A Gaussian Mixture Model Spectral Representation for Speech Recognition

Matthew Nicholas Stuttle

Hughes Hall
and
Cambridge University Engineering Department

July 2003

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

Summary

Most modern speech recognition systems use either Mel-frequency cepstral coefficients or perceptual linear prediction as acoustic features. Recently, there has been some interest in alternative speech parameterisations based on using formant features. Formants are the resonant frequencies in the vocal tract which form the characteristic shape of the speech spectrum. However, formants are difficult to reliably and robustly estimate from the speech signal and in some cases may not be clearly present. Rather than estimating the resonant frequencies, formant-like features can be used instead. Formant-like features use the characteristics of the spectral peaks to represent the spectrum.

In this work, novel features are developed based on estimating a Gaussian mixture model (GMM) from the speech spectrum. This approach has previously been used sucessfully as a speech codec. The EM algorithm is used to estimate the parameters of the GMM. The extracted parameters: the means, standard deviations and component weights can be related to the formant locations, bandwidths and magnitudes. As the features directly represent the linear spectrum, it is possibly to apply techniques for vocal tract length normalisation and additive noise compenstation techniques.

Various forms of GMM feature extraction are outlined, including methods to enforce temporal smoothing and a technique to incorporate a prior distribution to constrain the extracted parameters. In addition, techniques to compensate the GMM parameters in noise corrupted environments are presented. Two noise compensation methods are described: one during the front-end extraction stage and the other a model compensation approach.

Experimental results are presented on the Resource Management (RM) and Wall Street Journal (WSJ) corpora. By augmenting the standard MFCC feature vector with the GMM component mean features, reduced error rates on both tasks are achieved. Statistically significant improvements are obtained on the RM task. Results using the noise compensation techniques are presented on the RM task corrupted with additive “operations room” noise from the Noisex database. In addition, the performance of the features using maximum-likelihood linear regression (MLLR) adaptation approaches on the WSJ task is presented.

Keywords

Speech recognition, feature extraction, speech parameters, formants, formant-like features, expectation maximisation, noise compensation, gravity centroids, vocal tract length normalisation, speaker adaptation.

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 material has been presented at international conferences [101] [102].

The length of this thesis, including footnotes and appendices is approximately 49,000 words.

Acknowledgements

First, I would like thank my supervisor Mark Gales for his help and encouragement throughout my time as a PhD student. His expert advice and detailed knowledge of the field was invaluable, and I have learnt much during my time in Cambridge thanks to him. Mark was always available, and I thank him for all the time he gave me.

Thanks must also go to Tony Robinson for help during the initial formulation of ideas, and also to all those who helped during the writing-up stages, particularly Konrad Scheffler and Patrick Gosling.

There are many people who have helped me during the course of my studies. I am also grateful to all of those who made the SVR group a stimulating and interesting atmosphere to work in. There are too many people to acknowledge individually, but I would like to thank both Gunnar Evermann and Nathan Smith for their friendship and help. I am also grateful to Thomas Hain for the useful discussions we have had. This work would also not be possible without the efforts of all those involved with building and maintaining the HTK project. Particular thanks must go to Steve Young, Phil Woodland, Andrew Liu and Lan Wang.

My research and conference trips have been funded by the ESPRC, the Newton Trust, Softsound, the Rex Moir fund and Hughes Hall, and I am very grateful to them all.

I must also thank Amy for the limitless patience and unfailing love she has shown me. Finally, I would also like to thank my family for all of their support and inspiration over the years. Suffice to say, without them, none of this would have been possible.

Table of Notation

The following functions are used in this thesis:

$p(x)$	the probability density function for a continuous variable x
$P(x)$	the discrete probability of event x, the probability mass function
$\mathcal{Q}(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}})$	the auxiliary function for original and reestimated parameters $\boldsymbol{\theta}$ and $\hat{\boldsymbol{\theta}}$
$\mathcal{E}\{x\|g\}$	The expected value of x over g

Vectors and matrices are defined:

$\mathbf{A}$	a matrix of arbitary dimensions
$\mathbf{A}^T$	the transpose of the matrix $\mathbf{A}$
$\|\mathbf{A}\|$	the determinant of the matrix $\mathbf{A}$
$\mathbf{X}$	an arbitary length sequence of vector-valued elements
$\mathbf{X}_N$	a sequence of vectors length N
$\mathbf{x}$	an arbitary length vector
$\mathbf{x}_i$	the i^{th} vector-valued element of a sequence of vectors $\mathbf{X}$
x_j	the j_{th} scalar element of a vector, or sequence of scalars, $\mathbf{x}$

The exception to this notation is for:

$\mathbf{q}_T$	a sequence of HMM states length T
$\mathbf{W}_L$	the sequence of words of length L

Other symbols commonly used are:

$\mathbf{y}(t)$	a general speech observation at time t
$\mathbf{Y}_T$	A sequence of T speech observations
$\Delta\mathbf{y}(t)$	the first-order (velocity) dynamic parameters at time t
$\Delta\Delta\mathbf{y}(t)$	the second-order (acceleration) dynamic parameters at time t
$\mathbf{s}(t) = [s_1(t) \dots s_N(t)]^T$	set of FFT points at time T
$\boldsymbol{\theta}$	a set of Gaussian mixture model parameter values
$\boldsymbol{\theta}^{(n)}$	the set of GMM parameters for the noise model
$\boldsymbol{\theta}^{(c)}$	the noise-compensated GMM parameters
$\boldsymbol{\theta}_m$	a set of parameter values for mixture component m

Acronyms used in this work

ASR	Automatic Speech Recognition
RM corpus	Resource Management corpus
WSJ corpus	Wall Street Journal corpus
HMM	Hidden Markov Model
CDHMM	Continuous Density Hidden Markov Models
ANN	Artificial Neural Net
HMM-2	Hidden Markov Model - 2 system
MFCC	Mel Frequency Cepstral Coefficients
PLP	Perceptual Linear Prediction
GMM	Gaussian Mixture Model
EM	Expectation Maximisation
WER	Word Error Rate
MLLR	Maximum Likelihood Linear Regression
CMLLR	Constrained Maximum Likelihood Linear Regression
SAT	Speaker Adaptive Training
LDA	Linear Discriminant Analysis
FFT	Fast Fourier Transform
CSR	Continuous Speech Recognition
DARPA	Defence Advanced Research Projects Agency
PDF	Probability Density Function
HTK	HMM Tool Kit
CUED HTK	Cambridge University Engineering Department HTK
CRSNAB	Continuous Speech Recognition North American Broadcast news

Contents

List of Figures

List of Tables

1

Introduction

Automatic speech recognition (ASR) attempts to map from a speech signal to the corresponding sequence of words it represents. To perform this, a series of acoustic features are extracted from the speech signal, and then pattern recognition algorithms are used. Thus, the choice of acoustic features is critical for the system performance. If the feature vectors do not represent the underlying content of the speech, the system will perform poorly regardless of the algorithms applied.

This task is not easy and has been the subject of much research over the the past few decades. The task is complex due to the inherent variability of the speech signal. The speech signal varies for a given word both between speakers and for multiple utterances by the same speaker. Accent will differ between speakers. Changes in the physiology of the organs of speech production will produce variability in the speech waveform. For instance, a difference in height or gender will have an impact upon the shape of the spectral envelope produced. The speech signal will also vary considerably according to emphasis or stress on words. Environmental or recording differences also change the signal. Although humans listeners can cope well with these variations, the performance of state of the art ASR systems is still below that achieved by humans.

As the performance of ASR systems has advanced, the domains to which they have been applied has expanded. The first speech recognition systems were based on isolated word or letter recognition on very limited vocabularies of up to ten symbols and were typically speaker dependent. The next step was to develop medium vocabulary systems for continuous speech, such as the Resource Management (RM) task, with a vocabulary of approximately a thousand words [91]. Next, large vocabulary systems on read or broadcast speech with an unlimited scope were considered. Recognition systems on these tasks would use large vocabularies of up to 65,000 words, although it is not possible to guarantee that all observed words will be in the vocabularly. An example of a full vocabulary task would be the Wall Street Journal task (WSJ) where passages were read from the Wall Street Journal [87]. Current state of the art systems have been applied to to recognising conversational or spontaneous speech in noisy and limited bandwidth domains. An example of such a task would be the SwitchBoard corpus [42].

The most common approach to the problem of classifying speech signals is the use of *hidden*

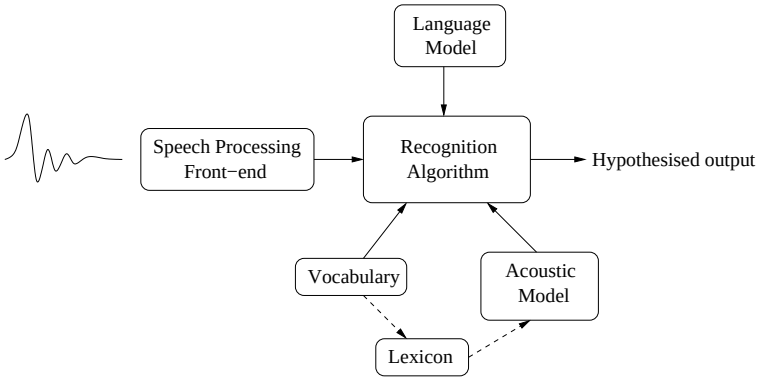


Figure 1.1 *General speech recognition system*

Markov models (HMMs). Originally adapted for the task of speech recognition in the early 1970s by researchers at CMU and IBM [64], HMMs have become the most popular models for speech recognition. One advantage of using HMMs is that they are a statistical approach to pattern recognition. This allows a number of techniques for adapting and extending the models. Furthermore, efficient recognition algorithms have been developed. One of the most popular alternative approaches to acoustic modelling used in ASR is the combination of an artificial neural net (ANN) with a HMM to form a hybrid HMM-ANN system [93] [9]. However, this thesis will only consider the use of HMM based speech recognition systems.

1.1 Speech recognition systems

Statistical pattern recognition is the current paradigm for automatic speech recognition. If a statistical model is to be used, the goal is to find the most likely word sequence $\hat{\mathbf{W}}$, given a series of T acoustic vectors, $\mathbf{Y}_T = \{\mathbf{y}(1), \ldots, \mathbf{y}(T)\}$

$$\hat{\mathbf{W}} = \arg\max_{\mathbf{W}} P(\mathbf{W}|\mathbf{Y}_T) \tag{1.1}$$

Applying Bayes rule to the above equation yields

$$\begin{aligned} \hat{\mathbf{W}} &= \arg\max_{\mathbf{W}} \left[\frac{P(\mathbf{W})p(\mathbf{Y}_T|\mathbf{W})}{p(\mathbf{Y}_T)}\right] &(1.2)\\ &= \arg\max_{\mathbf{W}} \left[P(\mathbf{W})p(\mathbf{Y}_T|\mathbf{W})\right] &(1.3)\end{aligned}$$

where the most likely word sequence is invariant of the likelihood of the acoustic vectors $p(\mathbf{Y}_T)$. The search for the optimal word sequence comprises two distributions: the likelihood of the acoustic vectors given a word sequence $p(\mathbf{Y}_T|\mathbf{W})$, generated by the *acoustic model* and the likelihood of a given string of words $P(\mathbf{W})$ given by the *language model*. An overview of a speech recognition system is given in figure 1.1.

In most systems, there is insufficient data to estimate statistical models for each word. Instead, the acoustic models are formed of sub-word units such as phones. To map from the

sub-word units to the word sequences, a *lexicon* is required. The language model represents the syntactic and semantic content of the speech, and the lexicon and acoustic model handle the relationship between the words and the feature vectors.

1.2 Speech parameterisation

In order to find the most likely word sequence, equation 1.3 requires a set of acoustic vectors $\mathbf{Y}_T$. Recognising speech using a HMM requires that the speech be broken into a sequence of time-discrete vectors. The assumption is made that the speech is quasi-stationary, that is, it is reasonably stationary over short (approximately 10ms) segments.

The goal of the feature vector is to represent the underlying phonetic content of the speech. The features should ideally be compact, distinct and well represented by the acoustic model. State of the art ASR systems use features based on the short term Fourier transform (SFT) of the speech waveform. Taking the SFT yields a frequency spectrum for each of the sample periods. These features model the general shape of the spectral envelope, and attempt to replicate some of the psycho-acoustic properties of the human auditory system. The two most commonly used parameterisations of speech are Mel-frequency cepstral coefficients (MFCCs) and perceptual linear prediction (PLP) features. There have been a number of studies examining useful features for speech recognition, to replace or augment the standard MFCC features. Such alternative features include formants [114], phase spectral information [97], pitch information [28], and features based on the speech articulators [27].

When examing spectral features, it is worth considering models of the speech production mechanism to evaluate the properties of the signal. One such example would be the *source-filter* model. In the source-filter model of speech production, the speech signal can be split into two parts. The *source* is the excitation signal from the vocal folds in the case of voiced speech, or noisy turbulence for unvoiced sounds. The *filter* is the frequency response of the vocal tract organs. By moving the articulators and changing the shape of the vocal tract, different resonances can be formed. Thus, the shape of the spectral envelope is changed. The resonances in the frequency response of the filter are known as *formants*. In English, the form of the excitation is not considered informative as to the phonetic class of the sound, except to distinguish different intensities of sounds [15].

The formants or resonances in the vocal tract are also known to be important in human recognition of speech [61]. This has motivated the belief that formants or formant-like features might be useful in ASR systems, especially in situations where the bandwidth is limited or in noisy environments. In the presence of background noise, it is hoped that the spectral peaks will sit above the background noise and therefore be less corrupted than standard spectral parameterisations.

There has been much work in developing schemes to estimate the formant frequencies from the speech signal. Estimating the formant frequencies is not simple. The formants may be poorly defined in some types of speech sound or may be completely absent in others. The labelling of

formants can also be ambiguous, and the distinction between whether to label a peak with a single wide formant or two seperate formants close together is sometimes not clear. Recently, some research has been focused on using statistical techniques to model the spectrum in terms of its peak structure rather than searching for the resonances in the speech signal. For example, approaches parameterising spectral sub-bands in terms of the first and second order moments, (also known as gravity centroids) have provided features complementary to MFCCs on small tasks [84] [16].

This work develops a novel statistical method of speech parameterisation for speech recognition. The feature vector is derived from the parameters of a Gaussian mixture model (GMM) representation of the smoothed spectral envelope. The parameters extracted from the GMM, the means, variances and component mixture weights represent the peak-like nature of the speech spectrum, and can be seen to be analogous to a set of formant-like features [125]. Techniques for estimating the parameters from the speech are presented, and the performance of the GMM features is examined. Approaches to combine the GMM features with standard MFCC and PLP parameterisations are also considered. In addition, the performance of the features in noise corrupted environments is studied, and techniques for compensating the GMM features are developed.

1.3 Organisation of thesis

This thesis is structured as follows: the next chapter gives a basic review of the theory of HMMs and their use as acoustic models. The theory of training and decoding sequences with HMMs is detailed, as well as how they are extended and utilised in ASR. The fundamental methods of speaker adaptation and noise compensation are also outlined.

Chapter 3 presents a review of methods for parameterising the speech spectrum. The most popular speech features, namely PLPs and MFCCs, are described and their relative merits discussed. Alternative parameterisations are also described, with particular emphasis placed on formant and spectral-peak features. Possible options of combining different speech parameterisations are also presented.

In chapter 4, the theory of extraction and use of the GMM features is presented. Issues in extracting the parameters and extensions to the framework are shown. A method previously proposed for combining formant features with MFCCs using a confidence metric is adapted for the GMM features, and extended to the case of a medium or large vocabulary task. Two techniques to compensate the GMM features in the presence of additive noise are described: one at the front-end level, the other a model-compensation approach.

Experimental results using the GMM features are presented in chapters 5, 6, 7 and 8. Chapter 5 presents results using the GMM features on a medium-vocabulary task. Chapter 6 details work using the GMM features in combination with an MFCC parameterisation on medium and large vocabulary tasks. Results using the GMM features in the presence of additive noise are described in chapter 7, and the performance of the compensation techniques described in chapter 4 are

presented. Finally, the GMM features are tested using MLLR speaker adaptation approaches on the large vocabulary Wall Street Journal corpus in chapter 8.

The final chapter summarises the work contained in this thesis and discusses potential future directions for research.

2

Hidden Markov models for speech recognition

In this chapter the basic theory of using Hidden Markov models for speech recognition will be outlined. The algorithms for training these models are shown, together with the algorithms for pattern recognition. In addition, techniques used in state of the art systems to improve the speech models in noise-corrupted environments are discussed. Finally, methods for speaker adaptation using maximum likelihood linear regression (MLLR) are covered, along with front-end feature transforms.

2.1 Framework of hidden Markov models

Hidden Markov models are generative models based on stochastic finite state networks. They are currently the most popular and successful acoustic models for automatic speech recognition. Hidden Markov models are used as the acoustic model in speech recognition as mentioned in section 1.1. The acoustic model provides the likelihood of a set of acoustic vectors given a word sequence. Alternative forms of an acoustic model or extensions to the HMM framework are an active research topic [100] [95], but are not considered in this work.

Markov models are stochastic state machines with a finite set of N states. Given a pointer to the active state at time t the selection of the next state has a constant probability distribution. Thus the sequence of states is a stationary stochastic process. An n^{th} order Markov assumption is that the likelihood of entering a given state depends on the occupancy in the previous n states. In speech recognition a 1^{st} order Markov assumption is usually used. The probability of the state sequence $\mathbf{q}_T = (q_1, \ldots q_T)$ is given by:

$$P(\mathbf{q}_T) \quad = \quad P(q_1) \prod_{t=2}^{T} P(q_t | q_1, \ldots, q_{t-1})$$

and using the first-order Markov assumption this is approximated by:

$$P(\mathbf{q}_T) \quad \simeq \quad P(q_1) \prod_{t=2}^{T} P(q_t | q_{t-1}) \tag{2.1}$$

The observation sequence is given as a series of points in vector space $\mathbf{Y}_T = \{\mathbf{y}(1), \ldots, \mathbf{y}(T)\}$ or alternatively as a series of discrete symbols. Markov processes are generative models and each state has associated with it a probability distribution for the points in the observation space. The extension to "hidden" Markov models is that the state sequence is hidden, and becomes an underlying unobservable stochastic process. The state sequence can only be observed through the stochastic processes of the vectors emitted by the state output probability distributions. Thus the probability of an observation sequence can be described by:

$$p(\mathbf{Y}_T) = \sum_{\mathbf{Q_T}} p(\mathbf{Y}|\mathbf{q}_T)P(\mathbf{q}_T) \tag{2.2}$$

where the sum $\sum_{\mathbf{Q_T}}$ is over all possible state sequences $\mathbf{q}_T$ through the model and the probability of a set of observed vectors, $p(\mathbf{Y}_T|\mathbf{q})$, can be defined by:

$$p(\mathbf{Y}_T|\mathbf{q}_T) = \prod_{t=1}^{T} p(\mathbf{y}(t)|q_t) \tag{2.3}$$

Using a HMM to model a signal makes several assumptions about the nature of the signal. One is that the likelihood of an observed symbol is independent of preceding symbols (the *independence assumption*) and depends only on the current state q_t. Another assumption is that the signal can be split into stationary regions, with instantaneous transitions in the signal between these regions. Neither assumption is true for speech signals, and extensions have been proposed to the HMM framework to account for these [124] [82], but are not considered in this thesis.

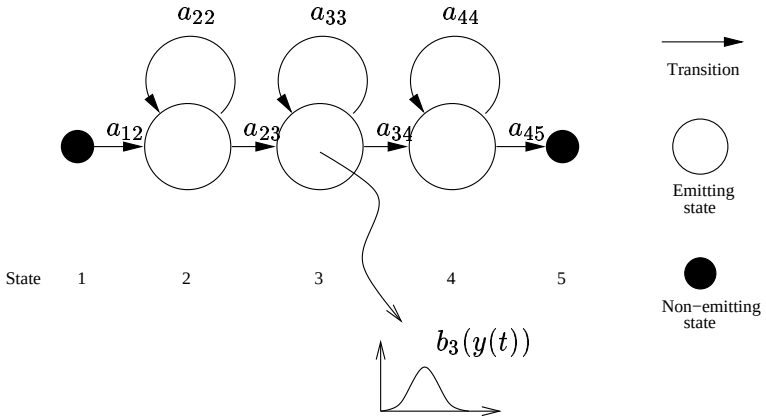


Figure 2.1 *3 state HMM having a left-to-right topology with beginning and end non-emitting states*

Figure 2.1 shows the topology of a typical HMM used in speech recognition. Transitions may only be made to the current state or the next state, in a left-to right fashion. In common with the standard HMM toolkit (HTK) terminology conventions, the topology includes non-emitting states for the first and last states. These non-emitting states are used to make the concatenation of basic units simpler.

The form of HMMs can be described by the set of parameters which defines them:

- **States** HMMs consist of N states in a model; the pointer $(q_t = i)$ indicates being in state i at time t.

- **Transitions** The transition matrix **A** gives the probabilities of traversing from one state to another over a time step

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

  The form of the matrix can be constrained such that certain state transitions are not permissible, as shown in figure 2.1. Additionally, the transition matrix has the constraint that

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

  and

$$a_{ij} \geq 0 \tag{2.6}$$

- **State Emissions** Each emitting state has associated with it a probability density function $b_j(\mathbf{y}(t))$; the probability of emitting a given feature vector if in state j at time t:

$$b_j(\mathbf{y}(t)) = p(\mathbf{y}(t) | q_t = j))) \tag{2.7}$$

An initial state distribution is also required. In common with the standard HTK conventions, the state sequence is constrained to begin and end in the first and last states, with the models begin concatentated together by the non-emitting states.

2.1.1 Output probability distributions

The output distributions used for the state probability functions (state emissions PDFs) may assume a number of forms. Neural nets may be used to provide the output probabilities in the approach used by hybrid/connectionist systems [9]. If the input data is discrete, or the data has been vector quantised, then discrete output distributions are used. However, in speech recognition systems continuous features are most commonly used, and are modelled with continuous density output probability functions.

If the output distributions are continuous density probability functions in the case of continuous density HMMs (CDHMMs), then they are typically described by a mixture of Gaussians function [76]. If a mixture of Gaussians is used, the emission probability of the feature vector $\mathbf{y}(t)$ in state j is given by

$$b_j\left(\mathbf{y}(t)\right) = \sum_{m=1}^{M} P(\omega_{jm}) \mathcal{N}\left(\mathbf{y}(t); \boldsymbol{\mu}_{jm}, \boldsymbol{\Sigma}_{jm}\right) \tag{2.8}$$

where the number of components in the mixture model is M, and the means, covariance matrices and mixture weights of each component are $\boldsymbol{\mu}_{jm}$, $\boldsymbol{\Sigma}_{jm}$ and $P(\omega_{jm})$ respectively. The mixture of Gaussians has several useful properties as a distribution model: training schemes exist for it in the HMM framework and the use of multiple mixture components allows for the modelling of more abstract distributions.

The covariance matrices for the Gaussian components can also take a number of different forms, using identity, diagonal, block diagonal or full covariance forms. The more complex the form of covariance modelled, the larger the number parameters to estimate for each component.

If the features are correlated, rather than estimating full covariance matrices a larger number of mixture components can be used in the model. As well as being able to approximately model correlations in the data set distributions, using multiple components can also approximate multimodal or arbitrary distributions.

Other work has studied the use of alternative distributions, such as the Richter or Laplace distributions in the emission probability functions [37] [2]. Rather than using a sum of mixture components, the use of a product of Gaussians has also been investigated [1]. Another approach is to use semi-continuous HMMs where the set of mixture components has been tied over the set of all states, but the component weights are state-specific [60]. However, in this work, GMMs are used to model the output PDFs in the HMMs.

2.1.2 Recognition using hidden Markov models

The requirement of an acoustic model in a speech recognition system is to find the probability of the observed data $\mathbf{Y}_T$ given a hypothesised set of word models or units $\mathbf{W}$. The word string is mapped to the relevant set of HMM models $\mathcal{M}$ and thus the search is over $p(\mathbf{Y}_T|\mathcal{M})$. As the emission probabilities are given by continuous probability density functions, the goal of the search is to maximise the likelihood of the data given the model set.

The probability for a given state sequence $\mathbf{q}_T = \{q_0, \cdots, q_T\}$ and observations $\mathbf{Y}_T$ is given by the product of the transition and output probabilities:

$$p(\mathbf{Y}_T, \mathbf{q}_T) = a_{q_1,q_2} \prod_{t=2}^{T} b_{q_t}(\mathbf{y}(t)) a_{q_{t-1}q_t} \tag{2.9}$$

The total likelihood is given by the sum of all possible state sequences (or paths) in the given model that end at the appropriate state. Hence the likelihood of the observation sequence ending in the final state N is given by:

$$p(\mathbf{Y}_T|\mathcal{M}) = \sum_{\mathbf{q}_T \in \mathbf{Q}} a_{q_T N} \prod_{t=2}^{T} a_{q_{t-1}q_t} b_{q_t}(\mathbf{y}(t)) \tag{2.10}$$

where $\mathbf{Q}$ is the set of all possible state sequences, $\mathcal{M}$ is the model set and q_t the state occupied at time t in path $\mathbf{q}_T$.

2.1.3 Forward-backward algorithm

The forward-backward algorithm is a technique for efficiently calculating the likelihood of generating an observation sequence given a set of models. As mentioned previously, the independence assumption states that the probability of a given observation depends only on the current state and not on any of the previous state sequence. Two probabilities are introduced: the forward probability and the backward probability. The forward probability is the probability of a given model producing an observation sequence $\mathbf{Y}_t = \{\mathbf{y}(1), \cdots, \mathbf{y}(t)\}$ and being in state j at time t:

$$\begin{aligned} \alpha_j(t) &= p(\mathbf{y}(1), \mathbf{y}(2), \cdots, \mathbf{y}(t), q_t = j|\mathcal{M}) \\ &= \left[\sum_{i=1}^{N} \alpha_i(t-1)a_{ij}\right] b_j(\mathbf{y}(t)) [\text{ for } (1 < t < T) \text{ and } (2 < j < N-1)] \end{aligned} \tag{2.11}$$

The initial conditions for the forward probability for a HMM are given by:

$$\alpha_1(0) = 1 \tag{2.12}$$

$$\alpha_j(0) = 0 \text{ if } j \neq 1 \tag{2.13}$$

and the termination is given by:

$$\alpha_N(T) = \sum_{i=2}^{N-1} \alpha_i(T)a_{iN} \tag{2.14}$$

The backward probability is defined by:

$$\begin{aligned} \beta_i(t) &= p(\mathbf{y}(t+1), \mathbf{y}(t+2), \cdots, \mathbf{y}(T)|q_t = 0, \mathcal{M}) \\ &= \sum_{j=1}^{N-1} a_{ij} b_j(\mathbf{y}_{t+1}) \beta_j(t+1) \end{aligned} \tag{2.15}$$

with initial and terminating conditions:

$$\beta_j(T) = a_{jN} \text{ for } 1 < j < N \tag{2.16}$$

$$\beta_N(t) = 0 \tag{2.17}$$

Thus, the likelihood of a given observation sequence can be given by:

$$p(\mathbf{Y}_T|\mathcal{M}) = \alpha_N(T) = \beta_1(0) = \sum_{j=1}^{N} \alpha_j(t)\beta_j(t) \tag{2.18}$$

Additionally, it is possible to calculate the probability of being in state i at time t by:

$$L_i(t) = \frac{\alpha_i(t)\beta_i(t)}{p(\mathbf{Y}_T|\mathcal{M})} \tag{2.19}$$

Hence, the forward-backward algorithm yields an efficient method for calculating the frame/state alignments required for the training of HMM model parameters using the EM algorithm.

2.1.4 Parameter estimation

The HMM model sets have been characterised by two sets of model parameters: the transition probabilites a_{ij} and the emission probabilities $b_j(\mathbf{y}(t))$. If Gaussian mixture models are to be used for the distributions then the second set of parameters comprises the state and mixture means $\boldsymbol{\mu}_{jm}$, covariances $\boldsymbol{\Sigma}_{jm}$ and mixture weights $P(\omega_{jm})$.

The objective of training the HMMs is to estimate a set of parameters which matches the training data well, according to a training criterion. The most commonly used optimisation criterion is the *Maximum Likelihood* (ML) function [4]. This is the training criterion used for the HMMs throughout this work.

Other criteria have also been successfully implemented to train HMMs for use in speech recognition algorithms. *Maximum Mutual Information* (MMI) training not only maximises the likelihood of the correct model, but also minimises the likelihood of "wrong" sequences with an optimisation function [3] [90]. Schemes which take the competing classes into account whilst training a class are known as discriminative schemes. Another alternative is a Bayesian technique, *Maximum a-posteriori* estimation [41]. The MAP approach assumes that the estimated parameters are themselves random variables with an associated prior distribution. The parameter vector is selected by the maximum of the posterior distribution. If the prior is uniform over all parameters the MAP solution is identical to the ML solution. The main issue with MAP training is the problem of obtaining meaningful priors.

The ML estimator is often chosen in preference to these schemes due to its relative simplicity, low computational complexity and wide range of algorithmic solutions and techiques. The aim of maximum likelihood training schemes is to maximise the likelihood of the training data given the model, i.e. maximise the function $\mathcal{F}_{mle}$:

$$\mathcal{F}_{mle}(\mathcal{M}) = p(\mathbf{Y}_T|\mathcal{M}) \tag{2.20}$$

Unfortunately, there exists no closed form solution for the optimisation of the function above for HMMs. There does exists a general iterative training scheme, the *Baum-Welch* algorithm. The Baum-Welch algorithm is an iterative approach to estimating the HMM parameters which is guaranteed not to decrease the objective function $\mathcal{F}_{mle}$ at each step [5]:

$$\mathcal{F}_{mle}(\hat{\mathcal{M}}) \geq \mathcal{F}_{mle}(\mathcal{M}) \tag{2.21}$$

where $\hat{\mathcal{M}}$ is the new estimate of the model parameters. The Baum-Welch training scheme maximises the auxiliary function, $\mathcal{Q}(\mathcal{M}, \hat{\mathcal{M}})$ of the current model set $\mathcal{M}$ and re-estimated set $\hat{\mathcal{M}}$ at each step:

$$\mathcal{Q}(\mathcal{M}, \hat{\mathcal{M}}) = \sum_{\mathbf{q}_T \in \mathbf{Q}} p(\mathbf{Y}_T, \mathbf{q}_T|\mathcal{M}) \log\left(p(\mathbf{Y}_T, \mathbf{q}_T|\hat{\mathcal{M}})\right) \tag{2.22}$$

Unlike the ML function, there is a closed form solution to optimise the auxiliary function with respect to the model parameters. The increase in the auxiliary function can be shown to be a lower bound on the increase in log-likelihood of the training data [5]. The algorithm estimates

the *complete* set of data $\{\mathbf{y}(1), \ldots, \mathbf{y}(T), \mathbf{L}(1), \ldots, \mathbf{L}(T)\}$, where $\mathbf{L}(\tau)$ is the matrix of frame/state alignment probabilities $L_{jm}(\tau)$. The probability $L_{jm}(\tau)$ is defined as the probability of being in state j and mixture m at time τ.

Once the complete dataset has been estimated, it is simple to obtain the new model parameters $\hat{\mathcal{M}}$ which maximise the auxiliary function. The estimation of the alignments and maximisation of the auxiliary function can then be iteratively repeated. Each iteration is guaranteed not to decrease the objective function.

The frame/state alignment and frame/state component alignments are given by:

$$\begin{aligned} L_{jm}(\tau) &= P(q_{jm}(\tau)|\mathbf{Y}_T, \mathcal{M}) && (2.23)\\ &= \frac{1}{p(\mathbf{Y}_T, \boldsymbol{\theta}|\mathcal{M})} U_j(\tau) P(\omega_{jm}) b_{jm}(\mathbf{y}(\tau)) \beta_j(\tau) && (2.24)\\ L_j(\tau) &= P(q_\tau = j|\mathbf{Y}_T, \mathcal{M}) && (2.25) \end{aligned}$$

where $q_{jm}(\tau)$ indicates being in state t and component m at time τ and

$$U_j(\tau) = \begin{cases} a_{1j} & \text{if } \tau = 1 \\ \sum_{i=2}^{N-1} \alpha_i(\tau - 1) a_{ij}, & \text{(otherwise)} \end{cases} \tag{2.26}$$

Using the auxiliary function, the estimates of the updated means, variances and mixture weights are given by:

$$\hat{\boldsymbol{\mu}}_{jm} = \frac{\sum_{\tau=1}^{T} L_{jm}(\tau) \mathbf{y}(\tau)}{\sum_{\tau=1}^{T} L_{jm}(\tau)} \tag{2.27}$$

$$\hat{\boldsymbol{\Sigma}}_{jm} = \frac{\sum_{\tau=1}^{T} L_{jm}(\tau) (\mathbf{y}(\tau) - \hat{\boldsymbol{\mu}}_{jm})(\mathbf{y}(\tau) - \hat{\boldsymbol{\mu}}_{jm})^T}{\sum_{\tau=1}^{T} L_{jm}(\tau)} \tag{2.28}$$

$$\hat{c}_{jm} = \frac{\sum_{\tau=1}^{T} L_{jm}(\tau)}{\sum_{\tau=1}^{T} L_j(\tau)} \tag{2.29}$$

The transition probabilities for $(1 \leq i < N)$ and $(1 \leq j < N)$ are given by:

$$\hat{a}_{ij} = \frac{\sum_{\tau=1}^{T-1} \alpha_i(\tau) a_{ij} b_j(\mathbf{y}(\tau + 1)) \beta_j(\tau + 1)}{\sum_{\tau=1}^{T-1} \alpha_i(\tau) \beta_i(\tau)} \tag{2.30}$$

and probability of the exits to and from the non-emitting states are given by:

$$\hat{a}_{1j} = \frac{1}{p(\mathbf{Y}_T|\mathcal{M})} \alpha_j(1) \beta_j(1) \tag{2.31}$$

$$\hat{a}_{iN} = \frac{\alpha_i(T) \beta_i(T)}{\sum_{\tau=1}^{T} \alpha_i(\tau) \beta_i(\tau)} \tag{2.32}$$

The Baum-Welch algorithm thus provides a method for iteratively updating the model parameters of a HMM. The HMM must still have a set of initial parameters prior to performing the Baum-Welch training. This issue will be dealt with for HMMs based on speech in section 2.2.3. The next section presents a technique for estimating the frame/state alignment, $L_j(t)$.

2.2 HMMs as acoustic models

As mentioned previously, there are several fundamental assumptions in the use of HMMs for speech recognition which are not valid for speech signals. One assumption is that the speech input can be broken up into a series of stationary segments or states, with instantaneous transitions between states. This is not true due to the smooth transitions between speech sounds caused by the movement of the speech articulators. Another is the independence assumption, which states that the emission probabilities are dependent only on the current feature vector, and not on any previous features. Neither assumption is correct for speech signals, and a number of extensions to the speech recognition framework have been proposed to correct these. Variable frame rate analysis can be used to compensate for the non-stationary behaviour of speech, in particular the effects of different speaking rates on the signal. [124]. The independence assumption has been addressed by the application of segment models which partially deal with the correlations between successive symbols [82]. However, even though the assumptions made in the model may not be valid, HMMs still form the basis for the most successful current speech recognition systems.

2.2.1 Speech input for HMM systems

Implementing a HMM for speech recognition makes the assumption that the features can be broken up into a series of quasi-stationary discrete segments. The segments are treated independently and in isolation. The frame rate must be sufficiently large such that the speech is roughly stationary over any given frame. Speech features are usually based upon the short-term Fourier transform of the input speech. For full bandwidth data, such as that of the RM or WSJ tasks, the speech will have been sampled at a rate of 16kHz. This gives the speech spectrum a bandwidth of 0-8kHz. For applications such as a telephone-based systems, the speech is sampled at a rate of 8kHz, giving a bandwidth of 0-4kHz. However, the bandwidth of the speech will have been limited to an effective range of 125-3800Hz by the telephony system.

Figure 2.2 shows the process of extracting overlapping windows of speech segments in order to form the feature vectors. Usually, the frames are extracted at a uniform time step. Some work has investigated the use of variable-frame rate analysis [124]. Most systems, however, use a fixed frame rate. A typical system would take frames of speech 25ms long every 10ms [122]. The process of extracting features from the speech frames is discussed in more detail in chapter 3.

The independance assumption that HMMs use is not applicable for speech since observation frames are dependent to some degree on the preceding observations due to the fixed trajectories of the articulators generating the signal [58]. Hence, it is desirable to incorporate some measure of the trajectories of the signal or of the correlations between frames. The simplest method to do this without changing the structure of the HMMs is to include dynamic coefficients into the feature vector [115] [29]. The dynamic coefficients, or delta parameters $\Delta\mathbf{y}(\tau)$ can be

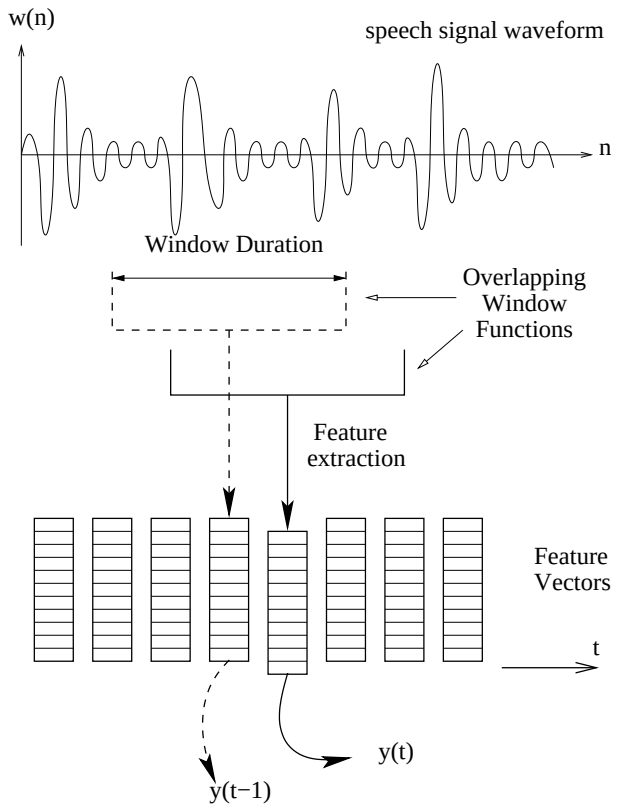


Figure 2.2 *Extraction of input vector frames by use of overlapping window functions on speech signal*

calculated as:

$$\Delta \mathbf{y}(t) = \frac{\sum_{\tau=d}^{D} \tau(\mathbf{y}(t+\tau) - \mathbf{y}(t-\tau))}{2\sum_{\tau=d}^{D} \tau^2} \tag{2.33}$$

Linear regression delta parameters are calculated if d=1. If the start and end frames distances are equal, i.e. d=D, simple difference parameters are calculated as the regression is taken over only a single time-step. By taking the dynamic coefficients again over the resulting delta coefficients, acceleration, or Δ^2 parameters are obtained.

2.2.2 Recognition units

For very small vocabulary recognition tasks, it would be possible to build a HMM model for each word. However, this presents problems of identifying adequate HMM topologies and establishing the optimal number of states for each word. In addition, with a medium or large vocabulary there will be insufficient data to robustly estimate parameters for each whole word model. The most commonly used approach is to split words up into smaller subword units, such as syllables or phones [121] [122]. A pronunciation dictionary or lexicon is used to map from the words to a sequence of sub-word units. Word-based HMM models are formed by concatentating the subword models together. Thus all examples of a given subword unit in the training data will be *tied* together, and share the same distribution parameters [123].

Phones are elementary sound units and represent the abstract notion of a sound as opposed to a particular realisation of it. Models based on phonemes are referred to as phone models. The use of the full set of phones without taking context into account is referred to as a *monophone* model set. However, the distributions of the acoustic features will change given the preceding and following phones. These effects of *coarticulation* are due to the finite trajectories of the speech articulators. To model these variations, *context dependent* models can be built. In a context model set, phone models are tied together depending on the preceding and/or following phones. For example, a *triphone* model ties together all occurances of a phone unit with the same preceding and following phone context. It is possible to build up larger contexts using an arbitarily large number of phones (e.g. for quinphone units [118]) either side of the current phone, but only triphones are considered in this work.

The full set of all possible triphones will be too large for there to be sufficient data to train each robustly in most systems. Furthermore, there will be some examples of triphones that will not be present in the training data. To obtain good estimates of model parameters it is necessary to share or tie the parameters over the full set of triphones. The most common approach is to tie parameters at the HMM state level, such that certain states will share the same model parameters. One method would be to cluster the states using a data-driven approach in a bottom-up fashion to merge triphone models which are acoustically similar until a threshold is reached. The problem with this approach is that it will be unreliable for contexts for which there is little training data and it cannot handle contexts with no training data.

The solution to the problem of state clustering with unseen contexts is to use a phonetic decision tree approach instead. 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 [123]. Figure 2.3 shows an example section of a context decision tree for triphone models. 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 maximise the likelihood of the training data whilst ensuring that each clustered state also has a minimum amount of data observed. The disadvantages of the decision tree clustering are that the cluster splits are only the local maximisation, and not all questions that could split the state clusters are considered [122].

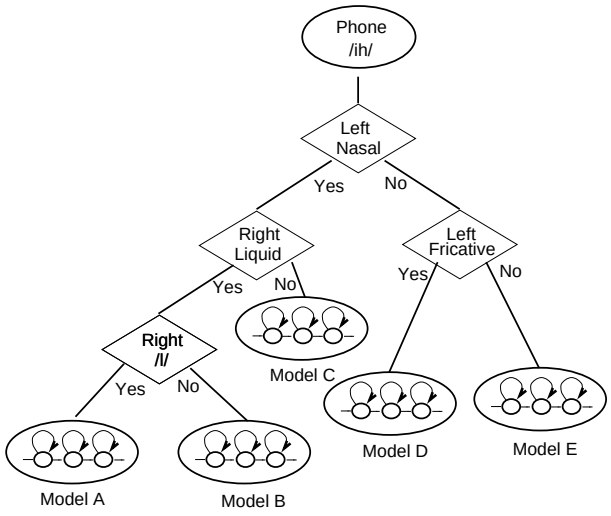


Figure 2.3 *Example of a context dependency tree for a triphone model (from [123])*

2.2.3 Training

The theory of ML parameter estimation for a HMM system has been outlined in section 2.1.4. However, the implementation of HMMs as acoustic models in speech recognition presents some additional issues. The EM algorithm is sensitive to the initialisation of the parameters. The optimisation function will have many different local maxima which may be found depending on the initial conditions. Initial parameters can be chosen in a number of ways. An existing segmentation of the data can be used for the state/model alignment if present. Alternatively, the models can also be *flat started* using identical models for each subword unit. Another option is to use an existing model set from another task to initialise the system. Following the initialisation, further iterations of the Baum-Welch training algorithm are required.

Using multiple component Gaussian mixture models in the emission PDFs requires both a frame/state alignment and a frame/component alignment. The complexity of the training steps will be increased and the search for the maximum likelihood of the training data will be more

complex. One approach is iterative mixture splitting (or *mixing up* [122]) of the components in the state emission PDFs. Mixing up progressively increases the number of components in the system during training. The component with the highest prior in the model is split and the means of the resulting components perturbed. Several iterations of the EM parameter estimation algorithm are then used after each increase in the number of components per state.

In typical system training, the initial model set is a monophone system. The set of monophone states are split into all possible triphones, and are then clustered using a decision tree. The number of components in the state emission PDFs are then gradually increased. Alternatively, if the models are trained from an existing multiple component triphone system, it may be desirable to repeat some or all of the training steps. Reclustering the triphone classes or repeating the mixing-up procedure may yield improvements to the system if there is a mismatch between the initialisation and the target system.

One system for rapidly training a model set on a new set of data given an existing parameterisation and model is single pass retraining (SPR) [122]. In SPR an existing model and training parameterisation is used to retrain a system on a second parameterisation. The first system is used to calculate the state/model and state/component alignments in equations 2.24 and 2.25. These alignments are then used in the parameter estimation calcuations of section 2.1.4 using the data from the second parameterisation. This yields a model set with the same set of states but updated means and variances for the second parameterisations. The component weights and transition matrices will be the same as those calculated if the first set of data was used to re-estimate the first model set. Single pass retraining requires that the two sets of training data be of identical length. The number of components and the mixture weights may not be optimal for the second model set. In addition, the alignment found by the first model set may not be ideal for the second model set. Hence, sometimes further training iterations are performed on the new model set.

2.2.4 Language models

In section 1.1 the search for the optimal word string was expressed as the maximisation of the product of two expressions. The first, the likelihood of the data given a word sequence was obtained from the acoustic model which is given by the HMM as detailed above. The second is the probability of the given word sequence, which is obtained from the language model. This section gives an outline of the language modelling problem. A more detailed description can be found in a review of the field [26].

Stochastic language models associate probabilities with given word strings. For a word sequence $\mathbf{W}_L = \{W_1, \ldots, W_L\}$ the probability of a given word sequence can be calculated by taking the product of the conditional probabilities of the words at each position l given their

histories $\mathbf{W}_{l-1}$.

$$P(\mathbf{W}_L) = P(W_1)P(W_2|W_1)\dots P(W_L|\mathbf{W}_{L-1}) \quad (2.34)$$

$$= \prod_{l=1}^{L} P(W_l|W_{l-1}, W_{l-2}, \dots, W_1) \quad (2.35)$$

However, for large vocabulary systems and systems with longer sentence structures, it is not possible to calculate or store estimates for word sequencies of arbitary length. Instead, the set of all possible word sequences can be clustered into equivalence classes to reduce the parameter space. The most simple form of this clustering is to truncate the word history after a fixed number of words. The assumption is made that the current word is only dependent on the previous N-1 words in the history:

$$P(\mathbf{W}_L) \approx \prod_{l=1}^{L} P(W_l|W_{l-1}, \dots, W_{l-N-1}) \quad (2.36)$$

For example, a trigram model can be build where the set of equivalence history classes is the set of all possible word-pairs. The estimates of probabilities are then:

$$P(\mathbf{W}_l) = \frac{N(W_l, W_{l-1}, W_{l-2})}{\sum_{\mathbf{W}} N(W_l, W_{l-1}, W_{l-2})} \quad (2.37)$$

Unigram models can be estimated from reference training documents or data. However, if a trigram model is to be built given a 60,000 word vocabulary, there are approximately 2.16×10^{14} different word triplets, and hence it is not possible to estimate, or even observe, all the possible triplets in a set of language data. To compensate for the data sparsity, it is possible to smooth the distribution of the word sequences [70]. The data can be *discounted* and all unseen events are given a small proportion of the overall probability mass. Another approach is to combine different length language models, interpolating the probabilities by using weighting functions.

An alternative strategy is not to consider the word sequence probabilities, but to use the language model to limit the set of permissable words which may follow the current word. Effectively, the language model forms a simplified bigram approach, and is referred to as a *word-pair* grammar.

One problem with the use of stochastic language models is that there is a considerable mismatch between the dynamic ranges of the language and acoustic models. The acoustic model and the language model are two separate information sources which are combined by the recognition system. The mismatch is due to the different training sets and ability to generate robust estimates of likelihoods or probabilities for each. The most commonly used solution is to scale the log-likelihood of the language model, usually by a constant factor for a given task. Another modification to the language model scoring is the use of a word insertion penalty. Hence the search for the optimum word sequence is over:

$$\hat{\mathbf{W}} = \arg\max_{\mathbf{W}} [\alpha \log P(\mathbf{W}) + \log p(\mathbf{Y}_T|\mathbf{W}) - \varphi N_{\mathbf{W}}] \quad (2.38)$$

where α is the language model scale factor, φ the word insertion penalty and $N_{\mathbf{W}}$ is the number of words in the sequence $\mathbf{W}$.

Using a word insertion penalty penalises the addition of words into the hypothesised word string, as word errors are frequently caused by the insertion of short words with wide contexts. Subtracting a word insertion penalty at the log-probability level is equivalent to scaling or discounting the word probabilities by a fixed amount.

2.2.5 Search techniques

The aim of recognition is to search for the most likely utterance over all possible word sequences. Thus it is necessary to calculate $p(\mathbf{Y}_T|\mathcal{M})$ for each word sequence. The likelihoods could be calculated by the Baum-Welch algorithm of equation 2.19, which requires the calculation of all paths through the model set. For training, where the word sequence is known this is not a problem. However, for the case of continuous speech recognition, all possible model sequences are considered. To make continous speeech recognition easier, the most likely state sequence associated with the observed data is used instead:

$$\phi_i(t) = \max_{\mathbf{q}_{t-1} \in \mathbf{Q}_{t-1}} [p(\mathbf{Y}_t, \mathbf{q}_t|\mathcal{M})] \tag{2.39}$$

where $\mathbf{Q}_{t-1}$ is the set of all valid partial paths of length $t-1$. The variable $\phi_i(t)$ can be calculated recursively:

$$\phi_j(t+1) = \max_{1<i<N} [\phi_i(t) a_{ij}] b_j(\mathbf{y}(t+1)) \tag{2.40}$$

This recursion forms the basis of the *Viterbi* algorithm. The search for the path with the highest likelihood may be performed using the *token passing* method [122]. In the token passing algorithm, for a given time step and feature vector, each state has a single token associated with it, and the token contains a word-end link and the value of $\phi_j(t)$. These tokens are updated for each time step and the most likely token at the end of each model is propagated onto all connecting models. A word-link record is kept with a pointer to the token's value of $\phi_j(t)$. 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 the phonetic context is considered across word boundaries. Using a language model can also expand the size of the decoding network since tokens can only be merged if the word histories are identical. If an N-gram language model is implemented, there must be a separate path through the network for each different word history.

The computational load of the search may be reduced by *pruning* or removing the tokens which fall below a given threshold. The most common method is to set the threshold, or *beam-width* a certain amount below the current most likely path, and delete all active tokens with a likelihood below that. Pruning can 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 small, the most likely path could be pruned before the token reaches the end of the utterance, resulting in

a *search error*. The choice of pruning beam-width is a trade off between avoiding search errors and increasing the speed of the system.

Rather than performing a full decoder search for each new system, it is possible to rescore a constrained set of alternative word hypothesises from the test data generated by a reference system. This approach is known as lattice rescoring [122]. Word lattices are constrained word networks, and can be searched using a Viterbi technique. By reducing the search space the use of lattice rescoring allows much more rapid evaluation of alternative systems and allows more complex language models and acoustic models to be considered. The assumption is that the lattice is sufficiently large and the system under test and the system which generated the lattice are sufficiently close.

2.2.6 Scoring and confidence

The performance quoted on experimental corpora is given as a percentage word error rate (WER). The hypothesised transcription from the recogniser is aligned with the correct transcription using a optimal string match dynamic programming step. Once the optimal alignment is found, the %WER can be calculated as

$$\%\text{WER} = 100 \times \left(1 - \frac{N - D - S - I}{N}\right) \tag{2.41}$$

where N is the total number of words, and D, S, and I are the number of deletions, substitutions and insertions respectively [122].

When comparing different performances of systems, it is useful to have a measure of confidence in the relative improvement or degradation in WER. The test used for the significance of results in this work is the McNemar test. The McNemar test gives a probability that the number of unique utterance errors is different for the two systems begin compared.

The confidence in the significance can be defined as

$$\text{Conf} = 100 \times [1 - P(\text{MIN}_{UUE}|T_{UUE})] \tag{2.42}$$

where MIN_{UUE} is the minimum number of unique utterance errors of the two systems under consideration. The number of unique utterance errors is obtained from a DP alignment of the hypothesised systems and the correct transcription. The total number of unique errors between the two systems is denoted by T_{UUE}. The assumption made is that the distribution of errors follows the binomial distribution for fair coin tosses. A result is considered significant if the confidence in the difference is 95% or above. If the confidence is low, then the number of unique errors in each system is not significantly different given the error rates of the two systems. This is the significance test used throughout this thesis.

2.3 Noise robustness

There are a number of uses for ASR in adverse acoustic environments, such as automotive applications, office environments, telephone speech or military scenarios. Environmental noise can

take a number of different forms. There may be a level of additive background noise corrupting the speech, and the channel or recording environment can introduce forms of convolutional noise to the signal. In addition to the external effects on the speech, speakers tend to alter their speech in the presence of noise to improve the intelligibility. This compensation is called the Lombard effect [43][47]. The Lombard effect can include alterations such as increasing formant frequencies, lowering lower frequency energies, increasing pitch and increasing the durations of certain phone types. The evaluation of noise robustness techniques has often been performed on data corrupted with additive noise. One example of an additive noise task would be the spoke ten (S10) addition to the ARPA 1994 CSRNAB evaluation data, which provided a set of test sentences corrupted with additive noise. More recently, the Aurora corpora have provided a set of data recorded in noisy environments with which to test systems [53].

Techniques for making a speech recognition system robust to environmental noise can be split into three broad classes:

1. Use features which are inherently noise robust;
2. Attempt to estimate the clean speech from the noise corrupted input at the front-end;
3. Compensate the speech models to represent the noise corrupted speech signal.

These techniques will be outlined in the following sections.

2.3.1 Noise robust features

Features can be used which are inherently noise robust. For instance, cepstral mean normalisation will remove some of the effects of convolutional channel noise. Convolutional noise can also be removed by the JRASTA and RASTA-PLP approaches [52]. Inherently noise robust approaches are desirable as they do not need to be adapted to a particular type or source of noise. However, most noise robust features can be further improved by other noise robustness techniques.

2.3.2 Speech compensation/enhancement

The speech can be compensated at the front-end extraction stage by estimating the clean speech parameters using the noise corrupted speech and a model of the noise. Speech compensation and enhancement approaches include spectral subtraction [7], adaptive noise cancellation [112] and probabilistic optimal filtering approaches [80].

Spectral subtraction is probably the simplest form of noise compensation [7]. Points in the N-point spectrum from the noise-corrupted speech $\mathbf{s}(t) = [s_1(t), \ldots, s_N(t)]^T$ are compensated to form the compensated spectral points $\hat{s}_i(t)$ given an estimate of the additive noise source spectrum $\mathbf{q} = \{q_1, \ldots, q_N\}$

$$\hat{s}_i(t) = d_i(t)s(t) \tag{2.43}$$

where

$$d_i(t) = \begin{cases} \left[\frac{|s_i(t)|^\gamma - \alpha|q_i(t)|^\gamma}{|s_i(t)^\gamma}\right]^\beta & (i < f \leq i+1) \text{ if } (\hat{s}_i(t) > k s_i(t) \\ k & \text{(otherwise)} \end{cases} \tag{2.44}$$

and γ, α and β can be set to effect various domains of subtraction. A maximum attenuation is used at k to prevent the spectral values becoming negative. Setting $\gamma = 2$, $\alpha = 1$ and $\beta = 0.5$ will implement power domain spectral subtraction. The parameter α can be set with an estimate of the signal-to-noise ratio and can be made time-dependent. One problem with spectral subtraction is that the phase of the corrupted speech is unknown [21] and thus the assumption that the noise sources will be additive in the magnitude spectrum domain is not necessarily valid.

2.3.3 Model compensation

Another method of compensating for noise in a speech recognition system is to adapt the clean model set or algorithm to the corrupted speech. Techniques using this approach include linear regression adaptation approaches [116], speech and noise decomposition [106] and parallel model combination [32].

Parallel model combination (PMC) attempts to combine the "clean" speech HMM models with a model of the noise distribution [39]. There are no closed-form solutions for the problem of combining the models. Various approximations can be made to perform the combination of the clean speech and noise model distributions [32]:

- samples can be drawn and a Monte-Carlo approach [10] adopted;
- the means in the HMM output PDFs can be combined by mapping the means from the log-cepstral to linear spectral domain and adding the noise and clean speech distributions together (log-add approximation);
- the cepstral speech and noise Gaussian distributions can be mapped to the linear-spectral domain where they are log-normal distributed. The two log-normal distributions can be summed and mapped back to the cepstral domain.

2.4 Feature transforms

If the features used in a system contain correlations, a number of different approaches can be used to model these correlations:

- **Full covariance matrices:** Full or block-diagonal covariance matrices can be estimated from the data. This approach requires a significant increase in the number of estimated parameters from the data.

- **More components:** more Gaussian components can be estimated from the data, and will model the correlations in the output PDFs. This is approach is a rough approximation, however.
- **Decorrelating transforms:** it is also possible to estimate a feature space transform such as PCA which will decorrelate the elements in the feature vector prior to estimating the model [10].

Linear transforms such as linear discriminant analysis (LDA) can also be estimated to improve the discriminative properties of the features and reduce the dimensionality.

2.4.1 Linear discriminant analysis

Linear discriminant analysis is a projection scheme which aims to find a set of feature vectors which have good discriminative properties, that is, the distributions are well separated in the feature space [46]. The technique attempts to maximise the between-class covariance $\mathbf{\Sigma}_B$ and minimise the within-class covariance $\mathbf{\Sigma}_W$ for a set of features. The assumptions made are that each transform class can be represented by a single Gaussian component. First, the feature space is transformed so that the within class covariance matrix has dimensions which are independent and of unit variance. In this transformed space the within class covariance is broken up using the eigenvalues $\mathbf{\Lambda}_w$ and eigenvectors $\mathbf{U}_w$. The between class covariance can then be described in this transformed space by:

$$\mathbf{\Sigma}'_B = \mathbf{\Lambda}_w^{-\frac{1}{2}} \mathbf{U}_w^T \mathbf{\Sigma}_B \mathbf{U}_w \mathbf{\Lambda}_w^{-\frac{1}{2}} \tag{2.45}$$

The between class covariance can also be diagonalised with the transform $\mathbf{U'}_B$ and the largest elements of the resulting diagonal between-class covariance matrix in the transformed space can be selected.

The full LDA transform $\mathbf{A}_{LDA}$ can be described as

$$\mathbf{A}_{LDA} = \mathbf{U}_w \Lambda_w^{-\frac{1}{2}} \mathbf{U}_B^T \tag{2.46}$$

The transformed features are:

$$\mathbf{y}_{LDA}(t) = \mathbf{A}_{LDA} \mathbf{y}(t) \tag{2.47}$$

The LDA transform can be truncated to select only the n largest eigenvalues, the transformed features with the largest ratios of between class covariance to within class covariance. By truncating the lower order LDA components, the dimensionality of the feature vector can be reduced. An LDA transform can also be used to incorporate temporal information from the surrounding frames and reduce the dimensionality rather than appending the standard dynamic parameters to each frame. Using an LDA transform will not necessarily yield an improvement in the performance of an ASR system [69].

2.4.2 Semi-tied transforms

The use of semi-tied covariance matrices is an extension to the use of Gaussian mixture models with CDHMMs [36]. Rather than calculating full covariance matrices for each Gaussian component, each component covariance matrix $\mathbf{\Sigma}_{jm}$ is comprised of two parts. First, there is a component-specific diagonal covariance element $\Sigma_{jm}^{(diag)}$ and second, a *semi-tied* class dependent matrix $\mathbf{H}^{(r)}$. The covariance used is then:

$$\mathbf{\Sigma}_{jm} = \mathbf{H}^{(r)} \Sigma_{jm}^{(diag)} \mathbf{H}^{(r)T} \tag{2.48}$$

The semi-tied matrix $\mathbf{H}^{(r)}$ may be tied over an arbitary set of components such as sets of context-independent classes. The problem of estimating the semi-tied matrix has been solved by an iterative EM approach on top of the estimation of the other HMM parameters which is guaranteed to increase the likelihood [36]. The semi-tied covariance transforms may take the form of full, diagonal or block diagonal structures.

2.5 Speaker adaptation

There exist many variations in speech production between speakers. Speaker adapatation schemes attempt to rapidly compensate an acoustic model to a given speaker. There exist many schemes of speaker adaptation, and it is beyond the scope of this work to present them all. The main techniques for speaker adaptation can be broadly classed as [116]:

1. *Speaker Clustering*: Speaker classes or clusters can be formed (e.g. gender) and appropriate model sets chosen for each test speaker [73];
2. *Feature Normalisation*: The speech input is transformed to a normalised space [92];
3. *Model Adaptation*: The parameters of the acoustic models can be transformed for a given speaker [75].

These methods are presented in the following sections.

2.5.1 Vocal tract length normalisation

One of the inter-speaker differences in speech can be associated with the differing physiology of the vocal tract between speakers. The effects of the varying length will move the resonances in the vocal tract and can be modelled by a transform of the frequency axis in the observed speech. Several transforms have been investigated, including linear and piecewise linear transforms [92] [49] and bilinear transforms [44]. Figure 2.4 shows the use of a vocal tract warping function.

The piecewise linear and bilinear warping functions are both constrained to warp the maximum and minimum frequencies to the same points. In addition, both are parameterised by a

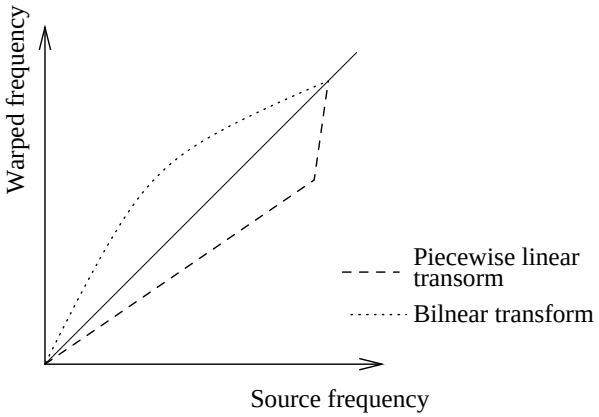


Figure 2.4 *Example of vocal tract length warping functions*

single warping factor for each speaker. The piecewise linear warping function warps the spectrum linearly, except at the highest and lowest regions of the spectrum. This is because the speech spectrum is band-limited and the warping function would otherwise warp the spectrum beyond the effective frequency range. The bilinear transform warps the lower spectral regions further than the higher frequency parts of the spectrum. In practice, neither model appears to outperform the other, but the linear or piecewise linear model is more commonly used for simplicity [105] [92].

The warping factors in the transforms can be estimated by performing a maximum-likelihood search over the speaker set on the training and adaptation data [92]. Alternatively the warping factors can be searched for using other frequency-domain parameterisations, such as formant frequencies [23].

2.5.2 Maximum likelihood linear regression

Maximum Likelihood Linear Regression is a technique used to adapt model parameters from a speaker-independent model to a given speaker with a set of labelled data [75]. The data can be a set of pre-labelled adaptation data, or the labels can be hypothesised by the speech recognition system. The goal is to maximise the likelihood of the adaptation data with a linear regression of the mean of a Gaussian component distribution in the HMM output PDF. The mean vector may be adapted by the $n \times n$ matrix $\mathbf{A}$ and the n-element bias vector $\mathbf{b}$, or alternatively, by considering the $n \times (n+1)$ transform $\mathbf{W}$. The transformed mean vector for a state j, $\bar{\boldsymbol{\mu}}_j$ is given by the unadapted mean $\boldsymbol{\mu}_j$ and the transform parameters:

$$\bar{\boldsymbol{\mu}}_j = \mathbf{A}\boldsymbol{\mu}_j + \mathbf{b} \tag{2.49}$$

$$= \mathbf{W}\boldsymbol{\xi}_j \tag{2.50}$$

where $\boldsymbol{\xi}_j$ is the extended mean array $[1, \mu_{j1}, \ldots, \mu_{jn}]^T$. MLLR seeks to find the transform $\bar{\mathbf{W}}$ which maximises the likelihood of the training data:

$$\bar{\mathbf{W}} = \arg\max_{W} \left\{ \sum_j \sum_t L_j(t) \left(\log \mathcal{N}(\mathbf{y}(t); \mathbf{W}\xi_j, \boldsymbol{\Sigma}_j) \right) \right\} \tag{2.51}$$

Maximisation of the auxiliary function in the Baum-Welch algorithm with respect to $\mathbf{W}$ is a linear regression problem with a closed form solution for $\mathbf{W}$ [75]. It is also possible to estimate an MLLR variance transform matrix $\mathbf{H}$ where the transformed variance $\bar{\boldsymbol{\Sigma}}_{jm}$ may be given by

$$\bar{\boldsymbol{\Sigma}}_{jm} = \mathbf{H}\boldsymbol{\Sigma}_{jm}\mathbf{H}^T \tag{2.52}$$

and solutions exist for the estimation of $\mathbf{H}$ [38].

MLLR uses regression classes to group together Gaussian components in the acoustic space. The assumption is made that Gaussian components that are close in acoustic space for a given speaker will also be close for others. Gaussian components close in the acoustic space are clustered together and organised into a *regression class tree* [34]. If sufficient data exists to estimate a transform, the lowest nodes in the tree are used as the classes to estimate the transforms together. If there is not sufficient data then the parent nodes will form the classes and a more global tying of transforms will be used.

2.5.3 Constrained MLLR and speaker adaptive training

Model-space constrained MLLR (CMLLR) is an extension of model space MLLR where the covariances of the Gaussian components are constrained to share the same transforms as the means. The transformed means and variances $\bar{\boldsymbol{\mu}}_j$ and $\bar{\boldsymbol{\Sigma}}_j$ are given as a function of the transform parameters:

$$\bar{\boldsymbol{\mu}}_j = \mathbf{A}\boldsymbol{\mu}_j - \mathbf{b} \tag{2.53}$$

$$\bar{\boldsymbol{\Sigma}}_j = \mathbf{A}\boldsymbol{\Sigma}_j\mathbf{A}^T \tag{2.54}$$

It has been noted that a duality exists between a constrained model-space approach and a feature-space transform since the two likelihoods are equivalent [35] [96]

$$p(\mathbf{y}(t); \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j, \mathbf{A}, \mathbf{b}) = \mathcal{N}(\mathbf{y}(t); \mathbf{A}^{-1}(\boldsymbol{\mu}_j + \mathbf{b}); \mathbf{A}^{-1}, \Sigma\mathbf{A}^{-1}) \tag{2.55}$$

$$= |\mathbf{A}'|\mathcal{N}(\mathbf{A}'\mathbf{y}(t) + \mathbf{b}'; \boldsymbol{\mu}, \boldsymbol{\Sigma}) \tag{2.56}$$

where $|\mathbf{A}|$ is the Jacobian of the feature space transform and $\mathbf{A}' = \mathbf{A}^{-1}$ and $\mathbf{b}' = \mathbf{A}'\mathbf{b}$. An iterative solution exists for computing the transform matrix.

It is possible to use the constrained MLLR transforms on the training data in a speaker adaptive training (SAT) approach. In the SAT system, CMLLR transforms for the training speakers are computed and the models retrained using the speaker transforms together with the speaker

data. These steps can be reiterated several times to yield a model based on the CMLLR transforms of the training data. The models estimated will be more appropriate estimates for the CMLLR transforms trained on the test data.

3

Acoustic features for speech recognition

The feature sets most commonly used in speech recognition are Mel frequency cepstral coefficients (MFCCs) and perceptual linear prediction (PLP) coefficients. These parameterisations are described in detail in this chapter. Various speech parameterisations have been proposed as alternatives to the spectral representations, and these are outlined and their relative merits discussed. Particular attention is made to features based on the spectral peaks or formant structures in speech.

In addition, techniques for combining different sets of features in the HMM framework are provided in the second section of this chapter. Methods for combining features at different levels in the system are shown and the appropriate features for each combination approach are discussed.

3.1 Human speech production and recognition

The production of speech sounds can be approximated to a source-filter model, where a sound *source* excites a vocal tract *filter*. The source can be split into various broad classes. The source can be periodic, due to the opening and closing of the vocal folds in the larynx. This form of speech is called voiced and the frequency of vibration of the vocal folds is called the fundamental frequency f_0, and is repeated at regular intervals in spectrum. An example of the source and filter for voiced speech is shown in figure 3.1. The excitation source in this idealised diagram exists as a series of impulses separated by the fundamental frequency. The vocal tract filter response is characterised by the series of formants or resonant frequencies. The attunuation of the source by the vocal tract response is obtained by multiplying the two frequency representations together. The excitation may also be obtained from an unvoiced source. In this case, the vocal folds can be adducted or a part of the vocal tract can be moved to create a narrow constriction. The aperiodic excitation source is filtered by the vocal tract and articulators in a similar fashion. The flow of air through this narrow aperture creates an aperiodic excitation signal. The vocal tract response filters the regular excitation of the vocal folds at the source. The resonances in the vocal tract cause peaks to be formed in the resulting speech spectrum. By interpolating the

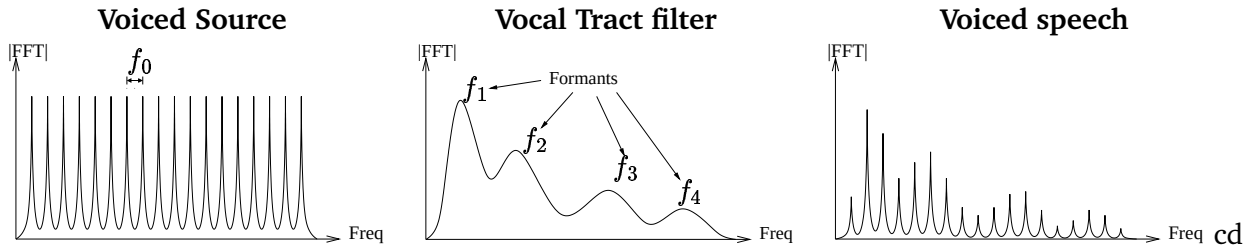


Figure 3.1 *The source and filter response for a typical vowel sound*

pitch peaks in the resulting speech, it is possible to recover the original vocal tract response or spectral envelope.

The vocal tract is a muscular tube which can be modelled as a resonant tube. The resonances of the vocal tract cause the attenuation and amplification of certain frequencies in the excitation signal. Along the vocal tract, there are several articulators, structures such as the lips and tongue which can be moved. The movement of the articulators changes the frequency response of the tract. In addition, the articulators can restrict the airflow to create turbulent or fricative sounds, or completely occlude the vocal tract to create stop sounds. The frequency response of the vocal tract can be characterised by the locations and amplitudes of the resonant frequencies (known as formants) and antiresonances.

The movement of the articulators and the change of source excitation determines the type of speech signal created. The complete set of speech sounds can be characterised by the manner and place of articulation and type of excitation source.

It has been hypothesised that the human speech production and recognition mechanisms evolved in tandem [81]. Thus, it is also important to consider the human auditory system in the speech recogntion process. The primary function of the human ear is to focus sound waves and convert them to electrical impulses in the cochlea. The cochlea is a liquid-filled concentric spiral tube in the inner ear. A cross-section is shown in figure 3.2. Next to the cochlea is the basilar membrane upon which lies the organ of Corti, which contains about 30,000 hair cells. Sound waves are carried here, and transmitted to the fluid by the middle ear. The hairs on the organ of Corti will vibrate in response to the movements in the fluid, and fire the neurons connected to them.

As the basilar membrane is tapered and varies in flexibility along its length, the hairs resonate at different characteristic frequencies. Hence, the neural signals transfer signals proportional to the energy levels in different frequency bands to the brain. The perception of frequency is uniform within certain frequency bands in the human ear, called *critical bands*. In each critical band sound is analyzed independently. Each band corresponds with an equal section of cochlea. The resolution is non-linear, with the most sensitive frequency resolution up to about 1kHz. Below 500 Hz bandwidths are constant, equal 100 Hz. Over 500 Hz the width of each next critical band is 20% larger than of the band below. It is possible to model the human auditory system as a set of band-pass filters with bandwidth of corresponding critical band.There are

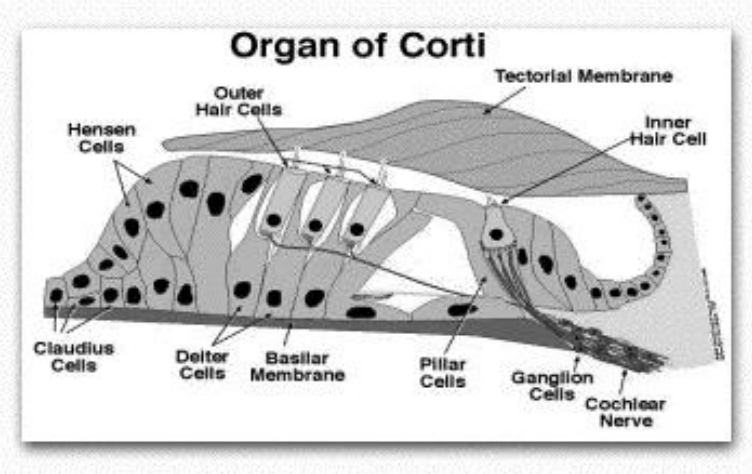


Figure 3.2 *The physiology of the inner ear (from [14])*

various psychoacoustic scales that can approximate the non-linear perceptual frequency scale, such as the Mel or Bark scales [43].

3.2 Spectral speech parameterisations

There are a number of desirable properties in features for ASR systems. First, they must adequately represent the speech. It is desirable that the features contain sufficient information to represent the phonetic content of speech. Second, they should be speaker independent and should not contain redundant or extraneous information. For example, in the recognition of spoken English, the excitation source is considered to be largely uninformative compared to the vocal tract response. Also, it is desirable that the feature set be of low dimensionality and as compact as possible to reduce the number of parameters estimated in the speech recognition system. Additionally, if the features are mostly uncorrelated then simpler forms of covariance modelling can be used, and there are fewer parameters to estimate in the system.

3.2.1 Speech Parameterisation

To process speech for ASR, the first stage is to capture the speech waveform with a microphone and convert it to a discrete signal for the computer. The sampling rate used varies, but typically the sampling frequency is either at 16kHz or 8kHz, to give a an effective frequency range from 0 to 8kHz or 4kHz. Telephone speech is usually sampled at a rate of 8kHz, but the effective frequency range will have been limited by the system to 125-3800Hz. The waveform is sampled at an accuracy of 16 bits in the case of uncompressed speech, but can also be compressed using u-law compounding down to 8 bits.

Frequency representations of the speech are considered to be more useful for speech recognition than the time-domain signal. In order to obtain a time-discrete representation, the digitised

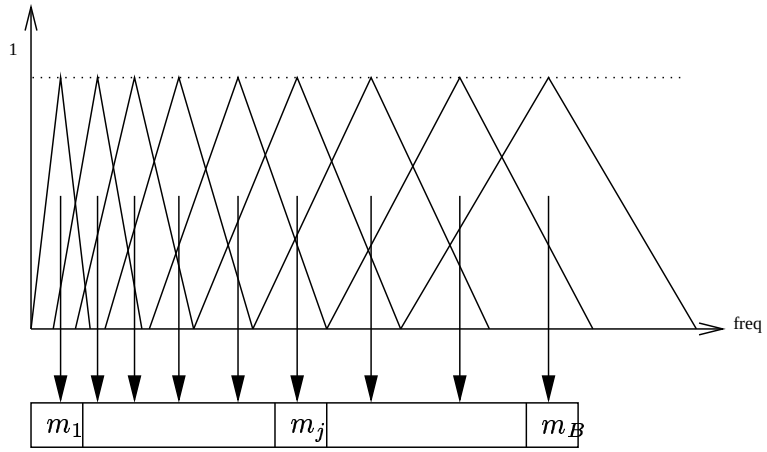


Figure 3.3 *Overlapping Mel-frequency bins*

waveform is split into overlapping frames. Speech signals are assumed to be quasi-periodic, stable over short periods and the frames are typically of width 25ms and at 10ms intervals. The speech waveform $w(n)$ is multiplied by a Hamming window of length T to reduce spectral distortion at the edges of the window

$$\hat{w}(n) = \left[0.54 - 0.46\cos\left(\frac{2\pi n}{T-1}\right)\right] w(n) \tag{3.1}$$

The FFT of the windowed speech is then taken. Thus, from the sampled speech waveform, a series of time discrete spectral frames is obtained.

3.2.2 Mel frequency cepstral coefficients

Mel frequency cepstral coefficients (MFCCs) are probably the most commonly used technique to represent the speech spectrum in ASR systems, and can be considered a baseline for performance comparison of feature sets [15].

The MFCCs are generated by first obtaining the speech spectrum as described above. A number (B) of triangular shaped filter bin functions equally spaced on the Mel scale (see figure 3.3) are taken from the magnitude spectrum:

$$f_{mel} = 1127\log[1 + \frac{f_{Hz}}{700}] \tag{3.2}$$

Usually around 24 filter-banks are used to represent the spectrum. The log-spectral filter-bank outputs could be used for speech recognition. The problem, however, is that a high energy in a given filter-bank corresponds to a high energy in the surrounding filters, and the features are highly correlated. The cepstral coefficients ($c_i(t)$) are then calculated by taking the discrete cosine transform (DCT) of the Mel-bin log energies.

$$c_i(t) = \sum_{b=1}^{B} \log(m_b(t)) \cos\left(\frac{i(b-0.5)\pi}{B}\right) \tag{3.3}$$

The lowest order cepstra represent the general shape of the spectrum and the higher orders represent the pitch voicing and the sharper changes in the frequency spectrum. In most applications, cepstra 1-12 are used for recognition, with a normalised log-energy term appended to the representation [122]. The MFCCs can also be transformed (or *liftered*) to emphasise different cepstral parameters [67]. There are several advantages to using MFCCs as features for ASR. First, the Mel-spaced bins used are analogous to the critical bands observed in the basilar membrane response in the human ear. Taking the logarithm of the bins will approximate the magnitude response of the ear. The DCT can been shown to approximate a set of principle components analysis (PCA) basis functions, meaning the cepstra are largely uncorrelated [51]. Thus diagonal covariance matrices may be used to model the distributions of the features.

The effects of any convolutional channel transfer functions are multiplicative in the spectral domain. However, in the log cepstral domain, this becomes a simple addition. By subtracting the mean of the MFCCs from the parameterisation of an utterance, the channel effects can be normalised. This technique is known as Cepstral Mean Normalisation (CMN).

3.2.3 Perceptual linear prediction

Perceptual Linear Prediction (PLP) coefficients have been proposed as an improved spectral representation [50]. The motivation of PLP is to closely model the psychoacoustics of hearing.

Three properties of the human auditory system are implemented in PLP: the nonlinear frequency response of the the human ear; the critical bands in the cochlea; and the non-linear amplitude response. In addition, linear predictive analysis is performed to exploit the resonant nature of the vocal tract function [77].

First, the nonlinear frequency response of the human ear is approximated by warping the spectrum to the Bark frequency scale f_{bark}.

$$f_{bark} = \log\{\frac{f_{Hz}}{600} + [(\frac{f_{Hz}}{600})^2) + 1]^{0.5}\} \tag{3.4}$$

The warped spectrum is then convolved with a series of critical band filters spaced in the Bark scale that roughly match the psychoacoustic information available for the human ear. These psychoacoustic techniques attempt to model the human auditory system's frequency response and masking effects. Using Mel-scaled triangular bins to model the critical bands has been equally successful in other implementations of PLP features [117].

To model the variations in perceived loudness in the human auditory response an equal loudness function $E(\omega)$ (eq. 3.5) is applied to the critical band filter-bank values. The equal loudness preemphasis function has a peak at about 3.5kHz and is based on human auditory response data.

$$E(\omega) = \frac{(\omega^2 + 56.8 \times 10^6)\omega^4}{(\omega^2 + 6.3 \times 10^6)^2(\omega^2 + 0.38 \times 10^9)(\omega^6 + 9.58 \times 10^{26})} \tag{3.5}$$

Finally, the cube root of the equal loudness bins is taken. This corresponds to the non-linear relationship observed between intensity of a sound and the perceived loudness. Once the spectrum is obtained it is then converted back into the time domain and an autocorrelative all-pole LP analysis is performed to obtain the PLP coefficients. The filter coefficients $\{a_1, \ldots, a_P\}$ form a prediction filter $A(z)$

$$A(z) = 1 - \sum_{k=1}^{P} a_k z^{-k} \tag{3.6}$$

The autocorrelative function can be obtained from the inverse Fourier transform of the power spectrum. Durbins recursion algorithm can then be applied to obtain the prediction coefficients which minimise the error [10]. The perceptual linear prediction coefficients $\mathbf{p}_N = [p_1, \ldots, p_N]^T$ are calculated from the prediction filter coefficients $[a_1, \ldots, a_n]$:

$$p_n = -a_n + \frac{1}{n} \sum_{i=1}^{T-1} (T - i) a_i p_{n-i} \tag{3.7}$$

where p_n is the n^{th} PLP coefficient. When used in speech recognition, these features slightly outperform MFCCs and have been shown to provide more noise robustness [50].

3.3 Alternative parameterisations

There has been much interest and research undertaken into other features vectors for speech recognition. Features based directly or indirectly on the positions of the articulators which produce the underlying sound have been proposed. Another area of interest is the use of the formant locations as features for speech recognition systems. Due to problems in robustly estimating the formants, other speech parameterisations have been proposed which represent the spectral peaks. These parameterisations are presented in this section and their use in ASR systems is discussed.

3.3.1 Articulatory features

Different speech sounds are produced by varying the positions of the articulators: the tongue position and velum; the degree of voicing or of nasalisation [81]. It has been proposed that if the positions of the articulators were known or estimated they could be useful representations of the class of speech sound [6]. Articulatory features can either be based directly on the movement of articulators (using medical imaging data or tags on the articulators) or by using pseudo-articulatory features [27]. Pseudo-articulatory features are based on extracting information on the articulators via a non-linear mapping from the original speech [47]. Both the direct measurement of the articulators' positions and the estimation of positions have been investigated for speech recognition systems.

Coarticulation effects between speech sounds are physically caused by the movement of the articulators between sounds. Hence the articulatory domain may be the best domain in which

to model those effects [25]. Additionally the much debated psychoacoustic phenomena of categorical perception can also support the use of articulatory features in a recognition framework [6].

One approach to using articulatory information in ASR used phones as the basic model unit, and allowed for asynchronous alignment between the features [24]. The structure of the models was ergodic and each state in the HMM represented a combination of features so that coarticulation effects were directly included in the model. The features used had strong articulatory correlates [19], and were assumed hidden by the model, therefore a stochastic (rather than deterministic) relationship between the acoustic features and the feature values existed.

Pseudo-articulatory features for vowels have been used in a linear regression system [62]. Prelabeled vowel sounds were used to establish a relationship with the cepstra from a frame and a set of 4 pseudo-articulators (high, back, round and tense). Then a search mechanism was used to to map this relationship to all phone classes. A dynamic programming step was used to find the best string of phones. Good recognition was obtained for vowel sounds, since these can be described well by the pseudo-articulators chosen. However, the plosive and fricative sounds were poorly represented by this model.

One problem with articulatory features is the extraction of the features. Extracting articulator positions directly using imaging equipment is not practical for most speech recognition tasks. The mapping from the speech signal to the articulator positions or pseudo articulatory labels is often modelled by a non-linear process and relies on appropriately labelled speech data to train neural nets or deterministic mapping systems [27]. Another problem is that there can be several mappings (methods of articulation) for a single speech sound [65].

Articulatory features can also be regarded as a recognition unit in the same way as a phone or a word [22]. Each phone can be subdivided into phonological features - for example, the phones /m/ and /n/ can be described as [*+labial, +nasal, +voiced*] and [*+velar, +nasal, +voiced*] respectively . Hence, models for these units can be trained in the same way as phones on labelled training data. Use of articulatory units has been shown to be effective on small tasks in noise corrupted environments [71]. The articulatory features have been combined with a "standard" phone model set by combining scores in an asynchronous streaming architecture [79]. Including these feature scores gives a slight improvement in combination with MFCCs, but larger improvements were yielded on hyperarticulated or overemphasised sounds [47].

Articulatory features also contain strong temporal correlations. As the articulatory features represent the positions of the articulators, it is believed that the articulatory domain is a natural domain to work in for trajectory and segmental models [27]. The mapping from the acoustic level to the underlying trajectories in a segment model has also been explored using an intermediate articulatory layer [63]. In this approach, the acoustic features were mapped to an articulatory layer with linear trajectories in a segmental HMM.

Many recognition systems using articulatory-based features implement an alternative topology of HMM to successfully utilise articulatory features. It appears difficult to extract articulatory features separately from the recognition process [20]. The application of articulatory features is

limited by the ability to estimate the parameters and finding a suitable framework by which to incorporate them.

3.3.2 Formant features

As mentioned previously, formants are peaks in the spectrum caused by the resonances of the vocal tract. They are believed to be representative of the underlying speech sequence [13] [88]. It has been shown [59] that formants have a smoother trajectory, and more importantly, these trajectories are more consistent for a given phone class, than MFCC parameters.

It has been shown that the human perceptive distance for speech sounds is partly based on spectral distances, but that relatively small shifts in formants could cause large shifts in the perceptive distance [72]. Other work has also supported the view that formants are extremely important for human recognition of speech, especially for noisy or band-limited channels [61]. Some results indicated that small shifts in formant position can change the perceived speaker identification (but not the perceived words) and a small relative shift of one formant with respect to the others was even possible to simulate a shift of accent [61]. From this it seems reasonable to conclude that formants can represent some fundamental features of speech.

There have been many proposed methods for calculating formants from the speech signal. The techniques can be broadly divided into three categories: analysis by synthesis, peak picking and solving the roots of the linear predictor polynomial. Note that strictly speaking, approaches using peak-picking algorithms which are not based on finding the resonant frequencies are not formant-based features.

The ESPS toolkit implements a formant tracker which solves the complex roots of the denominator polynomial of a linear predictor (LP) Z transform [77] [103]. The linear predictor is an all-pole model. This is reasonable for representing voiced sounds. When obtaining formants by LP analysis, results can often be poor if no form of continuity constraint or fixed trajectory is applied. However, these global continuity constraints can be too weak in the case of sonorants and too strong in the case of vowel-consonant boundaries. The ESPS formant tracker hypothesises more formants than are to be used, and performs a Viterbi search which optimises the local mapping cost and the transition cost of a given set formants in the utterance. The local mapping cost is a combination of the bandwidth, frequency and deviation of the formant, and the transition cost is combination of the relative formant change modulated by a measure of the signal's stationarity.

Another proposed formant recogniser uses a code book of spectra which have been hand-labelled with formant positions by a human expert [57]. The use of a code book of labelled spectra aims to reduce the problem of ambiguous peak structures or the inconsistent labelling of LP-based systems. The code book is searched to find the N most likely candidates using a spectral distance measure, then a dynamic programming alignment step is performed to map the hand-labelled formant positions to the best match with the input spectra. The model implements continuity constraints to create consistent formant trajectories, and also gives a measure of

confidence in the proposed candidates, together with alternative formant positions.

It has also been proposed that calculating formant positions may be aided by some knowledge of the phonetic class of the speech segment. An analysis by synthesis approach has been investigated using parallel digital resonators [111] to model formant positions. Also, the use of N-best lists in LPC analysis to delay selection of tracks until after phonetic search [98] has been investigated. The results using N-best lists of formants [99] performed no better than a MFCC representation, but when a human expert selected the correct hypothesis from the N-best list, performance was improved, which suggests formants may be promising for speech recognition if the identification can be made more consistent.

The formant positions contain information about the class of speech sound. Formants have been successfully used to improve a speaker adaptation system [120], by shifting formants to reduce the acoustic mismatches for different speakers. The shifts in formant positions between speakers have been used to calculate VTLN warping factors [74] [68].

There are a number of problems associated with the use of formants as features [56]. Formants are not always well defined in the spectra, for example in the cases of turbulent air flow (for fricatives) and unvoiced phones. The formant peaks may lie between the fundamental frequency harmonic peaks for speech with a high pitch. Formants can be labelled unreliably by LPC analysis, and applying continuity constraints can compromise the temporal resolution of the features. Formants frequencies alone do not contain amplitude information, which is required to discriminate between some nasalised sounds and voiced vowels which exhibit similar formant frequencies [56]. Formants alone cannot describe the general spectral shape, which means it is impossible to reconstruct the spectrum and some existing adaptation techniques cannot be applied. Another consideration for their use in ASR systems is that formant features possess a degree of correlation. Hence, the use of diagonal covariance matrices may not be based on a valid assumption, and more parameters may be required for the models to adequately represent the parameters.

It is desirable for recognition systems that the features represent the individual classes uniquely and consistently. Formants possess the problems that different phone types can have the same formant locations. In addition, a given phone type may have different formant locations depending on the context and the continuity constraints applied. However, if these problems can be overcome, it has been shown that formant or peak representations can be useful in combination with spectral representations.

3.3.3 Gravity centroids

Energy gravity centroids or spectral sub-band centroids [84] are an alternative approach to parameterising the peaks in the spectrum. The speech spectrum is split into a number of sub-bands by band-pass filters. The energy moments for each sub-band are then calculated to form the gravity centroid features. The first order moment will give an indication of the location of the peak in a given sub-band, and the second order moment will give information about distribution

around this peak.

Given a power spectrum $\mathbf{s}(t) = [s_1(t) \dots s_N(t)]^T$, the j^{th} moment $M_j^p(t)$ order p can be calculated

$$M_j^p(t) = \sum_{i=1}^{N} i^p h_j(i) s_i(t) \tag{3.8}$$

where $h_j(i)$ is the output of the band pass filter for the j^{th} moment. Approaches using bins equally spaced in linear, Mel and Bark frequency intervals have been applied, with the results yielding similar performance [12]. The optimal filter-bank shape was found to be rectangular. The zeroth moment gives the energy present in each sub-band From the moment $M_j^p(t)$, a normalised moment $\tilde{M}_j^p(t)$ can then be computed:

$$\tilde{M}_j^p(t) = \frac{M_j^p(t)}{M_j^0(t)} \tag{3.9}$$

The first and second order moments (p=1 and p=2) have been found to be useful for speech recognition and also in combination with MFCCs [104]. The first normalised moment corresponds to the mean of the sub-band filter reponse, and has been related to the location of the spectral energy peak in the region [84]. The second moment contains information about the spread or distribution of energy and can be related to the bandwidth of the peak in the region. Energy centroids have been applied to clean speech and noise corrupted environments [30]. Although the performance of the gravity centroids alone was poorer than MFCCs in clean speech environments, they were more robust than MFCCs for certain types of noise corrupted data. When combined with an MFCC parameterisation, Gravity Centroids have been shown to improve performance on limited domain tasks [16].

Gravity centroids do not have the trajectory continuity problems that formants present. Conversely, one problem is that the dynamic coefficients of the gravity centroids possess too small a dynamic range for them to be useful. This is most likely to be because the choice of sub-bands filters will strongly constrain the location of the peaks [12]. Another problem is that the results will be ambiguous or inconsistent if two spectral peaks are located in the same band. An alternative implementation has been made using an approximation to the continuous time differential instead of linear regressive dynamic parameters [11]. Subband centroids are a more direct spectral representation than formants. As such, it is easier to model additive noise sources and apply techniques for compensation and adaptation of the features.

3.3.4 HMM-2 System

The HMM-2 system is an extension to the HMM acoustic model framework [110] . Standard HMMs use continuous density Gaussian mixture models to estimate the probability of a feature vector given the current state. Each temporal state in the HMM-2 system has a second frequency HMM associated with it. At each time step, the frequency HMM generates a sequence of scalar

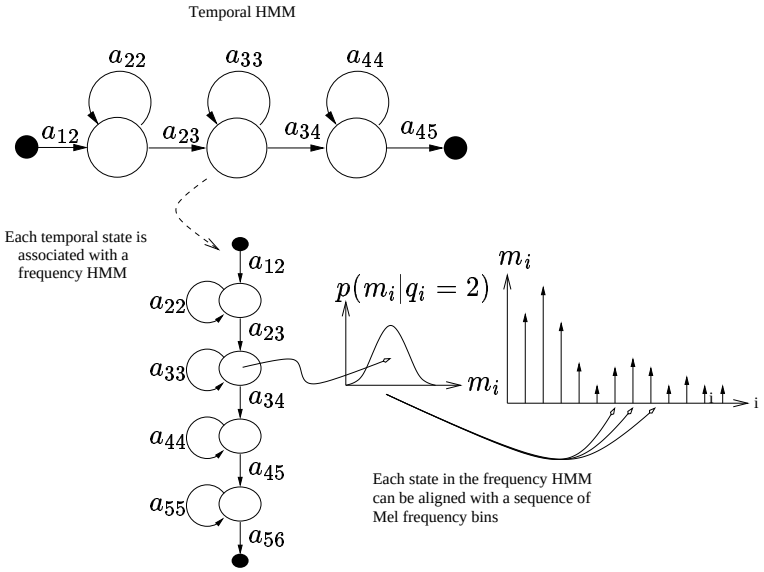


Figure 3.4 *Overview of the HMM-2 system as a generative model for speech*

values which represent the filter-bank energies and their derivatives. Thus, the emission probabilities for a given temporal state are estimated by the state-dependent frequency HMM. Each state in the frequency HMM can be mapped to a set of filter-banks and their dynamic parameters, as seen in figure 3.4.

The frequency HMMs are used to generate the likelihoods for a filter-bank feature vector. These likelihoods can be processed by the conventional temporal HMM as likelihoods generated by a normal output PDF. The HMM-2 system can alternatively be viewed as an expanded HMM system with synchronisation points enforced at the end of each time step. Thus, the system can be training with the EM algorithm in a similar fashion to the training of HMMs, estimating state/time and state/frequency alignments for the HMMs. The HMM-2 system can be used as an acoustic model, but when used in practise, the recognition results were poorer than those from a conventional HMM [107].

An alternative use has been proposed for the HMM-2 system. The state transitions in the frequency HMM may be used as features for speech recognition [108]. If the frequency states tend to match regions of similar energies, the segmentation of the frequency HMM may indicate the regions of similar energy levels or peaks. The segmentations have been called “formant-like” features and may then be used as features in a conventional HMM [107].

To ensure some continuity the filter-bank frequency can also be used as a feature in the frequency HMM layer. This effectively places a prior distribution on the expected positions of the formants for a given phone class and hence increases consistency and continuity in the extracted parameters.

The segmentations from a HMM-2 system have been used in combination with standard MFCC parameterisations. They improved recognition performance in the presence of additive factory noise from the noisex database [108]. The advantage of the HMM-2 systems over other

formant-like features is that the extraction process is expressly part of the recognition process, so the feature extraction process is class dependent. The extraction process also has a flexible statistical framework.

Though HMM-2 features can be seen to follow the locations of the formants, the features based on the segmentations are limited by the number of Mel-scaled filter-banks. Each HMM-2 feature in the current implementation [108] can only take one of 12 discrete values. The standard Δ and Δ^2 parameters give no improvement when added to the system [109]. This may be due to high level of quantisation making the dynamic coefficients unreliable. Increasing the number of Mel-scaled filterbanks would increase the resolution in the frequency domain, but will lead to an increase in the computational complexity. Also, although the features possess inherent noise robustness, due to the nature of the extraction process and the mapping between the spectral and feature domains, it is difficult to apply schemes to rapidly adapt the features.

3.4 Spectral Gaussian mixture model

A number of techniques for extracting formants from speech data were describedin section 3.3.2. However, there are a number of problems associated with the use of formants as features [56]. For example, formants do not extract any amplitude information from the speech signal, and are often poorly defined in certain types of phone. Recently, there has been interest in statistical methods to parameterise the spectrum in terms of its peak structure rather than searching for the resonances. As outlined in section 3.3.3, the use of Gravity Centroids to describe the first and second moments of spectral sub-bands has been shown to provide features complementary to MFCCs. However, one of the limitations of the gravity centroid features is that the choice of sub-band filters severly constrains the extracted parameters. For instance, the gravity centroid features will be ambiguous if there are two peaks within a sub-band.

Another statistical method of estimating spectral parameters from a speech spectrum is the Gaussian Mixture Model (GMM) proposed by Zolfaghari and Robinson [125]. Originally proposed as a method for parameterising the speech spectrum it was later developed as a low-bit speech codec [126] [127]. The technique assumes that a set of Gaussian components can represent a distribution based on the spectral envelope. The GMM parameters are iteratively estimated using the expectation maximisation (EM) algorithm. The posterior probabilities of each bin being generated by the target mixtures are estimated, then these values are used to calculate the Gaussian component parameters. These steps are iterated to converge upon a solution. The spectral GMM estimation can be viewed as an extension of the gravity centroid features, as shown in figure 3.5. The gravity centroid features will be related to the GMM parameters in the case where he posterior probabilities for each component are fixed and identical to the sub-band filter functions. The spectral GMM has more flexibility to model the spectrum than the gravity centroids parameters, but has more degrees of freedom.

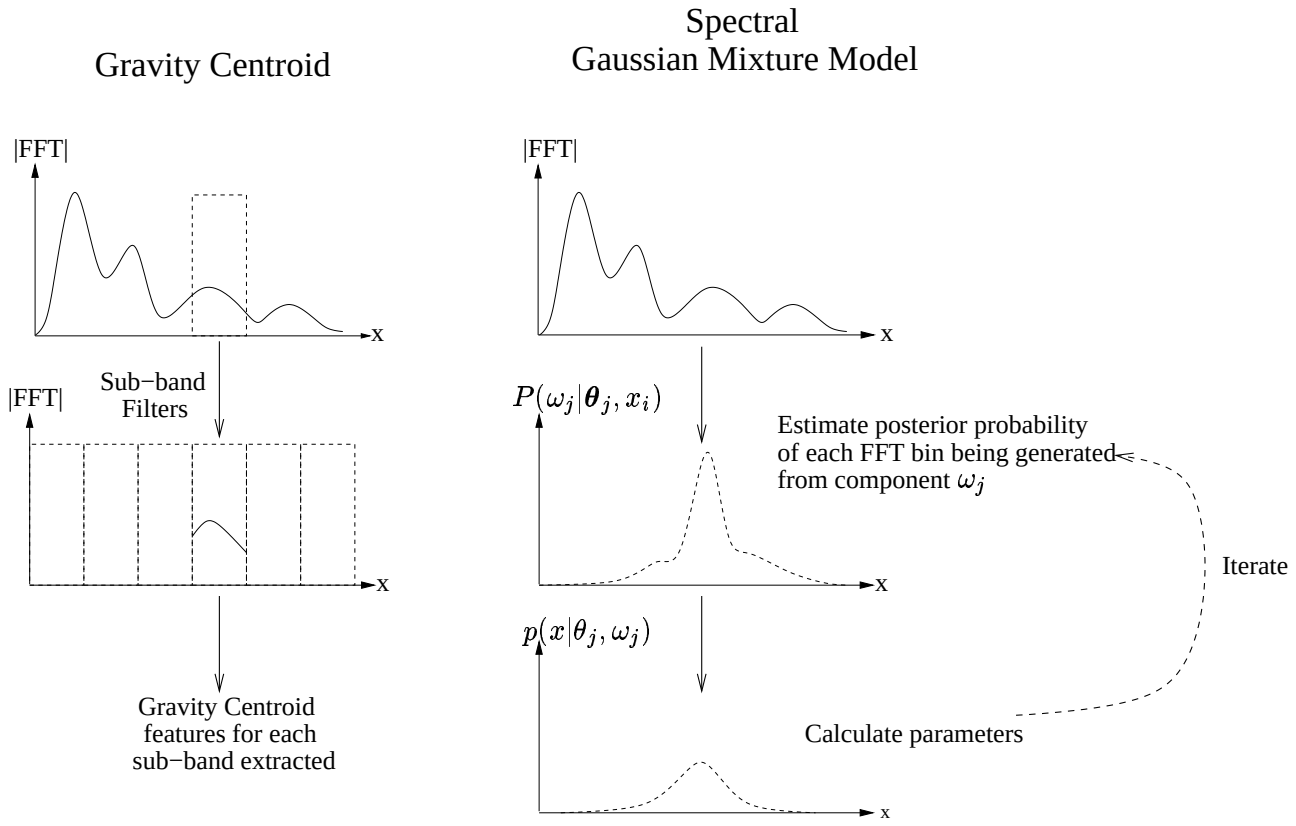


Figure 3.5 *Extracting gravity centroids and GMM parameters from a speech spectrum*

3.5 Frameworks for feature combination

Alternative speech parameterisations may provide information complementary to the standard MFCC or PLP parameterisations. Thus, given a suitable framework for combining the two information sources, improved performance could be attained. This section details some of the main approaches that have been proposed. The simplest method of combining features together is to concatenate them into a single feature vector. Another method is to split the different features into separate information streams which can be combined in a synchronous or asynchronous fashion. Different parameterisations or acoustic models can also be combined at the sentence or word level.

3.5.1 Concatenative

The simplest approach to incorporating different features is to append them onto the existing feature vector. This is refered to *concatenative* combination in this thesis. This is the simplest method of feature combination. After concatenating the features together, the standard dynamic parameters of all features can be appended to the feature vector. Feature concatenation has proved to be a successful method of combining MFCC parameterisations with a range of different speech features. The features from the gravity centroids [16], the HMM-2 system [108] and spectral phase features [97] have been successfully appended to an MFCC feature vector and yielded improved recognition results.

Simply concatenating features together increases the size of the feature vector. The alternative features may be highly correlated with the existing features. Furthermore, not all features appended to the spectrum will be useful. The use of Fisher ratios can give some indication of the discriminative information of a feature, but will not take into account the correlations between the feature sets. An increment information metric has been proposed which can also give some indication of the amount of improvement to be expected when including a feature into an existing system [83]. It is also possible to use a transform such as LDA or PCA on the concatenated feature vector prior to its use in the recogniser. Feature space transforms can remove the correlations between the feature types. The LDA matrix approach can be truncated to reduce the dimensionality of the data and remove the elements with the lowest Fisher ratios. Using this feature selection/extraction approach is sometimes adopted because adding features increases the size of the feature vector and hence the number of parameters to be estimated in the system. Increasing the dimensionality of the feature vector will also change the dynamic ranges of the state likelihood calculations. As a result, the search techniques will usually have to be run with a wider beam width on the pruning thresholds to keep the same number of active tokens. The optimal language model scale factors will be changed. The optimum number of components in the HMM emission PDFs may also change. If the features are correlated with respect each other, different forms of covariance matrix may be considered, or the number of Gaussian component mixtures may be increased to model the correlations. In conclusion, combining features by concatenating them into a single feature vector has been used to successfully incorporate

some alternative speech parameterisations. However, care must be taken when considering the usefulness of the additional features and the effects of increasing the size of the feature vector.

3.5.2 Synchronous streams

It is also possible to split the observations to the recogniser into multiple information streams. The feature vector $\boldsymbol{y}(t)$ can be divided into R separate information streams $\{\mathbf{y}_1(t), \cdots, \mathbf{y}_R(t)\}$. The observations are treated as conditionally independent information sources, that is, independent given the state generating the observation. Synchronous streams allow the number of "effective" full dimensional components to be increased without a large increase in the number of model parameters. The use of the stream weights also allows the contribution of different streams to the likelihood computation to be varied.

The output probability for a synchronous stream system for state j at time t is then given by:

$$b_j(\mathbf{y}(t)) = \prod_{r=1}^{R} \Big[\sum_{m=1}^{M_r} P(\omega_{jrm}) b_{jrm}\left(\mathbf{y}_r(t)\right) \Big]^{\gamma_r} \tag{3.10}$$

where $\boldsymbol{\gamma}_r$ is the stream weight of stream r, and M_r the number of mixtures in the r^{th} stream. The component priors $P(\omega_{jrm})$ will sum to one for each given stream ($\sum_{m=1}^{M_r} P(\omega_{jrm}) = 1$) and:

$$b_{jrm}(\mathbf{y}(t)) = \mathcal{N}\left(\mathbf{y}_r(t); \boldsymbol{\mu}_{jrm}, \boldsymbol{\Sigma}_{jrm}\right) \tag{3.11}$$

where $\boldsymbol{\mu}_{jrm}$ and $\boldsymbol{\Sigma}_{jrm}$ are the mean and covariance for stream r. By increasing the value of the exponent γ_r different observation sources can be given more emphasis.

Training of the system can be accomplished by optimising the parameters for each state, mixture and data stream, since the data streams are considered to be independent given the state.

The training is then based on the stream/frame alignment $L_{jrm}(\tau)$ where

$$L_{jrm}(\tau) = \frac{L_j(\tau) b_{jrm}(\mathbf{y}_r(\tau))}{\sum_{m=1}^{M_r} b_{jrm}(\mathbf{y}_r(\tau))} \tag{3.12}$$

and the state/frame and mixture/frame likelihood can be calculated as before. Hence, the updated means for the model set are given by:

$$\hat{\boldsymbol{\mu}}_{jrm} = \frac{\sum_{\tau=1}^{T} L_{jrm}(\tau) \mathbf{y}_r(\tau)}{\sum_{\tau=1}^{T} L_{jrm}(\tau)} \tag{3.13}$$

and the component weights are given by

$$\hat{P}(\omega_{jrm}) = \frac{\sum_{\tau=1}^{T} L_{jrm}(\tau)}{\sum_{\tau=1}^{T} \sum_{m=1}^{M_s} L_{jrm}(\tau)} \tag{3.14}$$

There does not exist an algorithm to train the stream weights in a maximum-likelihood framework. However it is possible to train using discriminative fashion using minimum classification error [78] or using gradient probability descent methods [89]. However, the simplest technique is to set a fixed global value for the stream weights. This value can be trained heuristically on a subset of the test or training data.

3.5.3 Asynchronous streams

In synchronous stream systems each observation is independent given the state that generated it, and each state is assumed to generate observations in the streams at the same time instance. Alternatively, each stream may be considered independent, and run separately in time as an *asynchronous* stream system. Recombination of the streams can occur at sentence, word or phone level. The recombination points are refered to as *anchor* points.

One approach to using asynchronous information streams formed a recogniser using parameters from frequency sub-bands as the different streams [8]. The anchor points were set between speech units to force some level of synchrony between the streams. Three strategies for recombination were considered: stream weights based on normalised phoneme-level recognition rates; stream weights based on normalised signal to noise ratios in each band and recombination using a neural net.

The use of an asynchronous stream can only be considered advantageous if the separate streams are not synchronous. This will determine whether the assumptions of the model are appropriate for the data. Other systems have also been built using an asynchronous streaming system to combine spectral sub-bands to yield improved recognition results on the TIMIT corpus [113].

3.5.4 Using confidence measure of features in a multiple stream system

When combining different features with MFCC or PLP features the complementary features may be more useful for certain classes. For example, spectral peak representations may be less reliable for certain classes of phones such as fricatives and nasalised sounds. In order to allow for this, the stream weights could be made to be class dependent and trained heuristically or in a discriminative fashion. However, this could effectively weight the probabilities of given phone classes and affect the recogniser performance.

An alternative approach is to make the stream weights a function of the observed speech signal at time t [114]. A measure of confidence $\xi_i(t)$ has been extracted from the spectrum when estimating the formant frequencies [56]. This confidence metric was based on the amplitude and degree of curvature of each formant or spectral peak. The formant features were combined with the MFCCs by applying the confidence measure as scaling factors on the state probability calculations. The likelihoods of the formant features were weighted by their confidence measure, and the higher order cepstra included were deweighted by the confidence metric to maintain the dynamic range of the system. As the system built used only a single component in the state output PDFs, the confidence scales can be viewed as a form of streaming system with time-dependent stream weights $\gamma_r(t)$:

$$b_j(\mathbf{y}(t)) = \prod_{r=1}^{R} \Big[\sum_{m=1}^{M_r} P(\omega_{jrm}) \mathcal{N}(\mathbf{y}_r(t); \boldsymbol{\mu}_{jrm}, \boldsymbol{\Sigma}_{jrm}) \Big]^{\gamma_r(t)} \tag{3.15}$$

The system used a representation of eight cepstral coefficients and three formant frequencies. The first five formant frequencies and their dynamic features had the stream weight fixed to one. The weights of the three formants were set to their respective confidences $\xi_i(t)$. The three higher order cepstra had a weight set such that each was weighted by a corresponding formant confidence $1 - \xi_i(t)$.

Using the stream system as defined above gave an improvement in performance on the TIMIT task over an MFCC system and a concatenative MFCC+formant system [114].

3.5.5 Multiple regression hidden Markov model

Some features which can be extracted from the speech signal can represent some of the inter-speaker variations. These features will be referred to as auxiliary features. The multiple-regression HMM (MR-HMM) is a method of incorporating auxiliary information features (such as the fundamental frequency f_0) to adapt the model parameters and thus better represent the standard acoustic features [28]. The means of the MFCC parameters $\boldsymbol{\mu}_j(t)$ are adapted based on the auxiliary information $\mathbf{y}_2(t)$ and a transform $\boldsymbol{A}$:

$$b_j(\mathbf{y}(t)) = \mathcal{N}(\mathbf{y}_1(t); \boldsymbol{\mu}_j + \boldsymbol{A}\mathbf{y}_2(t), \boldsymbol{\Sigma}_j) \tag{3.16}$$

The use of a MR-HMM incorporating pitch and a low-frequency energy term has been used to reduce error rates on a phone recognition task by more than 20%. The approach could be extended to allow the transform $\boldsymbol{A}$ to be tied over multiple classes. Features which could possibly be used to transform existing spectral parameterisations include pitch, energy, degree of voicing and formant features. The MR-HMM is more suited to using auxiliary information which models the intra-speaker correlations rather than features which could be used to provide discriminatory information. Formant features could be used as they possess both discriminatory information about the task and speaker-dependent information as well.

4

Gaussian mixture model front-end

In this chapter, a method for parameterising the speech spectrum based on estimating a Gaussian mixture model (GMM) from the speech spectrum is shown. The use of the EM algorithm to fit a set of Gaussians to a spectral histogram is described, together with issues in representing the speech using this model. In addition, techniques to extract features from the parameters of the GMM are discussed, along with the properties of the extracted features, and a measure of confidence in the GMM estimate is presented. Methods to apply temporal smoothing and for enforcing continuity constraints on the extracted parameters are also described. Finally, techniques to compensate the GMM features in noise-corrupted environments are presented.

4.1 Gaussian mixture model representations of the speech spectrum

This section outlines the basic theory of extracting parameters for a set of Gaussian components from the FFT of the speech signal. Unlike the previous work with spectral GMM estimation, the histogram is formed from continuous bin probability functions as opposed to the single impulse functions used in previous approaches [125].

4.1.1 Mixture models

The Gaussian mixture model for speech representation assumes that a M component mixture model with component weights $P(\omega_m)$ and parameters $\boldsymbol{\theta}_m$ can represent the spectral shape. The general form of a univariate mixture model is

$$p(x|\boldsymbol{\theta}) = \sum_{m=1}^{M} P(\omega_m) p(x|\omega_m, \boldsymbol{\theta}_m) \tag{4.1}$$

where $P(\omega_m)$ is the prior probability of component m and $\boldsymbol{\theta}_m$ the component parameters. The mixture components in this case are Gaussian, and hence can be described by

$$p(x|\omega_m, \boldsymbol{\theta}_m) \quad = \quad \frac{1}{\sqrt{2\pi\sigma_m^2}} \exp\left[-\frac{(x-\mu_m)^2}{2\sigma_m^2} \right] \tag{4.2}$$

where μ_m is the mean and σ_m the standard deviation for component ω_m. The first step in estimating a GMM from the speech is to form a continuous PDF based on the spectral representation. The distance between the spectral PDF and the GMM is then minimised with respect to the GMM parameters to yield the optimal GMM parameters.

4.1.2 Forming a probability density function from the FFT bins

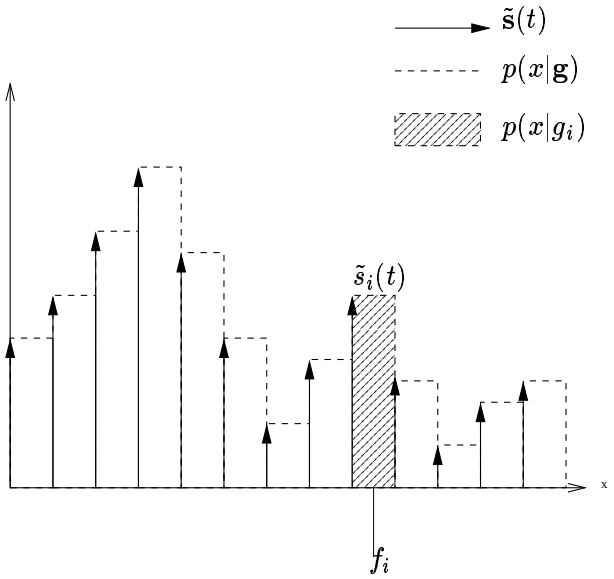


Figure 4.1 *Formation of a continuous probability density function $p(x|\mathbf{g})$ from FFT values*

As mentioned in the previous section, in order to estimate the GMM parameters from the spectrum, a PDF must be formed from the spectrum. From the N-point magnitude FFT representation of the speech $\mathbf{s}(t) = [s_1(t) \dots s_N(t)]^T$, a continuous probability density function is formed as the summation of functions based on the FFT points. Each point in the FFT $s_i(t)$ exists at a discrete frequency value. To form a continuous probability function, each point in the FFT is associated with a bin function $p(x|g_i)$[1] where g_i denotes the i^{th} bin, and x is the FFT bin frequency.

In previous work using the GMMs as a vocoder, to form a spectral histogram, each FFT frequency bin was represented by an impulse function weighted by the normalised FFT magnitude [125]. In this work, the bins have a width of 1 and are centred on the point f_i in the range $(f_i - \frac{1}{2} \leq x \leq f_i + \frac{1}{2})$, where $f_i = i + 0.5$, as shown in figure 4.1. A continuous probability density function is formed from the summation of the set of N FFT bins:

$$p(x|\mathbf{g}) = \sum_{i=1}^{N} P(g_i)p(x|g_i) \tag{4.3}$$

where $P(g_i)$ is the prior probability of the i^{th} bin g_i. The bin functions used could be trapezoids or linear interpolations of the FFT magnitudes. However, the assumption here is that the bins

[1]The time index t has been dropped for simplicity.

are simply rectangular functions centred at the value f_i:

$$p(x|g_i) = \begin{cases} 1 & (f_i - \frac{1}{2} \leq x \leq f_i + \frac{1}{2}) \\ 0 & \text{otherwise} \end{cases} \tag{4.4}$$

The prior probabilities are obtained from the normalised spectrum $\tilde{\mathbf{s}}(t)$

$$P(g_i) = \tilde{s}_i(t) \tag{4.5}$$

The normalised spectrum $\tilde{\mathbf{s}}(t)$ is computed from the points in the input FFT such that the histogram bins satisfy a sum to one constraint.

$$\tilde{s}_i(t) = \frac{s_i(t)}{\sum_{j=1}^{N} s_j(t)} \tag{4.6}$$

By forming a continous histogram in this fashion, it is possible to avoid some of the problems of data sparsity that may occur.

4.1.3 Parameter estimation criteria

Having obtained a function from the FFT bins which is a valid probability distribution from the speech spectrum, the next step is to estimate an optimal set of GMM parameters according to some criteria. The approach used in this work is to minimise the distance between the GMM and the smoothed distribution. In this case, the measure used is the Kullbeck Leibler (KL) divergence $\mathcal{D}$, where the KL distance between the two PDFs $p(x)$ and $q(x)$ can be defined as:

$$\mathcal{D}(p(x), q(x)) = \int p(x) \log \left(\frac{p(x)}{q(x)} \right) dx \tag{4.7}$$

$$= \int p(x) \log p(x) dx - \int p(x) \log q(x) dx \tag{4.8}$$

The distance between the GMM $p(x|\boldsymbol{\theta})$ and the spectral histogram $p(x|\mathbf{g})$ is given by:

$$\begin{aligned} \mathcal{D}(p(x|\mathbf{g}), p(x|\boldsymbol{\theta})) &= \int_{-\infty}^{+\infty} p(x|\mathbf{g}) \log \left(\frac{p(x|\mathbf{g})}{p(x|\boldsymbol{\theta})} \right) dx \\ &= \int_{-\infty}^{+\infty} p(x|\mathbf{g}) \log p(x|\mathbf{g}) dx - \int_{-\infty}^{+\infty} p(x|\mathbf{g}) \log p(x|\boldsymbol{\theta}) dx \end{aligned} \tag{4.9}$$

The first term does not vary with the parameters $\boldsymbol{\theta}$. The second term can be expressed as the sum of the expected value of the log likelihood over each histogram bin g_i. Since for a single bin (from equations 4.3 and 4.4):

$$\int_{-\infty}^{+\infty} P(x|g_i) \log p(x|\boldsymbol{\theta}) dx = \int_{f_i - \frac{1}{2}}^{f_i + \frac{1}{2}} P(x|g_i) \log p(x|\boldsymbol{\theta}) dx \tag{4.10}$$

$$= \mathcal{E} \left\{ \log p(x|\boldsymbol{\theta}) | g_i \right\} \tag{4.11}$$

and each bin is weighted by its probability mass $P(g_i)$, the second term in equation 4.9 can be written as the expected log likelihood of the Gaussian mixture model over all the histogram bins

$p(x|\mathbf{g})$ weighted by their prior probabilities:

$$\sum_{i=1}^{N} P(g_i) \int_{-\infty}^{+\infty} P(x|g_i) \log p(x|\boldsymbol{\theta}) dx = \sum_{i=1}^{N} P(g_i) \mathcal{E}\left\{\log p(x|\boldsymbol{\theta})|g_i\right\} \quad (4.12)$$

$$= \mathcal{E}\left\{\log p(x|\boldsymbol{\theta})|\mathbf{g}\right\} \quad (4.13)$$

The first term in equation 4.9 is dependent only upon the spectral histogram. The second term in equation 4.9 is the expected log-likelihood of the Gaussian mixture model in equation 4.13. Thus to minimise the KL distance with respect to the GMM model parameters, it is necessary to maximise the expected log-likelihood in equation 4.13.

4.1.4 GMM parameter estimation

There does not exist a closed form solution to the problem of maximising the likelihood in equation 4.13 when $p(x|\boldsymbol{\theta})$ is a GMM. However, it is possible to iteratively estimate the mixture component parameters using the expectation-maximisation (EM) algorithm. The EM algorithm is a general optimisation technique and provides a method to iteratively update model parameters such that the log-likelihood of the data is guaranteed not to decrease at each step.

One method to estimate the GMM parameters would be to use a Monte-Carlo approach and draw a sufficiently large number (D) of data points $\mathbf{x}_D = \{x_1, \ldots, x_D\}$ from the histogram, and then use the standard form of the EM algorithm for estimating GMMs from discrete datapoints - as outlined in appendix A.1 - to estimate the mixture parameters. In this case, the auxiliary function $\mathcal{Q}(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}})$ would be:

$$\mathcal{Q}(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}) = \frac{1}{D} \sum_{m=1}^{M} \left[\sum_{d=1}^{D} P(\omega_m|x_d, \boldsymbol{\theta}) \log\left(p(x_d|\omega_m, \hat{\boldsymbol{\theta}}_m)\right) \right] + \frac{1}{D} \sum_{m=1}^{M} \left[\sum_{d=1}^{D} P(\omega_m|x_d, \boldsymbol{\theta}) \log\left(\hat{P}(\omega_m)\right) \right] \quad (4.14)$$

However, as D increases, this method would become prohibitively computationally expensive for extracting speech features. Instead it is possible to make some approximations concerning the histogram data.

The first approximation made is that all the data points that are drawn from a given histogram bin g_i can be assigned the same posterior probabilities. For the data points $\mathbf{x}_d^{(g_i)}$ drawn from a given histogram bin g_i, denoted $x_d^{(g_i)}$, we can then approximate that all points drawn will share the same posterior probability:

$$P(\omega_m|x_d^{(g_i)}, \boldsymbol{\theta}) \approx P(\omega_m|g_i, \boldsymbol{\theta}) \quad (4.15)$$

Alternatively, the bins can be arbitarily sub-divided to reduce the power of this assumption, however, in practise this appears to make no difference to the GMM parameter estimates obtained.

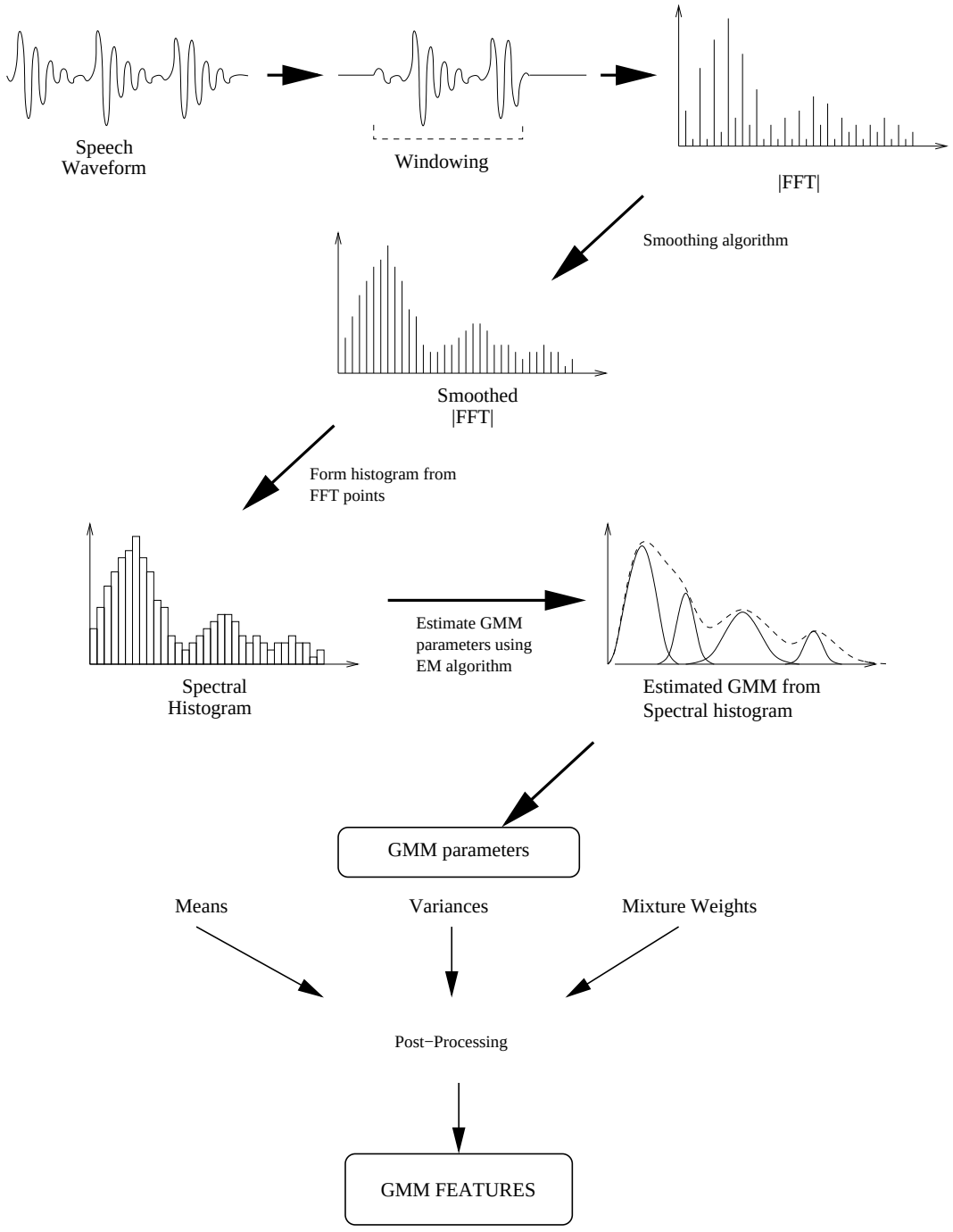


Figure 4.2 *Overview of the extraction of GMM parameters from the speech signal*

The auxiliary function can then be written:

$$\begin{aligned}\mathcal{Q}(\boldsymbol{\theta},\hat{\boldsymbol{\theta}}) &\approx \frac{1}{D}\sum_{m=1}^{M}\left[\sum_{i=1}^{N}P(\omega_m|g_i,\boldsymbol{\theta})\sum_{x_d^{(g_i)}}\log\left(p(x_d|\omega_m,\hat{\boldsymbol{\theta}}_m)\right)\right] \\ &\quad +\frac{1}{D}\sum_{m=1}^{M}\left[\sum_{i=1}^{N}P(\omega_m|g_i,\boldsymbol{\theta})\sum_{x_d^{(g_i)}}\log\left(\hat{P}(\omega_m)\right)\right] \end{aligned} \tag{4.16}$$

where the sum $\sum_{x_d^{(g_i)}}$ is taken over all the data points drawn from bin g_i. As the number of data points approaches infinity we can consider the prior probabilities of data coming from a given bin $P(g_i)$, and the expected log-likelihood of x given a mixture ω_m with parameters $\hat{\boldsymbol{\theta}}_m$:

$$\begin{aligned}\mathcal{Q}(\boldsymbol{\theta},\hat{\boldsymbol{\theta}}) &\approx \sum_{m=1}^{M}\left[\sum_{i=1}^{N}P(\omega_m|g_i,\boldsymbol{\theta})P(g_i)\mathcal{E}\left\{\log p(x|\omega_m,\hat{\boldsymbol{\theta}}_m)|g_i\right\}\right] \\ &\quad +\sum_{m=1}^{M}\left[\sum_{i=1}^{N}P(\omega_m|g_i,\boldsymbol{\theta})P(g_i)\log\left(\hat{P}(\omega_m)\right)\right] \end{aligned} \tag{4.17}$$

The expected value of the log-likelihood of the data over the bin g_i is given by:

$$\begin{aligned}\mathcal{E}\left\{\log p(x|\omega_m,\hat{\boldsymbol{\theta}}_m)|g_i\right\} &= \mathcal{E}\left\{\log\left(\frac{1}{\sqrt{2\pi\hat{\sigma}_m^2}}\right)|g_i\right\}+\mathcal{E}\left\{\left(-\frac{(x-\hat{\mu}_m)^2}{2\hat{\sigma}_m^2}\right)|g_i\right\} \\ &= \log\left(\frac{1}{\sqrt{2\pi\hat{\sigma}_m^2}}\right)-\frac{1}{2\hat{\sigma}_m^2}\left[\mathcal{E}\left\{x^2|g_i\right\}-2\mathcal{E}\left\{\hat{\mu}_m x|g_i\right\}+\mathcal{E}\left\{\hat{\mu}_m^2|g_i\right\}\right] \\ &= \log\left(\frac{1}{\sqrt{2\pi\hat{\sigma}_m^2}}\right)-\frac{1}{2\hat{\sigma}_m^2}\left[\mathcal{E}\left\{x^2|g_i\right\}-2\hat{\mu}_m\mathcal{E}\left\{x|g_i\right\}+\hat{\mu}_m^2\right] \end{aligned} \tag{4.18}$$

This expression for expected log-likelihood involves the first and second moments of the histogram bins. For a rectangular bin from $f_i-\frac{1}{2}$ to $f_i+\frac{1}{2}$ the first moment is the same as that of a single data point or impulse function:

$$\begin{aligned}\mathcal{E}\{x|g_i\} &= \int_{-\infty}^{+\infty}xp(x|g_i)dx \\ &= \int_{f_i-\frac{1}{2}}^{f_i+\frac{1}{2}}xdx \\ &= f_i \end{aligned} \tag{4.19}$$

The second moment of a histogram bin is given by:

$$\begin{aligned}\mathcal{E}\{x^2|g_i\} &= \int_{-\infty}^{+\infty}x^2p(x|g_i)dx \\ &= \int_{f_i-\frac{1}{2}}^{f_i+\frac{1}{2}}x^2dx \\ &= f_i^2+\frac{1}{12} \end{aligned} \tag{4.20}$$

Thus for observation data with a probability mass $P(g_i)$ centred at f_i, the expected value of the log-likelihood of data from a spectral histogram bin given a Gaussian component is:

$$\mathcal{E}\left\{\log p(x|\omega_m, \hat{\boldsymbol{\theta}}_m)|g_i\right\} = \log\left(\frac{1}{\sqrt{2\pi\hat{\sigma}_m^2}}\right) + \left[-\frac{(f_i - \hat{\mu}_m)^2 + \frac{1}{12}}{2\hat{\sigma}_m^2}\right] \tag{4.21}$$

This expression is similar to the likelihood of a single data point, save for the extra term of $\frac{1}{12}$ due to the variance of the rectangular bin function. This will have the effect of adding a floor to the value of expected log-likelihoods obtained. The above expression assumes that the value of $P(\omega_m|g_i, \boldsymbol{\theta}_m)$ is known. In practice, this must be estimated from the current model parameters. To calculate this exactly is not practical, as it would require an evaluation of the expected likehood of the Gaussian function $\mathcal{E}\{p(x|\omega_j, \boldsymbol{\theta}_j)|g_i\}$. Instead, the posterior probability that bin g_i was generated by component ω_j can be approximated by:

$$P(\omega_j|g_i, \boldsymbol{\theta}) \approx \frac{\hat{P}(\omega_j)\exp\left(\mathcal{E}\left\{\log\left(p(x|\omega_j, \boldsymbol{\theta}_j)\right)|g_i\right\}\right)}{\sum_{m=1}^{M}\hat{P}(\omega_m)\exp\left(\mathcal{E}\left\{\log\left(p(x|\omega_m, \boldsymbol{\theta}_m)\right)|g_i\right\}\right)} \tag{4.22}$$

The auxiliary function in equation 4.17 for the histogram is maximising with respect to $\hat{P}(\omega_j)$ and $\hat{\boldsymbol{\theta}}_j$. Differentiating equation 4.17 over $\hat{\boldsymbol{\theta}}_j$ and equating to zero, the following equation is obtained:

$$\frac{\partial\mathcal{Q}(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}})}{\partial\hat{\boldsymbol{\theta}}_j} = \sum_{i=1}^{N} P(g_i)P(\omega_j|g_i, \boldsymbol{\theta}_j)\frac{\partial}{\partial\hat{\boldsymbol{\theta}}_j}\left[\mathcal{E}\{p(x|\omega_j, \hat{\boldsymbol{\theta}}_j)|g_i\}\right] = 0 \tag{4.23}$$

Substituting equations 4.21 and and 4.22 into equation 4.23 the new parameter estimates $\hat{\mu}_j$ and $\hat{\sigma}_j^2$ are obtained:

$$\hat{\mu}_j = \frac{\sum_{i=1}^{N} P(g_i)P(\omega_j|g_i, \boldsymbol{\theta})f_i}{\sum_{i=1}^{N} P(g_i)P(\omega_j|g_i, \boldsymbol{\theta})} \tag{4.24}$$

$$\hat{\sigma}_j^2 = \frac{\sum_{i=1}^{N} P(g_i)P(\omega_j|g_i, \boldsymbol{\theta})\left[(f_i - \hat{\mu}_j)^2 + \frac{1}{12}\right]}{\sum_{i=1}^{N} P(g_i)P(\omega_j|g_i, \boldsymbol{\theta})} \tag{4.25}$$

The component priors have a sum to one constraint, and thus the new estimates can be given by considering the probability mass assigned to each component:

$$\hat{P}(\omega_j) = \frac{\sum_{i=1}^{N} P(g_i)P(\omega_j|g_i, \boldsymbol{\theta})}{\sum_{m=1}^{M}\sum_{i=1}^{N} P(g_i)P(\omega_m|g_i, \boldsymbol{\theta})} \tag{4.26}$$

The denominator is the sum over all mixtures and all bins, and hence will be equal to 1. The updated prior components can simply be written as:

$$\hat{P}(\omega_j) = \sum_{i=1}^{N} P(g_i)P(\omega_j|g_i, \boldsymbol{\theta}) \tag{4.27}$$

The EM algorithm converges upon a solution for maximising the auxiliary function by iterating two steps. The expectation (E) step is the estimation of the complete data by calculating the posterior bin probabilities. The maximisation (M) step then finds the optimal values of the model parameters to maximise the auxiliary function.

4.1.5 Initialisation

The EM algorithm is an iterative process. An important consideration is that the parameter estimates need to be initialised for the first iteration of the algorithm. The choice of initial parameters will constrain the solution found by the EM algorithm and hence is very important. There are several options for the choice of initial parameters.

The first option is to initialise the components equally across the spectrum. In this case, the component means will be distributed evenly and the component priors will be equal. The choice of the initial variances is also important. If the initial variances are small, the components will be more strongly constrained to the initial positions. Setting the variances too large will allow too much variation in the locations and may not yield a useful solution that represents the spectral PDF. For the initialisation of an M-component GMM when estimating a N-point histogram $p(x|\mathbf{g})$, small initial experiments suggested that good initialisation values for each component m in the GMM will be:

$$\mu_m = \frac{N(m+0.5)}{M} + 0.5 \tag{4.28}$$

$$\sigma_m^2 = \frac{N^2}{M^2} \tag{4.29}$$

$$P(\omega_m) = \frac{1}{M} \tag{4.30}$$

Another option used in previous work is to use the parameters from the final iteration of the previous frame as the initial parameters for the current frame [126]. It would also be possible to use the parameters estimates from another peak-picking algorithm to initialise the system. However, if the estimates from the other system are poor, the solutions found by the EM algorithm may also be poor estimates.

4.2 Issues in estimating a GMM from the speech spectrum

In the previous section the theory for estimating GMM parameters from a speech spectrum was presented. In this section, a number of issues with the implementation of the algorithm on a speech spectrum are examined.

4.2.1 Spectral smoothing

The characteristic shape of the speech spectrum can present problems for estimating a set of Gaussian components. The voiced speech spectrum is characterised by a number of pitch peaks separated by the fundamental frequency. As mentioned in the previous section, the choice of initial parameters can determine the maxima found by the EM algorithm. There exist many local maxima the EM algorithm could find at each of these pitch peaks. If the pitch peaks are separated by a high fundamental frequency, a maximum could be found estimating a Gaussian component to a single pitch peak, and ignoring the adjacent harmonics. The Gaussian component which models the harmonic will have a very small variance, but will not represent the general spectral

envelope, as seen in figure 4.3. In this figure two of the Gaussian components have converged upon the pitch peaks and are not modelling the general spectral shape. In order to represent the phonetic class of the speech spectrum it is desirable that the GMM model the spectral envelope and avoid the problem of components converging upon the pitch peaks.

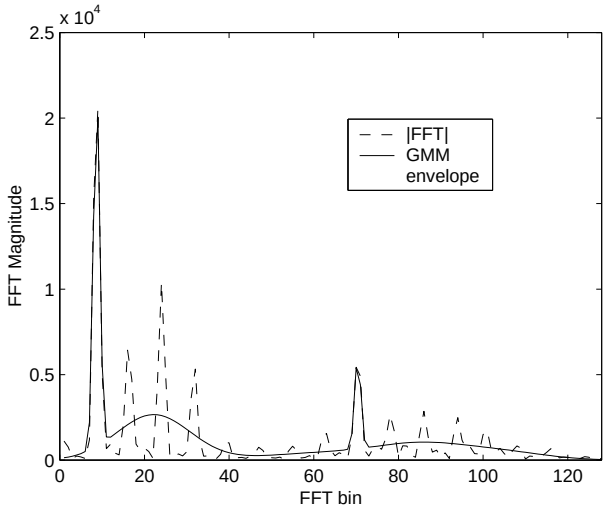


Figure 4.3 *EM algorithm finding a local maximum representing the pitch peaks in voiced speech*

There are two possible solutions to the problem of the Gaussians components representing the spectral harmonics:

- **Variance flooring**: applying a variance floor to the Gaussian components prevents any component becoming too narrow and representing only a single pitch peak.
- **Spectral smoothing**: using a smoothing algorithm to remove the pitch or voicing from the spectrum and estimate the vocal tract function.
- **Overlapping bin functions**: rather than the FFT bins being represented by a rectangular histogram function, the bins could be allowed to overlap each other. This technique can be related to a spectral smoothing approach.

The problem with using a variance floor is that the floor would have to be set suitably large to account for the highest possible fundamental frequency between the pitch periods in the spectrum. With a large variance floor the system will not be able to model narrow formants particularly well. In addition an unsmoothed spectrum is not modelled by a Gaussian mixture model very well.

In this work, the spectrum is smoothed prior to the GMM estimation. Three different forms of spectral smoothing which can be applied to the spectrum are presented below: cepstral liftering, estimating the SEEVOC envelope and applying a convolutional pitch filter.

4.2.1.1 Cepstral liftering

The cepstral representation of a spectrum is obtained by taking the inverse FFT of the log-spectrum. The cepstra can be approximated by taking the discrete cosine transform (DCT) of the log spectrum [17], as in equation 3.3.

The lower order cepstra represent the general spectral envelope and the higher order cepstra represent the pitch and voicing information. For a typical voiced speech signal, the cepstral representation will have most of the energy in the lower cepstral region with a single peak in the upper cepstra. The higher order cepstra can be removed by truncating the cepstral coefficients after a given point. After removing the higher order coefficients, the remaining cepstra can be used to reconstruct the spectral shape with the source removed. The spectrum is reconstructed by taking the inverse DCT of the exponents of the cepstra.

One possible problem is that the response of the cepstra is constrained by the maximum possible fundamental frequency set in the system. Thus, the frequency resolution of the reconstructed spectrum and the definition of the spectral envelop will be limited by the maximum fundamental frequency chosen.

4.2.1.2 SEEVOC envelope

The Spectral Envelope Estimate Vocoder (SEEVOC) is a sinusoidal model-based low bit rate codec [86]. The method has also been used in analysis/synthesis systems [94]. In this scheme, the spectral envelope is estimated by detecting the peaks in the speech spectrum and interpolating between them. The pitch peaks are found in the regions defined by multiples of the fundamental frequency f_0 in the spectral domain. The locations and magnitudes of the pitch peaks thus obtained are linearly interpolated to form an estimate of the spectral envelope function.

Given an estimate of the pitch f_0, the interval $[f_0/2, 3f_0/2]$ is searched to find the location and amplitude of the largest peak, denoted (A_1, f_1). The pitch intervals $[f_{l-1}+f_0/2, f_{l-1}+3f_0/2]$ are then searched to find the l^{th} peak amplitudes A_l and locations f_l. If no peak is found, the search is continued from the highest amplitude point in the region. The values obtained from the search are linearly interpolated and used to form the spectral envelope.

The pitch can be estimated by searching for a peak in the short-term autocorrelation function $\lambda(\tau)$ over the W width speech signal $w(n)$:

$$\lambda(\tau) = \frac{1}{W}\sum_{n=1}^{W} w(n)w(n+\tau) \tag{4.31}$$

The autocorrelation function indicates the degree of linear relationship between points separated by period τ in a sample waveform $w(n)$. The pitch period is estimated as the period of the largest peak in the autocorrelation function within the thresholds for minimum and maximum pitch periods, typically 30Hz and 400Hz respectively. The SEEVOC technique does not require an exact measurement of the pitch and doesn't require harmonic peaks whilst searching. It is thus reasonably robust.

This method of spectral smoothing has been used in preference to cepstral smoothing prior to estimating GMM parameters for speech coding [126]. The SEEVOC algorithm is useful in combination with a sinusoidal based coder, since the magnitudes of the pitch peaks are preserved, as is the quality of the coded speech.

4.2.1.3 Convolutional pitch filter

Another technique to remove voicing from the spectrum is to convolve the spectrum with a function to smooth the pitch periods. The width of the cosine filter n_{filt} is based directly on the pitch period n_{pitch} of the speech and the number of points N in the FFT

$$n_{filt} = (\frac{2N}{n_{pitch}} + 0.5) - 1 \tag{4.32}$$

where the pitch period n_{pitch} can be estimated by searching for the peak in the autocorrelation function as detailed in section 4.2.1.2. The convolutional pitch filter $\mathbf{h} = [h_1, \ldots, h_{n_{filt}}]^T$ is then defined as:

$$h_j = 0.5 - 0.5 \cos\left(2\pi \frac{j + 0.5}{n_{filt}}\right) \tag{4.33}$$

Hence the smoothed power spectrum $\hat{\mathbf{s}}(t)$ is given by:

$$\hat{s}_i(t) \quad = \quad \sum_{j=-n_{filt}}^{n_{filt}} h_j s_{i-j}(t) \tag{4.34}$$

The approach of using a convolutional pitch filter does not require an exact measurement of the pitch to set the width of the raised cosine window. It is thus also reasonably robust to the estimate of pitch. Convolution operations in the spectral domain are equivalent to multiplication in the temporal domain. Thus, the convolution with the raised cosine window in the spectral domain is equivalent to using a windowing function with a width based on the pitch period in the temporal domain. The pitch filtered spectrum will be smooth and is effectively based on fewer points (or a shorter window) of the source waveform.

This approach is closely related to the functions formed from the FFT bins as well. A similar technique could be implemented if the histogram bin functions $p(x|g_i)$ were represented by a raised cosine function instead.

4.2.2 Prior distributions

The technique for estimating a set of Gaussian components from a spectral histogram in section 4.1 places no prior constraint on the parameters extracted. However, small changes in the spectral histogram could lead to large changes in the solutions found by the EM algorithm. It is possible that the estimated parameters could vary greatly between two frames that appear similar. Some statistical spectral peak representations have constraints, or priors, on the locations of the peaks modelled. The locations of the gravity centroid parameters are explicitly constrained as each centroid provides information specific to a particular pre-defined spectral sub-band [84].

The HMM-2 system incorporates the location of the current frequency bin as a feature when estimating the second spectral HMM giving a class-dependent prior on the location of the spectral peaks [108].

It may be useful to be able to incorporate some form of constraint on the extraction of GMM parameters. The EM algorithm has been modified to use prior information on the data in the form of penalised log-likelihoods [45]. It would be possible to apply a component-dependent prior weighting to the data before the EM Gaussian estimates, effectively filtering the data observed by each component during the estimation process. Using a prior model in this fashion places a constraint on the observations for each mixture in a similar fashion to that used in the gravity centroid work. The EM algorithm can be constrained to give Gaussian parameters for the sub-bands the same as the gravity centroid system by fixing the component posterior probabilities $P(\omega_j|g_i,\theta_j)$ for each bin g_i to a sub-band filter function. Penalising or weighting the prior probabilities $P(g_i)$ or the bin functions on a per-component basis with a sub-band function will place a similar constraint on the auxiliary function of equation 4.17 and the maximisation function equation 4.23.

Weighting or penalising the data prior to the EM estimation would strongly constrain the locations of the components. Hence, applying a prior distribution to the data in this form will limit the range of the parameters estimated. Using a prior distribution can limit the flexibility of the technique to model the spectrum and reduce its usefulness in a recognition system. However, it may be helpful to weakly constrain the mixture components to certain regions in the spectrum, in order to maintain smoother trajectories and smaller discontinuities.

This section presents a method of including a prior distribution using a different technique to modify the log-likelihoods. A prior distribution is added directly to each of the components in the estimated model distributions. Effectively, the prior distribution will augment the data set of the histogram on a per-component basis. This approach gives a form of count smoothing on the EM estimation process. The prior distribution $\boldsymbol{\theta}^{(p)}$ of each of the components are modelled by Gaussians:

$$\boldsymbol{\theta}^{(p)} = \{\mu_1^{(p)}, \ldots, \mu_M^{(p)}, \sigma_1^{(p)2}, \ldots, \sigma_M^{(p)2}, P(\omega_1^{(p)}), \ldots, P(\omega_M^{(p)})\}$$

In this implementation, the prior distribution augments the observation set. The observations will either come from the spectral histogram, or from the prior distribution. The motivation is that the contribution from the prior distribution is constant over all frames. In spectral frames where the local maxima are found that differ greatly from similar frames, the prior distribution will alter the maxima found. Thus, it is hoped to obtain parameter estimates with smoother trajectories and fewer discontinuities.

The prior probabilities of the bins in the spectral histogram are discounted to ensure that the sum of the histogram bins and of the prior probability mixtures are one:

$$(\eta) \sum_{m=1}^{M} P(\omega_m^{(p)}) + (1-\eta) \sum_{i=1}^{N} P(g_i) = 1 \tag{4.35}$$

where η is defined as the discounting of the histogram probabilities due to the prior probabilities and is in the range 0 to 1. With η set to be 0, the prior distribution gives no contribution to the estimated parameterd.. When the prior weight is set to 1, the estimated parameters are based solely on the prior distribution.

The following expected log-likelihood is optimised:

$$\begin{aligned}\mathcal{E}\{\log p(x|\boldsymbol{\theta})|\mathbf{g},\boldsymbol{\theta}^{(p)}\} &= (1-\eta)\sum_{i=1}^{N} P(g_i)\mathcal{E}\left\{\log p(x|\boldsymbol{\theta})|g_i\right\} \\ &+ \eta\sum_{m=1}^{M} P(\omega_m^{(p)})\mathcal{E}\left\{\log p(x|\boldsymbol{\theta}_m)|\omega_m^{(p)}\right\} \end{aligned} \tag{4.36}$$

The auxiliary function can then be altered to take into account the two distributions, with data either being drawn from the spectral histogram bins or the prior components. The data points x drawn from a given bin g_i can be denoted $x_d^{(g_i)}$ and the data drawn from a prior distribution component $\omega_j^{(p)}$ are $x_d^{(\omega_j)}$. As in equation 4.15, the approximation is made that all data drawn from the same histogram bin or prior distribution has the same prior distribution. Following from equations 4.16 and 4.17 the auxiliary function with a prior distribution may be written:

$$\begin{aligned}\mathcal{Q}(\boldsymbol{\theta},\hat{\boldsymbol{\theta}}) &= \frac{1}{D}\sum_{m=1}^{M}\left[\sum_{i=1}^{N} P(\omega_m|g_i,\boldsymbol{\theta})\sum_{x_d^{(g_i)}}\log\left(p(x_d^{(g_i)}|\omega_m,\hat{\boldsymbol{\theta}}_m)\right)\right] \\ &+\frac{1}{D}\sum_{m=1}^{M}\left[\sum_{j=1}^{M} P(\omega_m|\omega_j^{(p)},\boldsymbol{\theta})\sum_{x_d^{(\omega_j)}}\log\left(p(x_d^{(\omega_j)}|\omega_m,\hat{\boldsymbol{\theta}}_m)\right)\right] \\ &+\frac{1}{D}\sum_{m=1}^{M}\left[\sum_{i=1}^{N} P(\omega_m|g_i,\boldsymbol{\theta})\sum_{x_d^{(g_i)}}\log\left(\hat{P}(\omega_m)\right)\right] \\ &+\frac{1}{D}\sum_{m=1}^{M}\left[\sum_{j=1}^{M} P(\omega_m|\omega_j^{(p)},\boldsymbol{\theta})\sum_{x_d^{(\omega_j)}}\log\left(\hat{P}(\omega_m)\right)\right] \end{aligned} \tag{4.37}$$

Using the same approach from section 4.1, taking the limit as the number of data points approaches infinity ($D\rightarrow\infty$) we can consider the prior probabilities that an observation came

from a given bin $(1-\eta)P(g_i)$ or from a component in the prior distribution $\eta P(\omega_m^{(p)})$:

$$\begin{aligned}\mathcal{Q}(\boldsymbol{\theta},\hat{\boldsymbol{\theta}}) &= \sum_{m=1}^{M}\left[\sum_{i=1}^{N} P(\omega_m|g_i,\boldsymbol{\theta})(1-\eta)P(g_i)\mathcal{E}\left\{\log\left(p(x|\omega_m,\hat{\boldsymbol{\theta}}_m)\right)|g_i\right\}\right] \\ &+ \sum_{m=1}^{M}\left[\sum_{j=1}^{M} P(\omega_m|\omega_j^{(p)},\boldsymbol{\theta})\eta P(\omega_j^{(p)})\mathcal{E}\left\{\log\left(p(x|\omega_m,\hat{\boldsymbol{\theta}}_m)\right)|\omega_j^{(p)}\right\}\right] \\ &+ \sum_{m=1}^{M}\left[\sum_{i=1}^{N} P(\omega_m|g_i,\boldsymbol{\theta})(1-\eta)P(g_i)\log\left(\hat{P}(\omega_m)\right)\right] \\ &+ \sum_{m=1}^{M}\left[\sum_{j=1}^{M} P(\omega_m|\omega_j^{(p)},\boldsymbol{\theta})\eta P(\omega_j^{(p)})P(\omega_m^{(p)})\log\left(\hat{P}(\omega_m)\right)\right] \end{aligned} \tag{4.38}$$

The posterior probabilities of data coming from a given bin are the same as in 4.1. The posterior probabilities of each prior component are fixed to the corresponding Gaussian mixture component being estimated from the histogram:

$$P(\omega_m|\omega_j^{(p)},\boldsymbol{\theta}) = 1(m=j) \tag{4.39}$$

$$P(\omega_m|\omega_j^{(p)},\boldsymbol{\theta}) = 0(m\neq j) \tag{4.40}$$

The expected log-likelihood of the bins can be calculated in a similar fashion to equation 4.18 to obtain the expected log-likelihood of a component ω_j over $\omega_i^{(p)}$:

$$\begin{aligned}\mathcal{E}\left\{\log p(x|\omega_j,\boldsymbol{\theta}_j)|\omega_i^{(p)}\right\} &= \log\left(\frac{1}{\sqrt{2\pi\hat{\sigma}_m^2}}\right) \\ &- \frac{1}{2\hat{\sigma}_m^2}\left[\mathcal{E}\left\{x^2|\omega_i^{(p)}\right\} - 2\hat{\mu}_m\mathcal{E}\left\{x|\omega_i^{(p)}\right\} + \hat{\mu}_m^2\right]\end{aligned} \tag{4.41}$$

The first and second moments of the prior distributions are given by:

$$\mathcal{E}\left\{x|\omega_j^{(p)}\right\} = \mu_j^{(p)} \tag{4.42}$$

$$\mathcal{E}\left\{x^2|\omega_j^{(p)}\right\} = \left[\mu_j^{(p)2} + \sigma_j^{(p)2}\right] \tag{4.43}$$

and substituting these into equation 4.41 gives:

$$\mathcal{E}\left\{\log p(x|\omega_j,\boldsymbol{\theta}_j)|\omega_i^{(p)}\right\} = \log\left(\frac{1}{\sqrt{2\pi\sigma_j^2}}\right) - \left[\frac{(\mu_i^{(p)}-\mu_j)^2+\sigma_i^{(p)2}}{2\sigma_j^2}\right] \tag{4.44}$$

By differentiating the auxiliary function with respect to the parameter updates and equating to zero, the estimates of the new parameters are obtained:

$$\hat{\mu}_j = \frac{\eta P(\omega_j^{(p)})\mu_j^{(p)} + (1-\eta)\sum_{i=1}^{N} P(g_i)P(\omega_j|g_i,\boldsymbol{\theta})f_i}{\eta P(\omega_j^{(p)}) + (1-\eta)\sum_{i=1}^{N} P(g_i)P(\omega_j|g_i,\boldsymbol{\theta})} \tag{4.45}$$

$$\hat{\sigma}_j^2 = \frac{\eta P(\omega_j^{(p)})\left[\mu_j^{(p)2}+\sigma_j^{(p)2}\right] + (1-\eta)\sum_{i=1}^{N} P(g_i)P(\omega_j|g_i,\boldsymbol{\theta}_j)\left[(f_i^2-\hat{\mu}_j^2)+\frac{1}{12}\right]}{\eta P(\omega_j^{(p)}) + (1-\eta)\sum_{i=1}^{N} P(g_i)P(\omega_j|g_i,\boldsymbol{\theta}_j)} \tag{4.46}$$

$$\hat{P}(\omega_j) = \eta P(\omega_j^{(p)}) + (1-\eta)\sum_{i=1}^{N} P(g_i)P(\omega_i|g_i,\boldsymbol{\theta}_j) \tag{4.47}$$

The use of the prior distribution allows a form of count smoothing to be utilised when extracting the GMM features. The prior weight η can be set to vary the contribution of the prior distribution.

4.3 Temporal smoothing

One of the problems associated with extracting formants from a spectrum, as mentioned in section 3.3.2, is that the formants cannot be consistently estimated [54]. As the formants are representative of the underlying articulator movements, they are expected to have mostly smooth trajectories when present [59]. However, inconsistencies in labelling formants or spectral peaks can lead to discontinuities in the formant tracks, giving noise in the extracted observations. Applying continuity constraints during the extraction process has been used to improve the robustness of other formant estimation algorithms [98] [103]. The process for extracting GMM features outlined in section 4.1 uses no frame to frame constraints on the locations of the Gaussian components. It may be desirable to impose some form of continuity constraint upon the features to ensure smooth trajectories are estimated. One simple implementation would be to use a moving average filter over the extracted parameters. However, applying a moving average filter would lead to a loss of temporal resolution, and is a relatively crude implementation. Another technique is to use the surrounding speech frames when estimating Gaussians from the spectral data. Rather than estimating parameters from the single dimensional histogram from the smoothed speech spectrum, a set of two-dimensional parameters can be estimated. The frames adjacent to the current spectrum can be used to make a 2-D histogram and the Gaussian components then form a two-dimensional representation of the data. This method is outlined in this section.

4.3.1 Formation of 2-D continuous probability density function

As in the previous technique on single dimensional data, the first step is to form a continuous PDF from the speech spectrum. The frames $\{\mathbf{s}(t-F),\ldots,\mathbf{s}(t+F)\}$ are used, where F is the frame width. Again, a normalised version of the spectrum $\tilde{s}(t)$ is used where:

$$\sum_{i=1}^{N}\sum_{\tau=-F}^{F}\tilde{s}_i(t+\tau)=1 \tag{4.48}$$

and a set of 2-dimensional histogram bins g_{ij} can be assembled with prior probabilities:

$$P(g_{ij}(t))=\tilde{s}_i(t-F+j) \tag{4.49}$$

The bin functions in this case are rectangular and are given by:

$$p(x,t|g_{ij})=\begin{cases}1 & (f_i-\frac{1}{2}\leq x\leq f_i+\frac{1}{2}) \text{ and } (j-1\leq t\leq j)\\ 0 & \text{(otherwise)}\end{cases} \tag{4.50}$$

where f_i is the centre frequency as before. Hence the 2-D probability distribution is thus given over the full set of bins $\mathbf{G}$:

$$p(x,t|\mathbf{G}) = \sum_{i=1}^{L} p(x,t|\mathbf{g}_i) \tag{4.51}$$

$$= \sum_{i=1}^{L}\sum_{j=1}^{N} P(g_{ij}(t))p(x,t|g_{ij}) \tag{4.52}$$

where

$$\mathbf{G} = \{\mathbf{g}_1,\ldots,\mathbf{g}_L\} \tag{4.53}$$

$$\mathbf{g}_l = [g_{l1},\ldots,g_{lN}]^T \tag{4.54}$$

and $L = 2F+1$. An example of the formation of a 2-D PDF can be shown in figure 4.4. The central frame is from the current time indexand the two frames preceeding and following the current frame are incorporated into the PDF.

4.3.2 Estimation of GMM parameters from 2-D PDF

The estimation of the GMM parameters proceeds in a similar fashion as for the 1-D case outlined previously. The approximation that all points drawn from a bin can be assigned with the same posterior probabilities is made again, so that a set of two-dimensional Gaussians may be estimated from the spectral histogram using the EM algorithm as before. The auxiliary function is similar to that of equation 4.17, but is taken over all the 2-D PDF bins:

$$\mathcal{Q}(\boldsymbol{\theta},\hat{\boldsymbol{\theta}}) \approx \sum_{m=1}^{M}\left[\sum_{i=1}^{N}\sum_{j=1}^{L} P(\omega_m|g_{ij},\boldsymbol{\theta})P(g_{ij})\mathcal{E}\left\{\log(x|\omega_m,\hat{\boldsymbol{\theta}}_m)|g_{ij}\right\}\right] + \sum_{m=1}^{M}\left[\sum_{i=1}^{N}\sum_{j=1}^{L} P(\omega_m|g_{ij},\boldsymbol{\theta})P(g_{ij})\log\left(\hat{P}(\omega_m)\right)\right] \tag{4.55}$$

The posterior probabilities of the components given the 2-D bin functions, $P(\omega_m|g_{ij},\boldsymbol{\theta})$, can be calculated from the expected bin log-likelihoods:

$$\mathcal{E}\left\{\log p(x|\omega_m,\hat{\boldsymbol{\theta}}_m)|g_{ij}\right\} = \log\left(\frac{1}{(2\pi)^{\frac{1}{2}}|\hat{\Sigma}_m|^{\frac{1}{2}}}\right) + \left[-\frac{1}{2}(f_i-\hat{\boldsymbol{\mu}}_m)^T\hat{\Sigma}_m^{-1}(f_i-\hat{\boldsymbol{\mu}}_m)+\frac{1}{12}\right] \tag{4.56}$$

and the priors can be calculated in the same fashion as the 1-D case in equation 4.22. The means of the second (temporal) dimension are fixed to the current (or central) temporal frame. This constraint is imposed to ensure that the features extracted are modelling the current input frame. Thus for a mixture component m:

$$\boldsymbol{\mu}_m = \begin{bmatrix} \mu_m 1 \\ 0 \end{bmatrix} \tag{4.57}$$

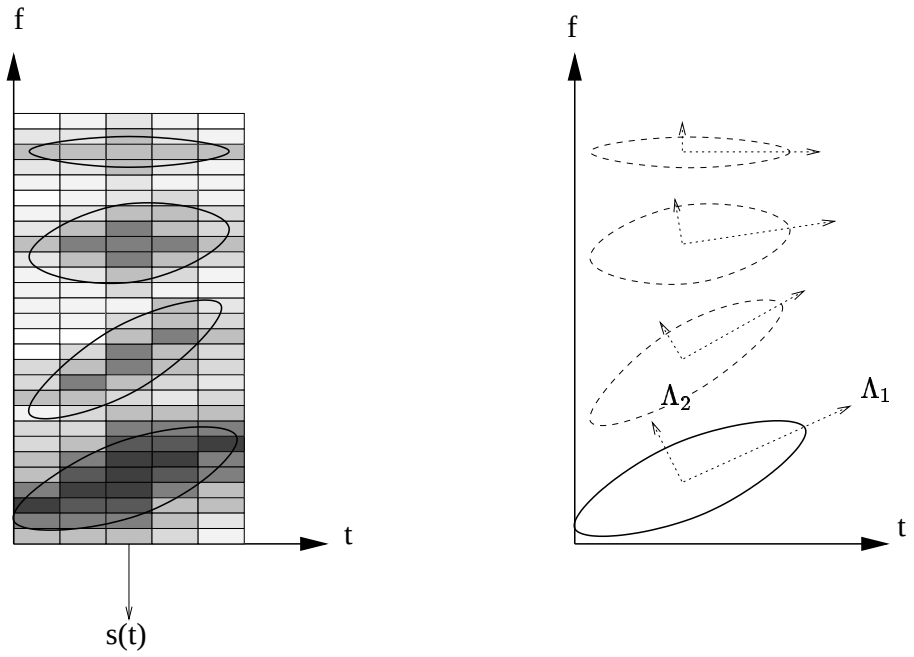


Figure 4.4 *Estimating Gaussians in two dimensions, and extracting eigenvectors of the covariance matrices*

In addition, the frames adjacent to the input frame can be de-weighted with a scale or windowing function $h_i(\tau)$

$$\tilde{s}_i(t+\tau) = h_w(\tau)s_i(t+\tau) \tag{4.58}$$

where $h_i(\tau)$ could be a raised cosine window or a triangular function, for example. The motivation for windowing in the temporal domain is twofold. First, by deweighting the surrounding frames, the parameters estimated are more dependent on the current frame. Second, by deweighting the data at the edges, the histograms will be better modelled by a set of Gaussian components as the distributions for Gaussian data are expected to tail off at the extremities.

4.3.3 Extracting parameters from the 2-D GMMs

The 2-D GMM estimates yields a similar set of parameters to that estimated for the 1-D GMM. From the two-dimensional Gaussian estimates a set of component priors, two dimensional means and two-dimensional covariance matrices are obtained.

The spectral dimension component means μ_1 can be used in the same fashion as the single dimensional GMM means. The component means in the temporal direction μ_2 are constrained to the central frame and thus will not vary over the spectral frames. The component priors can also be used in the same fashion as the 1-D component priors.

The covariance matrices could be used in two ways. The first approach would be to use the covariance element of the frequency dimension Σ_{11} in the same way as the variance in the single dimensional case is used. The second method is be to extract additional information from

the extra terms present in the full matrix. The full two-dimensional covariance matrix contains information about the correlations between the successive frames. By extracting the eigenvectors of the covariance matrix, it is possible to analyse the covariance of each mixture m in terms of the set of eigenvectors $\mathbf{U}_m = \{\mathbf{u}_{m1}, \mathbf{u}_{m2}\}$ and eigenvalues $\Lambda_m = \{\lambda_{m1}, \lambda_{m2}\}$. The directions of the eigenvector will be represent the degree of temporal correlation. As shown in figure 4.4, the eigenvectors can represent the trajectory of the Gaussian components. The eigenvector components give an indication of the velocity of the GMM parameters. The eigenvalues could be used as dynamic parameters rather than using the linear regression parameters used for conventional features. The scalar product of the eigenvector and the observation in one direction will yield a term representing the variance in the spectral dimension in the transformed space:

$$\sigma'_{m1} = \mathbf{u}_{m1} \cdot \begin{bmatrix} \Sigma_{11} \\ \Sigma_{21} \end{bmatrix} \tag{4.59}$$

The scalar product of the other elements of the covariance matrix with the second eigenvector will yield a term representing the variance in the temporal dimension which could be used in a similar fashion to the standard dynamic parameters for the position of the component:

$$\sigma'_{m2} = \mathbf{u}_{m2} \cdot \begin{bmatrix} \Sigma_{21} \\ \Sigma_{22} \end{bmatrix} \tag{4.60}$$

Thus an extra feature modelling the temporal correlations can be extracted from the 2-D GMM parameters.

4.4 Properties of the GMM parameters

The previous sections have outlined how to extract a set of GMM parameters to model a PDF formed from the speech spectrum. This section discusses the properties of the extracted parameters. The formant-like properties of the GMM parameters are discussed, along with their use as features for speech recognition. A measure of confidence in the extracted parameters is presented, together with a framework for its use in medium- and large-vocabulary systems. Finally, approaches for speaker normalisation of the GMM features are discussed.

4.4.1 Gaussian parameters as formant-like features

The GMM parameters can be considered to be analogous to a set of formant-like features [125]. The component means correspond to the formant locations, the standard deviations to the formant bandwidths and the component energies to the amplitudes. Once the GMM parameters have been extracted, they are ordered according to their frequency values. Thus the component with the lowest mean is the first component and so forth.

An example speech frame is shown in figure 4.5(a). The spectral envelope for a frame smoothed by cepstral deconvolution and the associated four component GMM estimates are

shown in figure 4.5(b). The speech comes from a section of a voiced vowel utterance and thus has a characteristic formant peak structure. The cepstral deconvolution has managed to remove the effects of the voicing from the speech. The GMM manages to represent the general spectral shape and follows the locations of the spectral peaks reasonably well. In addition, the overal structure of the spectrum is well modelled.

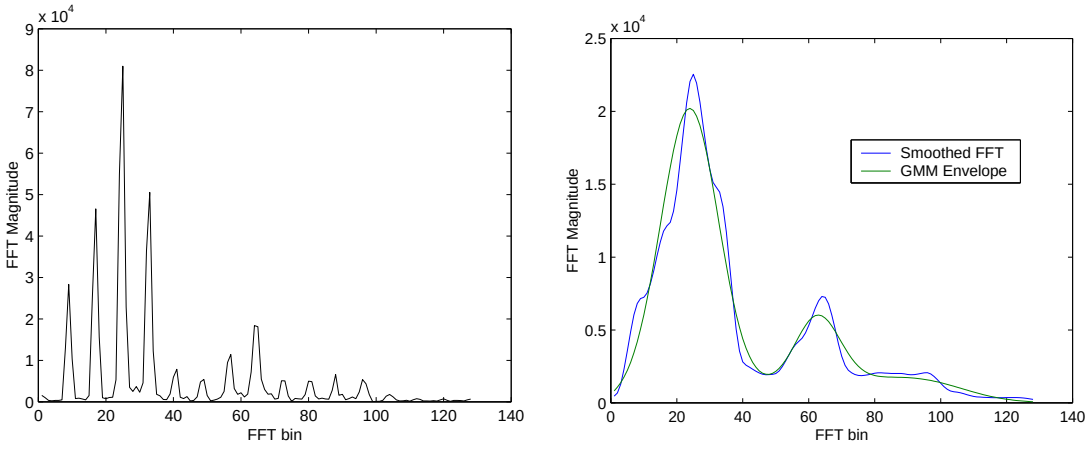


Figure 4.5 *Example plots showing envelope of Gaussian Mixture Model multiplied by spectral energy*

In figure 4.6 the Gaussian component means for each of the spectral representations have been plotted over the spectrogram for the all-voiced utterance "Where were you while we were away?" which possesses strong formant structures. The component means follow the observable formant structures in the speech. No frame-to-frame constraints were used to extract the parameters. Despite this, the trajectories of the component means are fairly smooth. During the silence periods the positions of the means vary slightly but mostly they stay close to their initialisation points. The mixtures do cross the boundaries of the spectral sub-bands they were initialised in and exhibit a large degree of freedom in their locations across the spectrum. This shows the flexibility the GMM features possess over the gravity centroid parameters.

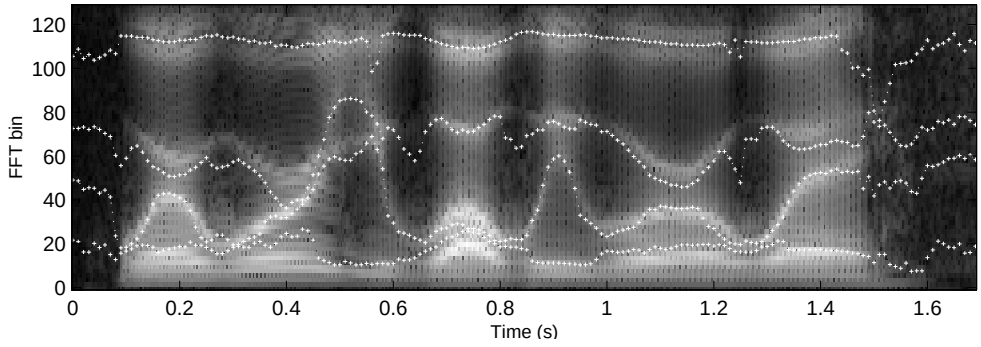


Figure 4.6 *Gaussian mixture component mean positions fitted to a 4kHz spectrum for the utterance "Where were you while we were away?", with four Gaussian components fitted to each frame.*

4.4.2 Extracting features from the GMM parameters

In section 3.2, a number of desirable properties of features for speech recognition were mentioned. Features should represent the underlying phonetic classes in a distinct fashion, and be separated in the feature space. The features should be well represented by the output distributions. They should be as speaker independent as possible. Ideally, to reduce the number of parameters to estimate, the features should be compact and uncorrelated.

The Gaussian parameters estimated by the EM algorithm are the means, variances and component priors. It is possible to apply post-processing to these features before using them in a speech recognition system, in order to yield features which better represent the phonetic class or are better represented by the emission probability distributions in the HMM states. In this case, the HMM output PDFs will also be Gaussian.

Scaling the component priors by the spectral energies yields the component energies. The component energies or log-energies $e_m(t)$ may be preferable to the component priors as the component log-energies contain spectral amplitude information required to distinguish certain types of phone which may have similar peak locations. This was the approach used when the spectral GMM was used as a vocoder. It is worth noting however that most spectral features (PLP and MFCC for instance) remove the spectral energy before calculating the parameters.

$$e_m(t) = \log[P(\omega_m) \sum_{i=1}^{N} s_i(t)] \tag{4.61}$$

If the component energies are used, then the log scale is a more appropriate domain for energy terms as it will compress the dynamic range of the parameters [122]. The log-energies can also be normalised on a per-utterance basis, with the maximum component log-energy term normalised to 1 and a silence floor can also be applied 50dB below this to all components, resulting in the range of component energies being $(1, -10.53)$. The maximum energy is defined as $\max_t e_m(t)$ and the normalised log energies $\tilde{e}_m(t)$ for each component are given by:

$$\tilde{e}_m(t) = \begin{cases} -10.53 & \text{if } e_m(t) < \max_t e_m(t) - 10.53 \\ 1.0 - (\max_t e_m(t) - e_m(t)) & \text{otherwise} \end{cases} \tag{4.62}$$

Alternatively, the energy amplitudes at the locations of the means may be used as features instead. The peak amplitudes may be preferable in the cases where two peaks are close together or the estimates are inconsistent and the component energy or prior does not represent the spectral peak.

Another consideration is the distribution of the parameters. The means and standard deviations will be constrained to be positive, and hence will not be Gaussian distributed. However, in most cases, the range of the parameter values in the distribution will tail off before reaching zero. However, applying a silence floor to the log-energy features will change the distribution of the extracted features. All of the log-energy features which would have been below the floor will now appear in the distribution at this point causing a peak in the distribution at the silence floor. Hence the log-energies may well not be Gaussian distributed. Also, the component mean

	μ_1	μ_2	μ_3	μ_4	σ_1	σ_2	σ_3	σ_4	$\tilde{e}_1$	$\tilde{e}_2$	$\tilde{e}_3$	$\tilde{e}_4$	r(t)
μ_1	1.00	0.34	0.12	0.24	0.86	0.26	-0.04	-0.21	-0.12	0.06	0.10	0.17	0.04
μ_2	0.34	1.00	0.58	0.35	0.46	0.86	-0.05	-0.26	-0.37	-0.28	0.00	0.09	-0.20
μ_3	0.12	0.58	1.00	0.41	0.28	0.59	0.26	-0.44	-0.38	-0.41	-0.17	0.06	-0.28
μ_4	0.24	0.35	0.41	1.00	0.37	0.31	0.14	-0.75	-0.43	-0.37	-0.36	-0.18	-0.34
σ_1	0.86	0.46	0.28	0.37	1.00	0.41	0.02	-0.32	-0.43	-0.25	-0.17	-0.08	-0.26
σ_2	0.26	0.86	0.59	0.31	0.41	1.00	-0.08	-0.23	-0.43	-0.40	-0.11	-0.02	-0.30
σ_3	-0.04	-0.05	0.26	0.14	0.02	-0.08	1.00	-0.20	-0.23	-0.23	-0.36	-0.22	-0.27
σ_4	-0.21	-0.26	-0.44	-0.75	-0.32	-0.23	-0.20	1.00	0.33	0.26	0.23	0.00	0.23
$\tilde{e}_1$	-0.12	-0.37	-0.38	-0.43	-0.43	-0.43	-0.23	0.33	1.00	0.95	0.89	0.81	0.96
$\tilde{e}_2$	0.06	-0.28	-0.40	-0.37	-0.25	-0.40	-0.23	0.26	0.95	1.00	0.91	0.84	0.97
$\tilde{e}_3$	0.10	0.00	-0.17	-0.36	-0.17	-0.11	-0.36	0.23	0.89	0.91	1.00	0.94	0.96
$\tilde{e}_4$	0.17	0.09	0.06	-0.18	-0.08	-0.02	-0.22	0.00	0.81	0.84	0.94	1.00	0.91
r(t)	0.04	-0.20	-0.28	-0.34	-0.26	-0.30	-0.27	0.23	0.96	0.97	0.96	0.91	1.00

Table 4.1 *Correlation matrix for a 4 component GMM system features taken from TIMIT database*

features are ordered by their location. Hence, the distribution will be constrained as a higher formant cannot have a value of frequency lower than the one below it.

The GMM features from the EM algorithm tend to have high degrees of correlation between the features compared to MFCC features. The correlation coefficient matrix for the GMM features for the Resource Management data for a four component GMM system estimated from a 4kHz spectrum is presented in table 4.1. The spectral log-energy feature for the frame, $r(t)$, is also shown in the table. The component mean position features are strongly correlated both with each other and with the corresponding standard deviation feature. For instance, μ_1 is most strongly correlated with σ_1, and less so with the other standard deviation features. This correlation between the standard deviation and the mean positions can be explained by considering that the higher in frequency the component mean is located, the less likely it is to be modelling a strong spectral peak rather than the general spectral shape. Thus if a component peak is located at a higher frequency, the larger the standard deviation will tend to be and the lower the corresponding energy term. However, the opposite is true for the higher order components, where a strong negative correlation is observable between μ_4 and σ_4. This suggests that if the fourth component is found higher in the spectrum, the bandwidth of that component will tend to be smaller. Hence, it appears that the more centrally located a component is, the more likely it will be wider and model the general spectral shape rather than a narrow peak. There are also negative correlations between the component means and the energy terms which could also be explained in a similar fashion as the lower energy sounds will not have strong peak structures and the components will have larger variances to model the general spectral structure. Component energies will tend to be correlated since if the energy of one component is high, the energies in the adjacent components will also tend to be higher as the overall energy in the sound is greater. Additionally, if the spectrum has a less defined formant structure, the components will be distributed higher in the spectrum and have larger variances to model the general spectral shape. The correlations between the mean features could also indicate some degree of inter-speaker correlation. The vocal tract length variations between speakers would lead to the components to be estimated uniformly higher or lower in the spectrum, and hence would be correlated.

The correlations may affect the performance when used with a speech recognition system. It is usually assumed that the elements of the feature vector are uncorrelated, and thus that diagonal covariance matrices can be used to represent the data in the HMM output PDFs. If highly correlated features are used, a diagonal covariance matrix may not be appropriate, and different approaches should be used to model the correlations. Such approaches could include increasing the number of mixtures or using a full covariance matrix.

4.4.3 Confidence measures

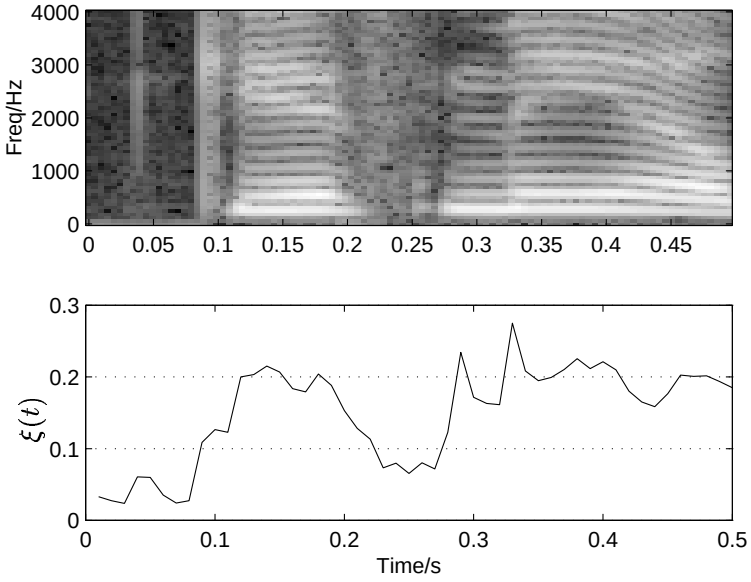


Figure 4.7 *Confidence metric plot for a test utterance fragment, with* $\beta = 0.3$

As mentioned in section 3.5.4, peak representations of the spectra may be less reliable for certain classes of phone, especially if the locations of the peaks alone are used. One proposed solution to the problem of different performance on different classes is to incorporate a confidence measure to combine formant locations with an MFCC parameterisation [114]. The system proposed implemented a measure of confidence to decrease the contribution of the formant features in regions where such features were not strongly defined. The confidence measure was used to scale the log-probabilities of the formant features when combining them with the MFCC features. As the HMMs only used single component output distributions, the confidence measures were effectively used as a time-dependent stream weight on the features. The confidence measure was derived from the Holmes formant estimator. The Holmes formant tracker gives a measure of confidence based on the amplitude and degree of local curvature for each hypothesised formant.

For a Gaussian component in the GMM, the amplitude can be represented by the log-normalised component energies $\tilde{e}_m(t)$. For a given Gaussian component in the GMM it has been shown that the 3dB bandwidth is proportional to the standard deviation [125]. Thus, for the GMM

confidence metric, the curvature term is replaced by the reciprocal of the component standard deviation.

The individual component confidence metrics were defined as

$$\xi_m(t) = \frac{\tilde{e}_m(t) + 10.53}{\sigma_m(t)} \tag{4.63}$$

The silence floor at 50dB (10.53) has been added to the log-normalised component energy $\tilde{e}_m(t)$ to constrain $\xi_m(t)$ to be positive. Hence the confidence measure will be high if the component has a narrow variance and high energy. The assumption made is that regions with pronounced peak structures will be estimated more reliably and consistantly. Components with lower energies and wider variances are more likely to be representing the spectral shape and will not be estimated consistantly.

The confidence measure could be applied to each GMM component mean probability separately in a similar fashion to the approach outlined in section 3.5.4. However, initial results using the confidence measures in this way gave only slight improvements. An alternative configuration was considered where all the component means and their derivatives could be placed into the same stream and a single confidence measure used on them. If the component means are to form a single stream weight, they must be combined in some fashion. The simplest approach to combining the scores is to take the arithmetic or geometric mean of the confidence measures. Small-scale initial experiments suggested that the geometric mean was preferable. The geometric mean has the advantage that if a single component is poorly defined, the combined confidence measure for the frame will be low. Hence, with a geometric mean, the confidence measure for the frame will only be high if most of the components appear to be well-defined. For a frame t, a confidence metric $\xi(t)$ can be defined taking the geometric mean of the component energies and curvatures:

$$\xi(t) = \beta \left[\prod_{m=1}^{M} \frac{\tilde{e}_m(t) + 10.53}{\sigma_m(t)} \right]^{\frac{1}{M}} \tag{4.64}$$

where β is a fixed scale factor. An example spectrogram and associated confidence $\xi(t)$ are shown in Figure 4.7. The confidence metric is high in regions with strong formant structures and low during unvoiced sounds, as expected. As mentioned previously, the confidence metric is constrained to be positive, hence the minimum value possible will be zero, during periods of silence. The upper limit of the metric is set by both the scale factor, maximum energy and standard deviations of the components. In practice, this means that the typical range of the metric is $(0, \beta)$.

For a synchronous stream system, the output probability distribution $b_j(\mathbf{y}(t))$ for an input vector $\mathbf{y}(t)$ divided into R streams $\{\mathbf{y}_1(t), \ldots \mathbf{y}_R(t)\}$ is calculated as

$$b_j(\mathbf{y}(t)) = \prod_{r=1}^{R} \Big[\sum_{m=1}^{M_r} P(\omega_{jrm}) \mathcal{N}(\mathbf{y}_r(t); \boldsymbol{\mu}_{jrm}, \boldsymbol{\Sigma}_{jrm}) \Big]^{\gamma_r(t)} \tag{4.65}$$

where $\gamma_r(t)$ is the time-dependent stream weight and $P(\omega_{jrm})$, $\boldsymbol{\mu}_{jrm}$ and $\boldsymbol{\Sigma}_{jrm}$ are the weight, mean and variance for component m of stream r for state j.

The framework previously mentioned combined the formant features with MFCCs by scaling the log-emission probabilities of the formant features by the confidence measure. However, the implementation was only performed with single-component HMM models on a phone recognition task. If the stream weights are to vary as a function of time, it is necessary to consider the fact that the feature streams may have different dynamic ranges. As mentioned in section 2.2.4, the acoustic model and the language model are separate information sources and will have differing dynamic ranges. Hence, a language model scale factor is used to compensate for the mismatch. Thus, in a medium or large vocabulary system, different parameterisations will have different optimal language model scale factors, α_1 and α_2. Simply using a sum-to-one constraint on the stream weights will not ensure that the weighted streams will have the same dynamic range. Hence the optimal grammar scale factor will vary depending on the confidence metric for a given utterance. The search for the optimal word string over each information stream[2] $\mathbf{Y}_T^{(1)}$ and $\mathbf{Y}_T^{(2)}$ separately is:

$$\hat{\mathbf{W}}^{(1)} = \arg\max_{\mathbf{W}} \left[\alpha_1 \log P(\mathbf{W}) + \log p(\mathbf{Y}_T^{(1)}|\mathbf{W})\right] \tag{4.66}$$

$$\hat{\mathbf{W}}^{(2)} = \arg\max_{\mathbf{W}} \left[\alpha_2 \log P(\mathbf{W}) + \log p(\mathbf{Y}_T^{(2)}|\mathbf{W})\right] \tag{4.67}$$

If the log-likelihood of the acoustic model in the second stream is scaled by $\frac{\alpha_1}{\alpha_2}$, the optimal language model scale factor for that information stream is also α_1. Thus, if the log-likelihoods of the second stream are scaled by $\frac{\alpha_1}{\alpha_2}$, both streams will approximately have the same dynamic range. Thus the two streams can be appropriately weighted or deweighted given the confidence metric $\xi(t)$, and the dynamic range will approximately be preserved. In a two-stream system using a confidence metric the stream weights are given by

$$\gamma_1(t) = 1 - \xi(t) \tag{4.68}$$

$$\gamma_2(t) = \left[\frac{\alpha_1}{\alpha_2}\right]\xi(t) \tag{4.69}$$

and these will be substituted into equation 4.65 during recognition.

4.4.4 Speaker adaptation

Formant features are not speaker independent, and can exhibit strong inter-speaker correlations [23]. The inter-speaker correlations have been exploited to estimate speaker adaptation transforms for speech recognition [44]. The positions of the formants for a given speaker can be used to estimate a vocal-tract length warping factor for a speaker [74] [68]. These warping factors have been used to warp the positions of the Mel or critical band filters in MFCC parameterisations to adapt the features and give improved recognition performance [44]. As shown in figure 2.4, the vocal tract warping function can assume different forms. The vocal tract variation can be approximated by a piecewise linear function or by a bilinear transform.

[2]The word insertion penalty has been neglected for simplicity

Work using formants and spectral peak estimates as acoustic features has sought to normalise these features on a per-speaker or per-utterance basis [104]. If the formant positions are uniformly scaled by a given value, the effects of a change in vocal tract length could be easily removed. Thus, for spectral features, the search for vocal tract normalising factors can be easier, since the formants are represented directly in frequency values.

An alternative approach is to apply feature mean normalisation to the formant features. This approach removes any linear shift from each formants and has been likened to a vocal tract normalisation transform [114]. The assumption made is that each formant should be distributed about its mean for a given utterance. Any shift on the mean position is assumed to be an an inter-speaker variation based on the change in vocal tract length, and can thus be removed. This approach is a linear subtraction and will have different effects from the application of a scaling factor to the extracted parameters. One difference is that observed range of the formants for a given speaker will be unchanged when a linear shift is applied, whereas using a linear scale would compress or expand the effective range of the formants. Applying feature variance normalisation could possibly compensate for this effect if it is an issue. As the means are shifted up for a speaker, the variances will also be expected to increase as the seperation of the peaks increases.

As mentioned previously, the GMM spectral features can be viewed as analogous to a set of formant eatures. As such, the approaches using formant features outlined above could prove useful for the GMM features as well. The extracted features for the speakers will allow for a VTLN transform to be simply estimated by linear regression. A linear warping of the component means will effect a simple VTLN normalisation approach directly on the features. Feature mean normalisation and feature variance normalisation will also help to normalise the speakers in terms of the vocal tract function. As the component means are scaled, the standard deviation features should be varied as well. As mentioned previously, the standard deviations can be related to the bandwidths of the components. If the locations of the components are to be linearly scaled, the standard deviations may also be expected to increase as the separation between the peaks increases.

The component log-energies in the GMM features will also exhibit inter-speaker variations. The arguments for cepstral (or log-spectral) normalisation will also apply to the GMM component log-means. Some of the effects of environmental noise or any speaker bias such as a spectral tilt could be removed by applying feature mean normalisation to these features.

4.5 Noise compensation for Gaussian mixture model features

Two approaches for compensating the GMM features in the presence of additive background noise are outlined in this section. The first operates during the feature extraction stage and attempts to estimate the clean speech parameters from the noise corrupted spectrum given a noise model. The second technique operates on the model set, attempting to form a noise compensated model set from the clean speech HMM and a noise model. Both compensation

schemes use a model derived from the average GMM features calculated from the additive noise source. This section does not address the issue of estimating the noise model itself. The noise model is assumed to be known, but could be obtained by similar approaches to those used when using the PMC techniques [40].

4.5.1 Spectral peak features in noise corrupted environments

Spectral peak features have been demonstrated to give some improvements when combined with MFCC-based systems on noisy data [16] [108]. Since the peaks represent the high energy regions of the speech, it is hoped for many noise sources that the formants or spectral peaks will sit above the level of the noise. If the spectral peaks sit above the level of the noise, the spectral peak features will be less corrupted by additive noise than the MFCC or PLP parameters would. The MFCC and PLP features represent the spectral shape. Adding a noise source will affect all of the parameters, although the lower order cepstra which represent the more general spectral shape will be worst affected [33]. However, noise sources with peak-structures in them can corrupt spectral peak or formant representations severely. For coloured noise of sufficient amplitude the spectral peak features will model the noise source rather than the speech signal.

The gravity centroid systems has shown improved recognition performance over MFCCs for certain additive noise conditions [12]. Specifically, significant reductions in WER were observed when using these features on speech corrupted with additive white Gaussian noise and car noise [31]. However, little improvement was gained from using gravity centroid features when the noise source was factory noise or background speech, since these sources contain strong peaks. In these circumstances, it is desirable to compensate the system to account for the mismatch between training and test environments. However, given the non-linear mapping between the spectral domain and the features extracted, this is not always simple. Additionally, some alternative parameterisations cannot map from the feature domain back to the spectral domain. In contrast, the GMM features have the advantage that it is possible to recover the speech spectrum from the feature set. In addition, the representation of the features directly in the log-spectral domain makes it simpler to implement a model of an additive noise sources.

4.5.2 Front-end noise compensation

This section presents an approach for compensating the GMM features during the feature extraction process. In this method, a set of fixed Gaussian components representing the additive noise are combined with the estimated GMM parameters in the EM process. The assumptions that are made are that the speech and noise are additive in the linear FFT magnitude spectrum, and that the noise source is stationary. By estimating the GMM parameters with the noise GMM added to the estimated distribution, the aim is to extract the clean speech GMM parameters from the noise corrupted speech. An overview of the use of a noise GMM to obtain estimates of the clean speech GMM parameters is shown in figure 4.8. The technique adds a set of noise mixtures to

model the additive noise during the EM estimation process, with the aim of estimating the clean speech models.

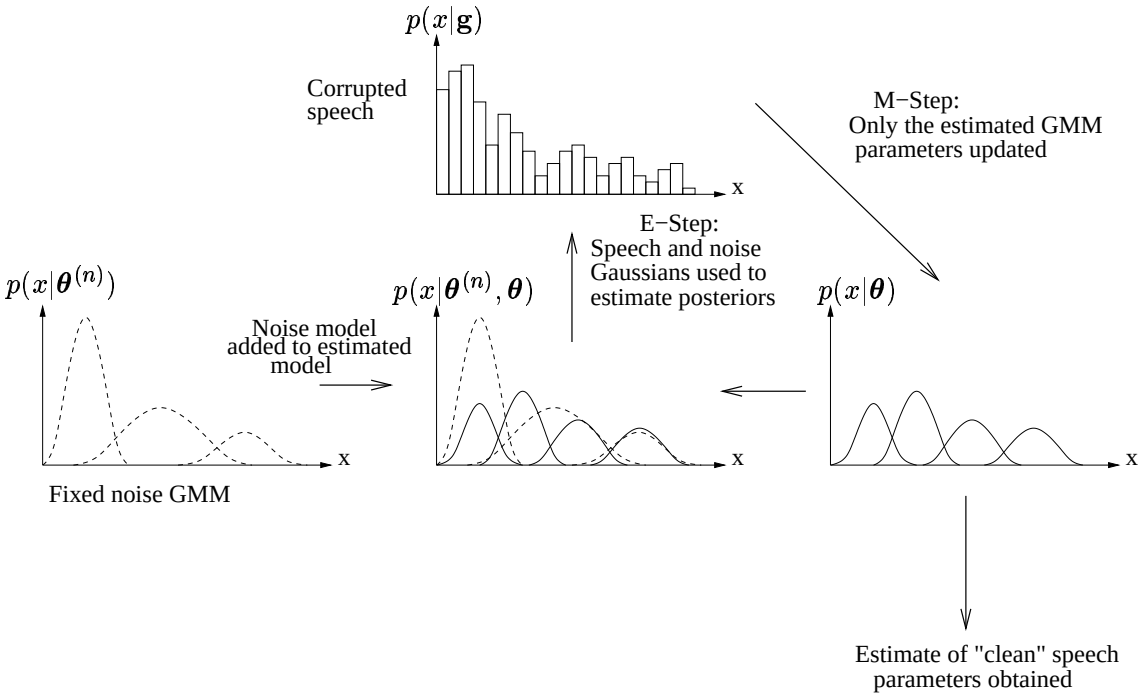


Figure 4.8 *Using a GMM noise model to obtain estimates of the clean speech parameters from a noise-corrupted spectrum*

This approach attempts to use a static noise model during the feature extraction process to estimate the clean speech parameters, assuming a model of additive noise. In this respect, it resembles the approach of spectral subtraction systems [7], but it avoids the problems of negative spectral values that can occur [21].

The aim of the EM step is, thus to optimise the log likelihood of the data given the clean and noise GMM parameters: with respect to the clean speech parameters $\boldsymbol{\theta}$:

$$\mathcal{E}\left\{\log p(x|\boldsymbol{\theta}, \boldsymbol{\theta}^{(n)})|\mathbf{g}\right\} = \sum_{i=1}^{N} P(g_i)\mathcal{E}\left\{\log p(x|\boldsymbol{\theta}, \boldsymbol{\theta}^{(n)})|g_i\right\} \tag{4.70}$$

The optimisation technique needs a model of the noise source. There are several approaches that could be used to estimate a noise model, such as using a voice activity detector. However, for simplicity in these systems, the noise model is assumed to be known, and a pre-calculated noise model is used. The noise model $\boldsymbol{\theta}^{(n)}$ for a given frame is formed from the average features $\bar{\mathbf{y}}^{(n)}$ of a series of T extracted features of a Q-component GMM estimated offline from the additive noise data:

$$\bar{\mathbf{y}}^{(n)} = \frac{1}{T}\sum_{t=1}^{T}\mathbf{y}(t) \tag{4.71}$$

where the average features comprise the means, standard deviations and the component energies of the noise spectrum:

$$\bar{\mathbf{y}}^{(n)} = [\mu_1^{(n)}, \ldots, \mu_Q^{(n)}, \sigma_1^{(n)}, \ldots, \sigma_Q^{(n)}, e_1^{(n)}, \ldots, e_Q^{(n)}]$$

The corresponding GMM parameters can be calculated from the average features. The noise model is assumed to be at a fixed energy level. Thus, the weight of the noise model is dependant on the spectral energy in the frame. For frames with low spectral energy the weighting of the noise model will be higher. The priors of the noise components will sum to one and are taken from the average noise features:

$$P(\omega_m^{(n)}) = \frac{e_m^{(n)}}{\sum_{q=1}^{Q} e_q^{(n)}} \tag{4.72}$$

and the weight of the noise distribution for a given frame is:

$$\psi = \frac{\sum_{q=1}^{Q} e_q^{(n)}}{\sum_{q=1}^{Q} e_q^{(n)} + \sum_{i=1}^{N} s_i(t)} \tag{4.73}$$

Hence, the weighted prior probablities for the speech and noise mixture components will sum to one:

$$(\psi) \sum_{q=1}^{Q} P(\omega_q^{(n)}) + (1-\psi) \sum_{m=1}^{M} P(\omega_m) = 1 \tag{4.74}$$

Using the approximation that data drawn from the same distribution can be assigned with the same posterior probabilities as before, the auxiliary function can be described as:

$$\begin{aligned}
\mathcal{Q}(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta}, \boldsymbol{\theta}^{(n)}) \quad \approx \quad & \sum_{m=1}^{M} \left[\sum_{i=1}^{N} P(\omega_m | g_i, \boldsymbol{\theta}, \boldsymbol{\theta}^{(n)}) P(g_i) \mathcal{E} \left\{ \log(x | \omega_m, \hat{\boldsymbol{\theta}}_m) | g_i \right\} \right] \\
& + \sum_{q=1}^{Q} \left[\sum_{i=1}^{N} P(\omega_q^{(n)} | g_i, \boldsymbol{\theta}, \boldsymbol{\theta}^{(n)}) P(g_i) \mathcal{E} \left\{ \log(x | \omega_q^{(n)}, \boldsymbol{\theta}^{(n)}) | g_i \right\} \right] \\
& + \sum_{m=1}^{M} \left[\sum_{i=1}^{N} P(\omega_m | g_i, \boldsymbol{\theta}, \boldsymbol{\theta}^{(n)}) P(g_i) \log \left(\hat{P}(\omega_m) \right) \right] \\
& + \sum_{q=1}^{Q} \left[\sum_{i=1}^{N} P(\omega_q^{(n)} | g_i, \boldsymbol{\theta}, \boldsymbol{\theta}^{(n)}) P(g_i) \log \left(P(\omega_q^{(n)}) \right) \right]
\end{aligned} \tag{4.75}$$

The posterior probabilities for the estimated components are calculated over the noise and speech models:

$$P(\omega_j | g_i, \boldsymbol{\theta}, \boldsymbol{\theta}^{(n)}) \approx \frac{(1-\psi) P(\omega_j) \exp\left(\mathcal{E}\left\{ \log p(g_i | \omega_j, \boldsymbol{\theta}_j) | g_i \right\} \right)}{\sum_{m=1}^{M} (1-\psi) P(\omega_m) \exp\left(\mathcal{E}\left\{ \log p(g_i | \omega_m, \boldsymbol{\theta}_m) \right\} \right) + \sum_{q=1}^{Q} (\psi) P(\omega_q^{(n)}) \exp\left(\mathcal{E}\left\{ \log p(g_i | \omega_q^{(n)}, \boldsymbol{\theta}_q^{(n)}) \right\} \right)} \tag{4.76}$$

The parameter update equations are then:

$$\hat{\mu}_j = \frac{\sum_{i=1}^{N} P(g_i)P(\omega_j|g_i,\boldsymbol{\theta},\boldsymbol{\theta}^{(n)})f_i}{\sum_{i=1}^{N} P(g_i)P(\omega_j|g_i,\boldsymbol{\theta},\boldsymbol{\theta}^{(n)})} \tag{4.77}$$

$$\hat{\sigma}_j^2 = \frac{\sum_{i=1}^{N} P(g_i)P(\omega_j|g_i,\boldsymbol{\theta},\boldsymbol{\theta}^{(n)})(f_i^2 - \hat{\mu}_j^2 + \frac{1}{12})}{\sum_{i=1}^{N} P(g_i)P(\omega_j|g_i,\boldsymbol{\theta},\boldsymbol{\theta}^{(n)})} \tag{4.78}$$

$$\hat{P}(\omega_j) = \sum_{i=1}^{N} P(g_i)P(\omega_i|g_i,\boldsymbol{\theta},\boldsymbol{\theta}^{(n)}) \tag{4.79}$$

The noise component GMM parameters are assumed to be fixed over all frames, and are not updated using the EM algorithm. Thus if the model of the noise is accurate, it is hoped that the GMM parameters estimated will represent the underlying clean speech.

4.5.3 Model based noise compensation

The previous section showed how to use a noise model to estimate "clean" GMM parameters from the noise corrupted speech. However, there are still some problems associated with this technique. One problem is that the components in the noise source can mask lower amplitude peaks in the clean speech. To avoid the problem of masking caused by the front-end noise compensation, the clean speech HMMs may be adapted to model noise corrupted speech by using an average noise model. A diagram showing the steps used to compensate the clean speech HMM using the noise model is shown in figure 4.9. The spectrum is reconstructed from the mean GMM featurs from a given HMM state component, then the noise model is added in the linear spectral domain. Next, the GMM parameters for the noise corrupted spectrum are estimated using the EM algorithm as before. Finally, the GMM parameters of the average spectrum for the state/component are transformed to yield the compensated average GMM features.

The approach is similar to that of the log-add approximation for MFCC or PLP features [33]. The GMM features are used to reconstruct the clean speech spectrum for a state in the HMM, then a noise model is added to form a noisy spectrum, and the parameters for the noise-corrupted spectrum are calculated.

Using the static means of the output PDF from HMM state j component u, it is possible to obtain the average GMM parameters for that state/component $\boldsymbol{\theta}_{ju}$. A set of data points at a uniform interval W can be calculated from these mixture models. In the original estimation process of section 4.1.4 the bins had a width or interval of 1. The arbitrary width allows for more rapid compensation schemes where the re-estimated histogram has fewer points than the original estimates. The number of points in the reconstructed spectrum is R where $R = N/W$ and N is the number of points used to originally estimate the spectrum. The spectrum $\mathbf{s}_{ju} = \{s_{ju1}, \ldots, s_{juR}\}$ can be generated from the GMM mean parameters $\bar{\mathbf{y}}_{ju}$ from the HMM output PDF for state j and mixture u. A noise spectrum $\mathbf{q} = [q_1, \ldots, q_R]$ can then be added to the the reconstructed spectrum. The reconstructed points are distributed uniformly, such that each

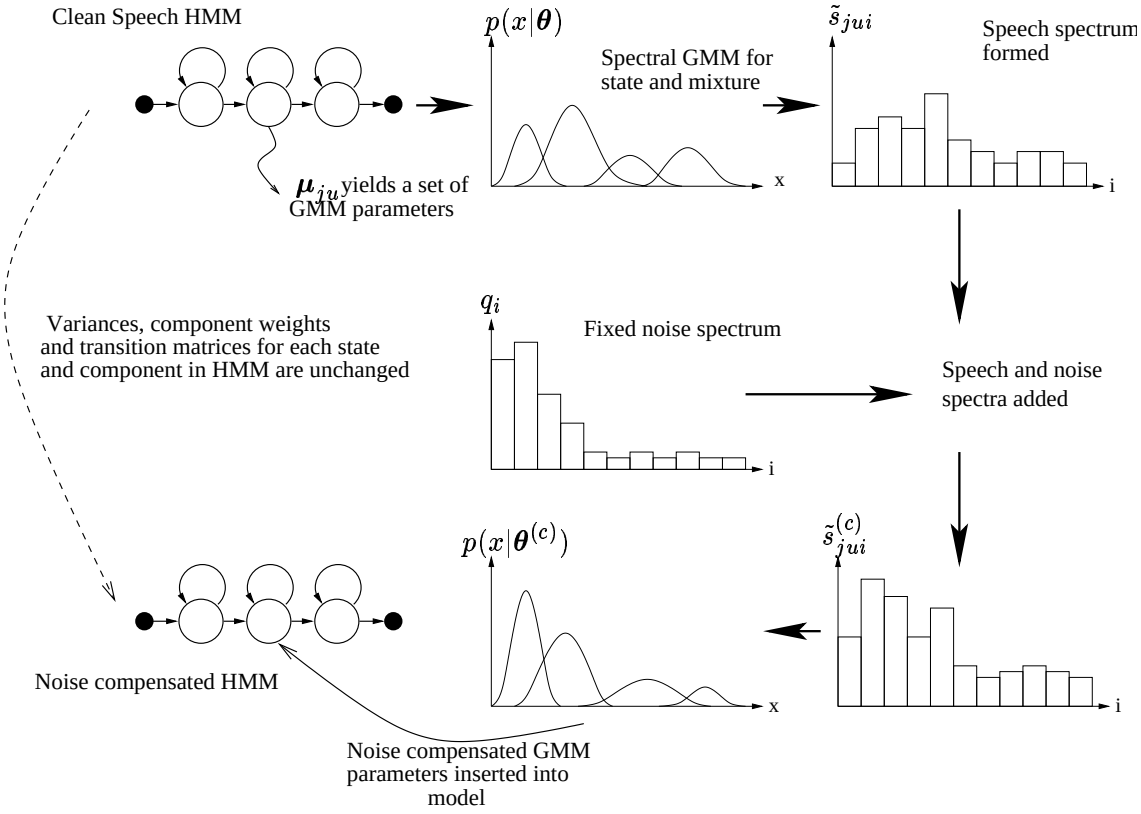


Figure 4.9 *Formation of a continuous probability density function $p(x|\mathbf{g})$ from FFT values*

point $s_{jui}^{(c)}$ is located at f_i where $f_i = W(i - \frac{1}{2})$. The noise-corrupted spectrum is given by:

$$s_{jui}^{(c)} = \sum_{m=1}^{M} \left[e_{jum} \mathcal{N}(f_i; \mu_{jum}, {\sigma^2}_{jum}) \right] + \sum_{q=1}^{Q} \left[\bar{e}_m^{(n)} \mathcal{N}(f_i; \bar{\mu}_m^{(n)}, (\bar{\sigma}_m^{(n)})^2) \right] \tag{4.80}$$

A piece-wise continuous PDF $p(x|\mathbf{g})$ is then formed from the noise-corrupted spectrum. The prior probabilities of the reconstructed data will be:

$$P(g_i^{(c)}) = \frac{s_{jui}^{(c)}}{\sum_{i=1}^{R} s_{jui}^{(c)}} \tag{4.81}$$

and the PDF functions form a histogram with each bin centered at f_i with a width of W:

$$p(x|g_i^{(c)}) = \begin{cases} \frac{1}{W} & (f_i - \frac{W}{2}) \leq x \leq (f_i + \frac{W}{2}) \\ 0 & \text{(otherwise)} \end{cases} \tag{4.82}$$

The noise corrupted Gaussian parameters $\boldsymbol{\theta}^{(c)}$ can then be estimated from the noise corrupted PDF. A new set of GMM parameters $\hat{\boldsymbol{\theta}}_{ju}^{(c)}$ for state j and mixture u in the output PDF of the HMM are estimated from the noise corrupted PDF.

By optimising the log-likelihood of the noise corrupted PDF for newly estimated noise compensated parameters $\boldsymbol{\theta}_{ju}^{(c)}$, a set of noise-compensated model features can be obtained. The likelihood to be optimised is:

$$\mathcal{E}\left\{ \log p(x|\boldsymbol{\theta}_{ju}^{(c)}) | \mathbf{g}^{(c)} \right\} = \sum_{i=1}^{R} P(g_i) \mathcal{E}\left\{ \log p(x|\boldsymbol{\theta}_{ju}^{(c)}) | g_i^{(c)} \right\} \tag{4.83}$$

For a bin of width W, the expectations of the data given the estimated bins are:

$$\begin{aligned} \mathcal{E}(x|g_i^{(c)}) &= \int_{f_i - \frac{W}{2}}^{f_i + \frac{W}{2}} x p(x|g_i) dx \\ &= f_i \end{aligned} \tag{4.84}$$

$$\begin{aligned} \mathcal{E}(x^2|g_i^{(c)}) &= \int_{f_i - \frac{W}{2}}^{f_i + \frac{W}{2}} f^2 p(x|g_i) dx \\ &= {f_i}^2 + \frac{W^2}{12} \end{aligned} \tag{4.85}$$

Substituting these values into the posterior probabilities and auxiliary function it is possible to calculate the expected log-likelihood:

$$\mathcal{E}\left\{ \log p(x|\omega_m, \hat{\boldsymbol{\theta}}_{jum}) | g_i \right\} = \log\left(\frac{1}{\sqrt{2\pi\sigma_{jum}^2}} \right) + \left[-\frac{(f_i - \mu_{jum})^2 + \frac{W^2}{12}}{2\sigma_{jum}^2} \right] \tag{4.86}$$

Thus, using an arbitary bin width reduces the computation overhead when compensating the models, but loses some of the resolution in the spectrum.

Rather than reconstructing the spectrum and forming a continuous histogram for each state, an alternative implementation exists. It is possible to directly add a set of noise mixture components directly to the spectral GMM from the HMM state output PDF. The clean speech GMM

can be added to a noise GMM to form a set of GMM components which represent the corrupted speech. However, since it is formed from the summation of two mixture models, the number of components in this corrupt-speech GMM will be larger than the from the clean speech model. From the corrupted speech set of spectral GMM components a number of components equal to that in the clean speech GMM can be estimated. The EM algorithm is then used to estimate the posterior probability of a given (target) Gaussian being generated from components in the noise corrupted speech model. The likelihood of the data set is then maximised given these posteriors. The expectation of the data given a Gaussian component in the noise corrupted speech is:

$$\mathcal{E}(x|\omega_i) = \mu_i \tag{4.87}$$

$$\mathcal{E}(x^2|\omega_i) = \mu_i^2 + \sigma_i^2 \tag{4.88}$$

And as above, it is posible to calculate the posterior probabilities of each (target) Gaussian being generated by each component in the speech+noise model. The expected log-likelihood of a given (target) Gaussian given a Gaussian in the speech+noise model is:

$$\mathcal{E}\left\{\log p(x|\omega_m, \hat{\boldsymbol{\theta}}_{jum})|\omega_i\right\} = \log\left(\frac{1}{\sqrt{2\pi\sigma_{jum}^2}}\right) + \left[-\frac{(f_i - \mu_{jum})^2 + \mu_i}{2\sigma_{jum}^2}\right] \tag{4.89}$$

The parameter update equations can also be calculated as before using the above values, in a similar fashion to the use of Gaussian priors in section 4.2.2. Using the speech and noise GMM to compensate the spectral GMM parameters for each state will be faster than reconstructing the spectrum. However, each noise+speech Gaussian component will have only a single posterior probability to reassign its probability mass, and the technique may suffer from the same problems as using a large bin width.

From the estimates of the noise corrupted parameters we can calculate the static means for GMM features for the given state and mixture output PDF in the HMM. The fewer data points that are generated from the source mixtures, the more rapidly the technique can be applied. However, if too few points are estimated from the combined speech and noise mixture models, then the technique will perform poorly. Using only a few points gives a low resolution to the frequency spectrum. All of the data represented by the histogram bin is assigned by the same posterior ($P(\omega_j|g_i^{(c)}, \boldsymbol{\theta}_{jum}$). With only a few data points used, the variances of the sample data points are larger and the points less distinct. Additionally, the assumption that the data from a histogram bin can be represented by the same posterior probability function becomes less valid as the bin widths increase. In practice, as fewer bins are used, the posterior probabilities will become more evenly shared across all components. As a result, some mixtures will tend towards the same point and have a larger variance to cover all the points with identical posterior probabilities for the source bins.

The computational cost of this technique is of the same order of magnitude as compensation using parallel model combination (PMC) with a log-add approximation [40].

5

Experimental results using a GMM front-end

In chapter 4 a new method for parameterising speech by describing its peak structure was presented. Using this technique, a GMM is estimated from the speech spectrum using the EM algorithm. By estimating a GMM from a normalised speech spectrum, the speech can be represented by the parameters of the GMM. The GMM features can be related to the gravity centroid features.

In this chapter, baseline experiments using GMM features are presented on a medium vocabulary task, the Resource Management (RM) corpus. The RM corpus is based on a naval management task and has an approximately 1000-word vocabulary. The corpus is described in more detail in appendix B.1. The aim is to evaluate a variety of techniques and configurations that estimate GMM features from the spectrum for speech recognition.

5.1 Estimating a GMM to represent a speech spectrum

This section presents a series of initial experiments performed using the GMM features on the RM task. First, a baseline system is presented. Next, experiments with a number of spectral smoothing techniques to remove the pitch from the spectrum are shown. Results implementing psychoacoustic transforms on the spectrum, using Mel-scaling or a pre-emphasis filter are also detailed.

5.1.1 Baseline system

The first experiment on the RM corpus was to build a baseline system using a standard parameterisation for comparative purposes. This system was built using an MFCC parameterisation together with the RM recipe from the HTK toolkit [122]. The feature vector was comprised of the first twelve Mel-cepstra ($[c_1(t), \ldots, c_{12}(t)]^T$) and a normalised log energy term ($r(t)$), with the Δ and Δ^2 terms appended. This gave a feature vector of length 39. The feature vectors were computed from frames of speech taken every 10ms. The initial model set was based upon monophone models and was initialised using a flat start. Several iterations of the training were

then performed. The model set was then clustered into cross-word context-dependent triphones using decision-tree based state-clustering. A total of 1605 distinct states were used in the model set. The number of components in the HMM output PDFs was increased until no further improvement was observed on the 'feb89' subset of the data. By this measure, the optimal number of components was six. The language model scale factor was also tuned on this subset, and then used for recognition on the other test sets. The average WER over all four sets was 4.19%.

5.1.2 Initial GMM system

An overview of the system block layout is shown in figure 4.2. An inital GMM system was built for the RM task using GMM features estimated from the speech signal. Frames of speech 25ms wide were taken every 10ms and a Hamming window was applied. The RM data was sampled at 16kHz, yielding an FFT window 400 samples long which was then zero padded to 512. A 256-point magnitude FFT was then obtained, truncated to the first 128 bits to yield a maximum frequency of 4kHz. This bandwidth has been found to yield the most reliable estimates of the GMM components [125] and was thus chosen as a baseline. The spectrum was then normalised and a continuous density PDF formed using the technique outlined in section 4.1.2. No form of smoothing was applied to the spectrum initially. From this spectral histogram a set of six means, variances and component priors for a Gaussian mixture model were iteratively estimated using the technique detailed in section 4.1.4. Twelve iterations of the EM algorithm were taken. From the parameters of means, variances and component priors, a set of features were estimated. The components were ordered according to the frequency values of their means. Thus, the component with the lowest mean is the first and so forth. These features were the means $\boldsymbol{\mu}(t) = [\mu_1(t), \ldots, \mu_M(t)]^T$, standard deviations $\boldsymbol{\sigma}(t) = [\sigma_1(t), \ldots, \sigma_M(t)]^T$ and component log energies $\tilde{\mathbf{e}}(t) = [\tilde{e}_1(t), \ldots, \tilde{e}_M(t)]^T$ with a normalised log energy term $r(t)$, for the frame appended. The component energies were estimated by multiplying the component priors by the energy in the frame. The features were appended with the dynamic parameters to give a feature vector $\mathbf{y}(t)$:

$$\mathbf{y}(t) = \begin{bmatrix} \left[\boldsymbol{\mu}^T(t), \boldsymbol{\sigma}^T(t), \tilde{\mathbf{e}}^T(t), r(t)\right]^T \\ \left[\Delta\boldsymbol{\mu}^T(t), \Delta\boldsymbol{\sigma}^T(t), \Delta\tilde{\mathbf{e}}^T(t), \Delta r(t)\right]^T \\ \left[\Delta^2\boldsymbol{\mu}^T(t), \Delta^2\boldsymbol{\sigma}^T(t), \Delta^2\tilde{\mathbf{e}}^T(t), \Delta^2 r(t)\right]^T \end{bmatrix} \tag{5.1}$$

The length of the feature vector for an M component GMM is $3(3M + 1)$.

Using the HTK RM recipe [122] as described in appendix B.1 with this new feature vector, a cross-word context dependent triphone HMM recognition system was built. A flat start was used to initialise the model set as before and decision tree based state-clustering was used to form cross word triphones. The optimal number of distinct states in the initial model was 2202, larger than the MFCC system. In the systems built in the following sections, the number of states was roughly constrained to be the same. The number of components in the HMM output PDFs was increased until no further improvement was observed on the "feb89" subset of the test data. The language model scale factor was also tuned on this subset of the data. The optimum

number of components per state in the HMM output PDFs was seven, slightly higher than the MFCC system which used six, possibly due to the correlations in the model set and the extended feature vector. All systems built on the RM task in this chapter were trained using individual state clusterings. It is worth noting that for the GMM systems both the optimal number of distinct states for the triphones was larger than that of the MFCC system. In addition, the size of the feature vector was larger than than of the MFCC system. The combined effect of these increases means that the total number of parameters to estimate in the HMMs for the GMM systems was higher than that of the MFCC system. This is something that was observed with a number of configurational changes in this chapter. However, care has been taken to ensure that the number of parameters and states in each system in the following sections are tuned to the optimal value (in terms of WER for a subset of the test data) to ensure that the systems are comparable and the best possible for a given parameterisation for the RM task. On the full set of test data, the GMM baseline system had a word error rate of 6.02%, significantly worse than that of the MFCC baseline system, which was 4.19%. The poorer performance of the GMM features is consistent with results using other formant or peak representations [12] [109] [111]. It may be that the GMM features do not represent the phonetic classes as well or provide as much discriminatory information as the MFCC features. Alternatively, it may be that the model does not represent the GMM features as well.

5.1.3 Spectral smoothing

One of the first considerations was to use some form of spectral smoothing to estimate the spectral envelope and remove the effects of the speech source. Three different techniques were investigated:

- A convolutional pitch filter was used as outlined in section 4.2.1.3. The pitch was estimated by searching for the peak in an autocorrelation function. The spectrum was then convolved with a raised cosine window centred on the fundamental frequency.
- Cepstral deconvolution was performed by taking the DCT transform of the DFT log-magnitude spectrum, then truncating it after a fixed number of bins (20 in this case), as presented in section 4.2.1.1. The spectrum was then reconstructed by taking the inverse of the log-cepstral representation.
- The SEEVOC envelope was extracted by searching for the pitch peaks at multiples of the fundamental frequency, as detailed in section 4.2.1.2. The locations and values of the pitch peaks were then interpolated to obtain the spectral envelope. An estimate of the pitch was obtained from the autocorrelation function as in the convolutional pitch filter.

The approaches used for estimating the vocal tract response or spectral envelope have different effects on the resulting spectrum, as shown in figure 5.1. In particular, the magnitudes and bandwidths of the formants, and the magnitudes of the anti-resonances differ greatly. The

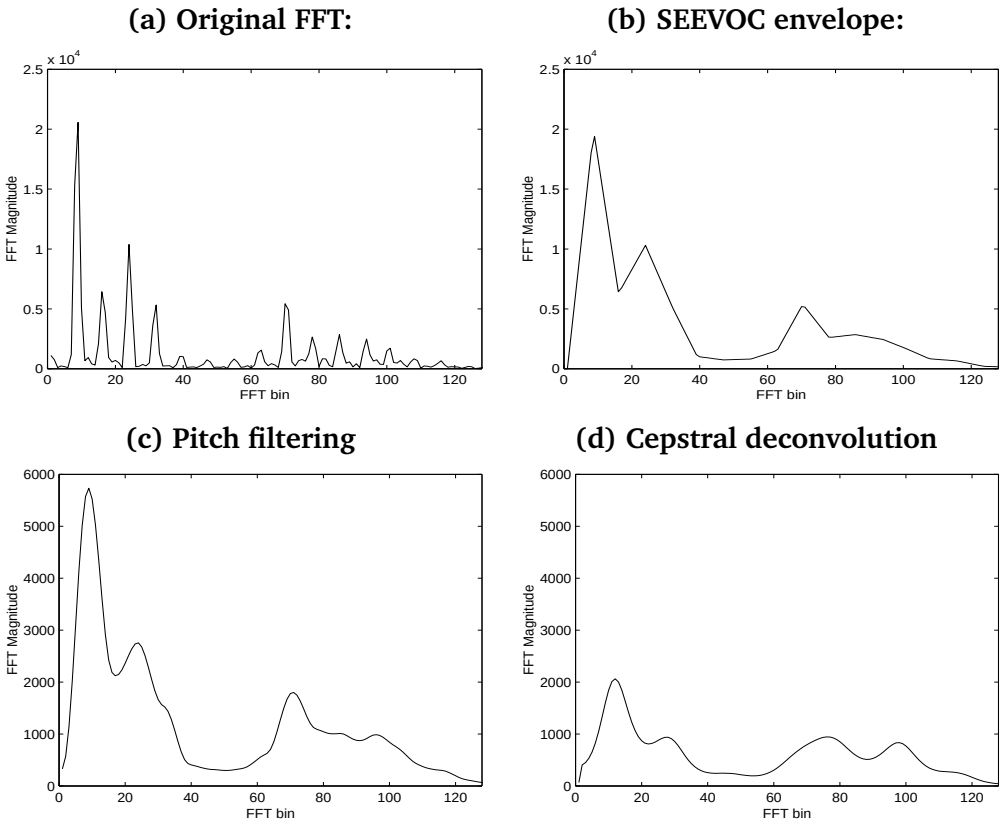


Figure 5.1 *Removing pitch from spectrum by different smoothing options*

SEEVOC smoothing finds the pitch peaks and interpolates between them to extract the envelope. Also, by interpolating between the pitch peaks the SEEVOC envelope will increase the total spectral energy and maintain the spectral magnitude at the locations of the pitch peaks. Thus, the envelope extracted can have wider peaks or formant structures and less defined peaks. Conversely, the convolutional pitch filter tends to extract more pronounced peak structures. With the SEEVOC envelope, more of the auxiliary function of the EM algorithm to be optimised is concerned with representing the lower-energy portions of the the spectrum. The SEEVOC envelope was used in the vocoder because it maintained the peak amplitudes of the partials within the spectrum. It is not the optimal smoothing technique if the GMM parameters are to be used in a recognition system, and the interpolated structure is not well represented by a GMM when the speaker pitch is relatively high. The cepstral filtering loses some of the definition of the peak structure when the high order cepstra are truncated. The strongly defined formant peaks can be attenuated by truncating the higher cepstra, as shown in figure 5.1.

The results of these experiments are presented in table 5.1. The optimal smoothing procedure in terms of reducing the error rate was the pitch-based convolutional filter. All other smoothing systems gave a similar performance on the RM task. The improvement of the pitch-filter over the SEEVOC window and the no smoothing case was significant at a confidence of not less than 95%. The SEEVOC and cepstral deconvolution approaches can be seen to remove the voicing effects from the spectrum. However, they also change the spectral representation in ways which degrade the extracted GMM features.

Smoothing Type	% WER
None	6.02
SEEVOC window	6.08
Pitch Filter	5.59
Cepstral liftering	5.90

Table 5.1 *Performance of parameters estimated using a six-component GMM to represent the data and different methods of removing pitch*

5.1.4 Feature post-processing

The energy levels vary on a speaker and channel basis, so using the log component energies directly may not be ideal. A simple technique to reduce the problems this presents is to normalise the log energies in a sentence. A standard approach used in the HTK environment was implemented in which the log energies were scaled such that the maximum log energy had a normalised value of 1 [122]. A silence floor was implemented 50dB below this, so the effective range of the component log energies was set at $(1, -10.53)$. Also, the energy value at each GMM mean position may be more useful than the energy of each component. Some components are used to represent the general spectral shape rather than the peaks, and have very large vari-

ances. Also, in the cases where two components or peaks are close together, the component energies will not represent the spectral amplitude correctly.

The best feature set obtained so far was from a six-component GMM estimate from a 4kHz spectrum smoothed with a pitch filter. The two techniques above (log energy normalisation, and use of log magnitude values at the means) were applied to this feature set. Applying the component log-normalisation gave a reduction in WER from 5.59% to 5.24%, a reduction of 6% relative. Using the log-magnitudes at the means rather than the component log-energies gave a further reduction the error rate to 4.90%, a relative improvement of 14% in total. This improvement can be attributed to both using the component mean energies and the log-component energy normalisation.

5.1.5 Psychoacoustic transforms

Psychoacoustic processing or transformations have been successfully applied in many feature extraction schemes [50]. These techniques can be applied to the GMM estimation by transforming the spectrum before extracting the parameters.

Work with other features such as MFCCs and PLPs has shown improved performance using a spectral pre-emphasis filter [122]. A pre-emphasis filter will increase the energy in the upper regions of the spectrum. The human ear has the greatest amplitude sensitivity in the region 1-5kHz. Thus applying a pre-emphasis filter on data sampled at a rate of 8kHz will emulate the non-linear response of the human ear. A pre-emphasis filter can be applied to the speech waveform $w(n)$:

$$w_{PE}(n) = w(n) - 0.97w(n-1) \tag{5.2}$$

Implementing a pre-emphasis filter on the spectrum raises the energy in the higher frequency regions. Hence probability mass in the higher frequency regions of the spectrum is increased. As a result, maximisation of the auxiliary function places more emphasis on modelling the spectral energy in the higher bands and less on the low-frequency peaks. The effects of using pre-emphasis on a sample spectral frame are shown in figure 5.2. Applying this filter to the speech prior to estimating the GMM increased the WER to 6.08%. This was a relative increase in WER of 25% relative to the performance of the best system so far (from section 5.1.4. The lower frequency regions' spectral peaks are believed to be more useful for speech recognition [61] and if too much emphasis is placed on the region about 2kHz, it seems reasonable that the recognition performance will suffer.

A psychoacoustic non-linear frequency warp can be applied either prior to the EM algorithm or after the features have been extracted. In the first case, Mel-scaled filter bins can be positioned over the spectrum. The responses of these filter-banks can then be normalised and used instead of the normalised magnitude spectrum as a PDF for the Gaussian estimation. This warping has the effect of widening the formants at the lower frequencies and narrowing those at the higher frequencies, as shown in figure 5.2. In addition, calculating the mel-scaled frequency bins are

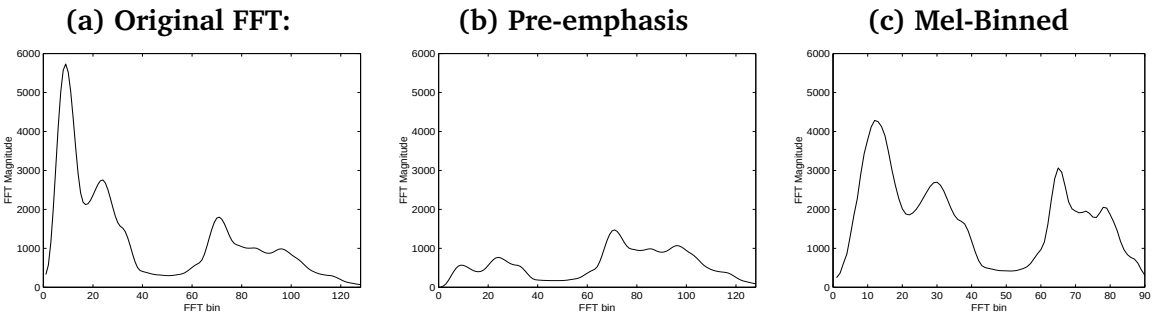


Figure 5.2 *Psychoacoustic transforms applied to a smoothed speech spectrum*

wider in the higher regions of the spectrum. This has the effect of increasing the magnitude of the higher frequency regions.Alternatively, the extracted mean positions from the magnitude spectrum GMM can themselves be Mel-scaled and a continuous histogram function formed from the mel-warped points. However, in the EM estimates of the GMMs, the lower frequency formant structures already have a large contribution to the EM estimation process, as they possess the greatest magnitude and therefore the largest amount of probability mass. The Mel-warping of the spectrum gives a further increase in the lower frequency probability mass by widening the formant structures there. The mel-scale binning increases the energy in the upper half of the spectrum. In practise, neither approach appears to improve the performance of the estimated features, suggesting that the balance of lower and upper region energy when using no warping is optimal.

Spectral Warp	% WER
none	4.90
Mel-binned spectra	7.49
Mel warping EM	7.19
Mel scaled mean positions	5.15

Table 5.2 *Warping frequency with Mel scale function, using a 4kHz system on RM task with GMM features estimated from the a six-component spectral fit*

Experiments were run using spectral Mel-warping and the Mel-scaling during the EM estimation on the RM task. The results are presented in table 5.2. Warping the spectrum with the Mel frequency scale degraded the performance of the GMM features considerably. The greater energy that exists in the lower frequency regions gives them a larger probability mass to contribute to the histogram. Hence increasing the probability mass in the lower regions by using Mel-warping does not help. Mel-scaling the estimated GMM component mean features before using them in an ASR system gives no significant degradation in the performance of the features. However, this scaling was not expected to yield an improvement, as Mel-scaling the component energy features after extraction does not change the relative importance of the frequency bands

but will merely alter the dynamic ranges of the features slightly.

5.2 Issues in the use of GMM spectral estimates

The configuration of the EM algorithm - the initialisation, the number of components estimates and the spectral bandwidth used will all affect the parameters estimated. These effects are explored in this section. In addition, the use of a prior distribution during the estimation procedure is examined.

5.2.1 Number of components

One variable to consider in estimating GMMs from the spectrum is the number of components used. A series of RM systems was built using features extracted from a six component GMM estimate of a 4kHz pitch-filtered system using the normalised log-magnitudes at the component means. Initially just the number of components estimated from a 4kHz spectrum was varied, and the results are presented in table 5.3.

Components Estimated	Number of parameters	% WER
3	30	7.95
4	39	6.53
5	48	6.12
6	57	5.59
7	66	6.66
8	75	6.96

Table 5.3 *Results on RM with GMM features, altering the number of Gaussian components in the GMM, using pitch filtering and a 4kHz spectrum*

Note that increasing the number of components increases the size of the feature vector, and hence the number of components in the system. The state tying during the formation of the cross-word triphones was optimised using the "feb89" subset of the test data to evaluate to obtain a reasonable number of parameters in the model set. The general rule for estimating formants from the spectrum is to assign one formant per kHz in the spectrum [81]. Estimating six components from the spectrum gave the best performance, improving on the performance of a 4 component system by 15% relative. The optimal number of components for the GMM features, 6, is higher than the typical number of expected formants in the spectrum. However, in the GMM estimate, some of the components are not modelling spectral peaks, but are just representing the general spectral shape. The components representing the general spectral shape do not model the peaks, but tend to have larger variances and smaller magnitudes at the means. The results using the spectral GMM as a vocoder [126], also gave optimal perceptual performance

when six components were used. The gravity centroid system also used the same number of spectral sub-bands [16]. Hence, although not all the structures extracted are formants or have a peak-like characteristic, it appears that the most consistent and robust strategy is to estimate six components in the mixture model.

5.2.2 Spectral bandwidth

Experiments presented so far only estimate parameters using a spectral bandwidth of 4kHz. The region 0-4kHz is believed to contain the most well-defined formant structures in speech [61]. Therefore, most formant or peak representation systems only use this frequency range. Beyond this region, the spectrum becomes more noise-like and formant features cannot model the spectrum well. However, there is information contained in the region above 4kHz which may be useful for recognition. For example, voiced affricative sounds have a significant proportion of their spectral energy above 4kHz. Hence, by only using the region 0-4kHz some potential discriminatory information in the spectrum may not be exploited by the GMM features. In order to estimate the degradation when only using this frequency range, a baseline MFCC system was built which was band limited to 0-4kHz. This system gave a WER of 4.30%, whereas the full 8kHz system had a error rate of 4.19%. Recognition systems built with MFCC or PLP features using only a 4kHz band limited signal perform worse than those using a full 8kHz spectrum.

The RM data was parameterised using a GMM system with pitch filtering, mean values and log-energy normalisation as before, but the full 8kHz spectrum was used. The components were initialised evenly across the spectrum. As previously, the size of the feature vector varies with the number of components. Hence, the optimal number of parameters in the HMM model will also vary. The number of parameters was tuned by changing the tying during the formation of the cross-word triphones and was tuned on the “feb89” subset of the test data. The results are in table 5.4. The best performance, 6.51% WER was obtained by estimating 8 components from an 8kHz spectrum, however, the performance of these features was worse than the WER obtained by estimating 6 components from a 4kHz spectrum.

Number of components	Numer of parameters	Word Error Rate
4	39	8.88
6	57	7.61
8	75	6.51
10	93	8.20
12	111	8.96

Table 5.4 *Varying number of components on a GMM system trained on a full 8kHz spectrum*

The problem with using an 8kHz spectrum in the system is that the extracted component means were still, on average, distributed evenly across the spectrum. Hence, the upper 4kHz

region had as many features dedicated to it as the lower 4kHz. Most speech recognition features such as MFCC or PLP are based on non-linear frequency scales which emphasis the lower frequency regions. It can therefore be expected that the majority of the upper component parameters are not useful for speech recognition. Additionally, there exists the problem that the upper band (4-8kHz) does not predominantly contain strong formant structures.

Nevertheless, the upper band does contain some useful information. Instead of initialising all the components evenly, it is possible to split the spectrum into sub-bands and perform separate GMM estimates for each region. Thus, the number of parameters dedicated to each band can be explicitly controlled by varying the number of components in the GMM in each region. The band-filtering of the spectrum in this fashion can be related to the work with gravity centroids [84]. The gravity centroid system is effectively the same as estimating a single set of Gaussian parameters from each spectral sub-band, or fixing the posterior probabilities of the GMM to the band-filter functions. Hence, each centroid has a fixed region to which it belongs. Splitting the spectrum into two bands and estimating the GMM parameters from each separately allows direct control of the frequency band each Gaussian represents, but is less severe than the constraints of the gravity centroid system.

In MFCC and some PLP features the Mel frequency scale is used to set the relative contribution of the frequency bands. The Mel-frequencies can also be considered when assigning components to frequency regions in the GMM estimates. The ratios of the Mel scaled frequencies at 4kHz and 8kHz from equation 3.2 is $4.2 : 1$. This suggests that for four or five lower band components one upper component should be estimated to give each band an amount of parameters proportional to its sensitivity in the human ear.

Components 0-4kHz	Components 4-8kHz		
	0	1	2
4	6.45	5.14	5.56
5	5.77	4.80	5.54
6	4.90	5.04	5.45

Table 5.5 *Estimating GMMs in separate frequency regions*

The results of the band-splitting experiments are in table 5.5. For a system using four or five components in the lower band, estimating extra components in the 4-8kHz band improves the performance. Estimating a single component in the upper band and five in the lower frequency band reduced the error rate by 3.3% relative to a system built using six components in the lower band only. This improvement is not statistically significant, however. The six component GMM system was not improved by adding extra components in the upper band. This may be due to the effects of over-estimating parameters or from increasing the size of the feature vector. The experiments agree with the ratios of Mel-frequencies hypothesised above. In addition, given the lack of formant structures in the upper region, it is only necessary to estimate a single Gaussian

to represent the general spectral shape and give an indication of the upper band energy.

5.2.3 Initialisation of the EM algorithm

The EM algorithm is sensitive to the values used to initialise the parameter estimates. The choice of initial parameters will constrain the local maximum found. As such, the choice of initialisation parameters for each frame is an important consideration.

The systems presented so far used a uniform distribution of the Gaussian components to initialise the GMM estimation. In previous work with the GMM features as a vocoder [126] the use of the previous frame values for the initialisation of the GMM algorithm was mentioned. Alternatively, it is possible to use the values from a formant plot or the gravity centroids as initialisation points. However, when features extracted from these systems were used as initialisation values, poor performance was obtained using the resulting GMM features.

Initialising the EM algorithm with the GMM parameters from the previous frame causes problems when the speech changes suddenly, such as a plosive sound. The estimated features respond poorly to rapid changes in the speech. The mixture weights of some components weights can approach zero and the variances become very large. When the values are passed onto the next frame the small component weights and large variances used to initialise will lead to the EM algorithm finding a local maximum with the variances approaching infinity and the priors approaching zero.

The system can also be initialised using the GMM parameters for each sub-band. These values can be related to the gravity centroid features. However, this limits the range of frequencies each component mean can occupy, since the initial estimates of variance are smaller than those previously used to initialise the EM algorithm. Hence, when GMM parameters were estimated using these values to initialise, the estimated components were much more limited in the regions they occupied. The parameters extracted using the sub-band did not vary significantly from the initialisation points. When used on the RM task, the GMM features extracted gave a WER of 6.70%. This is much higher than using the standard initialisations.

Examination of the extracted parameters suggested that the EM estimation is most sensitive to the initialisation of the component variances. If smaller variances are used in the initial parameters, the posterior probabilities of the histogram blocks will be mostly assigned to the closest Gaussian component. The locations of the Gaussian components will be constrained by their initialisation values, and it will be less likely that the component means parameters will vary greatly from the initialisation points. If the variances used to initialise are too large, the component weights can approach zero or the component will model the general spectral shape and not the peak structures. Using the values of the variances in equation 4.30 allows the components a sufficiently wide variance whilst still constraining the components to a rough frequency band. The estimates of the GMM parameters from the spectrum are not sensitive to the initialisation values of the component means. Implementing a peak picking algorithm or a formant estimator to initialise the EM algorithm does not yield any improvement in the features.

5.2.4 Number of iterations

The EM algorithm does not yield a closed form solution to the estimation of GMM parameters. However, the EM algorithm is guaranteed to converge upon a local maximum and the likelihood will not decrease with each successive iteration. The number of iterations used is important in determining the precise solution. Each successive iteration brings the parameters closer to a local maximum of the auxiliary function. There must be sufficient iterations to allow the algorithm to converge upon a solution and hence a good representation of the data in terms of the objective function. If too many iterations are used the model will not generalise well and will find a local maximum. There is a balance between allowing sufficient iterations to find a solution representative of the data, and not finding a specific maximum which generalises poorly onto other instances of the same class.

Another problem with using large numbers of iterations is that the histogram data is limited to the frequency range of the FFT. That is, the data lies only in a specific region. Thus beyond the boundary the Gaussians components will be modelling data which does not exist. This will affect the prior probabilities of the histogram bins assigned to the component. If a given component has a large bandwidth, it is possible that after a given point, the relative proportion of its probability mass outside of the data region can increase whilst the component weight decreases to zero.

In figure 5.3, the auxiliary function for a sample frame (t=1.85s from figure 5.4) has been plotted for 200 iterations. The auxiliary function levels off after about 10 iterations. Then, at around 100 iterations, there is another step in the function. The issue seems to be characteristic of over-training on the data, as a large change in the estimated parameters results in a small increase in the auxiliary function. The GMM parameters estimated after a large number of iterations are optimal in terms of the auxiliary function. However, they may be a local maxima specific to the given frame and not may not match the phonetic class well. In this case, the features extracted will perform poorly for recognition. By restricting the number of iterations, the parameters extracted may represent the general class of speech sound better.

The Gaussian means for the utterance “Where were you while we were away” (a mostly voiced utterance with strong formants) have been plotted in figure 5.4 estimated using 10 and 100 iterations. Although the trajectories follow roughly the same path there are some large discontinuities in the component mean trajectories when 100 iteration were used. Some of the component means are very different from the means estimated from the surrounding frames even though the spectra appear similar. Using a large number of iterations to estimate the GMM parameters gives erratic parameters which will not generalise very well for a given phone type.

In all the experimental results presented so far, the number of iterations of the EM algorithm has been fixed at 12 based on previous work using the GMM as a formant estimation algorithm [125]. To investigate varying the number of iterations, GMM parameters were extracted from 4kHz pitch filtered speech spectra using a varying number of iterations of the EM algorithm. From these GMM parameters, features based on the log-normalised component magnitudes,

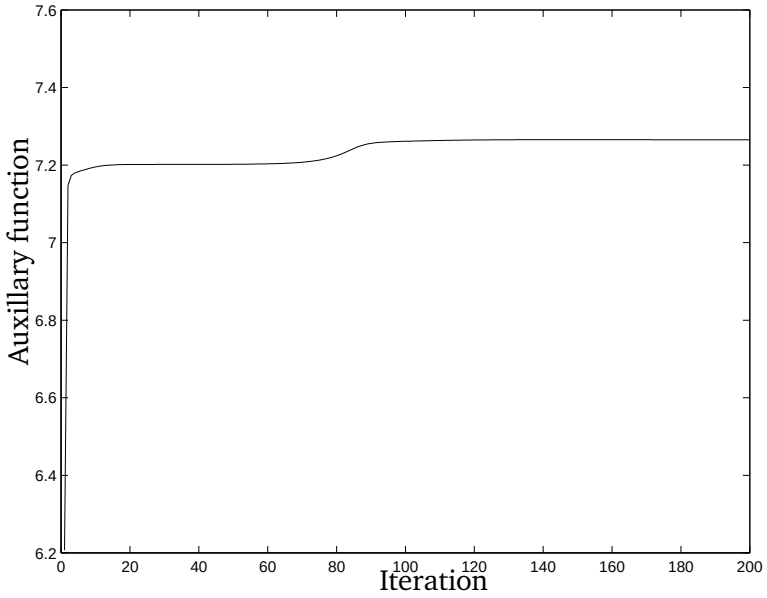


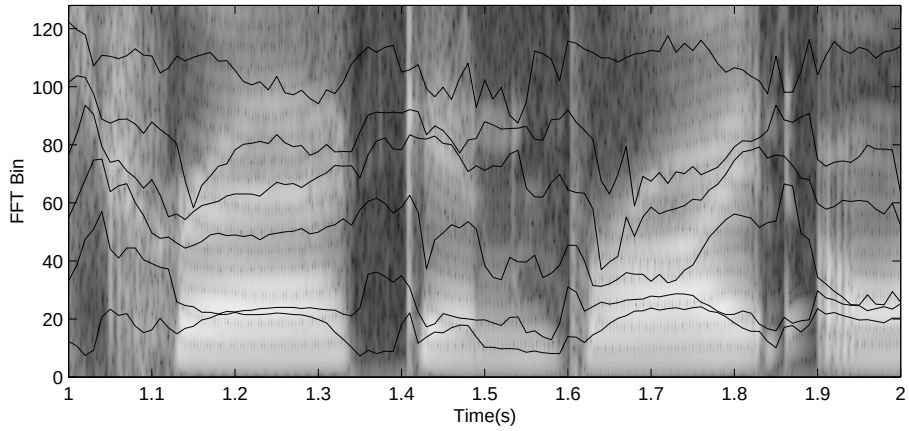
Figure 5.3 *Auxiliary function for 200 iterations, showing step in function*

Number Iterations	% WER
5	5.07
10	5.04
12	4.90
15	5.07
20	5.45
100	6.84

Table 5.6 *Number of iterations for a 4K GMM6 system*

mean positions and standard deviations were extracted. This was the best performing system so far out of the previous experiments. The results of varying the numer of iterations is in table 5.6. There is no significant variation in recognition performance until the number of iterations reaches 20. Additional iterations caused the error rate to increase. The optimal number of iterations was 12.

(a) 10 iterations:



(b) 100 iterations:

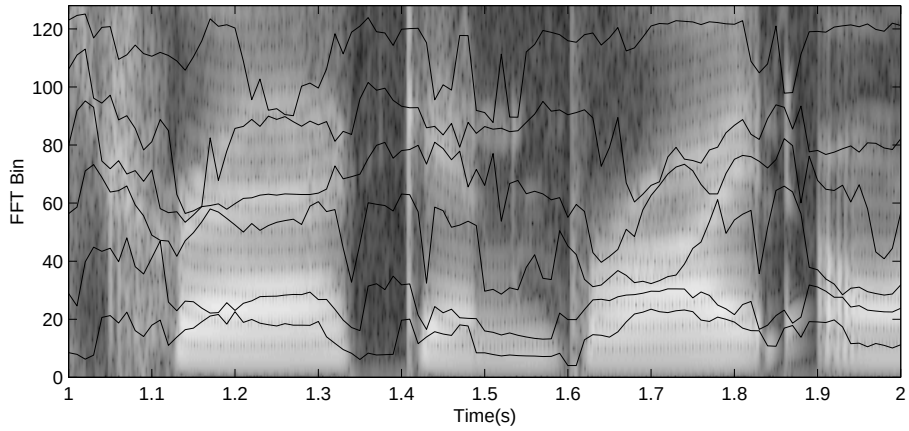


Figure 5.4 *Component Mean Trajectories for the utterance "Where were you while we were away?", using a six component GMM estimated from the spectrum and different iterations in the EM algorithm*

Rather than using a fixed number of iterations for the EM algorithm, it is possible to check the auxiliary function. Examining the auxiliary function $Q^{(i)}(\boldsymbol{\theta}|\hat{\boldsymbol{\theta}})$ after each iteration i it is possible to stop the EM algorithm if the increase in the auxiliary function is less than a certain fraction ρ of the previous iteration.

$$Q^{(i+1)}(\boldsymbol{\theta}|\hat{\boldsymbol{\theta}}) < (1+\rho)Q^{(i)}(\boldsymbol{\theta}|\hat{\boldsymbol{\theta}}) \tag{5.3}$$

This convergence criterion was applied to the EM process with several different values of ρ and the results are presented in table 5.7. The value $\rho = 0.0005$ gave the best performance and

corresponds to an average number of iterations similar to the optimal fixed number of iterations, 12. A small change in ρ will cause a large variation in the number of iterations taken. However, there is no significant variation in the WER between a system using a fixed number of iterations and one using a convergence criterion.

ρ	Average number of iterations	% WER
0.001	4	5.43
0.0005	13	5.10
0.0001	40	5.51
fixed	12	4.90

Table 5.7 *Results applying a convergence criterion to set the iterations of the EM algorithm, 6 component GMM system features on RM*

5.2.5 Prior distributions

The drop in performance when the number of iterations is increased could also suggest some inconsistencies or problems estimating the GMMs. Increasing the number of iterations used results in some large changes in the extracted parameters, as shown in figure 5.4 and 5.3. The plots of the component means show large discontinuities in regions where there are strong formant structures and the plots would be expected to be stable. The trajectories at 1.45s and at 1.8s in figure 5.4 show these discontinuities.

In section 4.2.2 a technique to incorporate a prior distribution during the GMM parameter estimation was discussed. The prior distribution was added on a per-component basis as a form of count smoothing. Adding a prior distribution during the extraction process should reduce the discontinuities and result in more consistent plots.

The prior distributions of the GMM component means were calculated from the full RM task training data. The global mean and variance of each of the GMM components was computed from a previous parameterisation of the data. This yielded a Gaussian prior distribution for the features. The parameter estimation was then performed adding the weighted prior distributions using the method in section 4.2.2.

Segments of speech parameterised using a prior weighting of 0.02 are shown in figure 5.5. The trajectories are smoother and more consistent than those observed in figure 5.4, but the locations of the means show less variation. Although the features obtained were smoother and more consistent, the estimates were less distinctive and provide less discriminatory information.

The prior distributions were added with different weights and the results are shown in table 5.8. The prior distributions prevented some of the problems associated with using higher numbers of iterations. Reduced error rates were achieved on systems using 100 iterations, although no improvement was gained over a system with fewer iterations and no prior distribution. Ap-

Prior weighting	0.00	0.005	0.01	0.02	0.05
12 iterations	4.90	5.30	5.44	5.60	5.63
100 iterations	6.84	6.51	6.21	5.96	6.08

Table 5.8 *Using a prior distribution during the GMM parameter estimation*

plying the prior distributions as a form of count smoothing effectively added a fixed observation on top of the speech data during the estimation process. Although adding the prior distribution gave smoother and parameter estimates, it appears that some of the discriminatory information is being lost. From these results, it appears that a more successful strategy is to limit the number of iterations rather than using explicit prior information during the estimation.

Another possible method to incorporate the priors would be to apply them using a phone-dependant basis. However, this would require a hypothesis of the phone class before incorporating the prior smoothing to the GMM estimation.

5.3 Temporal smoothing

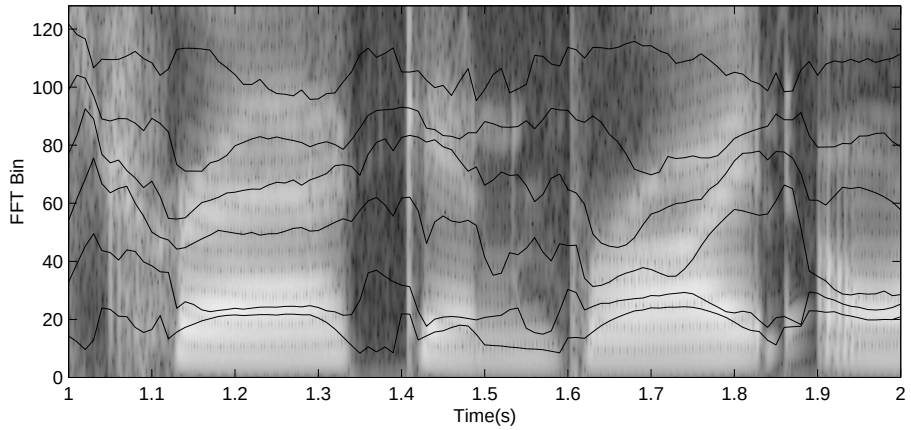
A technique was outlined in section 4.3 to incorporate the surrounding frames by performing a two-dimensional GMM estimation. By incorporating the surrounding frames it was hoped to obtain smoother and more consistent parameter trajectories. An alternative approach to smoothing the extracted features using a moving average filter was also proposed.

Taking a number of spectral frames around the current temporal frame, a two dimensional probability distribution can be formed. The second (temporal) dimension has the current input frame as the central frame with a number of frames taken around it. Once a 2-D histogram has thus been obtained, the EM algorithm can be used to estimate a two-dimensional GMM from the histogram. Using the surrounding temporal frames to estimate the GMM parameters rather than processing each frame seperately will enforce some smoothness on the extracted parameters

The 2-D GMM was initialised with a diagonal covariance matrix, and the GMM temporal mean was fixed to the central frame. The parameters extracted from the 2-D GMM were the spectral dimension means, the square root of the spectral element of the covariance matrix and the normalised log mean energies. Thus each feature was directly related to a corresponedng feature from the single dimensional GMM features. The temporal data were also windowed such that the frames around the central frame were deweighted.

Trajectories for the 2-D GMM system are show in figure 5.6. Using the 2-D GMM system gave smoother trajectories in the regions of voiced speech, for example around 1.2s in figure 5.6. However, there can be a tendency with the 2-D GMM estimates for the components to jump between stable regions and also to cluster together when a larger number of iterations are used, for example at 1.9s and 1.35s in the diagrams. The 2-D GMM system also gives large fluctuations in the mean positions in the regions of unvoiced sounds.

(a) 10 iterations:



(b) 100 iterations:

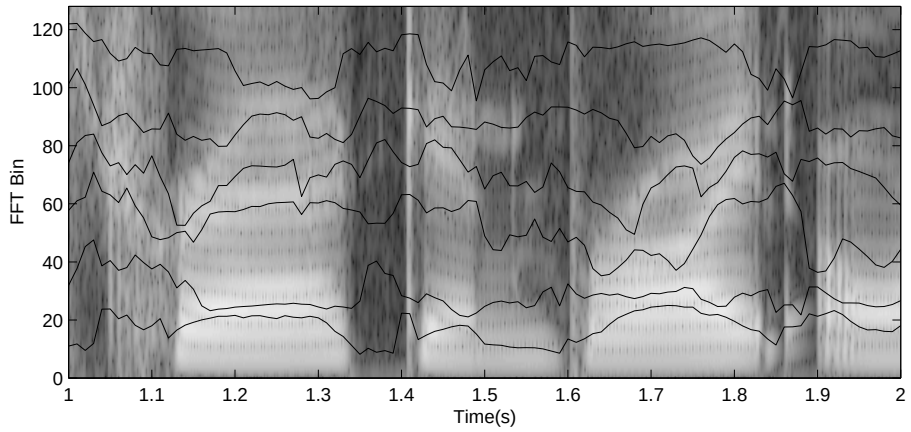
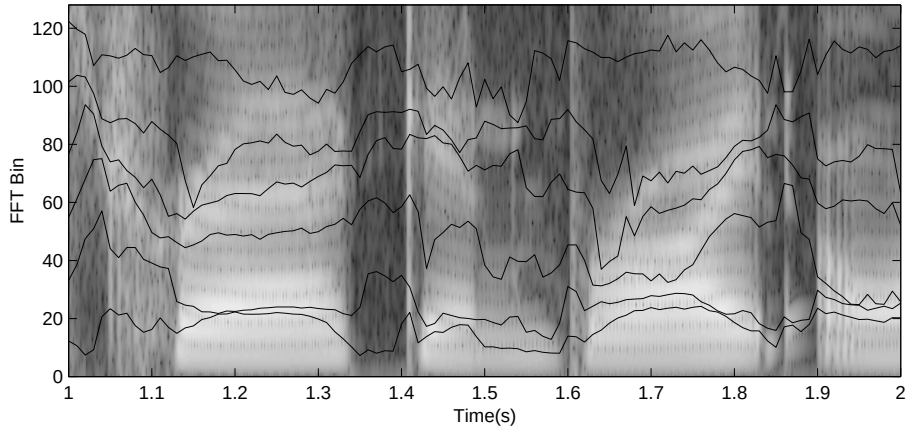


Figure 5.5 *Using a prior distribution model to estimate six GMM component mean trajectories from frames in a 1 second section of the utterance "Where were you while we were away?", using different iterations in the EM algorithm*

(a) 1 frame (1-D case) :



(b) 5 frames (2-D):

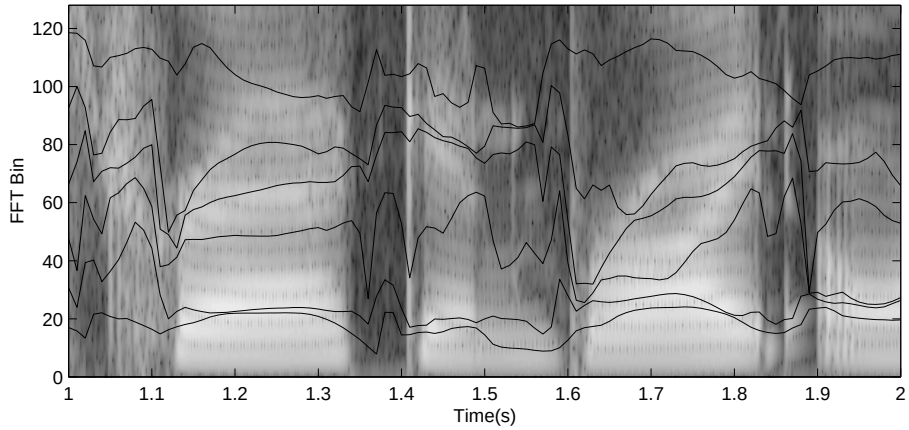


Figure 5.6 *GMM Mean trajectories using 2-D estimation with 5 frames of data from utterance "Where were you while we were away" with single dimesional case from figure 5.4a for comparison.*

To evaluate the performance of the two dimensional GMM features several different temporal frame widths were used in the estimation of the GMM parameters. Results on the RM task for the two dimensional estimates are presented in table 5.9. The RM data was parameterised using different numbers of temporal frames in a 2-D estimate. An additional experiment was performed using a 2-D GMM with three temporal frames, but with the frames surrounding the central frame windowed (or deweighted). The 2-D GMM systems gave poorer performance than the single-dimension GMM system in all cases. As the size of the window increased, so the error rate increased. In addition, there was a loss of temporal resolution from using the 2-D GMM estimates and hence discriminative detail was lost.

Rather than smoothing during the parameter extraction stage, the features can be smoothed directly after the estimation process using a moving average filter. Implementing a moving average filter of length $t = 3$ on the data gave drop in performance to 6.10% WER. This drop in performance is comparable with the degradation of the 3 frame 2-D GMM system. Hence, applying any form of temporal smoothing may yield more consistant trajectories, but will lose some of the resolution and discriminative properties of the features.

Temporal Frames	% WER
1 frame (1-D case)	4.90
3 frames	6.44
3 frames (+temporal window)	6.07
5 frames	6.79
7 frames	8.32

Table 5.9 *RM word error rates for different temporal smoothing arrangements on the GMM system*

5.4 Fisher ratios

In this section, all of the extracted GMM parameters were used as part of a speech feature vector. In this section, the elements in the feature vector are examined to compare the degree of discriminatory information each possesses. The measure of discriminatory information used is the *Fisher ratio*. The Fisher ratio is defined as the ratio of between-class variance to within-class variance.

Using Fisher ratios makes two assumptions about the distribution of the parameters. The first assumption is that the elements in the feature vector are uncorrelated. The second is that the features are Gaussian distributed within each class and that the within class covariances are the same.

The Fisher ratios were calculated for the best 4kHz system found in this section. This was the system presented in section 5.1.4 which extracted six Gaussian components from a 4kHz spectrum smoothed using a pitch-filter and extracted the normalised log-magnitudes at the means. The Fisher ratios were calculated on the full cross-word triphone system, using each component

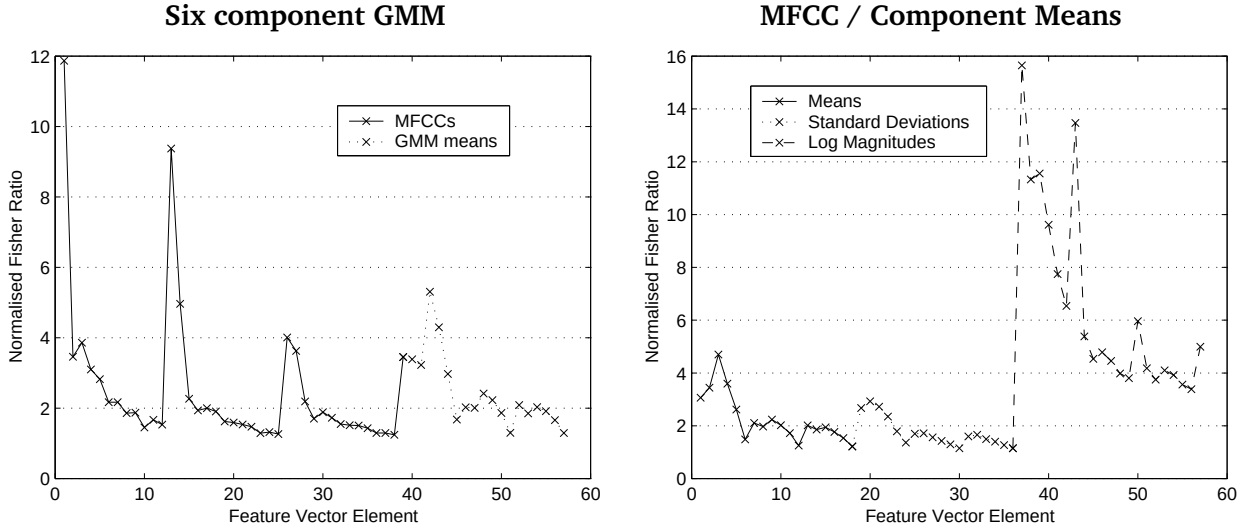


Figure 5.7 *Fisher ratios for the feature vector elements in a six component GMM system with a MFCC+6 component mean system for comparison*

mixture in the HMM output PDFs as a separate class. The results are in figure 5.7. The features are presented in the order $\{static, \Delta, \Delta^2\}$. The component log-magnitudes have the highest log-energies by far. However, there are very high degrees of correlation between the energy terms, as shown in table 4.1. The component means possess the next highest ratios, with the static terms in particular having high Fisher ratios. The standard deviation GMM features perform relatively poorly, especially the Δ and Δ^2 parameters.

As a comparison, the Fisher ratios for a system built with the GMM component means appended to a standard MFCC parameterisation is also shown. The MFCC features are also appended with a log-energy term. The component means have Fisher ratios lower than the first few cepstra and the energy term, but higher than the remaining cepstra. This suggest that the component mean features may be useful in combination with a MFCC parameterisation.

5.5 Summary

In this section a number of techniques and issues in estimating a set of Gaussian mixtures from a speech spectrum were presented. The optimal performance of a 4kHz band-limited signal was obtained by estimating six components with twelve iterations from a 4kHz spectrum smoothed with a convolutional pitch based filter. The most useful parameter set was formed by taking the component means, standard deviations and log-energy magnitudes at the means. For an 8kHz system, the optimal performance was achieved using the same smoothing technique and estimating five Gaussian components from the band up to 4kHz and one from the upper band. The technique for using a prior distribution gave more consistent parameter estimates, but the extracted features gave worse performance on the RM task. Estimating parameters from 2-D data using the surronding frames gave smoother parameter trajectories in voiced regions, but

did not do well in unvoiced regions and gave higher error rates overall.

6

Combining GMM features with MFCCs

Formants and formant-like features have been shown to be useful in combination with a MFCC parameterisation [55]. In this chapter, several methods for combining GMM features with a MFCCs are tested with concatenative and synchronous stream systems. The use of LDA feature space transforms to reduce the dimensionality and a semi-tied covariance matrix to handle possible correlations in the feature vectors are also tested and the results discussed.

Three speech recognition tasks are used to evaluate the performance of front-ends in this chapter. These are the medium vocabulary RM task, the large vocabulary Wall Street Journal (WSJ) task and the Switchboard corpus. All the RM systems were built from a flat started system as described by the HTK RM recipe outlined in appendix B.1. The exceptions to this are the synchronous stream systems, which were trained from the MFCC models. The WSJ and Switchboard systems were built using single-pass retraining from the MFCC or PLP baseline systems. As such, they contain the same set of states. This may not yield the optimal set of states for the target systems. However, this approach was used for simplicity on these more complex tasks.

6.1 Concatenative systems

In common with other formant-like front-ends, the performance of the GMM features alone was worse than a MFCC feature based system [111] [109]. However, formant information is considered to be complementary to MFCC features [57]. Other formant-like features have been successfully incorporated with MFCCs on small tasks [84] [108], and thus it seems likely that the mean positions from a GMM system can provide similarly useful information.

The simplest way to combine formant or peak information is to concatenate the MFCC and alternative features into a single feature vector as described in section 3.5.1. This has been used to successfully combine MFCCs with formants [56], gravity centroids [12] and band-pass filter-banks [83].

6.1.1 Adding features to MFCCs

Other types of spectral features have been shown to yield improvements when incorporated with MFCCs. A set of baseline experiments concatenating a MFCC parameterisation with different features (including Δ and Δ^2 parameters) were performed. The additional features were based on extra cepstra, formant means, PLPs coefficients and gravity centroids. The results are presented in table 6.1.

Additional Features	Number of parameters	%WER
None	39	4.19
Additional cepstra $\{c_{13} \cdots c_{16}\}$	51	4.29
4 PLP coefficients $\{p_1 \ldots p_4\}$	51	4.52
4 Formant frequencies from ESPS	51	4.89
4 Gravity Centroids	51	4.08
6 Gravity Centroids	57	5.02

Table 6.1 *Appending additional features to a MFCC system on RM*

One issue with appending formant-like features to MFCCs is that the dimensionality of the feature vector increases. As a comparison, experiments were run with a 16 Mel-cepstral representation. To compare the addition of other information sources, another system was built using 12 MFCCs combined with the first four PLP coefficients. Adding the higher order cepstral coefficients gave a slight degradation in performance, which is consistent with the belief that the standard MFCC parameterisation, using 12 cepstral coefficients gives the best representation of the speech [122]. Appending the first four PLP coefficients onto the MFCCs also gave no performance gains suggesting that although these are considered the most informative PLP coefficients, they add no complementary information to the MFCCs.

The gravity centroid parameters can be related to the GMM features. They have been successfully combined with MFCCs to reduce the WER on small tasks [16]. Experiments were performed to evaluate their performance with MFCCs on the medium vocabulary RM task. The gravity centroid features were calculated by splitting the 4kHz spectrum into rectangular subbands and calculating the first sub-band moment for each. A relative reduction in the the WER of 2.6% was obtained by adding four gravity centroids to the MFCCs. Adding six gravity centroids slightly degraded the performance of the MFCC features. This result differs slightly from results in other work with the gravity centroids on small vocabulary tasks which suggested that the optimal number of gravity centroids was six [16]. The difference may be due to the fact that standard delta parameters were appended to the parameterisation.

Formants are believed to be representative of the underlying spectral class. However, concatenating the MFCCs with a set of formant features has led to degradation on phone and digit recognition tasks [114] [57]. A state-of-the-art formant tracker was used to extract formant

frequencies on the RM task [103]. As described in section 3.3.2 this formant estimator uses a dynamic programming step to obtain smooth and consistent formant estimates. Four formants were estimated from the 4kHz spectrum. These estimates were combined with the MFCC parameters to give a new feature vector. Adding the formants estimated by the ESPS tracker yielded a degradation in performance.

6.1.2 Adding GMM features to MFCCs

The Fisher ratios for the GMM features were discussed in section 5.4. The component log-magnitudes had the highest Fisher ratios of the GMM features. However, the log-energy terms are highly correlated with the energy term already present in the feature vector. The next highest Fisher ratios were from the component means, and these compared favourably with the MFCC features.

To investigate the performance of adding the GMM means onto the standard MFCC feature vector was then augmented with the mean positions from four, five and six component GMM spectral estimates from a pitch-filtered 4kHz spectrum. Component means were used for two reasons. First, they can be related to the formant positions or gravity centroids which have been incorporated successfully in speech recognition systems. Second, these features had the highest Fisher ratios of all the GMM parameters studied, save for the energy features, which were highly correlated. The GMM component means will provide information about the spectral peak locations useful for discriminating between phone classes. The information about locations of spectral peaks is not directly available from the MFCC positions. The results of the concatenative GMM feature systems are shown in table 6.2.

Additional Features	Number of parameters	%WER
None	39	4.19
4 GMM Means	51	4.08
5 GMM Means	54	4.06
6 GMM Means	57	3.82
6 Means and 6 Variances	75	5.03
5 / 1 Means from split band at 4kHz	57	4.03

Table 6.2 *Concatenating GMM features onto a MFCC RM parameterisation*

Appending the GMM means from a six component estimate to the MFCCs gave a relative decrease in WER of 8.8%. Using the means from a four or five component estimate reduced the error rate by 2.6% and 2.4% respectively, a smaller improvement than using the six component means. This suggests that the GMM features may be complementary to the MFCC parameterisation. The decrease in WER relative to the MFCC system is significant at a confidence of 96%. A phone-level confusion matrix was examined for the MFCC and MFCC plus component means

systems. The gains in improvement tended to be spread over the vowel sounds and strong voiced sounds. The recognition of affricative and stop sounds was not improved by addition of the GMM component means to the feature vector.

Incorporating the component means from a split band GMM estimate with 5 components in the lower band and 1 component in the 4-8kHz band reduced the WER of an MFCC system. However, it did not outperform the MFCC system with six components estimated from a 4kHz spectrum. The mean of the GMM component estimated from the upper band did not add useful information to the MFCC system. The improvement in performance when adding features from estimating a component in the 4-8kHz band to a system based on a 5 component 4kHz GMM is most likely from the representation of the energy levels in the upper band.

The concatenative systems so far have added the extra parameters to the standard MFCC parameterisation. This yields an increase in the overall size of the feature vector. Rather than increasing the size of the feature vector an experiment was performed substituting the last four MFCCs by the four GMM means. This is similar to an approach used to incorporate MFCCs with formants [55]. Replacing the higher order cepstra gives a similar WER (4.14%) to the MFCC baseline with a similar number of features, and is an improvement over using only 8 MFCCs which gave a WER of 4.34%.

Optimal performance with the MFCC features was obtained using the GMM means from a six component fit. The gravity centroids gave their best performance when using four extracted means. This difference may be attributable to the restrictions the filter-banks in the gravity centroid system place on the locations of the peaks extracted. Thus the extracted parameters are less distinct. The gravity centroids have a strong prior on the location of each peak or centroid which restricts the ability of the features to adequately represent the spectrum. Hence, the GMM features have an advantage over the gravity centroid features.

6.1.3 Feature mean normalisation

Cepstral mean normalisation is a technique to remove the convolutional noise from a signal by subtracting the mean of the cepstral feature vector from the parameters, as discussed in section 3.2.2. This technique can equally be applied to other parameterisations than cepstra and its use is denoted here as feature mean normalisation to avoid confusion when it is being applied to the GMM parameters. Feature mean normalisation is a simple method of removing some speaker and channel effects from speech parameterisations on a per-speaker or per-utterance basis. Feature mean normalisation was applied to the systems built previously and RM systems were rebuilt from a flat start with the RM recipe as before.

It has been hypothesised that applying feature mean normalisation to formant features can have a vocal-tract length normalisation effect [114]. Subtracting the mean values from the GMM component mean features will remove any linear shifts on a per-utterance basis. As mentioned in sections 2.5.1 and 4.4.4, the effects of a vocal tract length variation may be modelled as a linear scaling of the frequency. Thus given a variation in the speaker, the GMM component mean

features will tend to be located higher or lower. Thus, subtracting the means of these features will remove some of the effects of the vocal tract length variation.

System Description	%WER FMN	%WER no FMN
MFCC	4.15	4.19
MFCC + 6 means	3.62	3.81
6 Component GMM	4.94	4.90

Table 6.3 *Using feature mean normalisation with MFCC and GMM features on RM task*

Results using feature mean normalisation are presented in table 6.3. There is no significant change in the WER of the MFCC system using cepstral mean normalisation on the RM task. Using feature mean normalisation on the full GMM set of features also yields no significant change from the results without the normalisation. This suggests that there is little benefit to be gained from removing the mean from a set of log-spectral features on the RM task. Mean normalisation on log-spectral features removes the effects of convolutional channel noise, and it seems likely there should be little effects from channel noise on the RM task.

Implementing mean normalisation on a system built with six GMM component means concatenated onto the MFCC feature vector gives a drop in WER of 13% relative to the MFCC features with CMN. A reduction in WER of 5.5% relative from a MFCC+6 means system with no mean normalisation was achieved. This improvement suggests that subtracting the means is having some normalising effect on the speakers in the system and boosting recognition performance.

6.1.4 Linear discriminant analysis

In section 3.5.1, the implications of concatenating extra parameters onto the feature vector were discussed. One disadvantage is that the size of the feature vector is increased, thus increasing the number of parameters to be estimated in the model. A solution to this would be to use a feature selection or projection scheme to remove the least discriminatory dimensions from the data. This requires an indication of the discriminative properties of each element in the feature vector. The Fisher ratio - the ratio of the within to between class covariances in the model - is one such measure. Linear discriminant analysis (LDA) was outlined in section 2.4.1 and is a projection scheme which attempts to maximise the between class covariance and minimised the within class covariance. Linear discriminant analysis has also been used to combine complementary features with an MFCC parameterisation [97]. The LDA generates an orthonormal transform matrix, and the directions of the vectors are based on the maximisation of the Fisher ratios. Thus, the lower LDA dimensions will have poor separation between classes and can be discarded.

The statistics for the within and between class matrices were generated on the cross-word triphone RM systems, with each component in the output HMM being regarded as a separate

Projected Dimensions	Features		
	MFCC	MFCC+6 Means	GMM6
10		8.34	10.13
20		5.99	7.36
30	4.90	5.18	6.89
39	4.49	4.88	7.25
50		5.07	7.06
57		5.26	8.07

Table 6.4 *RM results in % WER using LDA to project down the data to a lower dimensional representation*

class. These were used to compute the LDA transform. The RM systems were fully rebuilt using this transform on the features, truncating the vectors as required. Results from these experiments are in table 6.4.

The systems trained with an LDA transform performed significantly worse than the normal systems. LDA gave no improvement to an MFCC parameterisation on the RM task, although the MFCC system exhibited less performance degradation than the MFCC+6 Means system when LDA was applied. The LDA transforms seek to maximise the class separation by maximising the Fisher criterion. However, it has been shown that maximising the Fisher ratios will not necessarily increase the classification rate for speech recognition tasks [69]. Problems have also been noted using LDA for ASR systems if the extra feature vectors added are noisy or less useful for recognition. The assumption made by LDA that all classes share the same within-class covariance matrices is not a valid one. In addition, large amounts of data are necessary to generate robust transforms for LDA, and there may be insufficient data to robustly estimate transforms to separate the classes in the RM models.

6.2 Multiple information stream systems

Using a concatenative approach to combine the GMM means with an MFCC parameterisation improved the performance on the RM task in the previous section. Concatenating different features together assumes that the features were generated from the same data source. An alternative type of model is to consider the features as coming from separate information streams and combine them in a synchronous stream framework. This form of model allows different states, output distributions or numbers of components for each feature stream. In addition, different emphasis can be put on the feature streams. Thus, if a given stream is believed to be more or less informative, the stream weight can be correspondingly increased or decreased. As presented in section 3.5.2, the standard HTK method for doing so is to calculate the likelihoods of the individual information streams separately. The likelihood scores are then given different weights and the scores are combined together at the state level [122]. All the streams are constrained to have the same time-state alignment.

Synchronous stream systems were built with the first stream $\mathbf{y}_1(t)$ containing the first twelve MFCCs $\mathbf{c}(t)$, normalised log energy $r(t)$ and their respective dynamic parameters:

$$\mathbf{y}_1(t) = \begin{bmatrix} \left[\mathbf{c}^T(t), r(t)\right]^T \\ \left[\Delta\mathbf{c}^T(t), \Delta r(t)\right]^T \\ \left[\Delta^2\mathbf{c}^T(t), \Delta^2 r(t)\right]^T \end{bmatrix} \tag{6.1}$$

The second stream $\mathbf{y}_2(t)$ contained an alternative parameterisation and related dynamic parameters.

Three additional parameterisations of the speech were considered:

1. GMM features from a six component spectral estimate of a pitch filtered 4kHz spectrum (MFCC+GMM6);

$$\mathbf{y}_2(t) = \begin{bmatrix} \left[\boldsymbol{\mu}^T(t), \boldsymbol{\sigma}^T(t), \tilde{\mathbf{e}}^T(t), r(t)\right]^T \\ \left[\Delta\boldsymbol{\mu}^T(t), \Delta\boldsymbol{\sigma}^T(t), \Delta\tilde{\mathbf{e}}^T(t), \Delta r(t)\right]^T \\ \left[\Delta^2\boldsymbol{\mu}^T(t), \Delta^2\boldsymbol{\sigma}^T(t), \Delta^2\tilde{\mathbf{e}}^T(t), \Delta^2 r(t)\right]^T \end{bmatrix} \tag{6.2}$$

2. Component means from a six component GMM estimate from the spectrum (MFCC+6MEAN);

$$\mathbf{y}_2(t) = \begin{bmatrix} \boldsymbol{\mu}(t) \\ \Delta\boldsymbol{\mu}(t) \\ \Delta^2\boldsymbol{\mu}(t) \end{bmatrix} \tag{6.3}$$

3. For comparative purposes, a PLP parameterisation using the first twelve coefficients: $\mathbf{p}(t) = [p_1(t), \ldots, p_{12}(t)]^T$ and the zeroth cepstrum $c_0(t)$ (MFCC+PLP);

$$\mathbf{y}_2(t) = \begin{bmatrix} \left[\mathbf{p}^T(t), c_0(t)\right]^T \\ \left[\Delta\mathbf{p}^T(t), \Delta c_0(t)\right]^T \\ \left[\Delta^2\mathbf{p}^T(t), \Delta^2 c_0(t)\right]^T \end{bmatrix} \tag{6.4}$$

The systems were trained using the stream weight of the alternative features $y_2(t)$ set to zero throughout the training proceedure, so the HMMs were built using only the MFCC features for alignment in the Baum-Welch algorithm. This approach was taken as the optimal stream weight was not yet determined, and the MFCC features outperformed the GMM features on the RM task. Hence the MFCC features were used to provide the alignments. All three systems were built using the MFCC context decision tree. Hence, the MFCC stream in the model sets is identical to the MFCC model trained alone. The models were tested on the RM task with a range of stream weights, with the sum of the stream weights constrained to sum to one. The decision tree and state-component alignments used during training will be the optimal for the MFCC feature stream. They will not be the optimal for the second feature stream.

The performance of the three systems with the MFCC stream weight varied from 0 to 1 is shown in figure 6.1. The baseline MFCC performance is obtained when the MFCC stream is

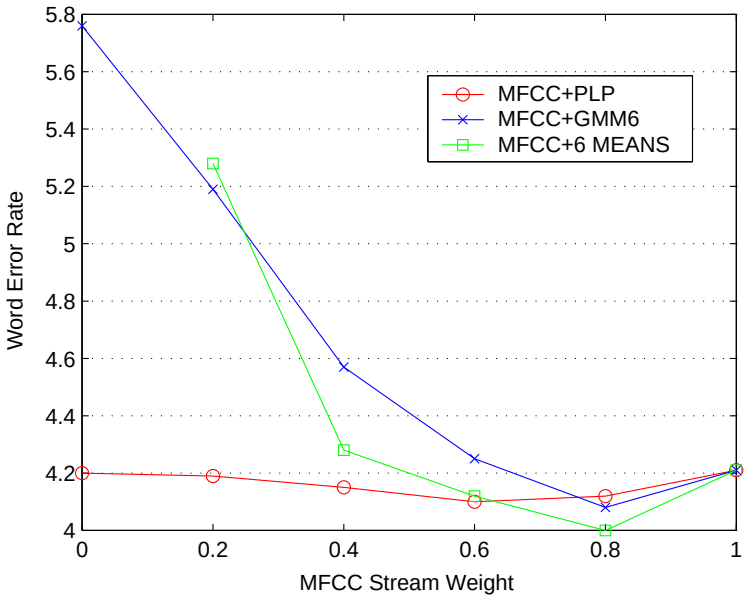


Figure 6.1 *Synchronous stream systems on RM with various stream weights, stream weights sum to 1*

given a weight of 1 and only the MFCC stream is used for recognition. This is because the model set was built using only the MFCC features in the probability calculations. The MFCC+6MEAN system with an MFCC stream weight of 0.8 gave the best performance. This system had a word error rate of 4.00%, or a relative improvement of 4.5% over the baseline MFCC performance with a confidence in the improvement of 99%. The MFCC+GMM6 with an MFCC gave a performance of 4.10% with a MFCC stream weight of 0.8, for a relative drop in WER of 2.2%. This improvement is not significant.

Incorporating MFCC and PLP features in a synchronous stream system gave an error rate of 4.11% with an MFCC stream weight of 0.6. This shows no significant improvement which suggests PLP and MFCC systems contain little or no complementary information. This not surprising as the PLP features are based on a similar (Mel-smoothed) representation of the spectrum.

Using a synchronous stream system to combine the MFCC features with the GMM means resulted in a 5.0% relative higher WER than concatenating the two into a single feature stream. This difference could be due to the assumption made by the stream model that the feature streams are independent. With the synchronous streams system the output distributions for each HMM state are independant. The distributions used above used GMMs with six mixture components for each stream in each state, although the number of components does not have to be the same.

Combining the full GMM features with the MFCCs using a synchronous stream system improved upon concatenating the two together. The difference is that the full GMM system had a much larger dynamic range than the MFCC system. The highly correlated energy terms in particular contribute to this. If an approximation is made that the dynamic range is roughly proportional to the optimal language model scale factor, then the dynamic range of the full GMM

system is roughly 2.5 times that of the MFCC system. The GMM means alone have a dynamic range of about 0.5 that of the MFCC system. Hence, using a synchronous stream system allows the GMM features to be deweighted so the MFCC features, which have a better general performance, can contribute more. Retraining the systems using the optimal stream weights of 0.8 and 0.2 as opposed to using only the MFCC features gave a WER of 3.98%, and no significant improvement in performance.

Using a MFCC stream weight of zero uses only the alternative feature set. The recognition results in this case are slightly worse than those of the baseline systems. Using only the GMM6 stream gave a WER 10% relative higher than a system built independently on the GMM6 features. This is due to the fact that the synchronous stream system uses the MFCC alignments to train the observation probability density functions and transition matrices. The MFCC state-component alignments will not be the ML solution for the other features. Using only the GMM6 means, an error rate of 9.23% was obtained with a total feature vector length of 18, which compares favourably with other systems using only formant features [111].

6.3 Combining MFCCs and GMM features with a confidence metric

In the previous section, using a fixed value for the weights was found to provide no significant gain over a concatenative system. However, other work with formant features [114] has obtained improved performance by using a measure of confidence of the assigned formant location to set the weight of the formant stream. A similar measure of confidence can be obtained from the GMM parameters for each frame directly as shown in section 3.5.4. A confidence metric, $\xi(t)$ was derived from the component energies and standard deviations, which would be high in regions with strongly defined peak structures. For an M-component GMM feature the confidence measure is given by:

$$\xi(t) = \beta \left[\prod_{m=1}^{M} \frac{\tilde{e}_m(t) + 10.53}{\sigma_m(t)} \right]^{\frac{1}{M}} \tag{6.5}$$

The GMM stream weight can then be set to be proportional to the confidence metric. Hence, in regions with strong formant-like structures, more of the likelihood score will be based on the GMM means.

To evaluate the performance of the confidence metric, a two-stream system was built. The first stream contained the MFCC features plus energy and the second stream the means from a GMM estimate from a pitch filtered 4kHz spectrum. This is the same as the MFCC+6MEAN system detailed above.

The stream weight of the GMM component means $\gamma_2(t)$ was set to zero during training, hence the MFCCs alone were used for alignment during the Baum-Welch training. A set of models were trained on the RM data and tested using the confidence metric to set the stream weights. As detailed in section 4.4.3 the ratios of the dynamic ranges can be approximated to

the ratio of the language model scale factors. For the GMM component means and the MFCC parameters this ratio is 2.0. The stream weights were set as:

$$\gamma_1(t) = 1 - \xi(t) \tag{6.6}$$

$$\gamma_2(t) = 0.5\xi(t) \tag{6.7}$$

Results using a range of scale factors for the confidence weight $\xi(t)$ are shown in Table 6.5.

Confidence weight β	% WER
0.0 (MFCC system)	4.19
0.1	3.95
0.2	3.94
0.3	4.12
0.4	4.32
MFCC+6Mean (concatenative)	3.81

Table 6.5 *Synchronous stream system with confidence weighting*

The confidence metric gives a 6% relative reduction in WER relative to the MFCC baseline system with a value of $\beta = 0.2$. This improvement over the MFCC baseline is significant at a confidence of 92%, and is an improvement over the synchronous stream with fixed stream weights at a confidence of 99%. Using a confidence metric to combine the information streams gives a better result than using fixed stream weights. However, it does not improve the performance of a MFCC and GMM component mean concatenative system. At this scale factor, the average value for the confidence metric is roughly 0.2. This is similar to the optimal value for the synchronous stream system with a fixed stream weight. The confidence stream system also gives a small but not significant improvement over the synchronous stream system with fixed stream weights which had a WER of 4.00%.

The confidence weights can also be used in training as well as testing, rather than using $\gamma(t) = 0$ for the state likelihood calculations. The confidence metrics for the training data were calculated. The stream weights in equations 6.6 and 6.7 were substituted into the emission probability calculations with the scale factor β set to 0.2. Retraining the data in this fashion and testing using the confidence metric to combine the scores yielded a WER of 3.95% on the RM task. Hence, no significant improvement was achieved by using the confidence metric during training.

In conclusion, although the confidence metric gave a small improvement over a synchronous stream system, there was no performance improvement over a system using the MFCCs and GMM means concatenated into a single feature vector.

Description	% WER
MFCC	9.75
MFCC+6 Means Concatenative	9.56
MFCC+6 Means fixed Stream weights	9.64
MFCC+6 Means confidence metric	9.52
GMM6 system	12.43
GMM6 system with mean normalisation	12.02

Table 6.6 *Results using GMM features on WSJ corpus and CSRNAB hub 1 test set*

6.4 Wall Street Journal experiments

The performance of the features was also investigated on a large vocabulary task, the Wall Street Journal (WSJ) corpus. Evaluation was performed on the CSRNAB Hub 1 test set. The WSJ corpus is based on extracts read from the Wall Street Journal. The SI-284 corpus, using 284 training speakers in approximately 60 hours of data was used to train the models. Further details can be found in appendix B.2.

Systems were built on the WSJ task using different feature parameters:

MFCC, a baseline MFCC system;

GMM6, system using the GMM means, standard deviations and log-magnitude terms from a six-component spectral estimate.

MFCC+6Mean concatenative, a concatenative feature vector formed from the GMM component means and the MFCCs in a single stream;

MFCC+6Mean fixed stream weights, a synchronous stream system using MFCCs and GMM component means as two synchronous feature streams with the stream weights fixed at 0.8 and 0.2 respectively;

MFCC+6Mean confidence metric, a synchronous stream system using a time-dependent confidence weight to combine the MFCC and GMM means feature streams;

The systems were built by single-pass retraining the MFCC model sets for the new features. The same context decision tree and set of states from the MFCC system was used in all the models. The synchronous stream systems were built using only the MFCC stream during training.

The GMM parameters were extracted using six components on a 4kHz spectrum smoothed with a pitch filter. The MFCC and synchronous stream systems used 12 component output PDFs for each HMM state, the concatenative system 16. The increased optimal number of mixtures could be required to model the correlations in the GMM features. Results for the systems based on rescoring the MFCC lattices on the CSRNAB hub 1 test sets are presented in Table 6.6.

The GMM features alone had a WER 23% higher than the MFCC parameterisation, albeit at a lower spectral bandwidth. Using feature mean normalisation decreases the WER by 2.2%

relative on the task. Since the component log-magnitudes have already been normalised during the extraction process, the improvement seen here is presumably due to the effect of normalising the component means. Normalising each component mean over an utterance removes any offset or linear bias that it may possess. If taken over sufficient data, this has the effect of acting as a speaker or utterance normalisation, similar to a vocal tract length normalisation as mentioned before in section 4.4.4 [114].

Adding the GMM component means to the feature vectors decreases the WER by 2.0% relative, an improvement which is not significant. Using the confidence metric to combine the features in a streaming system produces a small improvement compated to the performance of a system with fixed stream weights of 0.2 and 0.8. Combining the features in separate information streams with a confidence metric also gives a slight improvement. However, the performance increase over the concatenative system is not significant. Adding the GMM component means to the MFCC parameters on a large task gives a slight but not significant improvement to the system.

Results on the WSJ task track the results on the RM task. The GMM features performed 17% relative worse on RM and 23% worse on the WSJ corpus. The relative improvement in WER gained by using the GMM features in combination with MFCCs and feature mean normalisation was 2.0% relative on the WSJ corpus and 13% on the RM task.

Combining the MFCCs with the GMM features on WSJ gave relatively poor performance compared to the results on the RM task. This could be attributable to a number of factors. The state clusterings used were those generated for the MFCC features and may not have been optimal. It may be that the GMM features do not generalise well onto larger tasks and represent the classes poorly. Another possibility is related to the effects of cepstral mean normalisation. On the RM task applying cepstral mean normalisation gave no significant performance gains. However, on the WSJ task applying cepstral mean normalisation to the MFCC features gives a significant gain. Although the WSJ task has little environmental or channel noise, CMN can the remove the effects of speaker bias or spectral tilt. The extraction of the GMM features does not incorporate this normalising effect and hence the features may be giving relatively poorer performance on this task.

6.4.1 Semi-tied covariance matrices

One problem with the GMM features is that they possess a large degree of correlation. It could be possible to generate full covariance matrices in the HMM output PDFs to handle the correlations. Another method for modelling correlations is to use a semi-tied covariance matrix. The use of a semi-tied covariance matrix was discussed in section 2.4.2.

Semi-tied transforms are a form of covariance modelling with full or block-diagonal covariance matrices tied over multiple classes [36]. The matrices can be tied over all phones or certain phone classes and can be grouped into separate blocks of features as well.

Global covariance transforms were generated on the WSJ corpus. The transforms were es-

Form of block structure	Features		
	MFCC	MFCC+6Mean	GMM6
None	9.75	9.56	12.28
Features+Δ	N/A	9.13	11.94
Δ	9.03	9.55	12.90
Features	N/A	8.99	11.85
Full	8.85	9.67	13.02

Table 6.7 *WSJ results giving % WER using global semi-tied transforms with different block structures for different feature sets*

timated on the WSJ model sets, and then two further passes of EM training on the data were performed. The semi-tied transforms were tested with the transformed model sets and the word insertion penalties and language model scale factors were not altered. Different block diagonal structures for the semi-tied transform were considered, grouping features by type (component means, variances, component magnitudes or MFCCs), static and dynamic parameters, or both together.

The results of the experiments using these transforms are presented in table 6.7. Using a full transform with the GMM feature system increased the WER by 6% relative, compared to the 9.2% decrease in error observed when used with the MFCC system. Although the error rate went up, an increase in log likelihood was observed in the training data. Implementing a full semi-tied transform with the MFCC+6 GMM means system increased the error rate slightly as well. Constraining the semi-tied transform to a block-diagonal structure based on feature type led to improved performance in the case of the GMM feature system and in the GMM means in combination with the MFCC parameters. The best performance with the concatenative system was gained by using two blocks, one with the MFCCs and one with the GMM component means. However, the performance was still slightly lower than the baseline MFCC system with a full transform.

It can be concluded that the although a log-likelihood increase can be observed in the training data using a semi-tied feature-space transform, it does not significantly improve the results using GMM features. The only systems which showed a slight improvement with the MFCC features were when a block diagonal structure was used to split the features into separate blocks. The GMM features possess a high degree of correlation but appear not suited to the approach of the semi-tied covariance matrix.

6.5 Switchboard experiments

Combining MFCCs with GMM features on the Wall Street Journal gave a smaller relative gain than the corresponding experiments on the RM task. To explore the effect of combining MFCCs with GMM features on larger speech corpora, experiments were performed on the large vocab-

ulary Switchboard corpus.

The Switchboard corpus is a large corpus based on conversational telephone speech from north American speakers [42]. The speakers were asked to converse either freely or on given topics, and the speech was recorded at 8kHz. The speech can come from landlines or cellular connections. Due to the nature of the telephone channel, the effective frequency range of the speech is 125-3300kHz. The speech has been recorded in stereo and μ-law compounded with a resolution of 8 bits per sample. An echo cancellation algorithm has also been applied.

The experiments were run on a 68 hour training set *h5train03sub*. The training set contained data from 1118 conversation sides. The training data contained information from both normal and cellular calls.

The *h5train03sub* data was coded using PLPs normalised with a vocal tract length warping factor found for each speaker using a maximum-likelihood Brent estimation [49], and both cepstral mean and variance normalisation were used. The baseline PLP system used a model generated from the full (200+ hours) training data for the 2002 CU-HTK evaluation system[1]. This model was mixed down to have single component Gaussians in the output PDFs. The states were reclustered to yield a model with roughly 6000 unique states with single component PDFs. The models were iteratively re-estimated and the number of Gaussian components per state gradually increased. The number of Gaussian components in the output PDFs in the final model was twelve. The baseline system was then evaluated on the *dev01sub* subset of the *dev01* test set, which contains data from the cellular and normal call databases. The language model used was a 58K backoff trigram model, as used in the CU-HTK evaluation systems [48]. Testing was performed using a Viterbi search for the most likely word sequence (as opposed to the lattice rescoring used for the WSJ experiments). The WER achieved with the baseline system was 36.8%.

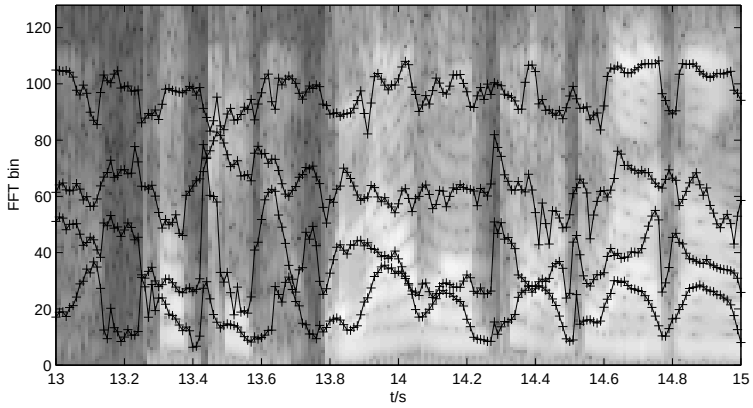


Figure 6.2 *GMM component mean features for a section of the data from the SwitchBoard corpus*

The GMM mean features were then evaluated in combination with the PLP features. The feature vectors were formed by concatenating the VTLN PLP coeffiecients with the GMM component

[1] see [48] for a related description

mean features to give a feature vector of length 57. Cepstral mean and variance normalisation was then applied to these features. The single component model with 6,000 states trained on the h5train03sub PLP data was then single pass retrained for the PLP+6Mean data. The model thus obtained was then iteratively re-estimated and the number of components in the HMM output PDFs was gradually increased. The system was tested as above, with increased beamwidths to account for the increase dynamic range to give roughly the same search time. The WER obtained using the system was 39.0%. Although the values extracted from the data appear reasonable, as shown in figure 6.2, a significant degradation in performance is obtained including them on this task. Examination of the Fisher ratios also suggests that the GMM features possess discriminatory information on this task. As mentioned in section 6.4, there may be a number of reasons why the GMM features gave poorer performance when combined with the PLP features on this task. The lack of a log-spectral (or cepstral) mean normalisation for the GMM features may be affecting the performance when combined with PLP features which do incorporate it. Experimental results show that implementing CMN on the PLPs improves the performance by around 7% on the Switchboard task. Hence, if some form of log-spectral normalisation could be implemented on the GMM features, the features extracted may perform better on the task. Alternatively, it may be that the GMM features do not generalise well for more complex large vocabulary tasks. The features may not be distinct between classes, or they may not be consistantly estimated. Another possibility is that the features are performing badly in the complex noisy environmental conditions of Switchboard.

6.6 Summary

In this chapter, results combining the features from a GMM estimated from the spectrum with an MFCC parameterisation have been presented. Specifically, the experiments focused on combining the GMM means - which can be compared to the formant positions - with the MFCCs. On the medium vocabulary RM task, appending the GMM means to MFCC features gives an improvement in WER of 8.8% relative over the MFCC system, and an improvement in WER of 13% relative when feature mean normalisation is applied. Using a synchronous stream system with a confidence metric to combine the parameterisations gives a small improvement over the MFCC parameterisation, but did not beat the performance of the concatenative system. Results on the larger WSJ task tracked the results on the RM corpus, but the improvements were not as large or as significant. Using an LDA transform on a concatenative system gave a drop in performance on the RM task, as did using a semi-tied covariance matrix on the WSJ corpus. Combining the GMM component means with PLP features on the Switchboard corpus gave a relative degradation in performance as well. This suggests that the GMM features perform poorer on complex tasks, and this may be due to the lack of log-spectral - or cepstral - mean normalisation with the GMM features.

7

Results using noise compensation on GMM features

In this section the behaviour of the GMM features in a noise-corrupted environment is discussed. The performance of models using GMM features in mismatched conditions is shown. Results using system using GMM features in noise-matched conditions are also shown. Experiments using the noise compensation techniques in section 4.5 are presented.

The noise corrupted speech in this section is formed by adding random segments of the Noisex database sound "Operations room" to the test data at the waveform level. This form of artificial noise corruption does not take into account other effects of recording speech in noise-corrupted environments such as the Lombard stress. However, it allows easier comparative evaluation of systems and training in a noise matched environment can be performed using single pass retraining methods.

All the noise compensation techniques discussed assume that a noise model is available. In this work, the noise model parameters were estimated by taking the average of a GMM parameter estimate of the noise source. In practice, this could be estimated using a voice activity detector on the corrupted speech signal.

7.1 Effects of noise on GMM features

This aim of this chapter is to evaluate the performance of the noise robustness techniques presented in section 4.5. Work with spectral peak features has shown that they possess some inherent noise robustness in white noise and car noise [31]. However, little or no improvement was observed when using spectral peak features on coloured noise (i.e. possessing a defined peak structure) such as factory noise or background noise [12]. The interfering noise source chosen in this section is the "Operations Room" (Op-Room) noise from the Noisex database. Previous work has shown that this form of noise severely corrupts MFCC parameters [40]. Figure 7.1 shows plot of the average noise spectrum of the OpRoom source. In addition, a GMM plot of a clean spectrum and one with additive Op-Room noise at a 18dB signal to noise (SNR) ratio is shown. This noise source was chosen because it will severely corrupt both the MFCC and GMM parameters. The Op-Room noise is coloured and possesses a strong low frequency spectral peak.

A spectral peak representation of the corrupted speech signal will model the noise rather than the speech in the low frequency regions.

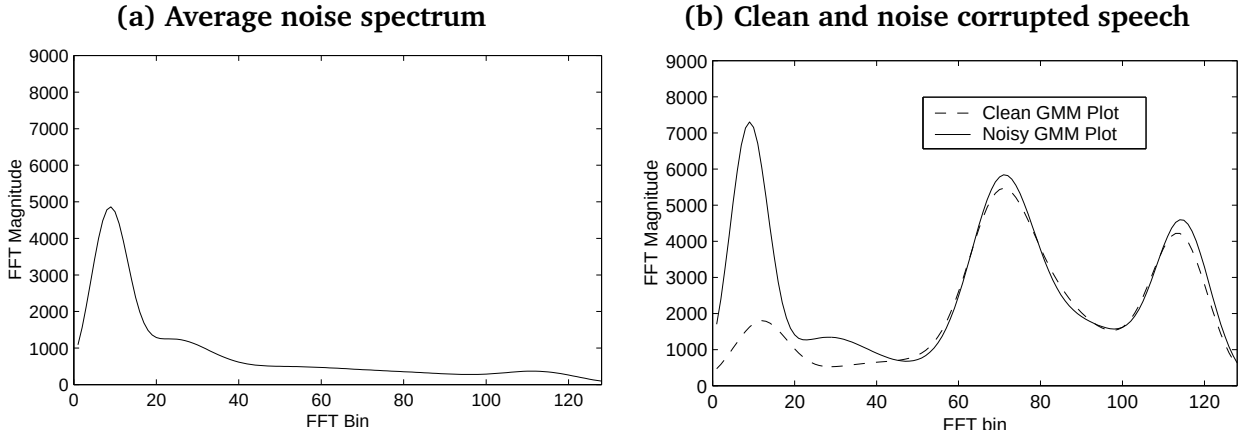


Figure 7.1 *Plot of average Op-Room noise spectrum and sample low-energy GMM spectral envelope corrupted with the Op-Room noise*

In figure 7.2 the component means for a section of an utterance have been plotted. The configuration is the same as that used for figure 5.4(a). During periods of high energy, the component mean trajectories extracted change very little from those in clean speech. However, in the periods of lower energy, the mean positions, especially those of the lower order components are severely corrupted.

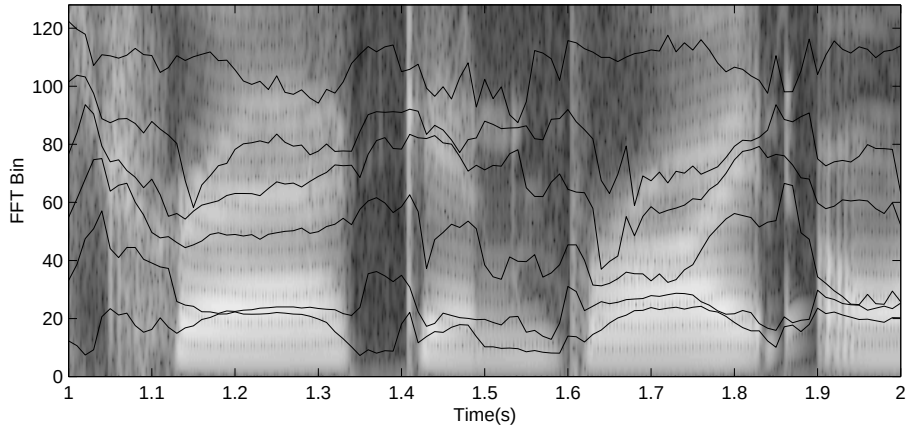
7.1.1 Model distances

Since the relationship between the spectrum and the extracted parameters is non-linear, the effects of additive noise one the elements in the feature vector will not be straightforward. However, it would be useful to examine the degree of noise corruption of the various elements in the feature vector. In this section the corruption of the elements of the feature vector is found by considering the difference between a model based on clean speech and one trained on noise corrupted data.

There are a number of different measures of closeness of two model sets, based on distance measures of the underlying distributions [66]. If the noise corrupted model set is built using a single pass retraining step of the clean model, then it will possess the same set of states, transition matrices and component priors. When evaluating the distance between the model sets, it would be preferable to use a measure based only on the parameters which have been altered - in this case the means and variances in the HMM output PDFs. Thus the KL distance between pairs of state/component Gaussian distributions can be considered rather than between complete models [33]. Using this approach it is possible to compare the distance of each parameter in the feature vector between the clean and noise corrupted feature sets.

The KL distance between two Gaussian distributions p and q with means μ_p and μ_q and

(a) Clean speech:



(b) Noise corrupted at 18dB SNR:

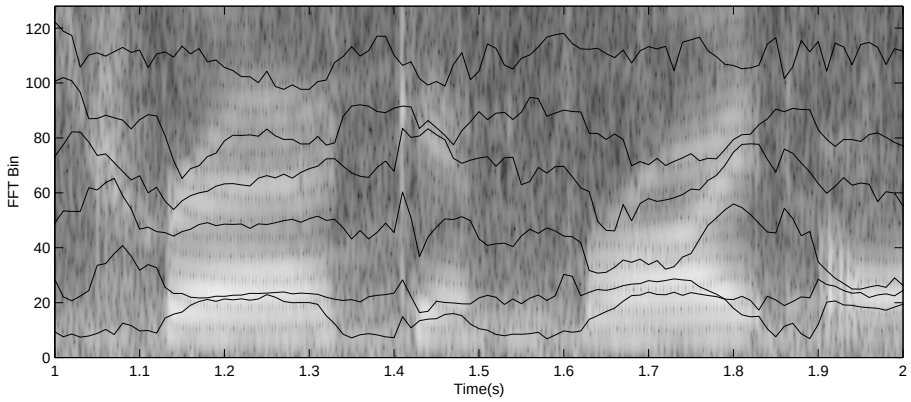


Figure 7.2 *GMM Mean trajectories in the presence of additive Op-Room noise for the utterance "Where were you while we were away" (cf fig 5.4)*

variances σ_q^2 and σ_q^2 is given by:

$$\mathcal{D}_{KL}\left(\omega_p, \omega_q\right) = \frac{1}{2}\left[log\left(\frac{\sigma_q^2}{\sigma_p^2}\right) + \frac{(\mu_p - \mu_q)^2}{\sigma_q^2} + \left(\frac{\sigma_q^2}{\sigma_p^2} - 1\right)\right] \tag{7.1}$$

And the average KL distance between two complete HMM model sets $\mathcal{M}$ and $\hat{\mathcal{M}}$ is taken as:

$$\bar{\mathcal{D}}_{KL}(p,q)(\mathcal{M}, \hat{\mathcal{M}}) = \frac{1}{P}\sum_{p=1}^{P}\mathcal{D}_{KL}\left(\mathcal{M}(\mathcal{M}(\omega_p), \hat{\mathcal{M}}(\omega_p)\right) \tag{7.2}$$

where $\mathcal{M}(\omega_p)$ is the p^{th} model in the model set $\mathcal{M}$ and P is the total number of mixture components in each model.

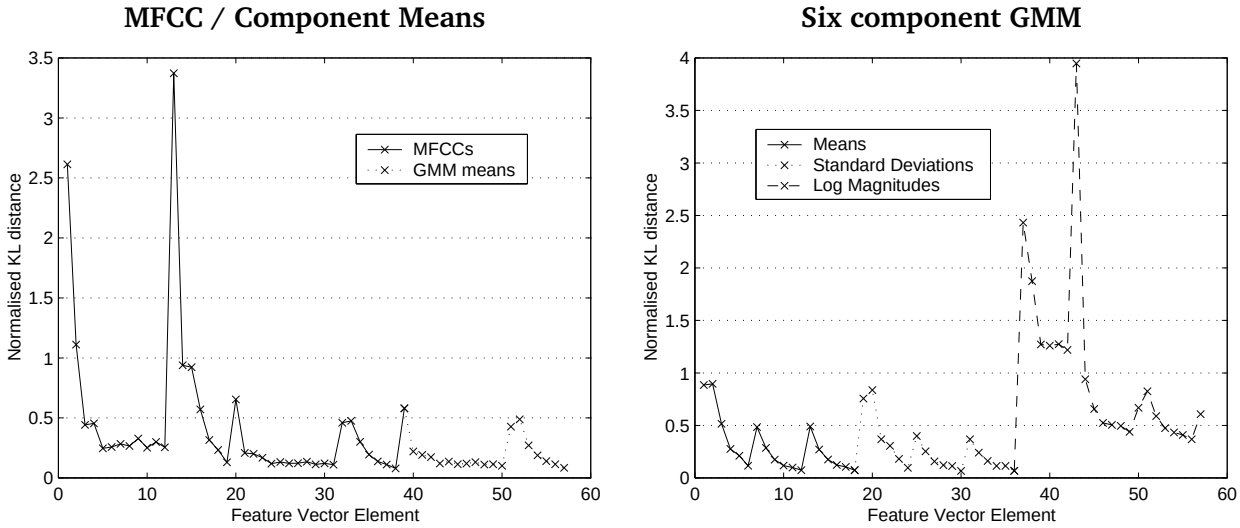


Figure 7.3 *KL model distances between clean speech HMMs and HMMs trained in noise corrupted environments for MFCC + 6 GMM component mean features, and a complete GMM system*

Figure 7.3 shows the KL distances between the clean model sets and those single pass retrained on noise corrupted data on the RM task. The KL distances for each element of the feature vector are given, where the features are presented in the order $\{static, \Delta, \Delta^2\}$.

The GMM component means positions in the MFCC+6Mean feature vector are corrupted less than the log-magnitude term and the first MFCCs, but worse than the other parameters. Due to the coloured nature of the noise the lower order means, corresponding to the lower frequency regions, have been worst affected. The parameters for higher order GMM component means are relatively close to those of models trained in noise matched conditions. In the GMM6 system, the standard deviations are corrupted to a similar degree to the component means. The log-magnitude terms in the feature vector are by far the worse affected by the noise.

7.1.2 Performance of uncompensated models in noise corrupted enviroments

In order to initially explore the performance of GMM features in noise, four systems were built. These were the same used in sections 6.4, namely:

MFCC, a baseline system using the standard MFCC parameterisation;

GMM6, GMM parameters from estimating six Gaussian components to a 4kHz spectrum smoothed using pitch filtering - the component positions, standard deviations and normalised log mean energies were used;

MFCC+6Mean concatenative, a feature vector formed by concatenating the MFCC features together with the component means from the GMM spectral estimates from the GMM6 system;

MFCC+6Mean Confidence Metric, a two stream synchronous stream system using the MFCCs and GMM6 component means in independent streams. The stream weights are time-dependent, set to the confidence metric described in section 4.4.3.

The systems were tested on data with additive Op-Room noise at a SNR of 18dB and the results are shown in Table 7.1 and Figure 7.4.

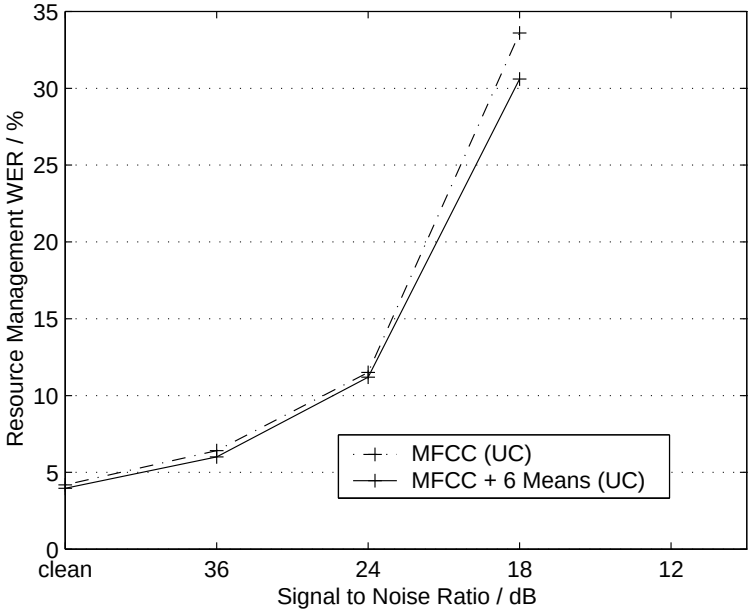


Figure 7.4 *WER on RM task for uncompensated (UC) MFCC and MFCC+6Mean systems on RM task corrupted with additive Op-Room noise*

The concatenative MFCC + GMM means system in additive Op-Room noise at 18dB SNR had a WER of 30.6%, a 5.5% relative improvement over the MFCC system. The Op-Room noise corrupts speech badly even at a relatively high SNR. The improvement suggests that the GMM means supply complementary information to MFCCs in coloured noise environments. However, the relative performance gain is less that that achieved on clean spech (8.8%). Figure 7.3 indicates that the GMM component mean features are affected by the OpRoom noise source to a similar degree as the higher order cepstra. The results in additive noise follow this, as the relative improvement adding the GMM component means exhibits only a small variation between

clean and noise mismatched conditions.The reduction in WER is similar to that achieved in clean speech conditions. This slight improvement is maintained over a range of SNRs.

The GMM system performed badly in the noise corrupted enviroment, with an WER of 66%, approximately twice that of the MFCC system. The GMM features perform badly in the noise mismatched conditions. Studying figure 7.3 suggests that the main drop in performance is due to the high degree of corruption in the component mean log-magnitude terms.

Using the confidence metric on noise corrupted speech also yields a slight improvement of 1% absolute in WER over the MFCC+6Mean concatentative system. The confidence measure will deweight the GMM features in regions where are not strongly defined peaks. These regions will correspond to the low-energy regions of speech which are worst affected by the noise. However, the confidence measure extracted from the speech can itself be corrupted by the peak-structure of the noise, limiting its effectiveness.

18 dB SNR	Uncompensated System	Noise Matched
MFCC	32.3	8.1
GMM6	66.7	12.3
MFCC+GMM Concat.	30.6	7.1
+ Confidence	29.6	7.1

Table 7.1 *Results using uncompensated and noise matched systems on the RM task corrupted with additive Op-Room noise at 18dB SNR*

7.1.3 Results training on RM data with additive noise

The performance of noise matched systems built on the RM task corrupted by additive noise is presented in this section. The aim is to see what the "optimal" performance of the speech compensation techniques can achieve if the models are adequately compensated.

A noise matched system was built using single pass retraining from the clean speech models using training data corrupted with additive noise, as described in section 2.2.3. The clean speech data was used together with the clean model set to generate the frame/state alignments. The alignments thus generated were then used in combination with the corrupted data to generate a noise-matched model set. The results using the above parameterisations with these systems are also presented in table 7.1 and figure 7.5.

In these noise matched conditions, the MFCC+6Mean concatenative system gives a reduction in WER to 7.1% from the MFCC system at 8.2%, an improvement at a confidence of 98%. This improvement suggests that if the GMM mean features can be adequately compensated in the model set, then they still possess complementary information in noise-corrupted environments.

Table 7.2 shows the performance of the system when the parameters from a noise matched model are used. The noise matched HMM parameters can be considered the "ideal" parameters

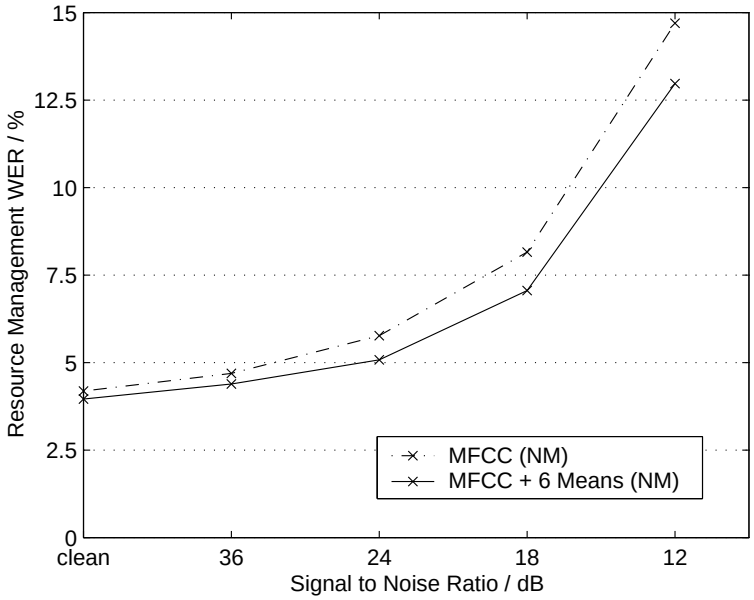


Figure 7.5 *WER on RM task for MFCC and MFCC+6Mean systems corrupted with additive Op-Room noise for noise matched models retrained with corrupted training data*

and should form an upper bound on the performance of any model compensation approach. The GMM6 models alone perform poorly on data trained on the Op-Room noise. However, with compensation the relative difference in performance between the GMM6 parameters and the MFCC system decreases.

Using the compensated values of the static means gives the largest improvement in the WER for all systems. Using the variances from the noise-matched model gives improvements typically about 15-20% relative over the systems only using the compensated means.

Compensating the parameters of the MFCC+6Mean system yields reductions in WER over those of the MFCC system. In particular, compensating only the static means of the MFCC+6Mean system yields an WER of 12.2%, and this result using the "ideal" static mean parameters should be considered a baseline for the results using the model-based noise compensation techique detailed later.

The performance of the compensated MFCC+6mean systems outperforms all of the MFCC systems. Thus, if the GMM parameters can be adequately compensated then significant performance advantages can be achieved.

7.2 Front-end noise compensation

The technique for front-end noise compensation presented in section 4.5.2 was applied to the feature extraction process for a GMM system. The average noise model was used during the extraction process to estimate the clean speech GMM parameters from the noise corrupted speech. A sample plot of component mean trajectories calculated using the front-end compensation

Parameters	MFCC		MFCC+6MEAN		GMM6	
Compensated	μ	$\mu + \Sigma$	μ	$\mu + \Sigma$	μ	$\mu + \Sigma$
Static	14.7	13.1	12.2	10.8	26.6	25.4
Static + Δ	11.9	9.5	9.9	8.3	18.88	15.9
Static + Δ^2	10.2	8.1	8.8	7.1	14.8	12.3

Table 7.2 *MFCC Results selecting model features from a noise matched system to complement a clean speech system on RM task corrupted with Op-Room noise at 18dB SNR*

scheme is shown in figure 7.6. When this approach was applied, the observed tracks for the GMM parameters were closer to the clean speech, but unfortunately exhibited large discontinuities between certain frames. These may have been caused by the noise model masking the low frequency speech signal during low intensity sounds. To counteract this effect, a moving average (MA) filter was also applied to smooth the parameters extracted using the front-end noise compensation. A four-component model of the noise was obtained offline by taking the average values of GMM estimates from the noise spectra. The compensated GMM means were combined with the uncompensated MFCCs and were tested with the clean system. A moving average filter of length 3 was applied over the GMM mean features after the front-end compensation as mentioned previously. The filter was applied prior to the calculation of dynamic parameters.

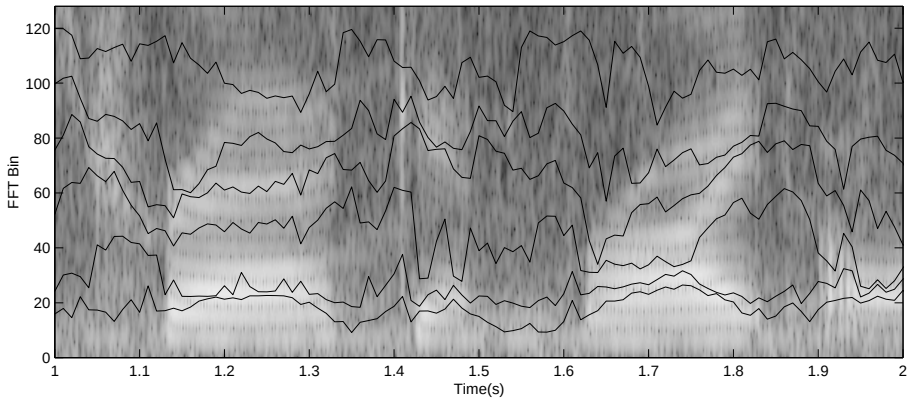


Figure 7.6 *GMM Mean trajectories in the presence of additive Op-Room noise using the front-end compensation approach for the utterance "Where were you while we were away"*

Using the front-end compensation technique improves the performance of the GMM6 system with a 22% reduction in WER. Applying the MA filter to smooth the extracted parameters yields a further improvement of 39% relative to the performance of the clean models on a noise corrupted enviroment. Using a MA filter on the clean speech data actually led to a degradation in performance when used in section 4.3. The improvement when adding the front-end compensated features to an MFCC parameterisation is relatively small compared to adding the uncompensated GMM means. The WER was actually slightly increased using the component

means from the front-end compensated GMM6 system. Applying the moving-average smoothing technique gives a slight decrease in WER of 7.6% relative to using the uncompensated GMM means, and the confidence in this improvement is 75%.

It is likely that the compensation technique is most effective on the log-magnitude terms in the feature vector which were badly corrupted by the noise source. When a moving average filter was applied to the features from the GMM system in section 5.3, performance was degraded. However, when used with the front-end compensation scheme, a small improvement was observed from the smoothing the GMM features.

Description	WER /%
MFCC (UC)	32.3
GMM (UC)	66.6
GMM (FC)	51.1
+smoothing	31.3
MFCC (UC) + GMM (UC)	30.6
MFCC (UC) + GMM (FC)	31.9
+smoothing	28.3

Table 7.3 *Word Error Rates (%) on RM task with additive Op-Room noise at 18dB SNR with uncompensated (UC) and front-end compensation (FC) parameters*

The reason the front-end compensation technique did not work as well as expected is most likely due to the same problems that spectral subtraction techniques face [21]. The noise source is time-varying and does not always have the same amplitude. Additionally, the phase of the noise signal is unknown, so the effects of the additive noise signal on the magnitude spectrum cannot be determined. During regions of low spectral energy, the noise model peaks can easily mask the speech signal, especially in the low frequency regions.

7.3 Model based noise compensation

In this section, the model compensation technique outlined in section 4.5.3 is used to compensate the HMMs trained on clean speech to the presence of additive noise. The technique presented in section 4.5.3 compensates the static mean parameters of the GMM features in the output PDFs in each HMM state. The technique is similar to compensating the MFCCs using a log-add approximation. The noise model used is the same as used in the previous section and taken from the average GMM parameters from the noise source.

The model compensation of the static mean MFCC parameters in the HMMs was simulated by replacing the values in a clean model by the those from the “ideal” noise matched model. In practice the MFCC static means in the HMM could be compensated by using a log-add PMC approach or similar. The important consideration is the relative improvement the compensated GMM means give over a compensated MFCC system. Using the ideal MFCC mean parameters

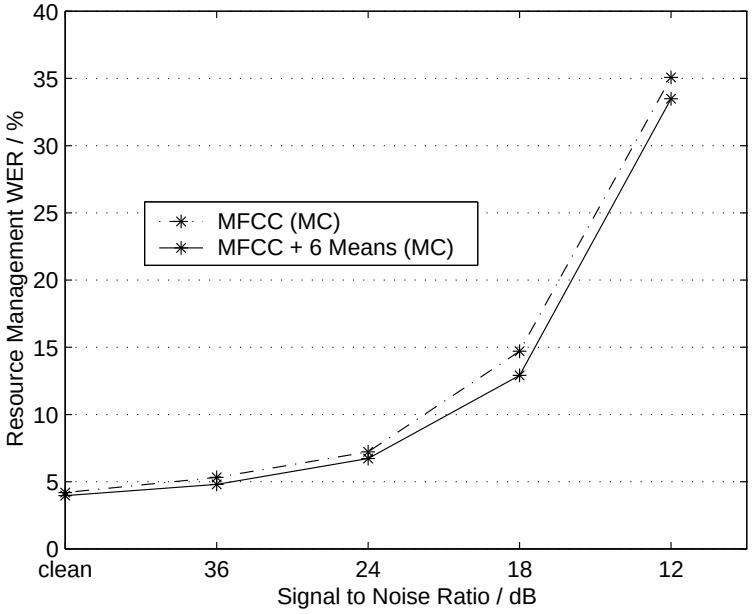


Figure 7.7 *WER on RM task for MFCC and MFCC+6Mean systems corrupted with additive Op-Room noise for models with compensated static mean parameters*

will give a lower bound on the relative improvements to be gained by compensating the GMM parameters.

The GMM parameters allow a compensation technique to work directly in the spectral domain, thus reducing the complexity of mapping linear cepstral domain and the log-add approximations that are made with PMC on MFCC features [40].

The results of the experiments are presented in table 7.4 and figure 7.7.

Description	WER /%
MFCC (MC)	14.7
GMM (MC)	32.6
MFCC (UC) + GMM (MC)	22.1
MFCC (MC) + GMM (MC)	12.9

Table 7.4 *Word Error Rates (%) on RM task with additive Op-Room noise at 18dB SNR with uncompensated (UC) and front-end compensation (FC) parameters*

Adding GMM mean features to a static mean compensated MFCC system reduced the WER by 7% relative at 18dB SNR, with the confidence on the improvement at 99%. The model compensation system gave results very close to the performance predicted by the "ideal" static mean HMM parameter compensated systems presented in table 7.2. As is the case with MFCC features, adapting the GMM model parameters yields better performance than compensation at the front-end level. A relative improvement of 31% was achieved when only the six static means

of the GMM component means were compensated. The model compensated systems were also tested using the confidence metric to combine the MFCC and GMM features, and the results are in Table 7.4. Using the confidence metric gives a 4% reduction in WER relative to a single stream system at 18dB. However, using the confidence metric with a noise matched system gave no reduction in error rate.

7.4 Summary

In this section recognition results using the GMM features on the RM task corrupted with additive noise were presented. The GMM features were shown to have some inherently noise-robust properties, and gave a slight improvement in performance in an uncompensated system. In addition, results using two techniques to compensate the performance of the system in additive noise were presented. The front-end compensation scheme improved the performance of the GMM features on a noise corrupted speech, although the improvement was mostly due to the compensation of the log-magnitude terms, and only a slight improvement was gained when using the compensated GMM means in combination with MFCC features. The model compensation technique managed to improve the performance of a MFCC+6 GMM means system in a noise corrupted environment. Using a system with compensated static means for MFCCs and GMM means, an improvement of 7% was achieved over a system built with only compensated MFCC features at 18dB SNR. The improvements gained using the model compensation approach were close to the "ideal" performance from a system trained in a noise matched enviroment. The improvements from a noise matched system show that further progress can be made if the dynamic parameters of the GMM features in the model set can be compensated.

8

Results using speaker adaptation with GMM features

In this section the results of using MLLR speakaer adaptation approaches on the GMM features alone and in combination with MFCCs are shown. Results using unconstrained MLLR adaptation on the test set are shown, as well as constrained transform on the test speakers and speaker adaptive training (SAT) schemes.

Three systems are considered in this section:

1. **MFCC**:a system built using a standard MFCC parameterisation;
2. **MFCC+6Mean**: a system built with a feature vector formed concatenating MFCC features with the GMM component mean features from a six-component spectral estimate;
3. **GMM6**: a system built with the full set of GMM features: means, standard deviations and log-component energies.

8.1 GMM features and vocal tract normalisation

One of the motivations of using spectral peak features is the fact that the peak locations are directly represented as frequency or bin values. Hence, linear scalings of the spectral peak locations can approximate the effects of vocal tract length variation.

In figure 8.1, the VTLN warp factors for the MFCC means have been calculated for the WSJ SI-284 speakers. The warp factors were calculated by using a Brent estimation training likelihood optimisation technique [48]. The technique performs an iterative search to find the VTLN warp factor which yields the maximum likelihood for each speaker on the training data. The MFCC warp factors are plotted against warp factors from the GMM system. The GMM targets were calculated by taking a single diagonal constrained MLLR transform of the GMM features for each speaker. The warp factors were then calculated as a linear regression of the scaling on the GMM component means from the global mean to the speaker target.

As can be observed, there is a reasonable degree of correlation between the GMM features and MFCC warp factors and the correlation index for the two sets of warp factors is 0.7. This

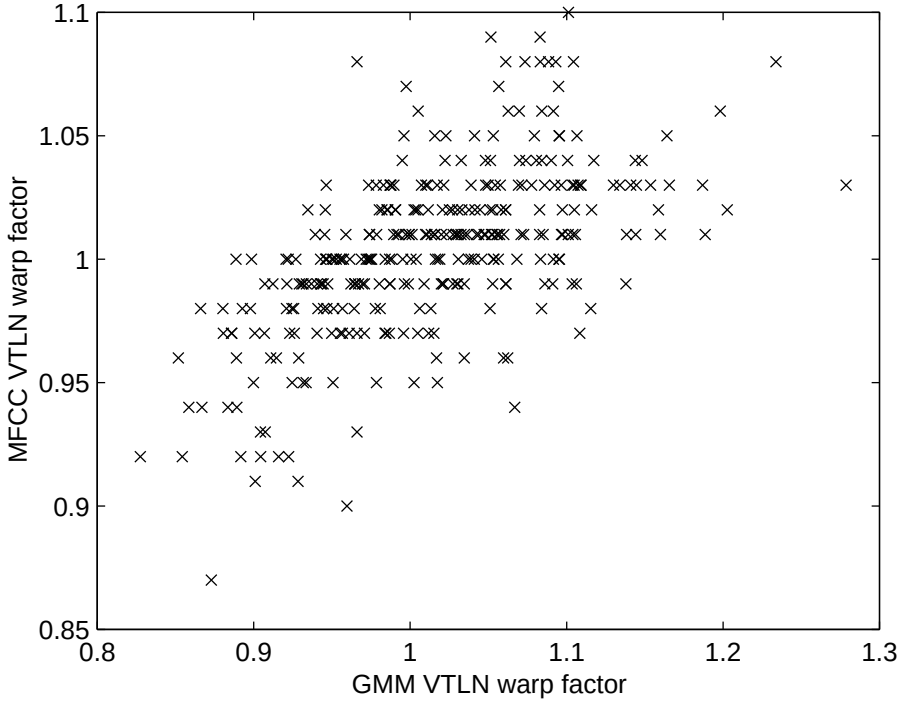


Figure 8.1 *VTLN warp factors for MFCC features calculated on WSJ speakers using Brent estimation against linear regression on GMM component means from CMLLR transforms*

correlation suggests that diagonal transforms of the GMM features are a fair approximation to other estimates of VTL functions.

8.2 Unconstrained maximum likelihood linear regression adaptation

The first type of adaptation investigated was a simple transform of the HMM model output PDF means using a MLLR transform. The transform was calculated using the speaker adaptation data from the CSRNAB-1 corpus. The CSRNAB-1 corpus provides forty adaptation sentences for each speaker.

Block Structure	*MFCC System*		
	Single	Speech Sil	512
Full	8.69	8.69	8.26
Δ	8.89	8.84	8.42
Diagonal	9.81	9.61	9.10

Table 8.1 *Using MLLR transforms on MFCC features to adapt the HMM means of WSJ systems, using full, block diagonal (based on Δ coefficients) and diagonal transforms*

There were two variables considered in making the transform: block structure and size of the regression class tree. Three forms of transform were used for the MFCC and GMM6 systems: a full transform; a diagonal transform, or a block transform based on grouping the dynamic parameters together. For the MFCC+6Mean system, two additional forms were considered: a block structure grouping the features together by type (MFCC/GMM Means) and one grouping by both dynamic parameters and feature type (MFCC/GMM Means/ΔMFCC/ΔGMM Means etc.). The transforms were calculated for each system and the transforms iteratively re-estimated twice. The model transforms were then tested using the MLLR transforms with the speaker independent HMMs to rescore the lattices.

Block Structure	*MFCC+6 Mean System*		
	Single	Speech	Sil
Full	8.56	8.36	7.98
Features (MFCC/GMM means)	8.66	8.60	7.96
Δ	8.74	8.56	8.09
Features+Δ	8.77	8.76	8.30
Diagonal	9.53	9.50	9.04

Table 8.2 *Using MLLR transforms on a MFCC+6Mean feature vector to adapt the HMM means of WSJ systems, using full, block diagonal (groupings based on features type and/or Delta coefficients) and diagonal transforms*

The results for the MFCC, MFCC+6Mean and GMM6 systems are in tables 8.1, 8.2 and 8.3. All systems stated were single stream (concatenative) systems. Appending the six GMM component means to a GMM system improves performance by 2-4% relative in almost all configurations. The best performance for an MFCC system was gained using a full variance transform with a 512 class regression tree for a WER of 8.26%. The best perfomance for the MFCC+6Mean system was achieved using a block structure based on the feature type and a 512 class regression tree. For the MFCC features, little improvement was observed between using a single global transform and one using two classes (speech and silence). On the MFCC+6Mean and GMM6 system small improvements can be seen using seperate speech/silence transforms as opposed to a single global transform. This may be due to the non-linear relationship between the spectrum and the GMM features as the GMMs extracted during periods of silence will experience different shifts as those during speech.

The full GMM6 systems exhibit similar relative performance improvements to the MFCC system for most of the systems tested. However, using a diagonal transform gave the GMM6 systems a relative reduction in WER of 4.5% using a single diagonal transform, whereas no improvement was observed on the MFCC features using this form of transform. Calculating MLLR variance transforms for each of the test speakers in the CSRNAB Hub 1 set also gave small improvements to all of the systems tested.

Block Structure	*GMM6 System*		
	Single	Speech Sil	512
Full	10.51	10.37	10.12
Δ	10.99	10.51	10.31
Diagonal	11.67	11.65	11.07

Table 8.3 *Experiments using MLLR transforms on GMM6 feature vector to adapt the HMM means of WSJ systems, using full, block diagonal (based on Δ coefficients) and diagonal transforms*

8.3 Constrained maximum likelihood linear regression

Constrained MLLR (CMLLR) transforms as presented in section 2.5.3 use the same transform for the mean and variance adaptation of the model set, and can be viewed as a feature-space transform.

In this section, CMLLR transforms were calculated for the speakers in the test set for the systems presented above. The same block structures were used, and transforms for two regression classes (speech/silence) were estimated. The results are in table 8.4.

The MFCC systems gave little or no change in the WER from the systems built using unconstrained MLLR in table 8.1 regardless of the block structure. The MFCC+6Mean system experienced no improvement in WER from the MFCC system when CMLLR was applied, except for the case of the diagonal transforms. Using a diagonal transform, the relative improvement of 2% WER over MFCC system was maintained. Compared to the using the unconstrained MLLR on the MFCC+6Mean system in table 8.2, there is actually a slight degradation in performance using the CMLLR systems.

Implementing a CMLLR transform on the GMM6 system gave a small reduction in WER over the baseline. However, the performance gain is much smaller than the improvements gained using unconstrained MLLR, especially compared to the gains using CMLLR on the MFCC system. There was very little improvement gained with using larger block sizes on the GMM6 system.

It is interesting to compare the results with those using the semi-tied systems in table 6.7. As in the case for semi-tied systems, the CMLLR transforms that worked the best used a block diagonal structure which split the feature types (MFCCs, component means, standard deviations) into separate blocks.

8.3.1 Speaker adaptive training

In order to evaluate the performance with speaker adaptive training, CMLLR systems were built for the WSJ system. For each speaker in the SI-284 training set, CMLLR speaker transforms were calculated. The same systems as presented in the previous section were used, and transforms for a two-class regression tree were built. The HMM models were then retrained using the training speaker transforms, and speaker transforms re-estimated using the new model sets and the previous speaker transforms. These steps were iterated five times, and the resulting transforms

Block Structure	*MFCC System*	*MFCC+6 Mean System*	*GMM6 System*
Full	8.80	8.84	11.26
Feature type (MFCC/GMM mean)		8.75	
Δ	8.71	8.78	11.36
Feature + Δ		8.67	
Diagonal	9.61	9.42	11.69

Table 8.4 *Experiments using constrained MLLR transforms for WSJ test speakers, using full, block diagonal (groupings based on features type and/or Delta coefficients) and diagonal transforms*

and model sets were tested.

The results of the SAT experiments are in table 8.5. The MFCC systems exhibit a consistent and significant improvement of around 9% using SAT in combination with a constrained MLLR test set adaptation for block diagonal and full transforms. Little improvement was gained using a full transform rather than a block-diagonal structure with the MFCCs.

Implementing SAT on a MFCC+6Means system yields a relative drop in WER of 4% over the test set CMLLR system in the previous section for block diagonal and full transforms. This relative improvement is much lower than that exhibited by the MFCC system, and the systems overall perform worse than the MFCC systems with SAT. The diagonal transform case performs slightly better than the MFCC system with a diagonal transform, possibly due to the VTLN normalising effects discussed in section 8.1.

Implementing SAT on the full GMM6 systems does not improve their recognition performance significantly from test-set only adaptation, except for the case of the diagonal transform, which improved by roughly 3% relative. Although the systems exhibit an increase in log-likelihoods, this does not guarantee an increase in the recognition rate. It may be that the high degree of correlations present in the GMM feature vector make them unsuited to the CMLLR approach.

Block Structure	*MFCC System*	*MFCC+6 Mean System*	*GMM6 System*
Full	7.98	8.45	11.32
Feature type (MFCC/GMM mean)		8.34	
Δ	8.05	8.43	11.34
Feature + Δ		8.31	
Diagonal	9.69	9.40	11.43

Table 8.5 *Experiments using constrained MLLR transforms incorporating speaker adaptive training on WSJ task, using full, block diagonal (groupings based on features type and/or Delta coefficients) and diagonal transforms*

8.4 Summary

This section presented results using MLLR supervised adaptation schemes on the large vocabulary WSJ task. Using model-based (unconstrained) transforms, a consistent improvement in performance of between $2-4\%$ was observed for all forms of transform when appending GMM component mean features to the MFCC parameterisation. However, when using a feature-space (constrained) MLLR approach, there were no performance gains observed, save for the case of using a diagonal transform. Using a SAT approach with CMLLR gave no significant gains for the MFCC+6Means system, and gave a comparative degradation in performance compared to the MFCC system using CMLLR and SAT.

9

Conclusions and further work

This thesis presents a novel speech parameterisation based on representing the spectral envelope with a Gaussian mixture model. Features derived from the GMM parameters were used as formant-like features for speech recognition. In particular, the values of the GMM component means can be related to the formant or spectral peak locations. Techniques for extracting the parameters using the EM algorithm were presented, along with frameworks for combining the GMM features with MFCC or PLP parameterisations. The performance of the features in the presence of additive background noise was examined, and techniques for compensating the GMM features were developed and tested. Finally, the use of MLLR adaptation techniques on the GMM features was investigated.

9.1 Review of work

There are several motivations for using spectral-peak or formant features. Formants are considered to be representative of the underlying phonetic content of speech. They are also believed to be relatively robust to the presence of noise, and useful in low-bandwidth applications. Additionally, it has been hypothesised that formants or spectral peak positions can be easily adapted to different speakers. However, the extraction of robust and reliable formant estimates is a non-trivial task. Recently, there has been increased interest in other methods for estimating spectral peaks, for example, using the HMM2 or gravity centroid features. The GMM features developed in this thesis bear some similarities to the gravity centroids. The GMM estimates for mean and variance are directly related to the first and second spectral sub-band moments if the posterior probabilities of the components are fixed to filter-bank functions rather than being iteratively updated. Hence, the GMM features possess more flexibility in the spectral modelling than the gravity centroid features. In addition, the features can be easily mapped into the linear spectral domain, giving them interesting properties for speaker adaptation approaches and noise compensation.

The theory of estimating the GMM parameters from a speech spectrum was presented in chapter 4. The EM algorithm was applied to the task of estimating a Gaussian mixture model

from a set of rectangular histogram bins. In order to impose some form of continuity constraints, the algorithm was also extended for the case of estimating a two-dimensional histogram using the surrounding spectral frames. The characteristic shape of the voiced spectrum was shown to be unsuitable for representing with a GMM. Hence, techniques for smoothing the spectrum to estimate the spectral envelope prior to estimating the GMM were also discussed. Another potential problem is that the extracted parameters will not generalise well. To address this, a method to incorporate a prior distribution to constrain the values of the extracted parameters was presented. It has been observed that formants or formant-like features do not represent unvoiced regions of speech which do not contain strong formant structures. A framework to combine MFCC parameters with the GMM component means using a measure of confidence in the estimated means was also presented, together with an extension to work on medium or large vocabulary tasks together with a language model. Another consideration for acoustic features is their robustness to additive noise, and whether they can be easily compensated to noise corrupted environments. Two techniques to compensate the GMM spectral features in additive noise using a noise model were presented in this thesis. The first added the noise model to the estimated GMM during the feature extraction stage to extract estimates of the clean speech parameters. The second combined the GMM parameters from the model set together with the noise model in the linear spectral domain, to obtain estimates of the noise corrupted GMM parameters.

Results using the GMM features alone were presented in chapter 5. The lowest WER for the GMM features was achieved on a 4kHz bandwidth system by estimating six components from a spectrum smoothed with a convolutional pitch-based filter. The best feature set extracted comprised of the GMM component means, standard deviations and the normalised log-energy at the component means. The performance of the best GMM system was below that of an MFCC system and had a WER 17% relative higher than that of the MFCC baseline. Using the surrounding frames in a two-dimensional estimate achieved smoother parameter trajectories during voiced speech, but lead to an increase in WER overall. Incorporating a prior distribution whilst estimating the spectrum increased the consistency of the estimated parameters but also did not lead to a decrease in WER.

In chapter 6, results combining the GMM component means with MFCC features were presented. The component means were chosen for their relatively high Fisher ratios and also their relationship to the formant positions. These features appear to possess some informantion complementary to the MFCC parameters. The GMM component mean features gave a small but significant improvement when combined with the MFCC parameters on a medium vocabulary task. A relative improvement of 8.8% was achieved by adding the six component mean features to an MFCC parameterisation. This improvement is significant at a confidence of 96%. When feature mean normalisation was implemented, the relative improvement over the MFCC baseline increased to 13%. Using a synchronous stream system to combine the parameterisations gave a small reduction in WER, but less than a concatenative system. Using the confidence metric to combine the systems improved the performance of the synchronous stream system,

but did not outperform a concatenative system. The results on the large vocabulary WSJ task mirrored the results on the RM task, but the relative improvements were smaller and not significant. Furthermore, adding the GMM component mean features to a PLP parameterisation on the SwitchBoard task led to a degradation in performance. No improvements were gained using a semi-tied covariance matrix or a LDA transform with the GMM features.

Chapter 7 detailed results using the GMM features in the presence of additive Op-Room noise. The Op-Room noise was used because it is coloured and corrupts both the GMM and MFCC features significantly. However, even without using any form of compensation, including the GMM mean features gave a small improvement to a MFCC systems. This section also presented results using the two noise compensation techniques described earlier. The front-end compensation technique gave a significant improvement on the full GMM system roughly halving the WER, mostly due to the correction of the component energy terms. The front-end compensated means only gave a slight improvement when added to the MFCC parameters: applying the front-end compensation technique reduced the WER of the concatenative MFCC and GMM component means system by 7.5% relative. The second noise compensation technique compensated the static means of the HMM states rather than the input features in a similar fashion to a log-add PMC approach. The model compensation technique gave a significant improvement on the RM task, reducing the WER of a static mean compensated MFCC by 12% relative. The decrease in WER observed was close to the predicted improvement from using the "ideal" parameters from a model set trained in noise-matched conditions.

Using the GMM features with MLLR transforms was examined in chapter 8. The small improvements gained by adding the six component means to an MFCC system were preserved when unconstrained MLLR adaptation transforms were estimated. However, the MFCC+GMM means systems performed poorly when constrained MLLR transforms were estimated.

In summary, the GMM features alone perform poorer than MFCCs but give some complementary information to MFCC features on a medium vocabulary tasks. The GMM features also reduced the WER when added to the MFCC features in noise corrupted environments, and can be rapidly adapted given a model of the noise. However, on the large vocabulary WSJ the relative improvements were smaller, and adding the GMM means onto a SwitchBoard system led to a degradation in performance. These results may be due to the lack of any form of cepstral (or log-spectral) normalisation for the GMM features. Applying MLLR to the systems preserved the small improvements on the WSJ task, but a relative degradation was observed when using constrained MLLR transforms.

9.2 Future work

The model-based noise compensation systems only allow the means of the HMM model components to be compensated for the effects of additive noise. On other schemes such as PMC and the noise-matched systems presented here, additional performance gains have been achieved by compensating the Δ and Δ^2 parameters and the variances of the models. Extending the

compensation scheme to the other model parameters is an interesting research direction. For example, it would be possible to apply the matrix approximation to the dynamic parameters. Alternatively, the continuous time approximation could be applied to compensate the dynamic parameters of the GMM.

The noise results presented were performed on a task artificially corrupted with an additive noise source. However, this neglects the Lombard effect which may degrade performance further. It would be useful to further consider the performance of the GMM features on a task recorded in noisy conditions, such as the Aurora corpus.

The use of the GMM features in combination with MFCCs provided smaller improvements on larger tasks. Further work could be conducted into the relative failure of the GMM features on the Switchboard task could also be undertaken. In particular, the incorporation of some form of log-spectral normalisation prior to estimating the GMM features could be investigated, as this yield significant improvements when applied to MFCC and PLP features on larger tasks.

Work with formant estimation techniques has achieved smoother and more consistent trajectories using continuity constraints. Since the EM algorithm is a statistical approach, it could be possible to apply similar techniques using cost functions to the estimation of the GMM components. A subset of the Gaussian components estimated from the spectrum could be selected using a DP alignment and a cost function based on the continuity and reliability of estimate. Further investigation could be performed into other methods for estimating the GMM parameters as well, using other forms of trajectory constraint or implementing class-dependant priors on the estimated features.

The technique for combining the GMM and MFCC features which yielded the lowest WER was the concatenative approach. However, it may be interesting to investigate other methods for combining the two features together. For example, the use of multiple-regression HMMs could make use of some of the inter-speaker information contained in the GMM features. It could also be possible to investigate alternative schemes to use the confidence metric when combining the features.

The use of constrained MLLR schemes suggests that these transforms are not appropriate for the GMM features. Further research could be performed into alternative transformations using non-linear adaptation schemes for the GMM features. Other transforms of the GMM features may also be possible.

A

Expectation-Maximisation Algorithm

The EM algorithm is a general iterative optimisation technique. It provides a method for successively updating the model parameters $\boldsymbol{\theta}$ at each iteration such that the log-likelihood of the training data increases at each step [18].

The EM algorithm is used when it is not possible to optimise the log likelihood $log[p(\mathbf{X}|\boldsymbol{\theta})]$ directly with respect to $\boldsymbol{\theta}$. Instead discrete random variables $\mathbf{Z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_D\}$ are introduced which are dependent on the set of observations $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_D\}$ and the model parameters $\boldsymbol{\theta}$

$$\sum_{d=1}^{D} \log p(\mathbf{x}_d|\boldsymbol{\theta}) = \sum_{d=1}^{D} \log p(\mathbf{x}_d, \mathbf{z}_d|\boldsymbol{\theta}) - \sum_{d=1}^{D} \log p(\mathbf{z}_d|\mathbf{x}_d, \boldsymbol{\theta}) \tag{A.1}$$

The expectation of the log likelihood of the complete data, the second term on the right hand of equation A.1, can be optimised instead. The increase in log-likelihood of the complete data $(\mathbf{X}, \mathbf{Z})$ forms a lower bound on the increase in log likelihood for the observed data $(\mathbf{X})$. The parameters $\hat{\boldsymbol{\theta}}$ which produce an increase in the expected log likelihood of the complete data given the current parameters $\boldsymbol{\theta}$ are found. The expected log likelihood given the complete data is the *auxiliary function*, $\mathcal{Q}(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}})$. Optimising the auxiliary function is guaranteed to increase (or not to decrease) the log-likelihood of the observed data, but does not yield a ML solution. Therefore it is necessary to iterate the steps of caluculating the auxilary function and maximising it until convergance.

The basis of the EM algorithm is that if the auxilary function increases, the log-likelihood of the observed data $\log[p(\mathbf{X}|\boldsymbol{\theta})]$ will not decrease. The auxilary function $\mathcal{Q}(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}})$ can be defined as the expectation of the complete data log-likelihood, conditional on the observed data $\mathbf{X}$ and the current values of the parameters $\boldsymbol{\theta}$:

$$\mathcal{Q}_d(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}) = \mathcal{E}[\log p(\mathbf{x}_d, \mathbf{z}_d|\hat{\boldsymbol{\theta}})|\mathbf{x}_d, \boldsymbol{\theta}] \tag{A.2}$$

$$\mathcal{Q}(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}) = \sum_{d=1}^{D} \mathcal{Q}_d(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}) \tag{A.3}$$

And so for the discrete set of observed data $\mathbf{X}$

$$\begin{aligned}\mathcal{Q}(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}) &= \sum_{\forall \mathbf{Z}} P(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}) \log\left(p(\mathbf{X}, \mathbf{Z}|\hat{\boldsymbol{\theta}})\right) \\ &= \sum_{d=1}^{D} \sum_{\forall \mathbf{z}_d} P(\mathbf{z}_d|\mathbf{x}_d, \boldsymbol{\theta}) \log\left(p(\mathbf{x}_d, \mathbf{z}_d|\hat{\boldsymbol{\theta}})\right) \end{aligned} \tag{A.4}$$

The EM algorithm iterates two stages:

- **Expectation:** given the current parameters $\boldsymbol{\theta}$ calculate the posterior probability mass function of the hidden variable, $P(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta})$. Using this PDF, calculate expected values of the log-likelihood of the complete data set as a function of the new model parameters $\hat{\boldsymbol{\theta}}$ given the current parameters.
- **Maximisation:** maximise the auxiliary function $\mathcal{Q}(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}})$ with respect to $\hat{\boldsymbol{\theta}}$

A.1 EM algorithm for fitting mixture components to a data set

The EM algorithm can be applied to the problem of fitting a set of Gaussian mixtures to a set of observed data $\mathbf{x} = \{x_1, \ldots, x_D\}$. The "hidden" data component is an *indicator variable* which indicates which mixture component generated the data.

$$z_{dj} = \begin{cases} 1 & \text{observation } x_d \text{ was generated by } \omega_j \\ 0 & \text{otherwise} \end{cases} \tag{A.5}$$

And so for the full set of data $\mathbf{x}$ there is a corresponding set of latent variables for each data point, related to the set of mixtures:

$$\mathbf{Z} = \{\mathbf{z} \ldots \mathbf{z}_D\} \tag{A.6}$$

$$\mathbf{z}_d = [z_{d1} \ldots z_{dM}]^T \tag{A.7}$$

For a single point known to be generated by component ω_j it is possible to calculate

$$p(\mathbf{z}_d, x_d|\boldsymbol{\theta}) = \prod_{m=1}^{M} [p(x_d|\omega_m, \boldsymbol{\theta}_m)P(\omega_m))]^{z_{dm}} \tag{A.8}$$

$$\log(\mathbf{z}_d, x_d|\boldsymbol{\theta})) = \sum_{m=1}^{M} z_{dm} \log[p(x_d|\omega_m, \boldsymbol{\theta}_m)P(\omega_m))] \tag{A.9}$$

and since all the data points are independent:

$$\log\left(p(\mathbf{Z}, \mathbf{x}|\boldsymbol{\theta})\right) = \sum_{d=1}^{D} \log(p(\mathbf{z}_d, x_d|\boldsymbol{\theta})) \tag{A.10}$$

The auxiliary function can be written as

$$\begin{aligned}
\mathcal{Q}(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}) &= \frac{1}{D} \sum_{\forall \mathbf{Z}} P(\mathbf{Z}|\mathbf{x}, \boldsymbol{\theta}) \log \left(p(\mathbf{x}, \mathbf{Z}|\hat{\boldsymbol{\theta}}) \right) \\
&= \frac{1}{D} \sum_{d=1}^{D} \sum_{\forall \mathbf{z}_d} P(\mathbf{z}_d|x_d, \boldsymbol{\theta}) \log \left(p(x_d, \mathbf{z}|\hat{\boldsymbol{\theta}}) \right) \\
&= \frac{1}{D} \sum_{d=1}^{D} \sum_{\forall \mathbf{z}_d} P(\mathbf{z}_d|x_d, \boldsymbol{\theta}) \sum_{m=1}^{M} z_{dm} \log \left(p(x_d|\omega_m, \hat{\boldsymbol{\theta}}_m) \right) \\
&\quad + \frac{1}{D} \sum_{d=1}^{D} \sum_{\forall \mathbf{z}_d} P(\mathbf{z}_d|x_d, \boldsymbol{\theta}) \sum_{m=1}^{M} z_{dm} \log \left(\hat{P}(\omega_m) \right) \qquad \text{(A.11)}
\end{aligned}$$

Since the summand is over all $\mathbf{z}_d (m = 1, M)$, $\mathbf{z}_d$ does not depend on d. Therefore $\mathbf{z}_d$ can be denoted by ω_m:

$$\begin{aligned}
\mathcal{Q}(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}) &= \frac{1}{D} \sum_{m=1}^{M} \left[\sum_{d=1}^{D} P(\omega_m|x_d, \boldsymbol{\theta}) \log \left(p(x_d|\omega_m, \hat{\boldsymbol{\theta}}_m) \right) \right] \\
&\quad + \frac{1}{D} \sum_{m=1}^{M} \left[\sum_{d=1}^{D} P(\omega_m|x_d, \boldsymbol{\theta}) \log \left(\hat{P}(\omega_m) \right) \right] \qquad \text{(A.12)}
\end{aligned}$$

The posterior probability that observation x_k was generated by component ω_j can be formalised by:

$$P(\omega_j|x_d, \boldsymbol{\theta}) = \frac{p(x_d|\omega_j, \boldsymbol{\theta}_j)\hat{P}(\omega_j)}{\sum_{m=1}^{M} p(x_d|\omega_m, \boldsymbol{\theta}_m)\hat{P}(\omega_m)} \qquad \text{(A.13)}$$

The mixture component probability functions are single-dimensional Gaussians in this case, and thus the probability of the data point is given by:

$$\begin{aligned}
p(x_d|\omega_j, \boldsymbol{\theta}_j) &= \mathcal{N}(x_d; \mu_j, \sigma_j^2) \qquad \text{(A.14)} \\
&= \frac{1}{\sqrt{2\pi\sigma_j^2}} \exp \left[- \frac{(x_d - \mu_j)^2}{2\sigma_j^2} \right] \qquad \text{(A.15)}
\end{aligned}$$

Maximisation of the auxiliary function is achieved by maximising each term in equation with respect to $\hat{P}(\omega_j)$ and $\hat{\boldsymbol{\theta}}_j$. Substituting equations A.15 and A.13 into eq. A.12 and differentiating with respect to $\hat{\boldsymbol{\theta}}_j$ and equating to zero, the following equation is obtained:

$$\frac{\partial \mathcal{Q}(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}})}{\partial \hat{\boldsymbol{\theta}}_j} = \sum_{d=1}^{D} P(\omega_j|x_d, \boldsymbol{\theta}_j) \frac{\partial}{\partial \hat{\boldsymbol{\theta}}_j} \left[\log \left(\mathcal{N}(x_d; \hat{\mu}_j, \hat{\sigma_j}^2) \right) \right] = 0 \qquad \text{(A.16)}$$

Differentiating for μ_j and σ_j in eq. A.16 and equating to zero, the new parameter estimates $\hat{\mu}_j$ and $\hat{\sigma}_j^2$

$$\begin{aligned}
\hat{\mu}_j &= \frac{\sum_{d=1}^{D} P(\omega_j|x_d, \boldsymbol{\theta}) x_d}{\sum_{d=1}^{D} P(\omega_j|x_d, \boldsymbol{\theta})} \qquad \text{(A.17)} \\
\hat{\sigma}_j^2 &= \frac{\sum_{d=1}^{D} P(\omega_j|x_d, \boldsymbol{\theta})(x_d - \hat{\mu}_j)^2}{\sum_{d=1}^{D} P(\omega_j|x_d, \boldsymbol{\theta})} \qquad \text{(A.18)}
\end{aligned}$$

The new prior estimates can be given by considering the probability mass assigned to each component:

$$\hat{P}(\omega_j) = \frac{1}{D} \frac{\sum_{i=1}^{N} P(\omega_j | x_d, \boldsymbol{\theta})}{\sum_{m=1}^{M} \sum_{i=1}^{N} P(\omega_m | x_d, \boldsymbol{\theta})} \tag{A.19}$$

The denominator is the sum over all mixtures and bins, and hence will be equal to 1. The updated prior components can simply be written as:

$$\hat{P}(\omega_j) = \frac{1}{D} \sum_{d=1}^{D} P(\omega_j | x_d, \boldsymbol{\theta}) \tag{A.20}$$

Thus the E-step calculates the total likelihood of the complete data set, and also calculates the posterior probability of each data point value being generated from each mixture. The M-step then calculates updated estimates for the GMM parameters using the expectations of the posteriors of each data point.

B

Experimental corpora and baseline systems

Two speech recognition tasks are used in this thesis. First, a medium vocabulary system, the Resource Management task is used to examine the basic performance. Second, the Wall Street Journal task is used in combination with the CSR North American Broadcast News task from the Hub 1 evaluations for 1994 to evaluate the performance of the best systems on a larger task. These corpora are described in more detail below, and the baseline systems built on each task are described.

B.1 Resource Management

The Resource Management (RM) task is a medium vocabulary task based on a naval resource management domain [91]. The RM task consists of 3990 training sentences with a 1000 word vocabulary. There are 109 training speakers in the corpus and four sets each of 300 test sentences from a total of 40 subjects. The DARPA evaluation sets are the February 1989, October 1989, February 1991 and September 1992 sets. The data was sampled at 16kHz and recorded in a sound isolated recording booth.

All recognition results were formed using hidden Markov models (HMMs) and using the HMM Toolkit (HTK) RM recipe [122], with the exception that the initial monophone models are flat started, as described below, rather than being initialised using the Dragon systems MFCC monophone model.

Initially monophone models with a single component Gaussian output PDF were trained from a flat start. Each of the state output PDFs in the monophone HMM was initialised with a mean of 0.0 and a variance of 1.0. Seven passes of the Baum-Welch retraining algorithm were performed. A variance floor, set at 0.01 of the global variance was used.

Cross-word triphone context-dependent HMMs were then made using a phonetic decision class tree for each parameterisation. The number of components in the state output probability density functions was increased by splitting the Gaussian components until no further recognition improvements were observed on the February 1989 subset of the test data. Four passes of the Baum-Welch retraining were performed after each splitting of the output Gaussian compo-

nents. A word-pair grammar was used for recognition with a perplexity of 60. The grammar scale factor was tuned on the 300 sentence February 1989 test data subset. The optimal value was then used to test all four evaluation sets and forms the results quoted.

As described in the HTK RM recipe, a baseline system using MFCC parameters from 25ms frames taken every 10ms using a full 8kHz spectrum built as described above was built and tested for comparison purposes. There were six components in the output Gaussian PDF in the HMM models and the word error rate obtained was 4.19%.

B.2 Wall Street Journal

In order to look at the performance of the features in a large vocabulary speech environment, experiments were run on the Wall Street Journal (WSJ) corpus. The WSJ corpus is an open vocabulary task based on speakers reading sentences from articles in the Wall Street Journal [87]. The full training set of 36,493 sentences from 284 speakers (SI-284) in the WSJ0 and WSJ1 sets was used. The test set was the development and evaluation test sets from the 1994 North American broadcast news set (CSRNAB) [85].

The baseline speech model was a gender independent, cross-word triphone HMM similar to that used in the 1994 CUED HTK evaluation system [118] [119]. The feature vector was taken as the cepstra $[c_1, \ldots, c_{12}]$ with a normalised log energy term and dynamic parameters appended to make a 39-dimension feature vector. Cepstral mean normalisation was applied on a sentence-level basis. The MFCC models were built with a decision tree state clustering to generate the sets of speech states, and the number of components in the output PDFs was mixed up with retraining to twelve components in the output PDF Gaussian. A variance floor, set to be 0.1 of the within-class covariance was used.

To generate models for the alternative parameterisations, alignments from the MFCC training data were used to retrain the MFCC model in a single pass retraining step with the new parameterisation. The models thus obtained had twelve mixtures in the state output PDF and the same context-dependent triphone set. The number of components in the output mixtures for the new models was reduced to one, then the number of components in the state output PDFs were increased with retraining until no further improvement was observed on the development test subset of the training data.

The test data was the ARPA 1994 CRSNAB hub 1 (H1) data set consisting of a development test (dt) and evaluation test (et) twenty speakers each, comprising test 626 sentences in all. A trigram language model from the 1994 evaluations was used during recognition. Results were generated rescoring lattices and language model scale factors and word insertion penalties were optimised on the development test subset of the training data. Using an system based on an MFCC parameterisation using cepstral mean normalisation on a per-utterance basis gave a WER averaged across both test sets of 9.75%.

For speaker adaptation and retraining, there is also a set of transcribed training data comprising forty sentences for each speaker in the test sets.

Bibliography

[1] S.S. Airey and M.J.F. Gales. Product of Gaussians and multiple stream systems. *Proceedings ICASSP*, pages 892–895, 2003.

[2] X. Aubert, H. Bourlard, Y. Kamp, and C.J. Wellekens. Improved hidden Markov models for speech recognition. *Phillips Journal of research*, 1(43):245–254, 1988.

[3] L.R. Bahl, P.F. Brown, P.V. de Souza, and R.L. Mercer. Maximum mutual information estimation of hidden Markov model parameters for speech recognition. *Proceedings ICASSP*, pages 49–52, 1986.

[4] L.R. Bahl, F. Jelinek, and R.L. Mercer. A maximum likelihood approach to continuous speech recognition. *IEEE Transactions on PAMI*, 5:179–190, 1983.

[5] L.E. Baum, T. Petrie, G Soules, and N. Weiss. A maximisation technique occurring in the statistical analysis of probabilistic functions of Markov chains. *Annals of Mathematical Statistics*, 41:164–171, 1970.

[6] C.S. Blackburn. *Articulatory Methods for Speech Production and Recognition*. PhD thesis, University of Cambridge, 1997.

[7] S. Boll. Suppression of acoustic noise in speech using spectral subtraction. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 27(2):113–120, 1979.

[8] H. Bourlard and S. Dupont. A new ASR approach based on independent processing and recombination of partial frequency bands. In *Proceedings ICSLP*, pages 426–429, 1996.

[9] H. Bourlard and N. Morgan. *Continuous Speech Recognition: A Hybrid Approach*. Kluwer Academic Publishers, 1993.

[10] J.M. Candy. *Signal Processing, the model-based approach*. McGraw Hill, 1986.

[11] J. Chen, Y. Huang, Q. Li, and K.K. Paliwal. Dynamic spectral sub-band centroid features for robust speech recognition. *IEEE Signal Processing Letters*, 2002.

[12] J. Chen, Y. Huang, and F.K. Soong. Recognition of noisy speech using normalised moments. In *Proceedings ICSLP*, pages 2441–2444, 2002.

[13] D.G. Childers and K. Wu. Gender recognition from speech. part ii: Fine analysis. *Journal of the Acoustic Society of America*, 90(4):1841–1856, 1991.

[14] Mayo Clinic. Department of otorhinolaryngology. Online at `http://mayoresearch.mayo.edu/mayo/research/ent_research/audiology.cfm`.

[15] S.B. Davis and P. Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions Acoustics Speech and Signal Processing*, 28:357–366, 1980.

[16] R. De Mori, L. Moisa, R. Gemello, F. Mana, and D. Albensano. Augmenting standard speech recognition features with energy gravity centres. *Computer Speech and Language*, 15:341–354, 2001.

[17] J. Deller, J. Proakis, and J. Hansen. *Discrete Time Processing of Speech Signals*. Macmillian, 1993.

[18] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of Royal Statistical Society Series B*, 39:1–38, 1977.

[19] L. Deng and K. Erler. Structural design of hidden Markov model speech recognizer using multivalued phonetic features: Comparison with segmental speech units. *Journal of the Acoustic Society of America*, 92(6), 1992.

[20] L. Deng and D.X. Sun. A statistical approach to automatic speech recognition using the atomic speech units constructed from overlapping articulatory features. *Journal of the Acoustic Society of America*, 95(5):2702, 1994.

[21] J. Droppo, A. Acero, and L. Deng. A non-linear observations model for removing noise from corrupted speech log mel-spectral energies. In *Proceedings ICSLP*, pages 1813–1816, 2002.

[22] E. Eide. Distinctive features for use in an automatic speech recognition system. In *Proceedings Eurospeech*, pages 1613–1617, 2001.

[23] E. Eide and H. Gish. A parametric approach to vocal tract length normalization. In *Proceedings ICASSP*, volume 1, pages 346–349, 1996.

[24] K. Erler and L. Deng. Hidden Markov model representation of quantized articulatory features for speech recognition. *Computer Speech and Language*, 7(3):265–2822, 1993.

[25] K. Erler and G.H. Freeman. An HMM-based speech recogniser using overlapping articulatory features. *Journal of the Acoustic Society of America*, 100(4):2500–2513, 1996.

[26] M. Federico. *Spoken Dialogues with computers*, chapter Language Modelling, pages 199–230. Academic press, 1998.

[27] J. Frankel and S. King. ASR - articulatory speech recognition. In *Proceedings Eurospeech*, pages 599–602, 2001.

[28] K. Fujinaga, M. Nakai, H. Shimodaira, and S. Sagayama. Multiple-regression hidden Markov model. In *Proceedings ICASSP*, pages 2311–2314, 2001.

[29] S. Furui. Speaker independent isolated word recognition using dynamic features of speech spectrum. *IEEE Transactions Acoustics Speech and Signal Processing*, 34:52–59, 1986.

[30] B. Gajic and K.K. Paliwal. Robust feature extraction using subband spectral centroid histograms. In *Proceedings ICASSP*, pages 2186–2189, 2001.

[31] B. Gajic and K.K. Paliwal. Robust parameters for speech recognition based on subband spectral centriod histograms. In *Proceedings ICSLP*, 2001.

[32] M.J.F. Gales. A fast and flexible implementation of parallel model combination. *Proceedings ICASSP*, pages 133–136, 1995.

[33] M.J.F. Gales. *Model-based techniques for noise robust speech recognition*. PhD thesis, Department of Engineering, University of Cambridge, 1995.

[34] M.J.F. Gales. The generation and use of regression class trees for MLLR adaptation. Technical report, CUED, 1996.

[35] M.J.F. Gales. Maximum likelihood linear transforms for HMM-based speech recognition. *Computer Speech and Language*, 12:75–98, 1998.

[36] M.J.F. Gales. Semi-tied covariance matrices for hidden Markov models. *IEEE Transactions on Speech and Audio Processing*, 7(3):272–281, 1999.

[37] M.J.F. Gales and P.A. Olsen. Tail distribution modelling using the Richter and power exponential distributions. *Proceedings Eurospeech*, pages 1507–1510, 1999.

[38] M.J.F. Gales and P.C. Woodland. Mean and variance adaptation within the MLLR framework. *Computer Speech and Language*, 10:249–264, 1996.

[39] M.J.F. Gales and S.J. Young. HMM recognition in noise using parallel model combination. *Proceedings Eurospeech*, pages 837–840, 1993.

[40] M.J.F. Gales and S.J. Young. Robust continuous speech recognition using parallel model combination. *IEEE Transactions on Speech and Audio Processing*, 4:352–359, 1996.

[41] J.L. Gauvain and C. H. Lee. Maximum a-posteriori estimation for multivariate Gaussian mixture observations of Markov chains. *IEEE Transactions on Speech and Audio Processing*, 1994.

[42] J.J. Godfrey, R.C. Holliman, and J. McDaniel. SWITCHBOARD: telephone speech corpus for research and development. *Proceedings ICASSP*, pages 517–520, 1992.

[43] B. Gold and N Morgan. *Speech and Audio Processing*. Wiley, 1999.

[44] E. Gouvea. *Acoustic-Feature-Based Frequency Warping for Speaker Normalization*. PhD thesis, ECE Department, CMU, 1999.

[45] P.J. Green. On the use of the EM algorithm for penalized likelihood estimation. *Journal of the Royal Statistical Society, Series B*, 52(3):443–452, 1990.

[46] R. Haeb-Umbach and H. Ney. Linear discriminant analysis for improved large vocabulary continuous speech recognition. *Proceeding ICASSP*, pages 13–16, 1992.

[47] S. Hagen, F Metze, and A. Waibel. Compensating for hyperarticulation by modeling articulatory properties. In *Proceedings ICSLP*, pages 841–844, 2002.

[48] T. Hain, P.C. Woodland, G. Evermann, and D. Povey. New features in the CU-HTK system for transcription of conversational telephone speech. In *Proceedings ICASSP*, pages 1973–1976, 2001.

[49] T. Hain, P.C. Woodland, T.R. Niesler, and E.W.D. Whittaker. The 1998 HTK system for transcription of conversational telephone speech. In *Proceedings ICASSP*, pages 57–60, 1999.

[50] H. Hermansky. Perceptual linear prediction (PLP) of speech. *Journal of the Acoustic Society of America*, 87(4):1738–1752, 1990.

[51] H. Hermansky. Should speech recognisers have ears? *Speech Communication*, 25(1-3):3–27, 1998.

[52] H. Hermansky and N. Morgan. RASTA processing of speech. *IEEE Transactions SAP*, 2:578–579, 1994.

[53] H.G. Hirsch and D Pearce. The Aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions. In *Proceedings ASR*, 2000.

[54] W.J. Holmes. Low bit rate speech coding using a linear-trajectory formant representation for both recognition and synthesis. *Proceedings of the IOA*, 20(6):179–186, 1998.

[55] W.J. Holmes and P.N. Garner. On the robust incorporation of formant frequencies into hidden Markov models for automatic speech recognition. In *IEEE Proceedings ICASSP*, pages 1546–1549, 1998.

[56] W.J. Holmes and J.N. Holmes. The use of formants as acoustic features for automatic speech recognition. *Proceedings of the IOA*, 18(9):275–283, 1996.

[57] W.J. Holmes, J.N. Holmes, and P.N. Garner. Using formant frequencies in speech recognition. In *Proceedings Eurospeech*, 1997.

[58] W.J. Holmes and M.J. Russell. Probabilistic-trajectory segmental HMMs. *Computer Speech and Language*, 13:3–37, 1999.

[59] Z. Hu and E. Barnard. Smoothness analysis for trajectory features. In *Proceedings ICASSP 97*, pages 979–982, 1997.

[60] X.D. Huang and M.A. Jack. Semi-continuous hidden Markov models for speech signals. *Computer Speech and Language*, 3:239–251, 1989.

[61] M.J. Hunt. Delayed decisions in speech recognition - the case for formants. *Pattern Recognition Letters*, 6:121–137, 1987.

[62] D.J. Iskra and W.H. Edmondson. Feature-based approach to speech recognition. *Proceedings of the IOA*, 20(6):83–89, 1998.

[63] P.J.B Jackson and M.J. Russell. Models of speech dynamics in a segmental-HMM recognizer using intermediate linear representations. In *Proceedings ICSLP*, pages 1253–1256, 2002.

[64] F. Jelinek. Continuous speech recognition by statistical methods. *Proceedings IEEE*, 64(4):532–557, 1976.

[65] K. Johnson, P. Ladefoged, and M. Lindau. Individual differences in vowel production. *Journal of the Acoustic Society of America*, 94(2):701–714, 1993.

[66] B.H. Juang and L.R. Rabiner. A probabalistic distance measure for hidden Markov models. *AT&T Technical Journal*, 64:391–408, 1985.

[67] B.H. Juang, L.R. Rabiner, and Wilpon J.G. On the use of bandpass liftering in speech recognition. *IEEE Transactions on Acoustics, Speech and Audio Processing*, 35:947–954, 1987.

[68] T. Kamm, G. Andreou, and J. Cohen. Vocal tract normalization in speech recognition: Compensating for systematic speaker variability. In *Proceedings of the 15th Annual Speech Research Symposium*, pages 161–167, 1995.

[69] M. Katz, H-G. Meier, H. Dolfing, and D. Klakow. Robustness of linear discriminant analysis in automatic speech recognition. *Proceedings ICPR*, 2002.

[70] S.M. Katz. Estimation of probabilities from sparse data for the language model component of a speech recogniser. *IEEE Transactions Acoustics, Speech and Signal Processing*, 35(3):400–411, 1087.

[71] K. Kirchoff. Combining articulatory and acoustic information for speech recognition in noisy and reverberant environments. In *Proceedings ICSLP*, pages 873–876, 1998.

[72] D.H. Klatt. Prediction of perceived phonetic distance from critical band spectra: A first step. In *Proceedings ICASSP*, pages 1278–1281, 1982.

[73] R. Kuhn, P. Nyugen, J-C. Junqua, L. Goldwasser, N. Niedzielski, S. Fincke, K. Field, and M. Contolini. Eigenvoices for speaker adaptation. *Proceedings Eurospeech*, pages 1155–1158, 1998.

[74] L. Lee and R. C. Rose. Speaker normalisation using efficient frequency warping procedures. In *Proceedings ICASSP*, volume 1, pages 353–356, 1996.

[75] C.J. Leggetter and P.C. Woodland. Maximum likelihood linear regression for speaker adaptation of continous density HMMs. *Computer Speech and Language*, 9(2):171–185, 1995.

[76] L.A. Liporace. Maximum likelihood estimation for multivariate observations of Markov sources. *IEEE Transactions Information Theory*, 28:729–734, 1982.

[77] J.D. Markel and A.H. Gray Jr. Linear prediction of speech. *Springer-Verlag*, 1976.

[78] P. McMahon, P. McCourt, and S. Vaseghi. Discriminative weighting of multi-resolution subband cepstral features for speech recognition. In *Proceedings ICSLP*, pages 1055–1058, 1998.

[79] F. Metze and A. Waibel. A flexible stream architecture for ASR using articulatory features. In *Proceedings ICSLP*, pages 2133–2136, 2002.

[80] L. Neumeyer and M. Weintraub. Probabilistic optimal filtering for robust speech recognition. In *Proceedings ICASSP*, pages 417–420, 1994.

[81] D. O'Shaughnessy. *Speech Communtication, Human and Machine*. Addison-Wesley, 1987.

[82] M. Ostendorf, V.D. Digilakis, and O.A. Kimball. From HMMs to segment models: A unified view of stochastic modelling for speech recognition. *IEEE Transactions on Speech and Audio Processing*, 4(5):360–378, 1996.

[83] M. Padmanabhan. Spectral peak tracking and its use in speech recognition. In *Proceedings ICSLP*, pages 1747–1750, 2000.

[84] K.K. Paliwal. Spectral subband centroid features for speech recognition. In *Proceedings ICASSP*, pages 617–620, 1998.

[85] D.S. Pallett, J.G. Fiscus, W.M. Fisher, J.S. Garofolo, B.A. Lund, A. Martin, and M.A. Przybocki. 1994 benchmark tests for the ARPA spoken language program. *Proceedings ARPA Workshop on Spoken Language Systems Technology*, pages 5–36, 1995.

[86] D. Paul. The spectral envelope estimation vocoder. *IEEE Transactions on Speech and Audio Processing*, 29(4):786–794, 1981.

[87] D.B. Paul and J.M. Baker. The design for the wall street journal-based corpus. *Proceedings ICSLP*, pages 899–902, 1992.

[88] G.E. Peterson and H.L. Barney. Control methods used in a study of the vowels. *Journal of the Acoustic Soceity, America*, 24(2):175–194, March 1952.

[89] G. Potamianos and H.P. Graf. Discriminative training of HMM stream exponents for audio-visual speech recognition. In *Proceedings ICSLP*, pages 3733–3736, 1998.

[90] D.P. Povey and P.C. Woodland. Frame discrimination training of HMMs for large vocabulary speech recognition. *Proceedings ICASSP*, pages 333–336, 1999.

[91] P. Price, W.M. Fisher, J. Bernstein, and D.S. Pallett. The DARPA 1000-word Resource Management database for continuous speech recognition. *Proceedings ICASSP*, pages 651–654, 1988.

[92] D. Pye and P.C. Woodland. Experiments in speaker normalisation and adaptation for large vocabulary speech recognition. In *Proceedings ICASSP*, pages 1047–1050, 1997.

[93] A.J. Robinson. An application of recurrent neural nets to phone probability estimation. *IEEE Transactions on Neural Networks*, 5(2):298–395, 1994.

[94] X. Rodet, P. Depalle, and G. Poirot. Speech analysis and synthesis methods based on spectral envelopes and voiced/unvoiced functions. In *European Conference on Speech Technology*, 1987.

[95] A-V.I. Rosti and M.J.F. Gales. Factor analysed hidden Markov models. *Proceedings ICASSP*, pages 949–952, 2002.

[96] G. Saon, G. Zweig, and M. Padmanabhan. Linear feature space projections for speaker adaptation. In *Proceedings ICASSP*, pages 2083–2086, 2001.

[97] R. Schlueter and H. Ney. Using phase information for improved speech recongition performance. In *Proceedings ICASSP*, pages 1010–1013, 2001.

[98] P. Schmid and E. Barnard. Robust, n-best formant tracking. In *Proceedings Eurospeech*, pages 737–740, 1995.

[99] P. Schmid and E. Barnard. Explicit, n-best formant features for vowel classification. In *Proceedings ICASSP 97*, pages 991–994, 1997.

[100] N.D. Smith and M.J.F. Gales. SVMs for speech recognition. *Proceedings ICASSP*, pages 2860–2683, 2002.

[101] M. Stuttle and M.J.F. Gales. A mixture of Gaussians front end for speech recognition. In *Proceedings Eurospeech*, pages 675–678, 2001.

[102] M. Stuttle and M.J.F. Gales. Combining a Gaussian mixture model front end with MFCC parameters. In *Proceedings ICSLP*, pages 1565–1568, 2002.

[103] D. Talkin. Speech formant trajectory estimation using dynamic programming with modulated transition costs. *Journal of the Acoustic Society of America*, 82:S55, 1987.

[104] S. Tsuge, T. Fukada, and H. Singer. Speaker normalised spectral subband parameters for noise robust speech recognition. In *Proceedings ICASSP*, pages 1686–1689, 1999.

[105] L.F. Uebel and P.C. Woodland. An investigation into vocal tract length normalisation. *Proceedings Eurospeech*, pages 2519–2522, 1999.

[106] A. Varga and R.K. Moore. Hidden markov model decomposition of speech and noise. In *Proceedings ICASSP*, pages 845–848, 1990.

[107] K. Weber, S. Bengio, and H. Bourlard. HMM-2 - a novel approach to HMM emission probability estimation. In *Proceedings ICSLP*, pages 147–150, 2000.

[108] K. Weber, S. Bengio, and H. Bourlard. HMM-2 extraction of formant structures and their use for robust ASR. In *Proceedings Eurospeech*, pages 607–610, 2001.

[109] K. Weber, F. de Wet, B. Cranen, L. Boves, S. Bengio, and H. Bourlard. Evaluation of formant-like features for ASR. In *Proceedings ICSLP*, pages 2101–2104, 2002.

[110] K. Weber, S. Ikbal, S. Bengio, and H. Bourlard. Robust speech recognition and feature extraction using HMM-22. *Computer, Speech, and Language*, 17(2-3):195–211, 2003.

[111] L. Welling and H. Ney. Formant estimation for speech recognition. *IEEE Transactions on Speech and Audio Processing*, 6(1):36–48, 1998.

[112] B. Widrow. Adaptive noise cancelling: Principles and applications. *IEEE Proceedings*, 63:1692–1716, 1975.

[113] N. Wilkinson and M.J. Russell. Progress towards improved spech modelling using asynchronous sub-bands and formant frequencies. In *Proceedings WISP*, pages 2121–2124, 2001.

[114] N.J. Wilkinson and M.J. Russell. Improved phone recognition on TIMIT using formant frequency data and confidence measures. In *Proceedings ICSLP*, 2002.

[115] J.G. Wilpon, C.H. Lee, and L.R. Rabiner. Improvements in connected digit recognition using higher order spectral and energy features. *Proceedings ICASSP*, pages 349–352, 1991.

[116] P.C. Woodland. Speaker adaptation: techniques and challanges. In *Proceedings ASRU*, 1999.

[117] P.C. Woodland, T. Hain, S.E. Johnson, T.R. Niesler, A. Tuerk, E.W.D. Whittaker, and S.J. Young. The 1997 HTK broadcast news transcription system. In *Proc. 1998 DARPA Broadcast News Transcription and Understanding Workshop*, 1998.

[118] P.C. Woodland, C.J. Leggetter, J.J. Odell, V. Valtchev, and S.J. Young. The 1994 HTK large vocabulary speech recognition system. In *Proceedings ICASSP*, pages 73–76, 1995.

[119] P.C. Woodland, J.J. Odell, V. Valchev, and S.J. Young. The development of the 1994 HTK large vocabulary speech recognition system. In *Proceedings ARPA workshop on spoken language systems technology*, 1994.

[120] T-Y. Yang, W-H. Shin, W-G. Kim, D-H. Youn, and I-W. Cha. On using formants to improve SCHMM speaker adaption. *IEEE Transactions on Speech and Audio Processing*, 7(2):226–230, 1999.

[121] P.C. Young, S.J. Woodland. The use of state tying in continuous speech recognition. In *Proceedings Eurospeech*, pages 2207–2210, 1993.

[122] S.J. Young, G. Evermann, D. Kershaw, G.L. Moore, J.J. Odell, D. Ollason, V. Valtchev, and P.C. Woodland. *The HTK Book, Version 3.1*. Cambridge University Engineering Department, 2002.

[123] S.J. Young, J.J. Odell, and P.C. Woodland. Tree based state tying for high accuracy acoustic modelling. In *ARPA Human Language Technology Workshop*, 1994.

[124] Q. Zhu and A. Alwan. On the use of variable frame rate analysis in speech recognition. *Proceedings ICASSP*, pages 3264–3267, 2000.

[125] P. Zolfaghari and A.J. Robinson. Formant analysis using mixtures of Gaussians. In *Proceedings ICSLP*, pages 904–907, 1996.

[126] P. Zolfaghari and A.J. Robinson. A formant vocoder based on mixtures of Gaussians. In *Proceedings ICASSP*, pages 1575–1578, 1997.

[127] P. Zolfaghari and A.J. Robinson. Speech coding using a mixture of Gaussians polynomial model. In *Proceedings Eurospeech*, pages 1495–1498, 1999.