 hour set minitrain04. Table 7.6 shows the performances of various complexity controlled systems built using marginalized MPE or MMI growth functions. Comparing the two global HLDA systems, using the MMI growth function, a slightly more complex system with 18.85 Gaussians per state was selected. This system was also outperformed by the MPE GFunc system by 0.2% absolute. However, after optimizing the number of useful dimensions the two systems gave roughly the same error rates. This is because

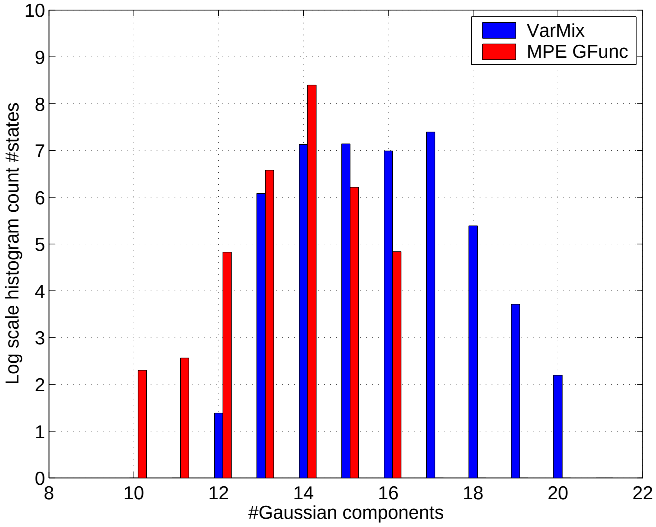

Figure 7.9 *Histogram distribution of #Gauss per state on CTS English 76 hour* h5etrain03sub

Complexity Control				Crit	#Trans	WER%
#Gauss		#Dim				
GFunc	18.34	Fixed	39	MPE	1	37.2
		GFunc	41.78		65	36.7
GFunc	18.85	Fixed	39	MMI	1	37.4
		GFunc	47.91		65	36.6

Table 7.6 *Marginalized MMI or MPE growth functions for* dev01sub *on CTS English 46 hour* minitrain04

more gains were obtained when using marginalized MMI growth functions to optimize the dimensionality for multiple HLDA. The MMI GFunc system gave a WER of 36.6% and also matched the performance of the best manually tuned BIC ($\rho = 0.5$) system in table 7.5. However this MMI GFunc system (18.85 components per state and 47.91 dimensions per Gaussian) was more complex than the MPE GFunc system in the table. Overall, with this setup the two approaches yielded similar performances and the MPE growth function tended to select a more compact system. Experiments in the following sections will still focus on using marginalized MPE growth functions for complexity control.

7.1.3.6 Discussion

In this section a series of complexity control experiments were conducted on an LVCSR task. Four training sets of CTS English data were used. Their sizes were increased approximately log-linearly so as to fully investigate marginalized discriminative growth functions for complexity control. Two attributes of an HLDA system, the number of Gaussian components per state and the number of useful dimensions per Gaussian were optimized on a local level. Important conclusions from these experiments may be summarized as:

- First, across different training sets and multiple forms of complexity attributes, the marginalized MPE growth function will at least select a compact system with approximately the lowest WER, if not giving further gains over the best manually tuned VarMix or BIC systems. As discussed in section 3.1, explicitly building and evaluating all possible systems is intractable for LVCSR tasks. The ability of automatically selecting the correct complexity without excessive tunning of free parameters is an important feature of a good complexity control scheme.

- Second, the correlation between marginalized discriminative growth functions and WER was examined for all four CTS training sets when the model complexity is locally optimized. In figures 7.5, 7.6, 7.7 and 7.8, a fairly strong correlation between marginalized MPE growth functions and WER was observed. These illustrate that marginalized MPE growth functions are closely related to the the WER. Hence, this technique may be an alternative to standard complexity control techniques under the likelihood framework for current speech recognition systems.

- Third, for multiple HLDA systems, it is beneficial to optimize the number of useful dimensions for each projection locally using an appropriate model selection technique. The complexity control gains from the MPE GFunc systems in table 7.5 show that is it preferable to do so when building multiple HLDA systems.

7.2 Interaction with Other Techniques

State-of-the-art LVCSR systems are highly complex. Many techniques may be used to improve the recognition performance. In all the previous complexity control experiments only the performances of ML trained systems were considered. In this section the interaction between complexity control and two important acoustic modeling techniques, discriminative training and speaker adaptation, is investigated.

7.2.1 Interaction with Discriminative Training

As discussed in chapter 4, the model correctness assumption in the ML training may be too strong for HMM based current speech recognition systems. The majority of state-of-the-art LVCSR systems are built using discriminative training techniques. All the MPE GFunc systems so far were ML trained, although discriminative statistics were used to select the optimal structural configuration. In this section, after determining the optimal model structure, model parameters are further updated discriminatively using the standard MPE training [90, 93]. The aim was to investigate the interaction between discriminative training and complexity control. This interaction will be investigated on both the 76 hour CTS English corpus h5etrain03sub and the 297 hour full set h5etrain03, as were described in section 7.1.1. For evaluation the same test set used previously, dev01, was used. Other experimental conditions remain the same as in section 7.1.1.

Complexity Control				#Trans	WER%	
#Gauss		#Dim			MLE	MPE
VarMix	16	Fixed	39	1 65	35.7 35.9	33.0 32.8
BIC $\rho = 0.5$	15.57	BIC $\rho = 0.5$	49.36	65	34.9	32.4
MPE GFunc	14.52	MPE GFunc	36.67	65	34.6	31.9

Table 7.7 *MPE training of complexity controlled systems for* dev01sub *on CTS English 76 hour* h5etrain03sub

For the 76 hour set, three baseline systems were used. The 16 component VarMix system with a global HLDA transform in table 7.3 and the comparable multiple HLDA system in table 7.5 were selected as two standard configurations. Based on the WER performances, the best 65 transform BIC ($\rho = 0.5$) system in table 7.3 was also selected. The MPE GFunc system in table 7.3 was MPE trained to compare with these three baseline systems. Four iterations of MPE training were performed for each system with the HLDA transforms kept fixed. Table 7.7 shows that MPE training reduced the WER for all systems by more than 2.5% absolute. Most of the gain from the GFunc system was maintained after MPE training. There were still significant WER gains of 1.1% and 0.9% absolute from the GFunc system over the two VarMix baselines respectively. It is also interesting to find that some gain from the most complex BIC system was lost. This may be because compact systems are often preferred for MPE training to ensure good generalization [93].

Complexity Control				#Trans	WER%	
#Gauss		#Dim			MLE	MPE
VarMix	20	Fixed	39	1	33.9	30.3
				65	34.2	30.3
BIC $\rho = 0.5$	19.21	BIC $\rho = 0.5$	50.91	65	33.4	29.9
MPE GFunc	17.54	MPE GFunc	44.77	65	33.0	29.4

Table 7.8 *MPE training of complexity controlled systems for* dev01sub *on CTS English 297 hour* h5etrain03

Similarly for the 297 hour full set, the 20 component VarMix global and multiple HLDA system with 39 useful dimensions in table 7.3 and 7.5 were selected as the baseline standard configurations. Based on the WER performances, the best 65 transform BIC system ($\rho = 0.5$) in table 7.5 was also selected. These are to be compared with the GFunc system in table 7.5 after four iterations of MPE training. As is shown in table 7.8, MPE training led to large reduction of the error rates for all systems by more than 3.5% absolute. The MPE GFunc system still significantly outperformed both VarMix baselines by 0.9%, and the BIC system by 0.5% after MPE training. In contrast, some gain from BIC was lost after MPE, like the results shown in table 7.7 for the 76 hour subset. Again this is expected as compact systems are often preferred for discriminative training.

7.2.2 Interaction with Speaker Adaptation

As discussed in section 2.5, characteristics of speech signals vary substantially across different speakers and acoustic environments. The majority of the state-of-the-art LVCSR systems employ standard adaptation techniques like MLLR to remove such variability [126, 51, 23, 64]. So far all the complexity controlled systems considered in this thesis are speaker independent models. Hence, it is interesting to further investigate the interaction between complexity control and adaptation techniques.

Following the MPE training experiments in section 7.2.1, MLLR based speaker adaptation was performed for systems trained on the 76 hour set h5etrain03sub in table 7.7, and the 297 hour full set h5etrain03 in table 7.8. Each system's recognition output from up-adapted decoding was used as its own supervision. Two MLLR mean transforms, one for speech and one for silence Gaussian components were estimated for each system. During the estimation of the MLLR matrices, the diagonal variance approximation described in section 2.6 was used. Table 7.9 shows the adapted performances of four complexity controlled systems on the 76 hour set. Using MLLR the error rates were reduced by 1.9%$\sim$2.0% for all systems. The gains from the GFunc system was largely maintained after the adaptation over the VarMix and BIC baselines.

For the 297 hour full set, the four MPE trained systems in table 7.8 were adapted in the same

Complexity Control				#Trans	WER%		
#Gauss		#Dim			MLE	MPE	MLLR
VarMix	16	Fixed	39	1	35.7	33.0	31.0
				65	35.9	32.8	30.8
BIC $\rho = 0.5$	15.57	BIC $\rho = 0.5$	49.36	65	34.9	32.4	30.5
MPE GFunc	14.52	MPE GFunc	36.67	65	34.6	31.9	30.0

Table 7.9 *Adapted performances of complexity controlled systems for* dev01sub *on CTS English 76 hour* h5etrain03sub

Complexity Control				#Trans	WER%		
#Gauss		#Dim			MLE	MPE	MLLR
VarMix	20	Fixed	39	1	33.9	30.3	28.6
				65	34.2	30.3	28.6
BIC $\rho = 0.5$	19.21	BIC $\rho = 0.5$	50.91	65	33.4	29.9	28.1
MPE GFunc	17.54	MPE GFunc	44.77	65	33.0	29.4	27.7

Table 7.10 *Adapted performances of complexity controlled systems for* dev01sub *on CTS English 297 hour* h5etrain03

fashion as the 76 hour systems. Table 7.10 shows the adapted performances of these systems. Again all the adapted systems outperformed the unadapted by more than 1.5% absolute. Significant WER gains from the GFunc system, 0.9% over both VarMix baselines and 0.4% over the BIC system, were maintained after adaptation.

7.2.3 Discussion

In this section the interaction between model complexity control and standard acoustic modeling techniques was investigated on an LVCSR task for CTS English data. As shown in previous experiments of section 7.1, using marginalized MPE growth functions more compact models tend to be selected. This is particularly useful for discriminative training techniques, as good generalization to unseen data is desired. Hence, the gains from marginalized discriminative growth functions over standard complexity control schemes were found to be mostly additive to discriminative training and speaker adaptation. This indicates that marginalized discriminative functions based complexity control may be useful for state-of-the-art LVCSR systems that use large scale discriminative training and sophisticated adaptation procedures. This is further investigated in the following sections.

7.3 Generalization to Other Tasks

All the previous experiments in sections 7.1.2 and 7.1.3 were conducted on CTS English data. As a general form of complexity control technique, marginalized discriminative growth functions is expected to be applicable to other speech recognition tasks. In this section the generalization of this technique is examined on two very different LVCSR tasks, broadcast news (BN) English and CTS Mandarin Chinese data.

7.3.1 Experimental Conditions

A 72 hour CTS Mandarin Chinese training set, swmtrain04, was used. It consists of 200 Call Home Mandarin (CHM) and 84 Call Friend Mandarin (CFM) conversation sides collected by LDC, and another 500 sides by Hong Kong University of Science and Technology (HKUST). For performance evaluation two data sets were used: The two hour set dev04, also collected by HKUST contains 48 conversation sides; The one hour long 2003 DARPA Mandarin evaluation set, eval03, consists of a total of 24 CFM conversation sides. The audio data was manually segmented for dev04 and automatically segmented for eval03. Like the CTS English systems described in section 7.1, 52-dimensional PLP features were extracted by appending derivatives up to the third order, and then normalized using VTLN, mean and variance normalization on a conversation side basis. This feature vector was projected down to 39 dimensions using one or more HLDA projections. Then pitch parameters, their first and second derivatives were further appended, yielding construct a 42-dimensional feature vector. For multiple HLDA systems the same component assignment scheme described in section 7.1.1 was used. Continuous density, mixture of Gaussians, cross-word tonal triphone, gender independent HMM systems were used. After phonetic decision tree based tying, there were approximately 4000 speech states. Note that the decision tree was built only using the training data collected by HKUST. For the baseline system the global HLDA transform was also estimated only using the data from HKUST. There are 16 Gaussian components per state on average. More detailed description of the baseline system may be found in [35]. Unless otherwise stated ML training were used for training all systems. All recognition experiments used a 16k word based tri-gram language model for full decoding. As there is no deterministic word segmentation for the Chinese language, the character error rate (CER) is used as a performance measurement, rather than WER.

Experiments on a BN English task were also conducted to investigate the performance of complexity control schemes. A 144 hour training set bnetrain02 was used. It consists of the BN English data released by the LDC in 1997 and 1998. The 1997 data was annotated by the LDC to ensure that each segment was acoustically homogeneous, but the 1998 data was transcribed only at the speaker turn level without distinguishing background conditions [64]. In total, these amounted to approximately 144 hours of usable data. For evaluation, a set of approximately 2.7 hour of 2003 DARPA RT03 evaluation data, eval03, was used. The audio data was automatically segmented. A 52 dimensional cepstral acoustic feature was then generated by appending derivatives up to the third order. Like the previous CTS English experiments in section 7.1, this was projected down to a 39 dimensional feature vector using one or 65 HLDA projections. The same component to transform assignment scheme was also used. Continuous density, mixture of Gaussians, cross-word triphone, gender independent HMM systems were used. There are approximately 7k speech states after decision tree based state tying, and the basic system has 16 Gaussian components per state. All recognition experiments used a 59k word tri-gram language model.

For both tasks training lattices were generated using an ML trained VarMix system with 16 Gaussians per state. A pruned bi-gram language model was also used in generating these lattices. They were further marked with model alignment and kept fixed when using marginalized discriminative growth functions for complexity control. The same set of experiments conducted for the CTS English data in section 7.1.3 are investigated. Two complexity attributes of an HLDA system were optimized: the number of Gaussian components per state, and the number of useful dimensions per Gaussian. The same configurations for BIC and marginalized MPE GFunc systems described in section 5.6 and 7.1.1 were also used in the experiments. Again the smoothing constant C of the MPE growth function was always set to 2.0 and not altered.

7.3.2 Experiments on 72 Hour swmtrain04

Table 7.11 shows the performances of various global HLDA systems after complexity control for the 72 hour Mandarin training set swmtrain04. Based on the WER the 20 component Fixed system was used as the starting model for both BIC and GFunc systems. This is marked with a "$\star$" in the table. For the BIC approach, setting the penalization coefficient $\rho = 1.0$ gave both the lowest WER and a more compact system, compared with the setting of $\rho = 0.5$. Using this BIC system a WER reduction of 0.4% was obtained over the 20 componet VarMix baseline on both test sets. It is also interesting to find that for both the standard ($\rho = 1.0$), and penalized BIC ($\rho = 0.5$) systems, the number of parameters were very similar to that of the starting model, although the WER improvements were 0.3% and 0.5% for the two test sets respectively over the 20 component Fixed system.

Complexity Control		CER%	
		dev04	eval03
Fixed	16	40.4	52.8
	18	40.2	52.0
	20	$39.9^{\star}$	$51.5^{\star}$
VarMix	16	39.8	52.5
	18	40.1	52.0
	20	40.0	51.4
BIC (ρ = 0.5)	19.99	39.6	51.0
BIC (ρ = 1.0)	19.75	39.6	51.0
BIC (ρ = 2.0)	17.22	39.8	51.4
MPE GFunc	18.53	39.7	51.3

Table 7.11 *Optimizing #Gauss for global HLDA systems for mandarin* dev04 *and* eval03 *on CTS Mandarin 72 hour* swmtrain04

Using marginalized MPE growth functions, a system with 18.53 Gaussians per state on average was selected. This system gave a WER of 39.7% on dev04 and approximately matched the performance of the best BIC configurations. On eval03 the performance difference between the best BIC systems and the MPE GFunc system was as big as 0.3%. However, this difference is expected. As described in section 7.3.1, both the decision tree and HLDA projection were generated only using the data collected by HKUST. Furthermore, the eval03 set contains purely LDC CFM data that was not present during decision tree clustering and HLDA estimation. Because of the big mismatch between the LDC and HKUST data [35], for the eval03 set more complex system are favored in order to compensate it. This is clearly shown in table 7.11. For example, increasing the number of components from 18 to 20 for the VarMix system reduced the WER by 0.6% on eval03. In contrast, only a marginal 0.1% improvement was obtained on dev04.

To further explore complexity control schemes on this Mandarin task, the dual complexity control problem in previous CTS English experiments was investigated. Table 7.12 shows the CER performances of various multiple HLDA systems after complexity control on both dev04 and eval03. Like the results shown in table 7.11, increasing the number of components of the VarMix baselines while fixing the dimensionality significantly reduced the CER by 0.4%-1.0% on eval03. Smaller CER gains of 0.3%-0.5% on dev04 were also obtained. The best VarMix system was the

Complexity Control				CER%	
#Gauss		#Dim		dev04	eval03
VarMix	16	Fixed	39	39.9	52.3
	16		52	40.0	51.7
	20		39	39.6	51.3
	20		52	$39.5^{\dagger}$	$51.3^{\dagger}$
BIC $\rho = 0.5$	19.99	BIC $\rho = 0.5$	47.03	39.0	51.1
BIC $\rho = 1.0$	19.75	BIC $\rho = 1.0$	38.98	39.5	51.6
BIC $\rho = 2.0$	17.22	BIC $\rho = 2.0$	30.23	39.7	52.4
MPE GFunc	18.53	MPE GFunc	45.20	39.0	50.9

Table 7.12 *Optimizing #Gauss and #Dim of 65 transform HLDA systems mandarin* dev04 *and* eval03 *on CTS Mandarin 72 hour* swmtrain04

most complex configuration that has 20 component per state and 52 dimensions per Gaussian. This system is marked with a "†" in the table. Using the same starting model as in table 7.11, three BIC systems with varying values of ρ and an MPE GFunc system were built. The best BIC configuration ($\rho = 0.5$, 19.99 com and 47.03 dim) outperformed the best VarMix baseline by 0.5% on dev04 and 0.2% on eval03. Note the standard BIC system ($\rho = 1.0$) selected a system that led to a performance degradation of 0.3% on eval03 compared with the 20 component VarMix baseline. Again performances of all the VarMix and BIC systems in the table show that more complex systems are favored for the eval03 data to compensate for the bias toward the HKUST data. Using marginalized MPE growth functions, a system with 18.53 components per state and 45.20 dimensions per Gaussian was selected. The GFunc system outperformed all the manually tuned VarMix and BIC systems in table on both test sets. For example, the gains over the standard BIC system were 0.5% and 0.7% on dev04 and eval03 respectively.

7.3.3 Experiments on 144 Hour bnetrain02

Table 7.14 shows the performances of global HLDA systems after complexity control on the 144 hour BN English training corpus bnetrain02. In the table marginal gains were obtained from VarMix over the standard Fixed systems. Increasing the number of components from 16 to 20 for the VarMix systems led to saturated WER performances. Based on the WER performances the 20 component Fixed system (marked with a "$\star$" in the table) was used as the starting model for both BIC and MPE GFunc systems. All three BIC systems in the table gave very similar WER. In common with the previous CTS English experiments in table 7.3, setting the BIC penalization coefficient $\rho = 0.5$ gave the best BIC performance of 15.6% on eval03. However, the selected system was still quite complex had 19.56 Gaussian components per state on average. The MPE GFunc system had a more compact model structure with 18.22 Gaussians per state on average. It gave a WER of 15.7% that approximately matched the performance of the best manually tuned BIC system ($\rho = 0.5$) in table 7.13.

Complexity Control		WER%
Fixed	16	15.9
	18	15.9
	20	$15.7^{\star}$
VarMix	16	15.8
	18	15.8
	20	15.7
BIC (ρ = 0.5)	19.56	15.6
BIC (ρ = 1.0)	18.61	15.7
BIC (ρ = 2.0)	16.88	15.7
MPE GFunc	18.22	15.7

Table 7.13 *Optimizing #Gauss of global HLDA systems for* eval03 *on BN English 144 hour* bnetrain02

Table 7.14 also allows examination of the problem of dual complexity control for multiple HLDA systems. The table shows that increasing the number of components or useful dimensions only gave marginal WER gains. BIC and MPE GFunc systems were built using the same starting model as in table 7.13. The best BIC configuration ($\rho = 2.0$) gave a WER of 15.4%. This matched the performance of the most complex VarMix system (marked with a "†" in the table) but with much fewer model parameters. It is interesting that different from previous CTS experiments in section 7.1.1, with this setup increasing the value of the penalization coefficient, ρ, gave the best BIC performance. This may suggest the BIC criterion requires the penalization coefficient to be excessively tuned for different forms of parameters and also different tasks. Again the MPE GFunc approach did not suffer from this limitation on this setup. Using the same configuration as described in section 7.1.1, the MPE GFunc system had 15.14 components per state and 45.92 dimensions per Gaussian and outperformed all tuned systems in the table. In particular, WER gains of 0.1%-0.3% were obtained over the three BIC systems.

7.3.4 Discussion

In this section marginalized discriminative growth functions were used for complexity control on two different LVCSR tasks. The same set of experiments considered in section 7.1.3 for CTS English data were conducted. For both the BN English and CTS Mandarin tasks, marginalized

Complexity Control				WER%
#Gauss		#Dim		
VarMix	16	Fixed	39	15.5
	16		52	15.4
	20		39	15.3
	20		32	15.4†
BIC $\rho = 0.5$	19.55	BIC $\rho = 0.5$	50.70	15.6
BIC $\rho = 1.0$	18.61	BIC $\rho = 1.0$	46.38	15.4
BIC $\rho = 2.0$	16.88	BIC $\rho = 2.0$	35.66	15.4
MPE GFunc	18.22	MPE GFunc	45.92	15.3

Table 7.14 *Optimizing #Gauss and #Dim of 65 transform HLDA systems for* eval03 *on BN English 144 hour* bnetrain02

MPE growth functions outperformed, or at least approximately matched, the performance of the best manually tuned system with a minimum complexity. More importantly the same configurations for the MPE GFunc systems in the previous CTS English experiments were also used. No tuning of any free parameters was required for any of the different tasks considered. This shows that marginalized discriminative growth functions is a general approach for model selection and may be useful for a variety of speech recognition tasks.

7.4 Evaluation in 10 Real-time LVCSR System

In most previous experiments complexity controlled systems were evaluated using a standard single pass Viterbi decoding without speaker adaptation and relatively simple word based tri-gram language models. In contrast, state-of-the-art LVCSR systems often use multiple pass decoding, sophisticated adaptation and large scale language models [23, 64]. To further employ the complimentary effects between different systems, multiple systems' recognition outputs may be combined, using confusion networks (CN) combination [22], or recognizer output voting error reduction (ROVER) [25]. In this section complexity control using marginalized discriminative growth functions will be investigated in the framework of a state-of-the-art multi-pass LVCSR system using sophisticated adaptation, large scale language models and CN based system combination. Under this complex framework, it is possible to obtain a realistic comparison of how complexity control schemes perform in a state-of-the-art LVCSR system.

7.4.1 Experimental Conditions

The CTS English data set used for training, fsh2004sub, consists of 400 hours of Fisher conversations released by the LDC, with a balanced gender and line condition [24]. Quick transcriptions are provided by BBN, LDC and another commercial transcription service. A 6 hour DARPA RT-03 evaluation set, eval03, was used for performance evaluation. It contains 144 conversation sides from the LDC Fisher collection, fsh, and Switchboard II phase 5, s25. The baseline model set had approximately 6k physical states after decision tree based tying. Unless otherwise states, the number of components per state was tuned as 28 on average level using VarMix for all systems. All systems were MPE trained.

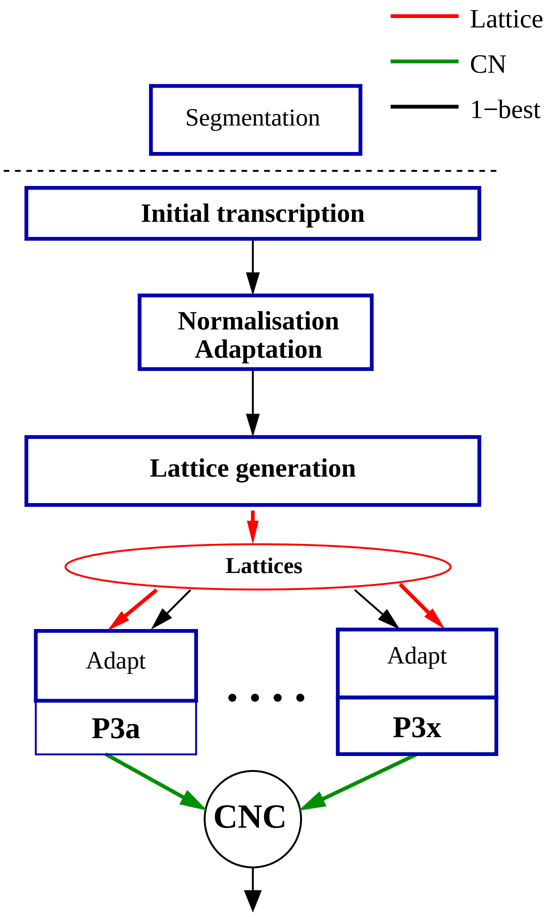

Figure 7.10 *CU-HTK 10xRT System for CTS English*

The CU-HTK 10 real-time multi-pass system was used to evaluate the performance of complexity controlled systems. It uses sophisticated adaptation and CN based system combination. The overall system structure consists of two main stages: the initial lattice generation stage and the rescoring stage using multiple model sets. The confusion network outputs from different rescoring passes were finally combined. This is shown in figure 7.10. More details of the overall system architecture can be found in [21]. The audio data is parameterized using 13 PLP features augmented with their first, second and third order derivatives. A 52 dimensional acoustic feature was projected down to 39 dimension using a global HLDA transform. All acoustic models were built using MPE training. VTLN was used in training and testing. Cepstral mean and variance normalization were also applied. Continuous density, mixture of Gaussians, cross-word triphone gender independent HMM systems were used. The two baseline model sets used in the lattice rescoring stage were a speaker adaptively trained (SAT) model employing constrained MLLR and an HMM set trained using a Single Pronunciation (SPron) dictionary [53]. These model sets were adapted using lattice based MLLR in addition to standard adaptation based only on the 1-best hypothesis. A word-based 4-gram language model was trained on the acoustic transcriptions and additional broadcast news data. The word-based 4-gram was then interpolated with a class-based tri-gram trained only on the associated acoustic transcriptions. The recognition dictionaries contain approximately 58k words. Each word had about 1.1 pronunciations on average level.

7.4.2 10 Real-time System Performances

Table 7.15 shows the baseline performance of the 10 time real-time CTS system. The 2-way combination between the SAT and SPron systems was the standard configuration used in the CUED CTS English evaluation system. Significant error rate reduction over individual branches was achieved after confusions networks combination. The final error rates were 20.5% on eval03.

System		WER%		
		s25	fsh	Avg
P2-cn	HLDA	26.6	18.4	22.6
P3a-cn	SAT	24.5	17.1	20.9
P3c-cn	SPron	24.7	17.6	21.3
P3a+P3c		23.9	16.8	20.5

Table 7.15 *10xRT system baseline performances for* eval03 *on CTS English 400 hour* fsh2004sub

Table 7.16 shows the CN decoding and system combination performances of three additional branches. The global HLDA system used for lattice generation in table 7.15 was also re-adapted as a rescoring branch. This is denoted by P3b in the table. In the previous experiments of section 7.1.3 marginalized MPE growth functions were found to always outperform, or at least match, the performance of BIC. Hence, in the experiments of this section only a 32 components per state 65 HLDA transform VarMix system was built as a baseline. The number of useful dimensions was set as 39 for all projections. The same component to transform assignment scheme as in section 7.1.1 was used. This is denoted by “P3d” in the table. Examining the single branch CN decoding performances, marginal improvement was obtained from the multiple HLDA system over the single transform P3b branch. No WER improvement was obtained if this system is combined with either the standard P3a or P3c branch over the standard two

System		WER%		
		s25	fsh	Avg
P3b-cn	HLDA	24.8	17.7	21.4
P3d-cn	MHLDA	24.5	17.8	21.3
P3e-cn	GFunc	24.5	17.5	21.1
P3a+P3d		23.8	17.0	20.5
P3c+P3d		23.9	16.8	20.5
P3a+P3e		23.7	16.9	20.4
P3c+P3e		23.8	16.9	20.4
P3a+P3c+P3b		23.9	16.6	20.4
P3a+P3c+P3d		23.5	16.7	20.2
P3a+P3c+P3e		23.6	16.5	20.1

Table 7.16 *Extended 10xRT system performances for* eval03 *on CTS English 400 hour* fsh2004sub

way combination in table 7.15. However, a WER reduction of 0.3% was obtained if a three way combination was performed between the two standard, and the P3d branches. This is expected as the multiple HLDA systems is structurally very different from the other two standard systems due to the use of multiple feature spaces. A complexity controlled multiple HLDA system was also built using marginalized MPE growth functions. The same configurations as described in section 7.1.1, were used when determining the optimal complexity. The starting model was a 32 components per state standard system. The GFunc system had 29.9 Gaussians per state and 42.6 useful dimensions per Gaussian. In the CN based word posterior decoding stage, the GFunc system outperformed both VarMix baselines by 0.2%-0.3%. Replacing either of the two standard branches in CNC combination with the P3e system gave marginal WER reduction. Adding the GFunc branch yielded the best system combination performance, which is 0.4% better than the baseline two way combination in table 7.15.

7.4.3 Discussion

In this section complexity control using marginalized discriminative functions was evaluated under a state-of-the-art 10 real-time LVCSR framework. Discriminative training, large scale language models, sophisticated adaptation and system combination were used to obtain the best WER performance. The gains from complexity controlled systems in terms of single branch performances were relatively smaller compared with previous experiments. Nevertheless marginalized discriminative growth functions was still found useful in combination with systems using standard complexity control schemes. This complimentary effect may be partly due to the fundamental difference between marginalized growth functions and standard techniques, as discussed in chapter 5. In previous experiments, for instance in figure 7.9 of section 7.1.3.4, such a difference was clearly reflected in the selected model structure during complexity control.

7.5 Summary

Experimental results using discriminative growth functions for model complexity control were presented in this chapter for LVCSR tasks. Two complexity control attributes of an HLDA system, the number of components per state and the number of useful dimensions per Gaussian were optimized. The global level complexity control considered in section 7.1.2 for a CTS task, allowed explicit construction and evaluation of all possible structural configurations. The correlation between WER and held-out data likelihood was found to be fairly weak for current ASR systems. This indicates that standard complexity control techniques under the likelihood framework may not be appropriate for these tasks. In particular, a limitation of BIC was found when used to simultaneously optimize multiple complexity attributes simultaneously. A strong correlation was observed between the WER and the marginalization of discriminative growth functions. They are more closely related to the recognition error, rather than likelihood.

In the main part of this chapter the same complexity control problem was considered on a local level. A series of experiments were conducted on four CTS English training sets of log-linearly increasing sizes in section 7.1.3. Using the same configurations, marginalized discriminative growth functions will at least select a compact system with approximately the lowest WER among all tuned systems, and in some cases may yield further gains. More importantly the same configurations were used throughout these experiments and no tunning of any free parameters was required. This is a desirable feature of a good complexity control technique. In addition, a strong correlation between marginalized discriminative growth functions and WER was observed. This technique was also found to generalize well to other LVCSR tasks. Furthermore, the gains from these growth function systems were found to be largely additive to other important acoustic modeling techniques including discriminative training and speaker adaption. In the final part of this chapter, marginalized discriminative growth functions was found to yield complimentary gains in a state-of-the-art 10 real-time LVCSR system. Therefore, it may be concluded that marginalized discriminative growth functions is a useful complexity control technique for current speech recognition systems.

8

Experiments on Discriminative Training of Linear Projections

In this chapter the performance of discriminatively trained linear projection schemes are evaluated on three LVCSR tasks. First, experimental results on a CTS transcription task for English data are presented. Then experimental results are presented for BN English and CTS Mandarin transcription tasks. These are followed by an investigation of the use of matched lattices for the discriminative training of multiple projection systems. Finally, both complexity control and parameter estimation are integrated into a consistent discriminative framework. The complexity of discriminatively trained model structures is optimized.

8.1 Experiments on CTS English

Discriminative training of linear projection schemes were initially evaluated for CTS English data. Two training sets, the 76 hour h5etrain03sub and the 297 hour full set h5etrain03, as described in section 7.1.1, were used. The 3 hour subset of 2001 development data, dev01sub, as described in section 7.1.1, was used for performance evaluation. The same component to transform assignment scheme described in section 7.1.1 was also used for multiple projections. Note that in all experiments, unless stated otherwise, neither the number of components per state nor the number of useful dimensions was optimized using any complexity control scheme. The number of useful HLDA dimensions was set as 39 for all projections. Like the complexity control experiments in section 7.1, the lattices used for MPE training on both training sets were generated using the standard 39 dimensional global HLDA systems. They were trained using the ML criterion and had 12 and 16 Gaussians per state respectively. A pruned bi-gram language model was used during decoding. These word lattices were further marked with model alignment and kept fixed for the MPE training of various systems. This was the “exact match” approach described in [124]. Four iterations of MPE training were performed after one re more HLDA transforms were estimated. During the MPE training for all experiments the smoothing constant is set $E = 2.0$, and the I-smoothing constant $\tau^I = 50$. The variance flooring described in section 6.4.1 was used. HLDA transforms may be updated in multiple iterations of MPE training. However, due to the intensive memory storage requirement for full covariance statistics during the transform estimation, the projections are updated only once and then fixed during subsequent MPE training of standard HMM parameters. Other details of the baseline systems were the same as in section 7.1.1.

8.1.1 Experiments on 76 Hour h5etrain03sub

Projection Schemes	#Trans	#Dim	WER%	
			MLE	MPE
HLDA	1	39	36.1	33.1
	65		35.5	32.7
MPE-HLDA	1	39	-	33.0
	65		-	32.4

Table 8.1 *Performances of HLDA systems for* dev01sub *on CTS English 76 hour* h5etrain03sub

Table 8.1 shows the performances of standard HLDA systems. The WER performances of linear projections that were optimized using the MPE criterion are also shown in the table. These are denoted by “MPE-HLDA” in the table. As discussed in section 6.3.1, the discriminative update of HLDA transforms requires the re-estimation of the Gaussian means and covariances using the EBW algorithm. Hence it is only fair to compare the performances of discriminatively trained projections with the ML baselines after MPE training of other HMM parameters. In the table the MPE-HLDA system outperformed the baseline HLDA system with a marginal WER improvement when using a global HLDA projection. Similarly, when using 65 HLDA transforms a marginal WER reduction of 0.3% was obtained from the MPE-HLDA system over the baseline multiple HLDA system. Compared to the baseline global HLDA system, the multiple transform MPE-HLDA system gave a total WER reduction of 0.7% absolute.

8.1.2 Experiments on 297 Hour h5etrain03

Projection Schemes	#Trans	#Dim	WER%	
			MLE	MPE
HLDA	1	39	34.9	30.9
	65		34.2	30.3
MPE-HLDA	1	39	-	30.5
	65		-	30.3

Table 8.2 *Performances of HLDA systems for* dev01sub *on CTS English 297 hour* h5etrain03

To further evaluate the performances of discriminatively trained linear projection schemes, a set of experiments similar to table 8.1 were conducted on the 297 hour full set h5etrain03.

In table 8.2 the global MPE-HLDA system outperforms the comparable HLDA baseline system by 0.4% absolute. Using 65 transforms, an absolute WER reduction of 0.6% was obtained over the global HLDA baseline. Unfortunately there is no performance difference between the two multiple transform systems. One possible reason may be that using the same set of lattices for the MPE training of all systems is inappropriate as the differences among systems are big. The mismatch between lattices and systems will increase as the model structural configurations, for instance the number of projections, and the underlying training criterion vary. This mismatch is further investigated in later sections.

8.2 Experiments on BN English

Experiments on a BN task were also conducted to investigate the performance of discriminative projections. The 144 hour training set bnetrain02, and the 2.7 hour of 2003 DARPA RT03 evaluation data, eval03, as described in section 7.3.1, were used in training and testing. All the other experimental conditions remained the same.

Projection Schemes	#Trans	#Dim	WER%	
			MLE	MPE
HLDA	1	39	15.9	14.1
	65		15.5	14.0
MPE-HLDA	1	39	-	13.9
	65		-	13.8

Table 8.3 *Performances of HLDA systems for* eval03 *on BN English on 144 hour* bnetrain02

Table 8.3 shows the performances of various HLDA systems on the 144 hour BN set bnetrain02. For both global and multiple HLDA systems, optimizing the transform parameters using the MPE criterion yield marginal 0.2% WER improvement. A total WER reduction of 0.3% is obtained over the global HLDA baseline system after four iterations of MPE training. Note that the gain from the multiple HLDA baseline system is greatly reduced from 0.5% to 0.1% after MPE training. Similar to the CTS English experiments in table 8.2, there may be a mismatch between the multiple HLDA systems and the the lattices generated by a system using a global projection.

8.3 Experiments on CTS Mandarin

To further investigate the performances of MPE-HLDA systems, experiments on a CTS Mandarin task were conducted. The 72 hour training set swmtrain94, as described in section 7.3.1, was used in training. The the two test data sets, dev04 and eval03, were also used. Note that the 16 components per state VarMix system was used as the baseline system for this setup. As discussed in section 7.3.1, the parameters of the HLDA projections were only estimated using 52 dimensional PLP features. Pitch parameters were then appended to the projected feature vector. These parameters present one issue for the MPE-HLDA systems. The pitch parameters need to be appropriately initialized after the projections are estimated and before the subsequent MPE update of Gaussian parameters. The approach used here is to take the pitch parameters from the global HLDA ML baseline system and to then append them to the MPE-HLDA systems. Other experimental conditions remain the same as in section 7.3.1. Table 8.4 shows the performances of various HLDA systems. For the ML baselines increasing the number of transforms actually led to marginal performance degradation. For the two systems using a global projection, the MPE-HLDA system outperformed the ML baseline by 0.3% on eval03 although the same WER was obtained on dev04. Comparing the two multiple HLDA systems, significant WER gains from the MPE-HLDA system were obtained, 0.7% on dev04 and 1.0% on eval03. Similarly, significant gains over the baseline global HLDA system were 0.5% on dev04 and 0.8% on eval03 respectively.

Projection Schemes	#Trans	#Dim	dev04		eval03	
			MLE	MPE	MLE	MPE
HLDA	1	39	39.8	36.2	52.5	47.9
	65		39.9	36.4	52.3	48.1
MPE-HLDA	1	39	-	36.2	-	47.6
	65		-	35.7	-	47.1

Table 8.4 *Performances of HLDA systems for* dev04 *and* eval03 *of Mandarin Chinese on 76 hour* swmtrain04

8.4 Experiments on Using Matched Lattices

In all previous experiments word lattices were generated only once using a global HLDA baseline system. These lattices were further marked with model alignment and kept fixed for MPE training of various systems. One important issue with this “exact match” approach is whether it is appropriate to use the same set of lattices for training systems that are very different from the one used to generate them. As discussed in section 6.4.3, ideally individual models should be used to generate the matched lattices for MPE training. Using multiple projections a significant structural difference to a standard global HLDA system is introduced. Furthermore, optimizing the HLDA transform parameters in a discriminative fashion, instead of using the ML criterion, also has a similar impact. Hence, the word level confusion and model alignment given by a ML trained global HLDA system may no longer be appropriate for MPE-HLDA systems. In this section this issue is investigated by using the matched lattices for MPE training of various HLDA systems. The lattices were either completely re-generated via full decoding of the training data, or only re-marked with model alignment using matched systems.

8.4.1 Experiments on 76 Hour h5etrain03sub

On the 76 hour CTS English training set h5etrain03sub, matched lattices were generated by completely re-decoding the training data using the matched acoustic models for the multiple HLDA baseline and the two MPE-HLDA systems in table 8.1 respectively. Matched lattices were then used for the subsequent MPE training of each system while fixing the HLDA projections. Note a different decoder, rather than the one used for training lattices generation in all previous experiments, was used to re-decode the training data.

Projection Schemes	#Trans	#Dim	WER%	
			MLE	MPE
HLDA	1	39	36.1	33.3
	65		35.5	32.7
MPE-HLDA	1	39	-	32.8
	65		-	32.3

Table 8.5 *Performances of HLDA systems for* dev01sub *on CTS English 76 hour* h5etrain03sub *using matched word lattices*

The lattices are found on average smaller than those used for experiments in table 8.1, by approximately 20% in terms of the number of lattice nodes. This may have led to the marginal performance degradation of the global HLDA baseline system in table 8.5. However, this still allows the effect of using matched lattices to be investigated. In the table it is shown that using completely matched training lattices, the WER gain from using MPE-HLDA projections was increased, compared to table 8.1. After four iterations of MPE training, 0.5% absolute WER reduction was obtained from the global transform MPE-HLDA system over the ML single transform baseline. The 65 transform MPE-HLDA system also outperformed the multiple transform ML baseline by 0.4% absolute, and the global HLDA baseline by 1.0% absolute in total.

8.4.2 Experiments on 144 Hour bnetrain02

Re-decoding the training data to obtain matched lattices is highly expensive for LVCSR systems. In order to reduce the computational cost, in this section only the model alignment was re-generated using matched acoustic models. It is therefore assumed that the word level confusion is used for multiple model sets. In this section on the 144 hour BN English training set bnetrain02, matched lattices were generated by re-model marking the same set of word lattices using the multiple HLDA baseline, and the two MPE-HLDA systems in table 8.3 respectively. As in the previous experiments, the same HLDA transforms in table 8.3 were used and kept fixed during the MPE update of standard HMM parameters. Table 8.6 shows the performances of various HLDA systems on eval03, after four iterations of MPE training using lattices with matched model alignment. Compared with previous results in table 8.3, where a single set of lattices was used for all systems, there was marginal improvement from both the multiple HLDA baseline and the global transform MPE-HLDA systems. Unfortunately, no performance improvement was obtained from the multiple transform MPE-HLDA system over the comparable multiple HLDA baseline.

Projection Schemes	#Trans	#Dim	WER%	
			MLE	MPE
HLDA	1	39	15.9	14.1
	65		15.5	13.9
MPE-HLDA	1	39	-	13.8
	65		-	13.8

Table 8.6 *Performances of HLDA systems for* eval03 *on BN English 144 hour* bnetrain02 *using matched phone lattices*

8.4.3 Discussion

In this section matched lattices were used for the subsequent MPE training of standard HMM parameters after linear projections were estimated. Marginal performance improvements were obtained on a CTS English transcription task by completely re-decoding the training data. To reduce the computational cost, for the BN English data the same set of word lattices were re-marked with model alignment using the matched acoustic models. Unfortunately, no significant WER reduction was obtained by only re-generating the model alignment. Re-generating the training data lattices can be very expensive for LVCSR systems. Given the small performance improvements observed, the mismatch between systems and the lattices may be ignored in practice for discriminative training of linear projections.

8.5 Integrated Model Complexity and Parameter Optimization

The complexity control systems considered in chapter 7 were trained using the ML criterion whilst discriminative statistics were used to select the optimal structural configuration. In contrast all the experiments in this chapter so far only considered discriminative training of HLDA systems while the model complexity was not controlled. As discussed in section 6.4.4, it is interesting to integrate both model selection and parameter estimation into a consistent discriminative learning process. When selecting the number of Gaussian components per state, for instance, Gaussian parameters will be considered to be discriminatively estimated for each candidate model structure. Similarly, when selecting the number of useful dimensions for multiple HLDA systems, the HLDA transforms and other model parameters will also be considered to be discriminatively updated. In this section two sets of experiments are conducted on the 46 hour CTS English corpus minitrain04 and the 76 hour set h5etrain03sub, as described in section 7.1.1.

8.5.1 Optimizing the Number of Gaussians

Table 8.7 shows the performances of two global HLDA systems after complexity control and MPE training on the 46 hour CTS training set minitrain04. The first system is the 20 component Fixed baseline in table 7.3 on this setup. The number of components per state was not optimized for this system and there was no merging of Gaussian components. After four iterations of MPE training, the WER was reduced to 34.6%. The second was the GFunc system in table 7.3. Its complexity was determined by considering the parameters to be ML trained during and after complexity control. As discussed in section 5.6.1, the parameters of the merged components are estimated in a standard ML fashion. The combined sufficient ML statistics derived from the merging operation were used to estimate the means and covariances of the merged components. Four iterations of MPE training were performed on top of the final MLE model and the WER was reduced to 34.3%. In contrast, for the second GFunc system the selection of complexity and the parameters update were both discriminative during model selection. Parameters of merged Gaussian components were considered to be MPE updated when determining the number of components for each state. The combined sufficient MPE statistics derived from the merging operation were used for this purpose. Like the baseline GFunc system in the table, a total of four iterations of complexity control were performed using marginalized MPE growth functions. Between iterations model parameters were updated using one iteration of standard MPE training. Other configurations were the same as the MPE GFunc system in table 7.3. Unfortunately, there was no performance improvement by consistently optimizing both the complexity and parameters using a discriminative measure, though the same complexity control gain of 0.3% was obtained over the Fixed baseline after MPE training. In addition, the two GFunc systems selected approximately the same number of components per state.

Complexity Control	Parameter Estimation for Merged Components	#Gauss	WER%	
			MLE	MPE
Fixed	-	20	37.5	34.6
MPE GFunc	MLE	18.34	37.2	34.3
MPE GFunc	MPE	18.23	-	34.3

Table 8.7 *Integrated complexity control and parameter estimation for global HLDA systems for* dev01sub *on CTS English 46 hour* minitrain04

8.5.2 Optimizing the Number of Dimensions

To further investigate the integration of complexity control and parameter estimation, experiments were also conducted on the 76 hour set CTS English h5etrain03sub for multiple HLDA systems. Table 8.8 shows the WER performances of four multiple HLDA systems on dev01sub using the 76 hour h5etrain03sub. The first two Fixed systems are the multiple transform HLDA and MPE-HLDA systems in table 8.1. For neither system was the number of useful dimensions

Complexity Control	HLDA Estimation during Complexity Control	HLDA Estimation after Complexity Control	#Dim	WER%	
				MLE	MPE
Fixed	-	MLE	39	35.5	32.7
Fixed	-	MPE	39	-	32.4
MPE GFunc	MLE	MLE	35.16	35.0	32.3
MPE GFunc	MPE	MPE	38.29	-	32.4

Table 8.8 *Integrated complexity control and parameter estimation for 65 transform HLDA systems for* dev01sub *on CTS English 76 hour* h5etrain03sub

controlled. The third baseline system is a comparable MPE GFunc system on the 76 hour corpus. During and after complexity control, its model parameters, including projections, were considered to be ML trained. After MLE training, four additional iterations of MPE training were performed while the ML trained HLDA projections were fixed. In contrast the second GFunc system table had an integrated complexity and parameter optimization. During and after complexity control, all model parameters, including the HLDA projections, were discriminatively estimated. Other experimental conditions remained the same as the baseline GFunc system. Unfortunately, this system gave slight performance degradation compared with the baseline GFunc system after MPE training. Furthermore, the two systems had rather similar complexity.

8.5.3 Discussion

In this section a consistent discriminative optimization of model complexity and model parameters was investigated on CTS tasks. Initial experimental results show that there is no clear advantage in constraining the parameter estimation to be discriminative during model selection using marginalized discriminative growth functions. This may indicate that the two distinct stages of model building, complexity control and parameter estimation, are independent of one another for current speech recognition systems.

8.6 Summary

In this chapter experimental results of discriminative training of linear projection schemes were presented on three LVCSR tasks. Performance improvements were obtained over standard systems that use ML trained projections. The use of matched lattices for the subsequent discriminative training, after linear projections were estimated, was also investigated. Marginal WER gains were obtained by completely re-decoding the training data using matched acoustic models. Finally, a consistent discriminative optimization of model complexity and parameters was evaluated. It was found that model selection and parameter estimation may be independent of one another for current speech recognition systems. Initial experimental results showed no advantage in constraining the criteria for model selection and parameter estimation to be of the same discriminative nature during complexity control.

9

Conclusion and Future Work

In this thesis the automatic complexity control and feature selection problems for HMM based recognition systems are investigated. First, a novel discriminative complexity control framework was proposed. Under this general framework, model selection is based on the marginalization of a discriminative measure. This should be more explicitly related to the recognition error rate than standard likelihood based criteria. Efficient approximation schemes were proposed to make the marginalization more tractable for HMMs. Second, the discriminative training of linear projections was investigated. These projections should yield a compact feature representation with improved discriminative power compared with the standard maximum likelihood approach. Finally, the performances of discriminative complexity control and linear projections were evaluated on a wide range of LVCSR tasks. In this chapter, a more detailed summary of the thesis is presented. Some possible directions for future research are also discussed.

9.1 Review of Work

The theory of model complexity control using the marginalization of a discriminative measure was presented in chapter 5. Most of the standard standard model selection techniques discussed in chapter 3 reply on an inherent assumption that the classification error is strongly correlated with the likelihood on unseen data. Hence increasing the likelihood on the unseen date, or equivalently the marginal likelihood on the observed data, should decrease the error rate. However, this strong assumption is not true for current speech recognition systems using HMMs, as discussed in section 3.6. This is due to the incorrect modeling assumptions about speech signals in these systems. As this correlation is weakened, the predicted performance ranking based on the likelihood will be increasingly poor. This is the rationale behind the discriminative complexity control framework proposed in this thesis. Since the ultimate aim of model complexity control for speech recognition is to minimize the recognition error rate on unseen data, it is more appropriate to marginalize a criterion that is more explicitly related to the error rate. Discriminative criteria are natural choices for this purpose. They are more directly related to recognition error rate than likelihood.

However, due to the sensitivity to outliers, a direct marginalization of these discriminative criteria may be inappropriate for complexity control. For instance, sentences with very low posteriors are heavily weighted for the MMI criterion. The performance ranking prediction may be distorted due to the presence of these outliers. To handle this problem, the proposed method is based on the marginalization of a discriminative growth function. It maintains some of the attributes of the original discriminative criterion and is less sensitive to outliers. The marginalization of this growth function is used to determine the appropriate model complexity. This discriminative framework for complexity control is a very different approach to the standard likelihood based schemes discussed in chapter 3. Bayesian model selection techniques are based on the marginalization of the training data likelihood, or the evidence. In contrast the discriminative model selection method proposed in this thesis is based on a "discriminative evidence" that directly measures the discriminative power of model structures. In section 5.3 a general form of growth function was introduced. Then two forms of discriminative growth functions were proposed for the MPE and MMI criteria in sections 5.4 and 5.5 respectively. To make the marginalization of the two growth functions more tractable, an EM-like approach was used to to yield a lower bound approximation. This lower bound was then marginalized efficiently using Laplace's approximation for complexity control. Finally, in section 5.6 some important implementation issues were discussed to make the marginalization of discriminative growth functions more efficiently for complexity control. In particular, detailed implementation issues for systems using HLDA style linear projections were discussed.

The discriminative training algorithms for linear projections were presented in chapter 6. An important aspect of a speech recognition problem is to derive a good, and compact, feature representation. This should contain sufficient discriminant information to minimize the classification error. One commonly used type of techniques is the linear projection schemes discussed in section 2.4. When using these schemes, the projections are normally trained using the ML criterion. As discussed in sections 4.1 and 6.1, there are incorrect modeling assumptions about speech signals in current HMM based ASR systems. For these systems merely increasing the likelihood on unseen or observed data does not necessarily improve the recognition performance. Hence, in addition to the discriminative control of subspace dimensions, it is also preferable to employ discriminative criteria to estimate linear projections. These criteria are more closely related to the recognition error rate than likelihood. This is the motivation of developing discriminative training schemes for linear projection schemes.

Unfortunately, the existing discriminative training algorithms may not be appropriate to use for linear projections: the EBW algorithm can only be used to optimize standard forms of HMM parameters; gradient descent base numerical techniques are inefficient for LVCSR training and have difficulty guaranteeing convergence in practice. The recently introduced weak-sense auxiliary function approach provides a flexible and intuitive derivation of the EBW algorithm [91, 89, 93]. This method may also be used to efficiently optimize a variety of forms of model parameters including linear projections. In sections 6.3.1 and 6.3.2 a weak-sense auxiliary function was further used for the discriminative estimation of linear projections, as examples of non-standard form of model parameters. Finally, in section 6.4 some implementation issues for the the discriminative estimation of linear projections were discussed. In particular, a consistent discriminative optimization of both model complexity and parameters was discussed by bridging the research in chapter 5 and chapter 6.

Experimental results on complexity control using marginalized discriminative growth functions were presented in chapter 7. As discussed, the key motivation of using the marginalization of a discriminative measure is that this method is more strongly correlated with the recognition error rate than likelihood. This correlation was initially investigated for the optimization of two complexity attributes of an HLDA system trained using the ML criterion. The number of components and number of useful dimensions were controlled globally on an LVCSR task in section 7.1.2. This allowed all possible systems to be explicitly built and evaluated to examine the correlation between WER and complexity control criteria. The correlation between the WER and the likelihood on unseen data was found to be fairly week for current HMM based ASR systems. A limitation of BIC was also found when optimizing multiple complexity attributes simultaneously. This is because the BIC approximation may become increasingly poor as the amount of observed data decreases. Furthermore, the differences in the form of model parameters is not considered by BIC. In the experiments the issues with a direct use of discriminative criteria was also clearly shown. The MMI criterion, for instance, was heavily influenced by outliers sentences with very low posteriors and led to a poor selection of model complexity.

To further investigate model selection using marginalized discriminative growth functions, the same two complexity attributes of HLDA systems were optimized on a local level for a wide range of LVCSR tasks. Experimental results on four CTS English training setups were presented in section 7.1.3. Across different training data sets, if not giving further gains over the best manually tuned system, the marginalized MPE growth function will at least select a compact system with approximately the lowest WER among all tuned systems. Furthermore, the same configurations described in section 5.6 were used throughout these experiments and no tunning of any free parameters was required. A strong correlation between marginalized discriminative growth functions and WER was observed in the experiments. These are desirable features of a good complexity control technique. Using marginalized MPE growth functions compact models tend to be selected. This is particularly useful for discriminative training techniques, as good generalization to unseen data is preferred. Hence, in section 7.2 the gains from these growth function systems were also found most additive to discriminative training, and furthermore, MLLR based speaker adaption. In section 7.3, complexity control using marginalized discriminative growth functions was also found to generalize well to other LVCSR tasks. Finally, in section 7.4 complexity control using marginalized discriminative growth functions was evaluated in a state-of-the-art 10 real-time LVCSR system. WER gains were obtained in both adaptation and system combination stages. Therefore, it may be concluded that marginalized discriminative growth functions is a general form of complexity control technique and may be useful for current speech recognition systems.

Experimental results for the discriminative training of linear projection schemes were presented in chapter 8. HLDA projections estimated using the MPE criterion were evaluated on three LVCSR tasks. Across different training sets and tasks, performance improvements were obtained over the baseline systems using the ML trained projections. Then the use of matched lattices for the subsequent discriminative training of standard HMM parameters after estimating linear projections was investigated. Unfortunately, only small WER gains were obtained by using matched lattices. Considering the trade-off between the computational cost and the relative performance improvement, the mismatch between systems and lattices may be ignored for linear projection schemes in practice. Finally, a consistently discriminative optimization of model complexity and parameters discussed in section 6.4.4 was evaluated. Initial experimental results showed no clear advantage in constraining the criteria for model selection and parameter estimation to be of the same discriminative nature during complexity control. This may indicate that model selection and parameter estimation may be fairly independent of one another for current speech recognition systems. In summary, it may be concluded that discriminatively estimated linear projection schemes are useful to improve the performances of current speech recognition systems.

9.2 Future Work

There are several aspects of the work presented in this thesis may require further investigation, either in terms of different application domains, or modifications to the existing approaches. These are summarized as below:

- Marginalized discriminative growth functions is a general form of model complexity control technique. In this thesis complexity attributes of HLDA systems were optimized. It would be interesting to further apply this technique to control the complexity of other forms of acoustic models, such as the dimensionality of the state space of factor analyzed HMMs [98], or the number of inverse covariance experts in precision matrix modeling [108].

- The discriminative growth functions investigated in this thesis are related to the MPE and MMI criteria. For other pattern classification tasks, alternative forms of error rate measurement, rather than word or sentence level error rate, may be required. In these cases, the marginalized discriminative growth functions based approach may also be used, as long as an appropriate form of growth function is selected for the underlying criterion. Again the growth function selected should still have reduced sensitivity to outliers and be in a relatively tractable form. This provides a flexible framework for model complexity control whichever cost function is used.

- Laplace's approximation was used to compute the marginalization of discriminative growth functions in this thesis. However, this only gives a second order expansion of the growth function integral. Hence, it would be preferable to explore other approximation schemes to incorporate more information from ignored higher order terms.

- Integrating model selection and parameter estimation under a discriminative framework was initially investigated for HLDA systems in this thesis. As this consistent discriminative learning process is a very different approach from ML, or Bayesian, learning, it may be interesting to further explore the advantage of this integration for other forms of statistical models and applications.

A

Derivations of MPE Growth Functions

This appendix details the derivation of the MPE growth function lower bound. The derivation starts from the MPE growth function given in equation 5.6. Finally, the lower bound in equation 5.9, the MPE auxiliary function in equation 5.10 and the statistics in equation 5.11 are derived. Following the definition of the MPE criterion in equation 4.6, the growth function in equation 5.6 may be re-written as

$$\begin{aligned}\mathcal{G}(\lambda) &= \sum_{\tilde{\mathcal{W}}} p(\mathcal{O},\tilde{\mathcal{W}}|\lambda)\mathcal{A}(\tilde{\mathcal{W}},\mathcal{W}) - \mathcal{F}_{\mathtt{mpe}}(\tilde{\lambda})p(\mathcal{O}|\lambda) \\ &\quad + C \sum_{\substack{\tilde{\mathcal{W}} \\ \mathcal{A}(\tilde{\mathcal{W}},\mathcal{W})<\mathcal{F}_{\mathtt{mpe}}(\tilde{\lambda})}} p(\mathcal{O},\tilde{\mathcal{W}}|\lambda)\left[\mathcal{F}_{\mathtt{mpe}}(\tilde{\lambda}) - \mathcal{A}(\tilde{\mathcal{W}},\mathcal{W})\right] \qquad \text{(A.1)}\end{aligned}$$

An important aspect of the growth function is its expansion, $\mathcal{G}(\psi,\lambda)$, over hidden variable sequences, $\{\psi\}$. Following equation A.1 above, this is given by

$$\begin{aligned}\mathcal{G}(\psi,\lambda) &= \sum_{\tilde{\mathcal{W}}} p(\mathcal{O},\psi,\tilde{\mathcal{W}}|\lambda)\mathcal{A}(\tilde{\mathcal{W}},\mathcal{W}) - \mathcal{F}_{\mathtt{mpe}}(\tilde{\lambda})p(\mathcal{O},\psi|\lambda) \\ &\quad + C \sum_{\substack{\tilde{\mathcal{W}} \\ \mathcal{A}(\tilde{\mathcal{W}},\mathcal{W})<\mathcal{F}_{\mathtt{mpe}}(\tilde{\lambda})}} p(\mathcal{O},\psi,\tilde{\mathcal{W}}|\lambda)\left[\mathcal{F}_{\mathtt{mpe}}(\tilde{\lambda}) - \mathcal{A}(\tilde{\mathcal{W}},\mathcal{W})\right] \qquad \text{(A.2)}\end{aligned}$$

All the following derivations are based on various forms of the expansion in equation A.2. To make the growth function marginalization more efficient, a lower bound on $\mathcal{G}(\lambda)$ may be derived using an EM-like approach via Jensen's inequality. In a similar fashion to the log-likelihood bound in equation 2.5, a distribution over the hidden state sequences, $\mathcal{P}(\psi,\tilde{\lambda})$, is required. The lower bound is given by

$$\begin{aligned}\log\mathcal{G}(\lambda) &= \log\sum_{\psi}\mathcal{P}(\psi,\tilde{\lambda})\frac{\mathcal{G}(\psi,\lambda)}{\mathcal{P}(\psi,\tilde{\lambda})} \\ &\geq \sum_{\psi}\mathcal{P}(\psi,\tilde{\lambda})\log\frac{\mathcal{G}(\psi,\lambda)}{\mathcal{P}(\psi,\tilde{\lambda})} \\ &= \mathcal{L}_{\mathtt{mpe}}(\lambda,\tilde{\lambda}) \qquad \text{(A.3)}\end{aligned}$$

In order to make the above bound valid, the hidden variable sequence "posterior" distribution $\mathcal{P}(\boldsymbol{\psi},\tilde{\lambda})$ must satisfy the non-negative and sum-to-one constraint. The form of posterior considered here is

$$\mathcal{P}(\boldsymbol{\psi},\tilde{\lambda}) \quad = \quad \frac{\mathcal{G}(\boldsymbol{\psi},\tilde{\lambda})}{\sum_{\boldsymbol{\psi}}\mathcal{G}(\boldsymbol{\psi},\tilde{\lambda})} \tag{A.4}$$

Note that $\mathcal{P}(\boldsymbol{\psi},\tilde{\lambda})$ is not the true hidden state sequence posterior as used in the standard EM algorithm for ML training. Nevertheless it may still be related to a term, $\gamma^{\texttt{mpe}}_{\boldsymbol{\psi}}(\mathcal{O})$, which may be viewed as the MPE hidden state sequence "occupancy". Following equation A.2, this is given by,

$$\mathcal{G}(\boldsymbol{\psi},\tilde{\lambda}) \quad = \quad p(\mathcal{O}|\tilde{\lambda})\gamma^{\texttt{mpe}}_{\boldsymbol{\psi}}(\mathcal{O}) \tag{A.5}$$

and

$$\begin{aligned}\gamma^{\texttt{mpe}}_{\boldsymbol{\psi}}(\mathcal{O}) \quad = \quad & \sum_{\tilde{\mathcal{W}}} P(\boldsymbol{\psi},\tilde{\mathcal{W}}|\mathcal{O},\tilde{\lambda})\mathcal{A}(\tilde{\mathcal{W}},\mathcal{W}) - \mathcal{F}_{\texttt{mpe}}(\tilde{\lambda})P(\boldsymbol{\psi}|\mathcal{O},\tilde{\lambda}) \\ & +C \sum_{\substack{\tilde{\mathcal{W}} \\ \mathcal{A}(\tilde{\mathcal{W}},\mathcal{W})<\mathcal{F}_{\texttt{mpe}}(\tilde{\lambda})}} P(\boldsymbol{\psi},\tilde{\mathcal{W}}|\mathcal{O},\tilde{\lambda})\left[\mathcal{F}_{\texttt{mpe}}(\tilde{\lambda}) - \mathcal{A}(\tilde{\mathcal{W}},\mathcal{W})\right]\end{aligned} \tag{A.6}$$

When C is sufficiently large the non-negative and sum-to-one constraint will hold for $\mathcal{P}(\boldsymbol{\psi},\tilde{\lambda})$. In order to derive the growth function lower bound in equation 5.9 by further re-arranging equation A.3, another form of $\mathcal{G}(\boldsymbol{\psi},\lambda)$, given in equation A.2, is required. This is given by

$$\begin{aligned}\mathcal{G}(\boldsymbol{\psi},\lambda) \quad = \quad & p(\mathcal{O},\boldsymbol{\psi}|\lambda)\left\{\sum_{\tilde{\mathcal{W}}} P(\tilde{\mathcal{W}}|\boldsymbol{\psi})\mathcal{A}(\tilde{\mathcal{W}},\mathcal{W}) - \mathcal{F}_{\texttt{mpe}}(\tilde{\lambda})\right. \\ & \left. +C \sum_{\substack{\tilde{\mathcal{W}} \\ \mathcal{A}(\tilde{\mathcal{W}},\mathcal{W})<\mathcal{F}_{\texttt{mpe}}(\tilde{\lambda})}} P(\tilde{\mathcal{W}}|\boldsymbol{\psi})\left[\mathcal{F}_{\texttt{mpe}}(\tilde{\lambda}) - \mathcal{A}(\tilde{\mathcal{W}},\mathcal{W})\right]\right\}\end{aligned} \tag{A.7}$$

because for HMMs given the state sequence, the likelihood of observations are independent of the words.

$$p(\mathcal{O},\boldsymbol{\psi},\tilde{\mathcal{W}}|\lambda) \quad = \quad p(\mathcal{O},\boldsymbol{\psi}|\lambda)P(\tilde{\mathcal{W}}|\boldsymbol{\psi}) \tag{A.8}$$

Now following equations A.4, A.5, and A.7, the lower bound in A.3 may be re-arranged as

$$\begin{aligned}\mathcal{L}_{\texttt{mpe}}(\lambda,\tilde{\lambda}) \quad = \quad & \log\mathcal{G}(\tilde{\lambda}) + \sum_{\boldsymbol{\psi}} \frac{\gamma^{\texttt{mpe}}_{\boldsymbol{\psi}}(\mathcal{O})}{\sum_{\boldsymbol{\psi}}\gamma^{\texttt{mpe}}_{\boldsymbol{\psi}}(\mathcal{O})}\log p(\mathcal{O},\boldsymbol{\psi}|\lambda) \\ & - \sum_{\boldsymbol{\psi}} \frac{\gamma^{\texttt{mpe}}_{\boldsymbol{\psi}}(\mathcal{O})}{\sum_{\boldsymbol{\psi}}\gamma^{\texttt{mpe}}_{\boldsymbol{\psi}}(\mathcal{O})}\log p(\mathcal{O},\boldsymbol{\psi}|\tilde{\lambda})\end{aligned} \tag{A.9}$$

and the only term associated with model parameters, λ, is given by

$$\sum_{\boldsymbol{\psi}}\gamma^{\texttt{mpe}}_{\boldsymbol{\psi}}(\mathcal{O})\log p(\mathcal{O},\boldsymbol{\psi}|\lambda) \quad = \quad \sum_{\boldsymbol{\psi}}\gamma^{\texttt{mpe}}_{\boldsymbol{\psi}}(\mathcal{O})\log p(\mathcal{O}|\boldsymbol{\psi},\lambda) + \sum_{\boldsymbol{\psi}}\gamma^{\texttt{mpe}}_{\boldsymbol{\psi}}(\mathcal{O})\log P(\boldsymbol{\psi}|\lambda)$$

For the complexity control problem considered in this work, the state transition probabilities and Gaussian component priors are kept fixed. Hence the term related to the hidden state sequence priors in equation A.9, $\sum_{\boldsymbol{\psi}} \gamma^{\texttt{mpe}}_{\boldsymbol{\psi}}(\mathcal{O}) \log P(\boldsymbol{\psi}|\lambda)$ may be canceled out by $\sum_{\boldsymbol{\psi}} \gamma^{\texttt{mpe}}_{\boldsymbol{\psi}}(\mathcal{O}) \log P(\boldsymbol{\psi}|\tilde{\lambda})$. Now the only term related to model parameters, λ, in equation A.9 is $\sum_{\boldsymbol{\psi}} \gamma^{\texttt{mpe}}_{\boldsymbol{\psi}}(\mathcal{O}) \log p(\mathcal{O}|\boldsymbol{\psi},\lambda)$. For HMMs, rather than using the state sequence posteriors, the hidden state occupancies are normally used. The aim is to to re-express the hidden state sequence posteriors, $\gamma^{\texttt{mpe}}_{\boldsymbol{\psi}}(\mathcal{O})$, given in equation A.6, as the state occupancies given in equation 5.11. To do so $\gamma^{\texttt{mpe}}_{\boldsymbol{\psi}}(\mathcal{O})$ needs to be re-written using the MPE word sequence occupancy defined in equation 4.22. This is given by [1]

$$\begin{aligned}\gamma^{\texttt{mpe}}_{\boldsymbol{\psi}}(\mathcal{O}) &= \sum_{\tilde{\mathcal{W}},\gamma^{\texttt{mpe}}_{\tilde{\mathcal{W}}}\geq 0} P(\boldsymbol{\psi}|\mathcal{O},\tilde{\mathcal{W}},\tilde{\lambda})\gamma^{\texttt{mpe}}_{\tilde{\mathcal{W}}} + \sum_{\tilde{\mathcal{W}},\gamma^{\texttt{mpe}}_{\tilde{\mathcal{W}}}<0} P(\boldsymbol{\psi}|\mathcal{O},\tilde{\mathcal{W}},\tilde{\lambda})\gamma^{\texttt{mpe}}_{\tilde{\mathcal{W}}} \\ &\quad -C\sum_{\tilde{\mathcal{W}},\gamma^{\texttt{mpe}}_{\tilde{\mathcal{W}}}<0} P(\boldsymbol{\psi}|\mathcal{O},\tilde{\mathcal{W}},\tilde{\lambda})\gamma^{\texttt{mpe}}_{\tilde{\mathcal{W}}}. \qquad \text{(A.10)}\end{aligned}$$

When considering HMMs by summing over all the sequences passing through the same state for each time instance, the MPE statistics, $\gamma^{\texttt{mpe}}_j(\tau)$, in equation 5.11 may be derived. Now the only term related to model parameters in equation A.9, $\sum_{\boldsymbol{\psi}} \gamma^{\texttt{mpe}}_{\boldsymbol{\psi}}(\mathcal{O}) \log p(\mathcal{O}|\boldsymbol{\psi},\lambda)$, may be re-written as

$$\sum_{\boldsymbol{\psi}} \gamma^{\texttt{mpe}}_{\boldsymbol{\psi}}(\mathcal{O}) \log p(\mathcal{O}|\boldsymbol{\psi},\lambda) \quad = \quad \sum_{j,\tau} \gamma^{\texttt{mpe}}_j(\tau) \log p(\boldsymbol{o}_\tau|\boldsymbol{\psi}_\tau = \mathcal{S}_j,\lambda).$$

This gives the MPE auxiliary function, $\mathcal{Q}_{\texttt{mpe}}(\lambda,\tilde{\lambda})$, in equation 5.10. Finally, given this form of $\mathcal{Q}_{\texttt{mpe}}(\lambda,\tilde{\lambda})$ the growth function lower bound in equation A.9 may be re-written as in equation 5.9.

[1] Note the binary partition of all possible word sequences with respect to the sign of $\gamma^{\texttt{mpe}}_{\tilde{\mathcal{W}}}$ was also used in the standard form of MPE statistics of equation 5.13 as proposed in [93] for discriminative training.

B

Derivations of MMI Growth Functions

This appendix details the derivation of the MMI growth function lower bound. The derivation starts from the MMI growth function given in equation 5.15. The lower bound in equation 5.17, the MMI auxiliary function in equation 5.18 and the statistics in equation 5.19 are finally derived. Following the definition of the MMI criterion in equation 4.1, the growth function in equation 5.15 may be re-written as

$$\mathcal{G}(\lambda) \;=\; p(\mathcal{O}|\lambda)\left[P(\mathcal{W}|\mathcal{O},\lambda) - P(\mathcal{W}|\mathcal{O},\tilde{\lambda}) + CP(\mathcal{W}|\mathcal{O},\tilde{\lambda})\right] \tag{B.1}$$

An important aspect of the growth function is its expansion, $\mathcal{G}(\boldsymbol{\psi},\lambda)$, over hidden variable sequences, $\{\boldsymbol{\psi}\}$. Following equation B.1 above, this is given by

$$\mathcal{G}(\boldsymbol{\psi},\lambda) \;=\; p(\mathcal{O},\boldsymbol{\psi},\mathcal{W}|\lambda) - P(\mathcal{W}|\mathcal{O},\tilde{\lambda})p(\mathcal{O},\boldsymbol{\psi}|\lambda) + CP(\mathcal{W}|\mathcal{O},\tilde{\lambda})p(\mathcal{O},\boldsymbol{\psi}|\lambda) \tag{B.2}$$

All the following derivations are based on various forms of the expansion in equation B.2. To make the growth function marginalization more efficient, a lower bound of $\mathcal{G}(\lambda)$ may be derived using an EM like approach via Jensen's' inequality. In a similar fashion to the log-likelihood lower bound in equation 2.5, a distribution over the hidden state sequences, $\mathcal{P}(\boldsymbol{\psi},\tilde{\lambda})$, is required. The lower bound is given by

$$\begin{aligned}\log\mathcal{G}(\lambda) &= \log\sum_{\boldsymbol{\psi}}\mathcal{P}(\boldsymbol{\psi},\tilde{\lambda})\frac{\mathcal{G}(\boldsymbol{\psi},\lambda)}{\mathcal{P}(\boldsymbol{\psi},\tilde{\lambda})}\\ &\geq \sum_{\boldsymbol{\psi}}\mathcal{P}(\boldsymbol{\psi},\tilde{\lambda})\log\frac{\mathcal{G}(\boldsymbol{\psi},\lambda)}{\mathcal{P}(\boldsymbol{\psi},\tilde{\lambda})}\\ &= \mathcal{L}_{\texttt{mpe}}(\lambda,\tilde{\lambda})\end{aligned} \tag{B.3}$$

In order to make the above bound valid, the hidden variable sequence "posterior" distribution $\mathcal{P}(\boldsymbol{\psi},\tilde{\lambda})$ must satisfy the non-negative and sum-to-one constraint. The form of posterior considered here is

$$\mathcal{P}(\boldsymbol{\psi},\tilde{\lambda}) \;=\; \frac{\mathcal{G}(\boldsymbol{\psi},\tilde{\lambda})}{\sum_{\boldsymbol{\psi}}\mathcal{G}(\boldsymbol{\psi},\tilde{\lambda})} \tag{B.4}$$

Note that $\mathcal{P}(\boldsymbol{\psi}, \tilde{\lambda})$ is not the true hidden state sequence posterior as used in the standard EM algorithm for ML training. Nevertheless it may still be related to a term, $\gamma^{\mathtt{mmi}}_{\boldsymbol{\psi}}(\mathcal{O})$, which may be viewed as the MMI hidden state sequence "occupancy". Following equation B.2, this is given by

$$\mathcal{G}(\boldsymbol{\psi}, \tilde{\lambda}) \;=\; p(\mathcal{O}, \mathcal{W}|\tilde{\lambda})\gamma^{\mathtt{mmi}}_{\boldsymbol{\psi}}(\mathcal{O}) \tag{B.5}$$

and

$$\gamma^{\mathtt{mmi}}_{\boldsymbol{\psi}}(\mathcal{O}) \;=\; P(\boldsymbol{\psi}|\mathcal{O}, \mathcal{W}, \tilde{\lambda}) - P(\boldsymbol{\psi}|\mathcal{O}, \tilde{\lambda}) + CP(\boldsymbol{\psi}|\mathcal{O}, \tilde{\lambda}). \tag{B.6}$$

When C is large enough the non-negative and sum-to-one constraint will hold for $\mathcal{P}(\boldsymbol{\psi}, \tilde{\lambda})$. To further re-arrange the lower bound in equation B.3, another form of $\mathcal{G}(\boldsymbol{\psi}, \lambda)$, given in equation B.2, is required. This is given by

$$\mathcal{G}(\boldsymbol{\psi}, \lambda) \;=\; p(\mathcal{O}, \boldsymbol{\psi}|\lambda)\left[P(\mathcal{W}|\boldsymbol{\psi}) - P(\mathcal{W}|\mathcal{O}, \tilde{\lambda}) + CP(\mathcal{W}|\mathcal{O}, \tilde{\lambda})\right] \tag{B.7}$$

because for HMMs given the state sequence, the likelihood of observations are independent of the words sequences as given in equation A.8. Now, following equations B.4, B.5, and B.7, the lower bound in B.3 may be re-arranged as

$$\begin{aligned}\mathcal{L}_{\mathtt{mmi}}(\lambda, \tilde{\lambda}) \;=\; & \log \mathcal{G}(\tilde{\lambda}) + \sum_{\boldsymbol{\psi}} \frac{\gamma^{\mathtt{mmi}}_{\boldsymbol{\psi}}(\mathcal{O})}{\sum_{\boldsymbol{\psi}} \gamma^{\mathtt{mmi}}_{\boldsymbol{\psi}}(\mathcal{O})} \log p(\mathcal{O}, \boldsymbol{\psi}|\lambda) \\ & - \sum_{\boldsymbol{\psi}} \frac{\gamma^{\mathtt{mmi}}_{\boldsymbol{\psi}}(\mathcal{O})}{\sum_{\boldsymbol{\psi}} \gamma^{\mathtt{mmi}}_{\boldsymbol{\psi}}(\mathcal{O})} \log p(\mathcal{O}, \boldsymbol{\psi}|\tilde{\lambda})\end{aligned} \tag{B.8}$$

and the only term associated with model parameters, λ, is given by

$$\sum_{\boldsymbol{\psi}} \gamma^{\mathtt{mmi}}_{\boldsymbol{\psi}}(\mathcal{O}) \log p(\mathcal{O}, \boldsymbol{\psi}|\lambda) \;=\; \sum_{\boldsymbol{\psi}} \gamma^{\mathtt{mmi}}_{\boldsymbol{\psi}}(\mathcal{O}) \log p(\mathcal{O}|\boldsymbol{\psi}, \lambda) + \sum_{\boldsymbol{\psi}} \gamma^{\mathtt{mmi}}_{\boldsymbol{\psi}}(\mathcal{O}) \log P(\boldsymbol{\psi}|\lambda)$$

For the complexity control problem considered in this work, the state transition probabilities and Gaussian component priors are kept fixed. Hence the term related to the hidden state sequence priors in equation B.8, $\sum_{\boldsymbol{\psi}} \gamma^{\mathtt{mmi}}_{\boldsymbol{\psi}}(\mathcal{O}) \log P(\boldsymbol{\psi}|\lambda)$ may be canceled out by $\sum_{\boldsymbol{\psi}} \gamma^{\mathtt{mmi}}_{\boldsymbol{\psi}}(\mathcal{O}) \log P(\boldsymbol{\psi}|\tilde{\lambda})$. Now the only term related to model parameters, λ, in equation B.8 is $\sum_{\boldsymbol{\psi}} \gamma^{\mathtt{mmi}}_{\boldsymbol{\psi}}(\mathcal{O}) \log p(\mathcal{O}|\boldsymbol{\psi}, \lambda)$. For HMMs, rather than using the state sequence posteriors, the hidden state occupancies are normally used. The aim is to to re-express the hidden state sequence posteriors, $\gamma^{\mathtt{mmi}}_{\boldsymbol{\psi}}(\mathcal{O})$, given in equation B.6, as the state occupancies, $\gamma^{\mathtt{mmi}}_{j}(\tau)$, given in equation 5.19. For HMMs, by summing over all the sequences passing through the same state for each time instance, the MMI statistics, $\gamma^{\mathtt{mmi}}_{j}(\tau)$, in equation 5.19 may be derived. The only term related to model parameters in equation B.8, $\sum_{\boldsymbol{\psi}} \gamma^{\mathtt{mmi}}_{\boldsymbol{\psi}}(\mathcal{O}) \log p(\mathcal{O}|\boldsymbol{\psi}, \lambda)$, may also be re-written as

$$\sum_{\boldsymbol{\psi}} \gamma^{\mathtt{mmi}}_{\boldsymbol{\psi}}(\mathcal{O}) \log p(\mathcal{O}|\boldsymbol{\psi}, \lambda) \;=\; \sum_{j,\tau} \gamma^{\mathtt{mmi}}_{j}(\tau) \log p(\boldsymbol{o}_\tau|\boldsymbol{\psi}_\tau = \mathcal{S}_j, \lambda).$$

This is the MMI auxiliary function, $\mathcal{Q}_{\mathtt{mmi}}(\lambda, \tilde{\lambda})$, in equation 5.18. Finally, given this form of $\mathcal{Q}_{\mathtt{mmi}}(\lambda, \tilde{\lambda})$ the growth function lower bound in equation B.8 may be re-written as in equation 5.17.

C

Derivations of MPE Training of HLDA

This appendix details the derivation of the gradient of the weak-sense auxiliary function against parameters of HLDA transforms on a row by row basis, as given in equation 6.5. The derivation starts from the gradient of the weak-sense auxiliary function in equations 6.2, and 6.4. Finally the gradient against rows of HLDA transforms in equation 6.5 is derived.

Substituting the gradient information in equation 6.4 into equation 6.2 gives the weak-sense auxiliary function's gradients agains the rows of HLDA projections that associated with the useful and nuisance dimensions repectively. These are given by

$$\begin{aligned}
\left.\frac{\partial \mathcal{Q}(\lambda,\tilde{\lambda})}{\partial \boldsymbol{a}^{(r)}_{i,i\le p}}\right|_{\lambda=\tilde{\lambda}} &= \left[\sum_{j\in r,\tau}(\gamma_j^{\mathtt{num}}(\tau)-\gamma_j^{\mathtt{den}}(\tau))+\sum_{j\in r}D_j\right]\frac{\tilde{\boldsymbol{c}}_i^{(r)\top}}{\tilde{\boldsymbol{a}}_i^{(r)}\tilde{\boldsymbol{c}}_i^{(r)}} \\
&\quad -\sum_{j\in r}\frac{\tilde{\boldsymbol{a}}_i^{(r)}}{\breve{\sigma}_i^{(j)2}}\left\{\sum_{\tau}\left(\gamma_j^{\mathtt{num}}(\tau)-\gamma_j^{\mathtt{den}}(\tau)\right)\left(\boldsymbol{o}_\tau-\boldsymbol{\mu}^{(j)}\right)\left(\boldsymbol{o}_\tau-\boldsymbol{\mu}^{(j)}\right)^\top\right. \\
&\quad \left.+D_j\int p(\boldsymbol{o}|\boldsymbol{\psi}_{\boldsymbol{o}}=\mathcal{S}_j,\tilde{\lambda})\left(\boldsymbol{o}-\boldsymbol{\mu}^{(j)}\right)\left(\boldsymbol{o}-\boldsymbol{\mu}^{(j)}\right)^\top d\boldsymbol{o}\right\} \\
\left.\frac{\partial \mathcal{Q}(\lambda,\tilde{\lambda})}{\partial \boldsymbol{a}^{(r)}_{i,i>p}}\right|_{\lambda=\tilde{\lambda}} &= \left[\sum_{j\in r,\tau}(\gamma_j^{\mathtt{num}}(\tau)-\gamma_j^{\mathtt{den}}(\tau))+\sum_{j\in r}D_j\right]\frac{\tilde{\boldsymbol{c}}_i^{(r)\top}}{\tilde{\boldsymbol{a}}_i^{(r)}\tilde{\boldsymbol{c}}_i^{(r)}} \\
&\quad -\sum_{j\in r}\frac{\tilde{\boldsymbol{a}}_i^{(r)}}{\breve{\sigma}_i^{(j)2}}\left\{\sum_{\tau}\left(\gamma_j^{\mathtt{num}}(\tau)-\gamma_j^{\mathtt{den}}(\tau)\right)\left(\boldsymbol{o}_\tau-\boldsymbol{\mu}^{(g,r)}\right)\left(\boldsymbol{o}_\tau-\boldsymbol{\mu}^{(g,r)}\right)^\top\right. \\
&\quad \left.+D_j\int p(\boldsymbol{o}|\boldsymbol{\psi}_{\boldsymbol{o}}=\mathcal{S}_j,\tilde{\lambda})\left(\boldsymbol{o}-\boldsymbol{\mu}^{(g,r)}\right)\left(\boldsymbol{o}-\boldsymbol{\mu}^{(g,r)}\right)^\top d\boldsymbol{o}\right\}. \qquad \text{(C.1)}
\end{aligned}$$

To simply the above equations, first let us the case of useful dimensions, $i \le p$, for example, and examine the following expression.

$$
\begin{aligned}
& \sum_{\tau}\left(\gamma_j^{\mathtt{num}}(\tau)-\gamma_j^{\mathtt{den}}(\tau)\right)\left(\boldsymbol{o}_\tau-\boldsymbol{\mu}^{(j)}\right)\left(\boldsymbol{o}_\tau-\boldsymbol{\mu}^{(j)}\right)^\top \\
& +D_j\int p(\boldsymbol{o}|\boldsymbol{\psi}_{\boldsymbol{O}}=\mathcal{S}_j,\tilde{\lambda})\left(\boldsymbol{o}-\boldsymbol{\mu}^{(j)}\right)\left(\boldsymbol{o}-\boldsymbol{\mu}^{(j)}\right)^\top d\boldsymbol{o} \\
= & \sum_{\tau}\left(\gamma_j^{\mathtt{num}}(\tau)-\gamma_j^{\mathtt{den}}(\tau)\right)\boldsymbol{o}_\tau\boldsymbol{o}_\tau^\top-2\boldsymbol{\mu}^{(j)}\sum_{\tau}\left(\gamma_j^{\mathtt{num}}(\tau)-\gamma_j^{\mathtt{den}}(\tau)\right)\boldsymbol{o}_\tau^\top \\
& +\sum_{\tau}\left(\gamma_j^{\mathtt{num}}(\tau)-\gamma_j^{\mathtt{den}}(\tau)\right)\boldsymbol{\mu}^{(j)}\boldsymbol{\mu}^{(j)\top}+D_j\left(\int p(\boldsymbol{o}|\boldsymbol{\psi}_{\boldsymbol{O}}=\mathcal{S}_j,\tilde{\lambda})\boldsymbol{o}\boldsymbol{o}^\top d\boldsymbol{o}\right. \\
& \left.-2\int p(\boldsymbol{o}|\boldsymbol{\psi}_{\boldsymbol{O}}=\mathcal{S}_j,\tilde{\lambda})\boldsymbol{o}d\boldsymbol{o}\cdot\boldsymbol{\mu}^{(j)\top}+\boldsymbol{\mu}^{(j)}\boldsymbol{\mu}^{(j)\top}\right)
\end{aligned}
$$

It is known that $p(\boldsymbol{o}|\boldsymbol{\psi}_{\boldsymbol{O}}=\mathcal{S}_j,\tilde{\lambda})=\mathcal{N}(\boldsymbol{o};\tilde{\boldsymbol{\mu}}^{(j)},\tilde{\boldsymbol{\Sigma}}^{(j)})$ is a Gaussian PDF, hence one may have

$$
\begin{aligned}
\int p(\boldsymbol{o}|\boldsymbol{\psi}_{\boldsymbol{O}}=\mathcal{S}_j,\tilde{\lambda})\boldsymbol{o}d\boldsymbol{o} &= \tilde{\boldsymbol{\mu}}^{(j)} \\
\int p(\boldsymbol{o}|\boldsymbol{\psi}_{\boldsymbol{O}}=\mathcal{S}_j,\tilde{\lambda})\boldsymbol{o}\boldsymbol{o}^\top d\boldsymbol{o} &= \tilde{\boldsymbol{\mu}}^{(j)}\tilde{\boldsymbol{\mu}}^{(j)\top}+\tilde{\boldsymbol{\Sigma}}^{(j)}
\end{aligned}
\tag{C.2}
$$

and then equation C.2 may be written as

$$
\begin{aligned}
& \sum_{\tau}\left(\gamma_j^{\mathtt{num}}(\tau)-\gamma_j^{\mathtt{den}}(\tau)\right)\left(\boldsymbol{o}_\tau-\boldsymbol{\mu}^{(j)}\right)\left(\boldsymbol{o}_\tau-\boldsymbol{\mu}^{(j)}\right)^\top \\
& +D_j\int p(\boldsymbol{o}|\boldsymbol{\psi}_{\boldsymbol{O}}=\mathcal{S}_j,\tilde{\lambda})\left(\boldsymbol{o}-\boldsymbol{\mu}^{(j)}\right)\left(\boldsymbol{o}-\boldsymbol{\mu}^{(j)}\right)^\top d\boldsymbol{o} \\
= & \sum_{\tau}\left(\gamma_j^{\mathtt{num}}(\tau)-\gamma_j^{\mathtt{den}}(\tau)\right)\boldsymbol{o}_\tau\boldsymbol{o}_\tau^\top-2\boldsymbol{\mu}^{(j)}\sum_{\tau}\left(\gamma_j^{\mathtt{num}}(\tau)-\gamma_j^{\mathtt{den}}(\tau)\right)\boldsymbol{o}_\tau^\top \\
& +\sum_{\tau}\left(\gamma_j^{\mathtt{num}}(\tau)-\gamma_j^{\mathtt{den}}(\tau)\right)\boldsymbol{\mu}^{(j)}\boldsymbol{\mu}^{(j)\top}+D_j\left(\tilde{\boldsymbol{\mu}}^{(j)}\tilde{\boldsymbol{\mu}}^{(j)\top}+\tilde{\boldsymbol{\Sigma}}^{(j)}-2\boldsymbol{\mu}^{(j)}\tilde{\boldsymbol{\mu}}^{(j)\top}+\boldsymbol{\mu}^{(j)}\boldsymbol{\mu}^{(j)\top}\right) \\
= & \sum_{\tau}\left(\gamma_j^{\mathtt{num}}(\tau)-\gamma_j^{\mathtt{den}}(\tau)\right)\boldsymbol{o}_\tau\boldsymbol{o}_\tau^\top+D_j\left(\tilde{\boldsymbol{\mu}}^{(j)}\tilde{\boldsymbol{\mu}}^{(j)\top}+\tilde{\boldsymbol{\Sigma}}^{(j)}\right) \qquad \text{(C.3)} \\
& +\left[\sum_{\tau}\left(\gamma_j^{\mathtt{num}}(\tau)-\gamma_j^{\mathtt{den}}(\tau)\right)+D_j\right]\boldsymbol{\mu}^{(j)}\boldsymbol{\mu}^{(j)\top}-2\boldsymbol{\mu}^{(j)}\left[\sum_{\tau}\left(\gamma_j^{\mathtt{num}}(\tau)-\gamma_j^{\mathtt{den}}(\tau)\right)\boldsymbol{o}_\tau^\top+D_j\tilde{\boldsymbol{\mu}}^{(j)\top}\right].
\end{aligned}
$$

Using the EBW update for Gaussian means and covariances in equation 4.16, and the numerator and denominator statistics defined in equations 4.17, 4.18, the above may be further simplified as

$$
\begin{aligned}
& \sum_{\tau}\left(\gamma_j^{\mathtt{num}}(\tau)-\gamma_j^{\mathtt{den}}(\tau)\right)\left(\boldsymbol{o}_\tau-\boldsymbol{\mu}^{(j)}\right)\left(\boldsymbol{o}_\tau-\boldsymbol{\mu}^{(j)}\right)^\top \\
& +D_j\int p(\boldsymbol{o}|\boldsymbol{\psi}_{\boldsymbol{O}}=\mathcal{S}_j,\tilde{\lambda})\left(\boldsymbol{o}-\boldsymbol{\mu}^{(j)}\right)\left(\boldsymbol{o}-\boldsymbol{\mu}^{(j)}\right)^\top d\boldsymbol{o} \\
= & \left[\sum_{\tau}\left(\gamma_j^{\mathtt{num}}(\tau)-\gamma_j^{\mathtt{den}}(\tau)\right)+D_j\right]\boldsymbol{\Sigma}^{(j)}
\end{aligned}
\tag{C.4}
$$

and $\boldsymbol{\Sigma}^{(j)}$ is the discriminatively updated full covariance using the EBW algorithm in equation 4.16.

In a similar fashion, examining the following expression for nuisance dimensions, $i > p$, gives

$$\begin{aligned}
& \sum_{j \in r} \left\{ \sum_{\tau} \left(\gamma_j^{\texttt{num}}(\tau) - \gamma_j^{\texttt{den}}(\tau) \right) \left(\boldsymbol{o}_\tau - \boldsymbol{\mu}^{(g,r)} \right) \left(\boldsymbol{o}_\tau - \boldsymbol{\mu}^{(g,r)} \right)^\top \right. \\
& \left. + \int p(\boldsymbol{o} | \psi_{\boldsymbol{O}} = \mathcal{S}_j, \tilde{\lambda}) \left(\boldsymbol{o} - \boldsymbol{\mu}^{(g,r)} \right) \left(\boldsymbol{o} - \boldsymbol{\mu}^{(g,r)} \right)^\top d\boldsymbol{o} \right\} \breve{\sigma}_i^{(j)-2} \\
= & \sum_{j \in r} \breve{\sigma}_i^{(j)-2} \left[\sum_{\tau} \left(\gamma_j^{\texttt{num}}(\tau) - \gamma_j^{\texttt{den}}(\tau) \right) + D_j \right] \boldsymbol{\Sigma}^{(g,r)}. \qquad \text{(C.5)}
\end{aligned}$$

Finally, substituting equations C.4 and C.5 into equation C.1, the gradient against rows of HLDA transforms given in equation 6.5 may be derived, where the sufficient discriminative statistics, $\boldsymbol{G}^{(r,i)}$, are accumulated as in equation 6.6.

Bibliography

[1] H. Akaike (1978). A Bayesian Analysis of the Minimum AIC Procedure, *The Annals of Institute of Statistical Mathematics*, vol. 30, pp. 9–14, April 1978.

[2] H. Attias (1999). Inferring Parameters and Structure of Latent Variable Models by Variational Bayes, in *Proc. of Uncertainty in Artificial Intelligence*, 1999.

[3] L. R. Bahl, P. F. Brown, P. V. de Souza, & L. R. Mercer (1986). Maximum Mutual Information Estimation of Hidden Markov Model Parameters for Speech Recognition, *Proc. ICASSP'86*, vol. 1, pp. 49-52, 1986.

[4] L. R. Bahl & M. Padmanabhan (1998). A Discriminant Measure for Model Complexity Adaptation, *Proc. ICASSP'98*, pp. 453–457, vol. 1, Seattle.

[5] J. K. Baker (1975). The DRAGON System - An Overview, *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 23, no. 1, pp. 24 – 29, February, 1975.

[6] A. R. Barron, J. J. Rissanen & B. Yu (1998). The Minimum Description Length Principle in Coding and Modeling, *IEEE Transactions on Information Theory*,pp. 2743–2760, vol. 44, no. 6, October 1998.

[7] L. Baum, T. Petrie, G. Soules, & N. Weiss (1970). A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains. *The Annals of Mathematical Statistics*, vol. 41, no. 1, pp. 164–171, 1970.

[8] J. R. Belleganda & D. Nahamoo (1990). Tied Mixture Continous Parameter Modelling for Speech Recognition, *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 38, no. 12, pp. 2033–2045, 1990.

[9] J. A. Bilmes, G. Zweig, T. Richardson, K. Flilali, K. Livescu, P. Xu, K. Jackson, Y. Brandman, E. Sandness, E. Holtz, J. Torres & B. Byrne (2001). Discriminatively Structured Graphical Models for Speech Recognition, *CSLP 2001 Summer Workshop Final Report*, Johns Hopkins University, 2001.

[10] N. Campbell (1984). Canonical Variate Analysis - a general formulation. *Australian Journal of Statistics*, Vol. 26, pp. 86 – 96, 1984.

[11] H. Y. Chan & P. C. Woodland (2004). Improving Broadcast News Transcription by Lightly Supervised Discriminative Training, *Proc. ICASSP'04*, Montreal, Canada..

[12] S. S. Chen & P. S. Gopalakrishnan (1998). Clustering Via the Bayesian Information Criterion with Applications in Speech Recognition,*IEEE Transactions on Speech and Audio Processing*,pp. 645–648, vol. 6, 1998.

[13] S. S. Chen & R. A. Gopinath (1999). Model Selection in Acoustic Modeling, *Proc. Eurospeech'99*, pp. 1087–1090, vol. 3, Budapest.

[14] W. Chou, C. H. Lee & B. H. Juang (1993). Minimum Error Rate Training Based On N-Best String Models, in *Proc. ICASSP'93*, pp. 652–655.

[15] W. Chou & W. Reichl (1999). Decision Tree State Tying Based on Penalized Bayesian Information Criterion, *Proc. ICASSP'99*, vol. 1, Phoenix.

[16] B. S. Clarke & A. R. Barron (1990). Information-Theoretic Asymptotic of Bayes Methods, *IEEE Transactions on Information Theory*, pp. 453–471, vol. 30, no. 3, May1990.

[17] S. B. Davis & P. Mermelstein (1980). Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences, *IEEE Transactions on Acoustic, Speech and Signal Processing*, vol. 28, pp. 357 – 366, 1980.

[18] J. R. Deller, J. H. L. Hansen & J. G. Proakis (1993). *Discrete-Time Processing of Speech Signals*, Macmillan, 1993.

[19] A. P. Dempster, N. M. Laird & D. B. Rubin (1977). Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of Royal Statistical Society*, pp. 1–38, vol. 39, 1977.

[20] V. Doumpiotis, S. Tsakalidis & W. Byrne (2004). Discriminative Linear Transforms for Feature Normalization and Speaker Adaptation in HMM Estimation. *IEEE Transactions Speech and Audio Processing*, to appear.

[21] G. Evermann & P. C. Woodland (2003). Design of Fast LVCSR Systems. *Proc. ASRU'03*, St. Thomas, U.S. Virgin Islands.

[22] G. Evermann & P.C. Woodland (2000), Large Vocabulary Decoding and Confidence Estimation using Word Posterior Probabilities. *Proc. ICASSP'00*, pp. 2366 – 2369, Istanbul, 2000.

[23] G. Evermann, H. Y. Chan, M. J. F. Gales, T. Hain, X. Liu, D. Mrva, L. Wang & P. C. Woodland, Development of the 2003 CU-HTK Conversational Telephone Speech Transcription System, *Proc. ICASSP'04*, Montreal, Canada..

[24] G. Evermann, H. Y. Chan, M. J. F. Gales, B. Jia, D. Mrva, P.C. Woodland & K. Yu (2004), Training LVCSR Systems on Thousands of Hours of Data, submitted to *Proc. ICASSP'05*.

[25] J. G. Fiscus (1997). A Post-Processing System to Yield Reduced Word Error Rates: Recogniser Output Voting Error Reduction (ROVER). *Proc. IEEE ASRU'97*, pp. 347 – 352, Santa Barbara, 1997.

[26] K. Fukunaga (1972) Introduction to Statistical Pattern Recognition, Academic Press, 1972.

[27] S. Furui (1986). Speaker Independent Isolated Word Recognition Using Dynamic Features of Speech Spectrum. *IEEE Transactions Acoustic Speech and Signal Processing*, vol. 34, pp. 52 – 59, 1986.

[28] M. J. F. Gales (1996). The Generation and Use of Regression Class Trees for MLLR Adaptation. *Technical Report CUED/F-INFENG/TR.263*, August 1996.

[29] M. J. F. Gales (1997). Adapting Semi-tied Full Covariance Matrix HMMs. *Technical Report CUED/F-INFENG/TR.298*, July 1997.

[30] M. J. F. Gales (1998). Maximum Likelihood Linear Transformations for HMM-based Speech Recognition. *Computer Speech and Language*, vol.12, pp. 75–98, 1998.

[31] M. J. F. Gales (1999). Semi-tied Covariance Matrices for Hidden Markov Models, *IEEE Transactions on Speech and Audio Processing*, pp. 272–281, vol. 7, 1999.

[32] M. J. F. Gales, K. M. Knill & S. J. Young (1999). State-based Gaussian Selectionin Large Vocabulary Continuous Speech Recognition Using HMMs, *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 2, pp. 37 – 47, 1999.

[33] M. J. F. Gales (2001). Maximum Likelihood Multiple Projection Schemes for Hidden Markov Models, *Technical Report CUED/F-INFENG/TR.365*, 2001.

[34] M. J. F. Gales (2002). Maximum Likelihood Multiple Projection Schemes for Hidden Markov Models, *IEEE Transactions on Speech and Audio Processing*, pp. 37–47, vol. 10, 2002.

[35] M. J. F. Gales, B. Jia, X. Liu, K. C. Sim & K. Yu (2004), Development of the CUHTK 2004 Mandarin Conversational Telephone Speech Transcription System, submitted to *Proc. ICASSP'05*.

[36] J. Gauvain & C. H. Lee (1994). Maximum a Posteriori Estimation for Multivariate Gaussian Mixture Observations of Markov Chains, *IEEE Transactions on Acoustic, Speech and Signal Processing*, vol. 2, no. 2, pp. 291 – 299, 1994.

[37] J. Gauvain, L. Lamel, G. Adda & M. Adda-Decker (1994). The LIMSI 93Nov WSJ System, *Proc. ARPA 1994 Spoken Language Technology Workshop*, Plainsboro, New Jersy, 1994.

[38] Z. Ghahramani & M. J. Beal (2000). Graphical Models and Variational Methods, *Advanced Mean Field Method–Theory and Practice*. MIT Press 2000.

[39] Z. Ghahramani & M. J. Beal (1999). Variational inference for Bayesian Mixtures of Factor Analyzers, *Neural Information Processing Systems*, vol. 12, 1999.

[40] Z. Gharamani & G. E. Hinton (1996). The EM Algorithm for Mixtures of Factor Analyzers, *Technical Report CRG-TR-96-1*, Department of Computer Science, University of Toronto, 1996.

[41] C. E. Rasmusse & Z. Ghahramani (2001). Occam's Razor, *Neural Information Processing Systems*, vol. 13, 2001.

[42] V. Goel, S. Kumar, & W. Byrne (2004). Segmental Minimum Bayes-risk Decoding for Automatic Speech Recognition. *IEEE Transactions on Speech and Audio Processing*, to appear.

[43] P. S. Gopalakrishnan, D. Kanevsky, A. Nádas, & D. Nahamoo (1989). Generalization of the Baum algorithm to Rational Objective Functions, *Proc. ICASSP'89*, pp. 631-634.

[44] P. S. Gopalakrishnan, D. Kanevsky, A. Nádas, & D. Nahamoo (1991). An Inequality for Rational Functions with Applications to Some Statistical Estimation Problems, *IEEE Transactions on Information Theory*, January, 1991.

[45] P. Olsen & R. A. Gopinath (2002). Modeling Inverse Covariances by Basis Expansion, *Proc. ICASSP'02*, Florida, USA.

[46] I. J. Good (1953). The population frequencies of species and the estimation of population parameters. *Biometrika*, vol. 40, pp. 237 – 264, 1953.

[47] P. D. Grünwald (1998). *The Minimum Description Length Principle and Reasoning under Uncertainty*, Ph. D. Thesis, University of Amsterdam, Amsterdam, 1998.

[48] A. Gunawardana & W. Byrne (2001). Discriminative Speaker Adaptation with Conditional Maximum Likelihood Linear Regression, *Proc. Euro Speech'01*, Denmark.

[49] A. Gunawardana (2001). *The Information Geometry of EM Variants for Speech and Image Processing*. PhD Thesis, John Hopkins University, April 2001.

[50] A. Gunawardana & W. Byrne (2002). General Extended Baum-Welch Algorithm for Parameter Estimation. Technical Report, Center for Language and Speech Processing, John Hopkins University, May 2002.

[51] T. Hain, P. C. Woodland, G. Evermann, M. J. F. Gales, X. Liu, G. L. Moore, D. Povey & L. Wang (2004). Automatic Transcription of Conversational Telephone Speech. *IEEE Transactions on Speech and Audio Processing*, to appear.

[52] T. Hain (2002). Implicit Pronunciation Modelliing in ASR, *Proc. ITRW PMLA'02*, Estes Park, Colorado, 2002.

[53] T. Hain (2001). *Hidden Model Sequence Model for Automatic Speech Recognition*. PhD Thesis, Cambridge University, April 2001.

[54] M. H. Hansen & B. Yu (2001). Model Selection and the Principle of Minimum Description Length, *JASA*,pp. 746-774 vol. 96, no. 454, 2001.

[55] H. Hermansky (1990). Perceptual Linear Prediction (PLP) of Speech. *Journal of the Acoustics Society of America*, vol. 87, no. 4, pp. 1738 – 1752, 1990.

[56] M. M. Hochberg, L. T. Niles, J. T. Foote & H. F. Silverman (1991). Hidden Markov Model/Neural Network Training Techniques for Connected Alpha-Digit Speech Recognition. *Proc. ICASSP'91*, pp. 109 – 112, Toronto.

[57] X. Huang, A. Acero & H. Hon (2001). Spoken Language Processing, Prentice Hall, 2001.

[58] F. Jelinek (1976). Continuous Speech Recognition by Statistical Methods. *Proc. IEEE* vol. 64, no. 4, April, 1976.

[59] T. Jitsuhiro & S. Nakamura (2004). Automatic Generation of Non-uniform HMM Structures Based on Variational Bayesian Approach, *Proc. ICASSP'04*, Montreal, Canada..

[60] B. H. Juang, W. Chou & C. H. Lee. (1997). Minimum Classification Error Rate Methods for Speech Recognition, *IEEE Transactions on Speech and Audio Processing*, vol. 5, pp. 266–277, 1997.

[61] T. Jung, S. Makeig, M. J. McKeown, A. J. Bell, T. Lee & T. J. Sejnowski (2001). Imaging Brain DynamicsUsing Independent Component Analysis, *Proc. IEEE*, vol. 89, no. 7, July, 2001.

[62] J. Kaiser, B. Horvat & Z. Kacic, A Novel Loss Function for the Overall Risk-criterion Based Discriminative Training of HMM Models, *Proc. ICSLP'00*, Beijing, China.

[63] S. M. Katz (1987). Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recognizer. *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 35, no. 3, pp. 400 – 401, 1987.

[64] D. Y. Kim, G. Evermann, T. Hain, D. Mrva, S. E. Tranter, L. Wang & P. C. Woodland (2003). Recent Advances in Broadcast News Transcription. *Proc. ASRU'03*, St. Thomas, U. S. Virgin Islands.

[65] D. Y. Kim, H. Y. Chan, G. Evermann, M. J. F. Gales, D. Mrva, K. C. Sim & P. C. Woodland (2004), Development of the CU-HTK 2004 Broadcast News Transcription Systems, submitted to *Proc. ICASSP'05*.

[66] N. Kumar (1997). *Investigation of Silicon-Auditory Models and Generalization of Linear Discriminant Analysis for Improved Speech Recognition*, PhD Thesis, John Hopkins University, Baltimore.

[67] N. Kumar & R. A. Gopinath, Multiple Linear Transforms, *Proc. ICASSP'01*, Salt Lake Cit y, Utah.

[68] C. J. Leggetter & P. C. Woodland (1995). Maximum Likelihood Linear Regression for Speaker Adaptation of Continuous Density HMMs. *Computer Speech and Language*, vol. 9, pp. 171 – 186, 1995.

[69] L. A. Liporace (1982). Maximum Likelihood Estimatin for Multivariate Observations of Markov Sources. *IEEE Transactions on Information Theory*, vol. 28, no. 5, pp. 729 – 734, 1982.

[70] X. Liu & M. J. F. Gales (2003). Automatic Model Complexity Control Using Marginalized Discriminative Growth Functions, *Proc. ASRU'03*, St. Thomas, U. S. Virgin Islands.

[71] X. Liu, M. J. F. Gales & P. C. Woodland (2003). Automatic Complexity Control for HLDA Systems, *Proc. ICASSP'03*, vol. 1, Hong Kong.

[72] X. Liu & M. J. F. Gales (2004). Model Complexity Control And Compression Using Discriminative Growth Functions, *Proc. ICASSP'04*, Montreal, Canada..

[73] X. Liu (2001). *Linear Projection Schemes for Automatic Speech Recognition*, MPhil thesis, Department of Engineering, Cambridge University, 2001.

[74] X. Liu & M. J. F. Gales (2004). *Discriminative Training of Multiple Subspace Projections For Large Vocabulary Speech Recognition*, Cambridge University Engineering Department Technical Report, CUED/F-INFENG/TR-489, August 2004.

[75] X. Liu & M. J. F. Gales (2004). *Automatic Model Complexity Control Using Marginalized Discriminative Growth Functions*, Cambridge University Engineering Department Technical Report, CUED/F-INFENG/TR-490, August 2004.

[76] X. Liu, M. J. F. Gales, K. C. Sim & K. Yu (2004), Investigation of Acoustic Modeling Techniques for LVCSR Systems, submitted to *Proc. ICASSP'05*.

[77] A. Ljolje (1994). The Importance of Cepstral Parameter Correlations in Speech Recognition. *Computer Speech and Language*, vol. 8, pp. 223 – 232, 1994.

[78] D. J. C. Mackay (1998). Choices of Basis for Laplace Approximation, *Machine Learning*, vol. 33, no. 1, October 1998.

[79] D. J. C. Mackay (1998). Introduction to Monte Carlo Methods, *Learning in Graphical Models*, NOTA Science Series, pp. 175–204, Kluwer Academic Press, 1998.

[80] J. McDonough, T. Schaaf & A. Waibel (2002). On Maximum Mutual Information Speaker-Adaptation Training, *Proc. ICASSP'02*, Orlando, U. S. A.

[81] A. Nádas (1983). A Decision Theoretic Formulation of a Training Problem in Speech Recognition and a Comparison of Training by Unconditional Versus conditional Maximum Likelihood. *IEEE Transactions on Speech and Audio Processing*, pp. 814–817, vol. 31, 1983.

[82] R. M. Neal (1993) Probabilistic Inference using Markov Chain Monte Carlo Methods, *Technical Report, CGT-TR-93-1*, Department of Computer Science, University of Toronto, 1993.

[83] H. Ney, U. Essen, & R. Kneser (1995). On the Estimation of "Small" Probabilities by Leavingone-out. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 12, pp. 1202 – 1212, 1995.

[84] Y. Normandin (1991). *Hidden Markov Models Maximum Mutual Information Estimation and the Speech Recognition Problem*, PhD thesis, McGill University, Canada.

[85] Y. Normandin (1995). Optimal Splitting of HMM Gaussian Mixture Components with MMIE Training, *IEEE Transactions on Speech and Audio Processing*, pp. 449–453, vol. 5, 1995.

[86] P. Olsen & R. A. Gopinath (2002). Modeling Inverse Covariances by Basis Expansion, *Proc. ICASSP'02*, Florida, USA.

[87] M. Ostendorf, V. Digalakis, & O. Kimball (1996). From HMM's to segment models: A unified View of Stochastic Modeling for Speech Recognition. *IEEE Transactions on Speech and Audio Processing*, vol. 4, no. 5, pp. 360 – 378, 1996.

[88] M. Padmanabhan & L. R. Bahl (2000). Model Complexity Adaptation Using a Discriminant Measure, *IEEE Transactions on Speech and Audio Processing*, pp. 205–208, vol. 8, no. 2, March 2000.

[89] D. Povey, M. J. F. Gales, D. Y. Kim & P. C. Woodland (2003). MMI-MAP and MPE-MAP for Acoustic Model Adaptation, *Proc. Euro Speech'03*, Geneva, Switzerland.

[90] D. Povey & P. C. Woodland (2002). Minimum Phone Error and I-smoothing for Improved Discriminative Training, *Proc. ICASSP'02*, Florida, USA.

[91] D. Povey, P. C. Woodland & M. J. F. Gales (2003). Discriminative MAP for Acoustic Model Adaptation, *Proc. ICASSP'03*, Hong Kong, China.

[92] D. Povey, B. Kingsbury, L. Mangu, G. Saon, H. Soltau & G. Zweig (2005). fMPE: Discriminatively Trained Features for Speech Recognition, *Proc. ICASSP'05*, Philadelphia, USA.

[93] D. Povey (2003). *Discriminative Training for Large Vocabulary Speech Recognition*, PhD thesis, Cambridge University Engineering Department, England.

[94] C. Rathinavalu & L. Deng (1997). HMM Based Speech Recognition Using State-dependent Discriminatively Derived Transforms on Mel-warped DFT features. *IEEE Transactions on Speech and Audio Processing*, pp. 243-256, vol. 3, 1997.

[95] J. J. Rissanen (2001). Strong Optimality of the Normalized ML Models as Universal Codes and Information in Data, *IEEE Transactions on Information Theory*, vol. 47, No 5, July 2001.

[96] J. J. Rissanen (1996). Fisher Information and Stochastic Complexity, *IEEE Transactions on Information Theory*, pp. 40–47, vol. 42, no. 1, January 1996.

[97] C. P. Robert & G. Gassela (1999). *Monte Carlo Statistical Methods*, Springer-Verlag, 1999.

[98] A-V. I. Rosti & M. J. F. Gales (2004). Factor Analysed Hidden Markov Models for Speech Recognition. *Computer Speech and Language*, pp. 181–200, vol. 2, 2004.

[99] A-V. I. Rosti & M. J. F. Gales (2002). Factor Analyzed Hidden Markov Models, *Proc. ICASSP'02*, Florida, USA.

[100] A-V. I. Rosti & M. J. F. Gales (2001). Generalized Linear Gaussian Models, *Technical Report CUED/F-INFENG/TR.420*, 2001.

[101] A-V. I. Rosti & M. J. F. Gales (2004). Switching Linear Dynamical Systems for Speech Recognition, *Technical Report CUED/F-INFENG/TR.461*, 2004.

[102] G. Saon, M. Padmanabhan, R. A. Gopinath & S. S. Chen (2000). Maximum Likelihood Discriminant Feature Spaces,*Proc. ICSLP'00*, Beijing, China.

[103] M. Saraclar, H. J. Nock & S. Khudanpur (2000). Pronunciation Modelling by Sharing Gaussian Densities across Phonetic Models, *Computer Speech and Language*, vol. 14 pp. 137 – 160, 2000.

[104] G. Schwartz (1978). Estimating the Dimension of a Model, *The Annals of Statistics*, pp. 461–464, vol. 6, no. 2, February 1978.

[105] K. Shinoda & T. Watanabe (1997). Acoustic Modeling based on the MDL Principle for Speech Recognition, *Proc. Eurospeech'97*,

[106] K. Shinoda & T. Watanabe (1995). Speaker Adaptation with Autonomous Model Complexity Control by MDL Principle, *IEEE Transactions on Speech and Audio Processing*, pp. 717–720, vol. 8, 1995.

[107] K. Shinoda & T. Watanabe (2000). MDL Based Context Dependent Subword Modeling for Speech Recognition, *Journal of Acoustic Society of Japan*, vol. 21, pp. 79–86, 2000.

[108] K. C. Sim & M. J. F. Gales (2004). Basis Superposition Precision Matrix Modeling For Large Vocabulary Continuous Speech Recognition, *Proc. ICASSP'04*, Montreal, Canada..

[109] R. Schlüter & W. Machery (1998). Comparison of Discriminative Training Criteria, *Proc. ICASSP'98*, USA.

[110] R. Schlüter (2000). *Investigations on Discriminative Training*, PhD thesis, Aachen University, Germany.

[111] L. F. Uebel & P. C. Woodland (2001). Discriminative Linear Transforms for Speaker Adaptation. *Proc. ISCA ITR-Workshop on Adaptation Methods in Speech Recognition*, Sophia-Antipolis.

[112] V. Valtchev, J. J. Odell, P. C. Woodland & S. J. Young (1997). MMIE Training of Large Vocabulary Speech Recognition Systems. *Speech Communication*, pp. 303-314, vol.22, 1997.

[113] V. Valtchev (1995). *Discriminative Methods for HMM-based Speech Recognition*, PhD thesis, Cambridge University Engineering Department, England.

[114] A. J. Viterbi (1967). Error Bounds for Convolutional Codes and an Asymptotically Optimal Decoding Algorithm. *IEEE Transactions on Information Theory*, IT-13:260 – 269, 1967.

[115] L. Wang & P. C. Woodland (2003). Discriminative Adaptive Training Using MPE Criterion, *Proc. ASRU'03*, St. Thomas, U. S. Virgin Islands.

[116] L. Wang & P. C. Woodland (2004). MPE-based Discriminative Linear Transform For Speaker Adaptation, *Proc. ICASSP'04*, Montreal, Canada..

[117] S. Watanabe, A. Sako & A. Nakamura (2004). Automatic Determination of Acoustic Model Topology Using Variational Bayesian Estimation and Clustering, *Proc. ICASSP'04*, Montreal, Canada..

[118] S. Watanabe, Y. Minami, A. Nakamura, N. Ueda (2004) Variational Bayesian Estimation and Clustering for Speech Recognition, *IEEE Transactions on Speech and Audio Processing*, pp: 365–381, Vol. 12, 2004.

[119] S. Watanabe, Y. Minami, A. Nakamura & N. Ueda (2002). Application of the Variational Bayesian Approach to Speech Recognition, *Proc. NIPS'02*, 2002.

[120] S. Watanabe, Y. Minami, A. Nakamura & N. Ueda (2002). Bayesian Acoustic Modeling for Spontaneous Speech Recognition, *Proc. SSPR'03*, 2003, Tokyo, Japan.

[121] A. Webb (1999) Statistical Pattern Recognition, Oxford University Press, 1999.

[122] A. H. Welsh (1996). Aspects of Statistical Inference, John Wiley & Sons, Inc., 1996.

[123] I. H. Witten & T. C. Bell (1991). The Zero-frequency Problem: Estimating the Probabilities of Novel Events in Adaptive Text Compression. *IEEE Transactions on Information Theory*, vol. 37, no. 4, pp. 1085 – 1094, 1991.

[124] P. C. Woodland & D. Povey (2002). Large Scale Discriminative Training of Hidden Markov Models for Speech Recognition. *Computer Speech and Language*, vol. 16, pp. 25-47, 2002.

[125] P. C. Woodland, T. Hain, G. L. Moore, T. R. Niesler, D. Povey, A. Tuerk & E. W. D. Whittaker (1999). The1998 HTK Broadcast News Transcription System: Development and Results. *Proc. DARPA Broadcast News Workshop*, pp. 265–270, Morgan Kaufman.

[126] P. C. Woodland, T. Hain, G. Evermann & D. Povey (2001). The CU-HTK March 2001 Hub5 System. *Presentation at 2001 NIST Large Vocabulary Conversational Speech Recognition Workshop*.

[127] P. C. Woodland, G. Evermann, M. J. F. Gales, T. Hain, X. Liu, G. Moore, D. Povey & L. Wang (2002). The CU-HTK April 2002 Switchboard System. *Presentation at 2002 NIST Large Vocabulary Conversational Speech Recognition Workshop*.

[128] P. C. Woodland (2001). Speaker Adaptation for Continuous Density HMMs: A Review. *ISCA ITRW Adaptation Methods for Automatic Speech Recognition*, pp. 11–19, 2001.

[129] P. C. Woodland & D. Povey (2000). Large Scale Discriminative Training for Speech Recognition,*Proc. ASR'00*, 2000.

[130] W. Xu, J. Duchateau, K. Demuynck, I. Dologlou, P. Wambacq, D. Van Compernolle, & H. Vanhammme (1999). Accuracy Versus Complexity in Context Dependent Phone Modeling, *Proc. Eurospeech'99*, pp. 1127–1130, vol. 3, Budapest.

[131] S. J. Young, D. Kershal, J. Odell, D. Olson, V. Valtchev & P. C. Woodland, The HTK Book Version 3.1, 2001.

[132] S. J. Young & P. C. Woodland (1993). The use of State Tying in Continuous Speech Recognition, *Proc. Eurospeech'93*, pp. 2207–2210, 1993.

[133] S. J. Young & P. C. Woodland (1994). Tree-based State Tying for High Accuracy Acoustic Modeling, *ARPA Human Language Age Technology Workshop*, pp. 307–312, Morgan Kaufman, 1994.

[134] B. Zhang & S. Matsoukas (2005). Minimum Phoneme Error Based Heteroscedastic Linear Discriminant Analysis for Speech Recognition, *Proc. ICASSP'05*, Philadelphia, USA.