TEL AVIV UNIVERSITY אוניברסיטת תל-אביב
FACULTY OF ENGINEERING הפאקולטה להנדסה

RANDOM TRAJECTORY SEGMENTAL MODELS FOR IMPROVED AUTOMATIC SPEECH RECOGNITION

Thesis submitted towards the degree of

"Doctor of Philosophy"

by

JACOB A. GOLDBERGER

Submitted to the senate of Tel-Aviv University

Nov 1997

This work was carried out under the supervision

of

Dr. DAVID BURSHTEIN

and

Prof. EHUD WEINSTEIN

Acknowledgments

First and foremost I would like to thank Dr. David Burshtein for his guidance and support throughout all my Ph.D studies and his helpful remarks and constructive reviews of my work. I would also like to thank Prof. Ehud Weinstein.

I would like to thank the members of the speech group at SRI international for the inspiring time during my visit at SRI in summer 1996. In particular I want to thank Horacio Franco and Yochai Konig. Special thanks to Sharon Ganot for his assistance and friendship. I thank Yona Eyal, our System Manager, who was always ready to help with any computer problems. Thanks to Sarah Yarkoni, Yonina Berglas and Gavriel Speyer for proofreading portions of my English manuscripts. Last but surely not the least, I thank my parents for their support and encouragement.

This thesis is dedicated to my parents Shlomo and Bracha Goldberger.

ABSTRACT

In this thesis we present alternative statistical models of phonetically based segments for use in continuous speech recognition. Segment models as opposed to hidden Markov models treat an entire phonetic segment as a single entity. This approach aims to capture the local behavior and statistical dependencies, as well as the long term variability existing in the acoustic feature sequence used to represent the speech waveform. The random trajectory segmental modeling approach tries to achieve these modeling goals by describing the mean trajectory of a phonetic unit along time as a random process instead of a fixed parameter. The main part of this work analyzes the short-comings of random trajectory segmental models suggested in the past and presents models that hopefully function better.

Random segmental models that have been proposed suffer from modeling problems caused by the fact that segment realizations of the same phone differ in length. A study described here shows the benefits of rescaling the model parameters according to the segment length. In the rescaled model the variance of the random mean trajectory is inversely proportional to the segment length. It is shown that, unlike previously suggested models, each frame contributes the same amount of information during the mean trajectory estimation process. A technical advantage of rescaling is the much simplified parameter estimation procedure. In a rescaled model there is no need to approximate the target function or to use an iterative procedure in order to find the maximum likelihood estimate. Instead, a closed form can be obtained. Phonetic classification experiment results support this approach. It was found that rescaling the model can improve recognition rate.

In order to capture the local dynamic behavior of speech, we propose a non-parametric description of the mean trajectory. This model does not impose any smooth structure on the random mean trajectory. Instead, it represents it by specifying a list of sample points along time. Classification experiments demonstrate the importance of modeling the local dynamic behavior.

The coarticulation effect approves the usage of triphone models. We concentrate on a segmental model which is based on a description of the mean trajectory along the feature space. We present a method for constructing a synthetic mean trajectory for each triphone context in order to enable model sharing across triphones.

Contents

1 Introduction **1**

1.1 Automatic Speech Recognition 1

1.2 Thesis Outline 4

2 Hidden Markov Models **5**

2.1 The Hidden Markov Model 5

2.2 Parameter Estimation 10

2.3 Using HMM for Speech Recognition 14

3 Segmental Models **19**

3.1 Limitations of the HMM Paradigm 19

3.2 The Concept of Segmental Modeling 24

3.3 Previous Segmental Models 25

4 Scaled Random Trajectory Segmental Models **35**

4.1 Introduction 35

4.2 Static Random Segmental Model 37

4.3 Scaled Random Trajectory Segment Models 43

4.4 Scaled Linear Random Trajectory Segmental Models 48

4.5 Baum-Welch re-Estimation 53

4.6 Experimental Results 58

4.7 Conclusions 62

5 Continuous Mixture of Segmental Models **69**

5.1 Motivations 69

5.2 Model Formulation 71

5.3 Recognition and Training Algorithms 74

5.4 A Simplified Version of the Model 79

5.5 Experimental Results 85

5.6 Conclusions 88

6 Synthetic Segmental Triphone Models **91**

6.1 Introduction 91

6.2 Synthetic Modeling of the Mean Trajectory 94

6.3 Experimental Results 96

6.4 Conclusions 98

7 Conclusions **100**

7.1 Thesis Contribution 100

7.2 Future Work 103

A Corpus and Signal Processing **105**

List of Figures

2.1 Modeling the mean trajectory of the phone 'p' using 3 states HMM. . 16
2.2 Modeling the mean trajectory of the phone 'p' using 3 states 3 mixture HMM. 16

3.1 Sixth cepstral coefficient of the acoustic vector and its HMM approximation. 21
3.2 Time derivative of the sixth cepstral coefficient of the acoustic vector and its HMM approximation. 21
3.3 Piecewise constant approximation of sin and its derivative cos. 22
3.4 HMM with 5 states and 3 mixtures that models the sixth cepstral coefficient. 23
3.5 HMM with 5 states and 3 mixtures that models the derivative of the sixth cepstral coefficient. 23
3.6 Modeling the mean trajectory of the phone 'l' with 3 states HMM. . . 28
3.7 Modeling the mean trajectory of the phone 'l' with 3 states linear trended HMM. 28

4.1 The inverse of the variances of the sampled trajectory as a function of the segment length. 45

5.1 Realizations of first cepstral coefficient of the phone 's' in the triphone context ih-ow. 70
5.2 Data after nonlinear time warping. 70

5.3 Data after nonlinear time warping and displacement elimination. . . . 71
5.4 Non-parametric model of the mean trajectory. 78

6.1 Mean trajectories of the phone 'l' for different triphone contexts. . . . 92
6.2 Mean trajectories of the phone 'ao' in various triphone contexts. . . . 95
6.3 Mean trajectory of the triphone $s-ao-r$. 97
6.4 Synthetic construction of the mean trajectory of the triphone $s-ao-r$ from simpler elements. 97

List of Tables

3.1 Transition matrix of the three mixtures of the second state presented in figure 3.4. 24

3.2 Same as Table 3.1, consecutive self transitions is counted only once. . 25

4.1 Phoneme classification rate results 60

4.2 Triphone classification rate results 61

5.1 Word error rate results without language model. 86

5.2 Word error rate results with language model. 86

5.3 Triphone recognition rate results. 87

6.1 Phone classification results. 97

6.2 Triphone classification results. 98

A.1 List of phone symbols and examples. 106

Chapter 1

Introduction

1.1 Automatic Speech Recognition

Today, as the twentieth century draws to a close, it is argued that computers can perform most data processing tasks more accurately and quickly than humans. There are, however, areas where the human mind still outperforms automated machines. Processing human language is still an intriguing challenge in computer science. For example, we have yet to develop an automatic procedure for translation from one language to another which yields results comparable to that of a professional human translator. One explanation for the exceptional difficulties in automatic manipulation of human language is that language is not only a means of communication between individuals but an outgrowth of human perception. This idea was formulated by Ludwig Wittgenstein in his book Tractatus Logico Philosophicus in the famous phrase : "the limits of my language mean the limits of my world". In automatic speech recognition there is another level of complexity. In addition to the philosophical and linguistic problems, there is the technical problem of extracting the linguistic information from the acoustic waveform. Most of the information conveyed in recorded speech is irrelevant and can even be misleading for the purpose of

automatic determination of the word transcription from a given utterance.

The progress in computer architecture over the past twenty years has encouraged the interest in a new research area which deals with developing accurate speech recognition systems. A speech recognition system is composed of several elements. The first element is signal processing. The recorded speech waveform is processed in order to extract a sequence of feature vectors that preserve the linguistic information incorporated in the signal which is necessary for recognition. The feature vectors are associated with phonetic units of the language via acoustic models that describe how abstract linguistic elements are represented by acoustic events. Another element of speech recognition is the formulation of efficient algorithms that operate on an entire sentence. These algorithms integrate the local information obtained from acoustic models in order to find the best word transcription of the recorded waveform. Other knowledge sources can be integrated in the system. The most important source is language modeling. Knowledge of grammatical rules and the statistics of words in the language can greatly reduce the uncertainty about the message conveyed in the speech signal.

Speech recognition is largely focused on acoustic modeling. The modeling difficulty is caused by the fact that on the one hand a model is expected to exhibit a precise and informative description of the data. On the other hand, however, the acoustic model must be relaxed enough to manage the acoustic variability exhibited by different speakers, dialects, recording equipment and adverse conditions. Several approaches to acoustic modeling have been proposed over the last twenty years. The template based approach was the first to be suggested. This approach is based on the Dynamic Time Warping (DTW) algorithm [60]. An unknown utterance is compared against a list of pre-recorded words in order to find the best match. A drawback of template comparison is that there is no attempt to understand the structure of the speech. Therefore, there is no possibility of generalizing the information conveyed in the template to various speech situations. This approach, however, is still the best known in scenarios of user dependent, small vocabulary,

isolated word recognition systems where there is a demand for training a word from a small number of occurrences. In recent years there has been an attempt to model speech acoustics using artificial neural networks. Neural networks are well suited for acoustic scoring, as they can accept continuous valued inputs without making any assumption as to the parametric shape of the density function. The most successful approach for acoustic modeling is, however, the statistical one. The introduction of hidden Markov models (HMM) into speech recognition in the late seventies [2] was a revolutionary step. Today HMM is the dominant paradigm in the area and most state-of-the-art large vocabulary systems are HMM based.

The enormous success of HMM caused a reaction of critical analysis of the shortcomings that still remain in this approach. The HMM presents a considerable success in automatic segmentation of speech into a sequence of phonetic-acoustic events. Mixture densities HMM is a relaxed and flexible model that can survive speech variability. The local acoustic modeling of HMM, however, suffers from the fact that the frame serves as the basic modeling unit. A frame oriented point of view is too local. Many speech effects last for an entire utterance or at least along the pronunciation of a phonetic unit. The HMM theory does not provide tools for modeling effects that remain fixed over a sequence of consecutive frames.

In this thesis we shall discuss in detail the shortcomings of HMM. We present the segmental modeling approach which intends to overcome these shortcomings by modeling an entire acoustic event as a single entity. In this study we concentrate on a particular family of segmental models known as random segmental models. The essence of random models is treating the mean trajectory along the feature space as a random process instead of a fixed parameter. It is assumed that this random process is updated at a slower rate than the frame producing rate and therefore it preserves the long term variability of the speech signal. The main part of this work analyzes the short-comings of random models suggested in the past and suggests random models that hopefully function better.

1.2 Thesis Outline

The first two chapters of this thesis provide some essential background and a summary of related work in speech recognition. Chapter 2 describes the hidden Markov model and explains how it is trained and used in speech recognition tasks. Chapter 3 describes the segmental modeling concept and reviews segmental models which are related to our work. The remainder of this thesis describes our own research on random segmental models. Chapter 4 presents alternative algorithms for training the random segmental model and introduces the concept of scaled random modeling. Chapter 5 extends the random segmental theory to the case of non-parametric representation of the mean trajectory. Chapter 6 presents a method for sharing segmental models, which describe the mean trajectory, across triphone contexts. Chapter 7 draws overall conclusions and suggests possible future work.

Chapter 2

Hidden Markov Models

In this chapter we shall present the concept of hidden Markov models (HMM), which is the predominant technique employed in automatic speech recognition. We shall first define the probabilistic model and present algorithms for decoding. Then, we re-derive the Baum-Welch training algorithm which can be considered a special case of the EM algorithm. Finally, we shall discuss how the speech recognition problem can be handled with HMM.

2.1 The Hidden Markov Model

A Hidden Markov Model(HMM) is a parametric probabilistic model that was found to be suitable for describing speech events. One of the reasons for the success of HMMs in speech recognition tasks is the existence of computationally efficient algorithms for decoding a given unknown utterance and for estimating the model parameters. The performance of speech recognizers that employ hidden Markov modeling has been proven superior to alternative recognition methods in a variety of real life applications, and in particular for speaker independent large vocabulary recognition tasks. HMM was introduced by Baum [3] in the early seventies. The

first implementation of HMM for speech recognition was conducted by Baker [2] and the IBM Speech Group [36]. This chapter includes a brief survey of the HMM theory. A detailed treatment of the hidden Markov model and its applications can be found in [53] [55] [32].

A discrete stochastic process has a Markovian distribution if the conditional probability of the current event, given all the past events, depends only on the most recent event. The probability of a sample sequence from a Markovian source can be written as :

$$p(s_1, ..., s_n) = p(s_1)p(s_2|s_1) \cdots p(s_n|s_{n-1})$$

A stochastic process x has a hidden Markov distribution, if there exists a Markov process s such that the probability of x can be written as :

$$p(x_1, ..., x_n) = \sum_{s=s_1,...,s_n} p(s) \prod_{t=1}^{n} p(x_t|s_t)$$

In other words, the process is HMM if the samples are independent given the hidden Markov states. We can consider the HMM as a two step experiment. In the first step, a state sequence is realized according to the Markov model. In the second step, an output variables sequence is realized according to the state sequence in such a manner that the output at time t is sampled according to the distribution associated with the state of the Markov process at time t. At the end of the experiment only the output is reported and the state sequence remains unobserved.

The parametric representation of the hidden Markov model consists of the following components :

1. Initial state distribution : $\lambda_j = p(s_1 = j)$.

2. State transition probabilities of the unobserved stationary Markov process :

$$P(k, j) = p(s_{t+1} = j|s_t = k)$$

3. Output probability distribution associated with each state which determines the likelihood of the observations generated by this state :

$$f_j(x_t) = f(x_t|s_t = j)$$

Several approaches exist for modeling the output distribution. One possibility is to use discrete distributions [46]. Each observed vector is quantized into a symbol from a finite symbol set. A vector quantizer selects a vector closest to the observed vector from a pre-defined code-book. A discrete distribution of the code-book entries is associated with each state, and the probability of the observation is defined to be the probability of the chosen entry. A disadvantage of this approach is the quantization noise added to the data.

Another commonly used distribution is the Gaussian and more generally a mixture of several Gaussian distributions [38] [54]. The term mixture stands for a convex combination of distributions. The concept of mixture of densities fits naturally into the framework of HMM. The choice of a particular member of the distribution set, that composes the mixture distribution, can be considered as part of the unobserved Markov process. Hence, output density can remain relatively simple.

In real situations there are many states, and there is insufficient training data for reliable parameter estimation of the mixture distribution associated with each state. In these cases a combination of continuous and discrete distributions can be used. This modeling approach is usually termed semi-continuous or tied mixture [31]. A tied mixture model is composed of a set of Gaussian densities which are common to all states. Each state is only characterized by its own set of mixture weights. The concept of tied mixtures was a key step in constructing large vocabulary systems. Many tying methods have been suggested. We shall return to this subject in chapter 6.

There are several efficient algorithms which are an important part of the HMM theory. The following algorithm enables efficient computation of the likelihood of

an observation sequence. The definition of the hidden Markov model implies that the likelihood of a sample sequence $x_1, .., x_n$ is

$$p(x_1, ..., x_n) = \sum_s \lambda(s_1) \prod_{t=2}^{n} p(s_t|s_{t-1}) \prod_{t=1}^{n} f(x_t|s_t) \tag{2.1}$$

where the summation extends over all possible length-n state sequences. Direct computation of this expression involves summation over an exponential number of terms which is not feasible. However, there exists a recursive approach which can significantly facilitate the computation.

For each t define the following column vector :

$$A_t(j) = p(x_1, ..., x_t, s_t = j)$$

A recursion formula can be derived for $A_t(j)$ in the following way :

$$\begin{aligned} A_t(j) &= p(x_1, ..., x_t, s_t = j) = \sum_k p(x_1, ..., x_t, s_{t-1} = k, s_t = j) \\ &= \sum_k p(x_1, ..., x_{t-1}, s_{t-1} = k) \cdot p(s_t = j|x_1, ..., x_{t-1}, s_{t-1} = k) \cdot \\ &\qquad p(x_t|x_1, ..., x_{t-1}, s_{t-1} = k, s_t = j) \\ &= \sum_k A_{t-1}(k)\, p(s_t = j|s_{t-1} = k)\, f(x_t|s_t = j) = \sum_k A_{t-1}(k)\, P(k, j)\, f(x_t|s_t = j) \end{aligned}$$

Matrix notation can be used to express this recursion :

$$\begin{aligned} A_1 &= \lambda^T M_1 \\ A_t &= A_{t-1}\, P\, M_t \qquad t = 2, ..., n \end{aligned}$$

where λ denotes the initial state distribution column vector, P denotes the transition matrix of the Markov process and M_t is a diagonal matrix with the following diagonal

elements :

$$M_t(j,j) = \ f(x_t|s_t = j)$$

Note that

$$p(x_1, ..., x_n) = \sum_j p(x_1, ..., x_n, s_n = j) = A_n \, 1$$

where 1 is an all ones column vector. From the recursive relation we can derive an explicit expression for the likelihood function :

$$p(x_1, ..., x_n) = \lambda^T M_1 \, P \, M_2 \ \ \cdots \ P \, M_n \, 1$$

The complexity of this matrix multiplication is linear in the length of the observation sequence and quadratic in the number of states. Hence, the likelihood can be efficiently computed. Further technical issues can be found in [53] [32].

The likelihood of the observations is obtained from summing over all the state sequences. The single most likely state sequence may also be required in algorithms which approximate the exact likelihood. The single best state sequence can be computed using the Viterbi algorithm [64] as follows :

$$\begin{aligned}
\delta(j,1) &= \lambda_j \, f(x_1|s_1 = j) \\
\delta(j,t) &= \max_k \{\delta(k,t-1) \, P(k,j)\} f(x_t|s_t = j) \qquad t = 2, ..., n \\
\psi(j,t) &= \arg\max_k \{\delta(k,t-1) \, P(k,j)\} \qquad t = 2, ..., n
\end{aligned}$$

The probabilistic interpretation of $\delta(j,t)$ is :

$$\delta(j,t) = \max_{s_1, ..., s_{t-1}} p(x_1, ..., x_t, s_1, ..., s_{t-1}, s_t = j)$$

Therefore, the joint probability of the observations and the most likely state sequence

is :

$$\max_s p(x,s) = \max_j \delta(j,n)$$

The best state sequence can be found in a traceback pass :

$$\begin{aligned} \hat{s}_n &= \arg\max_j \delta(j,n) \\ \hat{s}_t &= \psi(\hat{s}_{t+1}, t+1) \qquad t = n-1, ..., 1 \end{aligned}$$

2.2 Parameter Estimation

Training HMM systems involves estimating the model parameters. The most popular estimation criterion is maximum likelihood (ML). There is no known closed-form solution to the problem of finding ML estimators for the HMM parameters. There exists, however, an elegant iterative algorithm for this problem, namely the Baum-Welch algorithm. This algorithm can be considered a special case of the EM algorithm which is described in its general form in the appendix of this chapter. To simplify the derivation of the training algorithm we assume that the output probability function associated with each state is Gaussian. Training algorithms for other output distributions can be found in [53]. Denote the Gaussian density function associated with state j by $N(\mu_j, \Sigma_j)$. The model parameters are, therefore, the initial distribution, the Markov transition matrix and the parameters of the Gaussian distributions associated with each state. We shall now describe the derivation of the EM re-estimation equations. Denote the current estimate of the parameters by θ_0. In each iteration of the EM, the parameters are re-estimated. The general theory of the EM algorithm ensures an increase in the likelihood in each iteration. Let $x_1, ..., x_L$ be L sequences consisting the training data-base. Denote the length of x_i by n_i. The elements of the sequence x_i are denoted by $x_{i,1}, ..., x_{i,n_i}$. Denote by $s_{i,t}$ the underlying state at time t for the sequence x_i. Using EM terminology we can

refer to the sequences $s_1, ..., s_L$ as the missing data. The iterative algorithm relies on the fact that if these sequences were known, we could have obtained closed-form expressions for the ML estimation. From equation (2.1) we obtain :

$$\begin{aligned}
&\log p(x_1, s_1 ..., x_L, s_L, \theta) = \\
&\sum_{i=1}^{L} \sum_{j} I_{\{s_{i,1}=j\}} \log \lambda_j \quad + \quad \sum_{i=1}^{L} \sum_{t=2}^{n_i} \sum_{j,k} I_{\{s_{i,t-1}=k \ , \ s_{i,t}=j\}} \log P(k,j) \\
&-\frac{1}{2} \sum_{i} \sum_{t} \sum_{j} I_{\{s_{i,t}=j\}} ((x_{i,t} - \mu_j)^T \Sigma_j^{-1} (x_{i,t} - \mu_j) + \log |\Sigma_j|)
\end{aligned}$$

where I_A is the indicator random variable associated with the event A.

$$I_A(\omega) = \begin{cases} 1 & \omega \in A \\ 0 & \omega \in A^c \end{cases}$$

Denote :

$$\begin{aligned}
w_i(j,t) &= E(I_{\{s_{i,t}=j\}} | x_i, \theta_0) \\
u_i(k,j,t) &= E(I_{\{s_{i,t-1}=k \ , s_{i,t}=j\}} | x_i, \theta_0)
\end{aligned}$$

$w_i(j,t)$ is the aposteriori probability that while producing the sequence x_i, state j was visited at time t. The EM auxiliary function in this case is :

$$\begin{aligned}
Q(\theta, \theta_0) &= \sum_{i} E(\log p(x_i, s_i, \theta) | x, \theta_0) \\
&= \sum_{i} \sum_{j} w_i(j,1) \log \lambda_j \quad + \quad \sum_{i} \sum_{t=2}^{n_i} \sum_{j,k} u_i(k,j,t) \log P(k,j) \\
&\quad -\frac{1}{2} \sum_{i} \sum_{t} \sum_{j} w_i(j,t) ((x_{i,t} - \mu_j)^T \Sigma_j^{-1} (x_{i,t} - \mu_j) + \log |\Sigma_j|)
\end{aligned}$$

Maximization of $Q(\theta, \theta_0)$ yields the following re-estimation equations :

$$\begin{aligned}
\hat{\lambda}_j &= \frac{\sum_i w_i(j,1)}{L} \\
\hat{\mu}_j &= \frac{\sum_i \sum_t w_i(j,t) x_{i,t}}{\sum_i \sum_t w_i(j,t)} \\
\hat{\Sigma}_j &= \frac{\sum_i \sum_t w_i(j,t)(x_{i,t}-\mu_j)(x_{i,t}-\mu_j)^T}{\sum_i \sum_t w_i(j,t)} \\
\hat{P}(k,j) &= \frac{\sum_i \sum_t u_i(k,j,t)}{\sum_i \sum_t w_i(j,t)}
\end{aligned}$$

Efficient computation of the terms $w_i(j,t)$ and $u_i(k,j,t)$ which appear in the re-estimation equations can be done using the forward-backward algorithm that will now be presented. Applying the Bayese rule yields :

$$w_i(j,t) = E(I_{\{s_{i,t}=j\}}|x_i,\theta_0) = p(s_{i,t}=j|x_i,\theta_0) = \frac{p(x_i\ ,\ s_{i,t}=j|\theta_0)}{p(x_i|\theta_0)}$$

$$u_i(k,j,t) = E(I_{\{s_{i,t-1}=k\ ,\ s_{i,t}=j\}}|x_i,\theta_0) = \frac{p(x_i\ ,\ s_{i,t-1}=k\ ,\ s_{i,t}=j|\theta_0)}{p(x_i|\theta_0)}$$

Define the following diagonal matrices :

$$M_{i,t}(j,j) = f(x_{i,t}|s_{i,t}=j,\theta_0) \qquad t=1,...,n$$

Denote the current estimate of the initial state distribution by λ_0 and the current estimate of the transition Markov matrix by P_0. From the likelihood equation (2.1) we obtain :

$$p(x_i|\theta_0) = \lambda_0^T\, M_{i,1}\, P_0 \cdots P_0\, M_{i,t-1}\, P_0\, M_{i,t} \cdots M_{i,n_i}\, 1$$

Define the following forward-backward equations :

$$\begin{aligned} A_{i,1} &= \lambda_0^T \, M_{i,1} \\ A_{i,t} &= A_{i,t-1} \, P_0 \, M_{i,t} \qquad\qquad t = 2, ..., n_i \\ B_{i,n_i} &= 1 \\ B_{i,t} &= P_0 \, M_{i,t+1} \, B_{i,t+1} \qquad\qquad t = n_i - 1, ..., 1 \end{aligned}$$

Note that $A_{i,t}$ is a row vector and $B_{i,t}$ is a column vector, both having the following probabilistic interpretation :

$$\begin{aligned} A_{i,t}(j) &= p(x_{i,1}, ..., x_{i,t}, s_t = j|\theta_0) \\ B_{i,t}(j) &= p(x_{i,t+1}, ..., x_{i,n}, s_t = j|\theta_0) \end{aligned}$$

Using the above definitions it can be seen that :

$$\begin{aligned} p(x_i|\theta_0) &= A_{i,n_i} \, 1 \\ p(x_i \, , \, s_{i,t} = j|\theta_0) &= A_{i,t}(j) \, B_{i,t}(j) \\ p(x_i \, , \, s_{i,t-1} = k \, , \, s_{i,t} = j|\theta_0) &= A_{i,t-1}(k) \, P_0(k,j) M_{i,t}(j,j) \, B_{i,t}(j) \end{aligned}$$

Therefore using the results of the forward-backward algorithm we can derive the following efficiently computed expressions :

$$\begin{aligned} w_i(j,t) &= (A_{i,t}(j) \, B_{i,t}(j)) \; / \; (A_{i,n_i} \, 1) \\ u_i(k,j,t) &= (A_{i,t-1}(k) \, P_0(k,j) \, M_{i,t}(j,j) \, B_{i,t}(j)) \; / \; (A_{i,n_i} \, 1) \end{aligned}$$

An alternative training method is the segmental k-means algorithm [37], which

approximates the E-step of the Baum-Welch. The maximization step remains the same. The E-step consists of the computation of the aposteriory distribution of the state variables given the observation sequences. The segmental k-means uses the Viterbi algorithm in order to determine the most likely state sequence. Denote by $\hat{s}_{it}$ the best suited state value at time t in the i-th sequence. The segmental k-means uses the following approximation :

$$p(s_{i,t} = j|x_i, \theta_0) \quad \approx \quad I_{\{\hat{s}_{i,t}=j\}}$$

Experiments have shown that using this approximation does not cause any significant degradation in the performance of HMM based recognition systems.

2.3 Using HMM for Speech Recognition

A speech signal is built of a sequence of linguistic units that are transformed into acoustic events. Speech is a product of the vocal tract, a continuously varying system. There is always continuous movement from one phonetic unit to the next. It is, therefore, not easy to find the boundaries between acoustic realizations of phonetic units. Moreover, in continuous speech an exact transition point between phonetic units can not be determined even manually and actually does not exist at all.

When speech is modeled using HMM, each state is associated with a phonetic unit. The model assumes that within a state's boundaries, each observation is dependent only upon that state. In other words, the output distribution associated with the state describes the feature sequence as an IID (independent, identically distributed) process. Although this assumption is far from reality, it can function as a satisfactory approximation. The unobserved state sequence implies a specific segmentation of the utterance into phonetic units. The algorithm, that computes the likelihood, automatically checks all possible state segmentations and chooses the

best phonetic unit sequence and the best locations for the transition points between those units.

HMM was successfully implemented for small-vocabulary word base tasks [54]. For each word in the vocabulary, a hidden Markov model is trained. The Markov topology which is commonly used is a left to right one, such that each state is connected to itself and its successor. The states of the model are supposed to describe the acoustic events during the word pronunciation.

A word is the basic linguistic unit, and therefore words are the most natural units for acoustic modeling. In large vocabulary systems, however, word modeling is not possible. Training data can not be shared between words, and therefore, each word must be individually trained. Many examples of each word are needed necessitating an enormous quantity of training data. Other problems are the memory storage needed and the high complexity of the recognition procedure. A possible solution is to use small acoustic-phonetic units in order to allow sharing across words. Acoustic realizations of words are composed from a small set of basic acoustic sounds named phones. In English there are approximately 50 phones. A phone is an acoustic realization of a phoneme which is an abstract basic linguistic unit. Each word can be presented in phonetic transcription as a concatenation of phonemes. Phone models can be easily trained. They can be sufficiently trained with only few hundred sentences.

A phone is a simple acoustic event compared to a full word. Therefore, the topology of the HMM representing a phone can be simple as well. A typical structure of hidden Markov modeling of a phone is composed of three states. The first state models the transition into the phone; the second models the steady state part of the phone and the third state models the transition to the next phone. Other topologies for phone based HMM have been proposed [46]. The concept of phone based HMM made it possible to build large vocabulary continuous speech recognition systems with satisfactory results.

In order to illustrate phone modeling we present a concrete example. The data

we used was a set of realizations of the phone 'p' in the triphone context ow-eh taken from the *Wall Street Journal* corpus. In Figure 2.1 we show the mean trajectory of the first cepstral coefficient and the approximation of this trajectory with a three state HMM. In Figure 2.2 we show how the approximation is improved by using three mixtures.

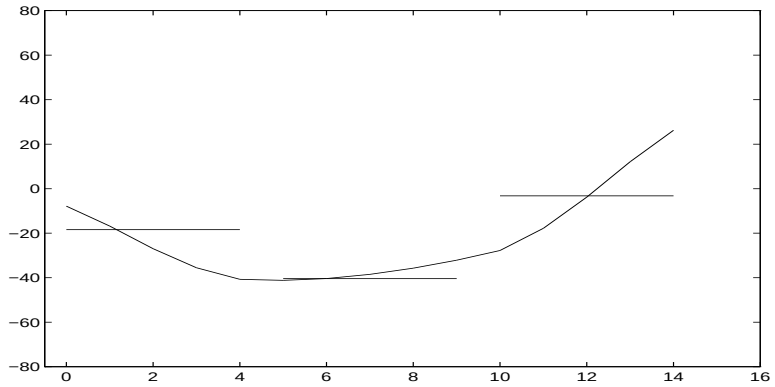

Figure 2.1: Modeling the mean trajectory of the phone 'p' using 3 states HMM.

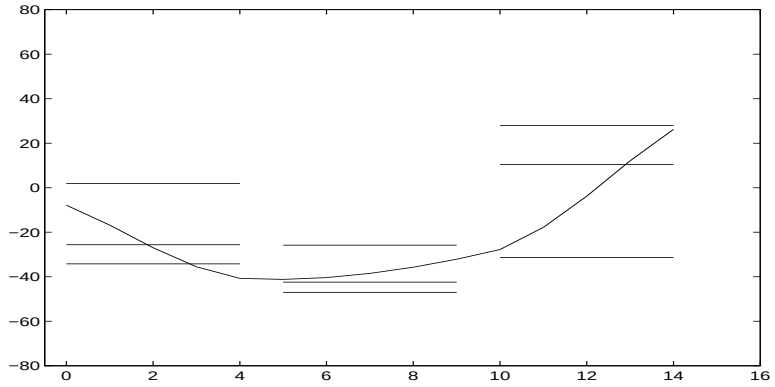

Figure 2.2: Modeling the mean trajectory of the phone 'p' using 3 states 3 mixture HMM.

Appendix 2.A

The EM algorithm is a major component in the theory of acoustic modeling. It is an iterative numerical procedure for finding a maximum likelihood estimation for a given model. The EM was first formulated by Laird, Dempster and Rubin in 1977 [5]. The Baum-Welch algorithm used for training hidden Markov models is a special case of the EM algorithm . In this appendix we give a brief presentation of the EM algorithm.

Let X and Y be two sample spaces such that X is mapped into Y by a non-invertible function F. We call $x \in X$ the complete data and $y = F(x) \in Y$ the incomplete data. The information needed to reconstruct x from y is called the missing data. In each experiment we sample $x \in X$ and report $F(x) \in Y$. Assume that $f_X(x, \theta)$ is a family of densities on X depending on a parameter $\theta \in \Theta$. A densities family on Y is defined by $f_Y(y, \theta) = \int_{F^{-1}(y)} f_X(x, \theta) dx$. Given a random sample $y_1, ..., y_n$, the EM algorithm finds $\hat{\theta} = \arg\max f_Y(y_1, ..., y_n, \theta)$ using the fact that in some cases it is much more convenient to deal with $f_X(x, \theta)$ than with $f_Y(y, \theta)$. From the definition stated above we obtain :

$$f_X(x, \theta) = f_Y(y, \theta) f_{X|Y=y(x)}(x, \theta)$$

Hence,

$$\log f_Y(y, \theta) = \log f_X(x, \theta) - \log f_{X|Y=y(x)}(x, \theta)$$

By taking the conditional expectation given $Y = y$ according to parameter value θ_0 we obtain :

$$\begin{aligned} \log f_Y(y, \theta) &= E(\log f_X(x, \theta)|Y = y, \theta_0) - E(\log f_{X|Y=y(x)}(x, \theta)|Y = y, \theta_0) \\ &= Q(\theta, \theta_0) - H(\theta, \theta_0) \end{aligned}$$

Q is refereed as the auxiliary function of the EM algorithm. Applying Jensen's inequality we obtain that $H(\theta, \theta_0)$, considered as a function of θ, has a maximum point at $\theta = \theta_0$. Therefore $Q(\theta, \theta_0) > Q(\theta_0, \theta_0)$ implies $f_Y(y, \theta) > f_Y(y, \theta_0)$. This enables us to define an iterative algorithm for computing a maximum likelihood estimation. Each iteration is composed of the following two steps :

$$\begin{aligned} \underline{\text{E}}\text{xpectation} \quad &: \qquad Q(\theta, \theta_n) = E(\log f_X(x, \theta) | Y = y, \theta_n) \\ \underline{\text{M}}\text{aximization} \quad &: \qquad \theta_{n+1} = \arg\max_{\theta} Q(\theta, \theta_n) \end{aligned}$$

Chapter 3

Segmental Models

In this chapter we shall discuss some of the shortcomings of the HMM paradigm. We present the segmental modeling concept and explain how some of the difficulties in HMM modeling can be solved. In the remainder of the chapter we review the main segmental models related to our work.

3.1 Limitations of the HMM Paradigm

The standard left to right HMM provides a technique for modeling the acoustic feature vector sequence, that represents some speech utterance, by a piecewise stationary process. The model assumes the existence of states, in which the observations are locally independent and identically distributed (IID) within each one. Several HMM variants exist. The simplest variant employs a discrete output probability distribution function (PDF) to describe the acoustic feature vector at each HMM state. A refinement of the above is obtained by replacing the discrete PDF with a continuous PDF, which is usually a mixture of Gaussians with a diagonal covariance matrix. All these variants share the common assumption that the probability of an acoustic vector in a particular state is not dependent on the other vectors in that

state. This simplifying assumption assures computationally efficient algorithms for system training and recognition.

The IID assumption is reasonable for some of the HMM states (e.g., states that correspond to a steady state vowel in a user dependent system). Most states, however, clearly violate this assumption (e.g., states corresponding to vowel-consonant transition, diphthongs, etc.) and are in fact characterized by a highly correlated and non-stationary speech signal. The consequence is a reduced accuracy of speech acoustic modeling which in practical terms corresponds to a reduction in the recognition rate.

The traditional formulation of a single Gaussian HMM has been based on a piecewise constant fitting of the acoustic feature vector data sequence [53], [55]. This model assumes that the sequence of observation vectors within a state $x = x_1, ..., x_n$ is generated according to :

$$x_t = \mu + \epsilon_t \qquad t = 1, ..., n$$

where μ is a state dependent parameter that represents the mean vector and ϵ_t is an additive, zero mean, white noise vector (i.e., its covariance matrix is diagonal) with state dependent variances. In Figure 3.1 we present the sixth cepstral coefficient of the acoustic vector sequence of the word 'seven'. The smooth curve is obtained from empirical averaging of utterances of the word 'seven' taken from the database TIDIGITS. A nonlinear warping is applied to these utterances to synchronize them. The piecewise constant function is a five state HMM approximation.

The most common method of encoding the dependency between consecutive frames is to extend the feature vector to include the first and at times the second derivative of the static features. In Figure 3.2 we present the same data that was presented in Figure 3.1, for the first derivative of the sixth cepstral coefficient.

In a standard stationary-state HMM, the trajectories of the feature vectors are approximated by a piecewise constant function. Each region of constant value cor-

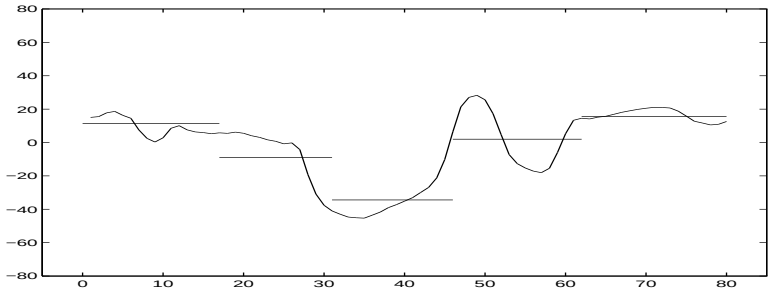

Figure 3.1: Sixth cepstral coefficient of the acoustic vector and its HMM approximation.

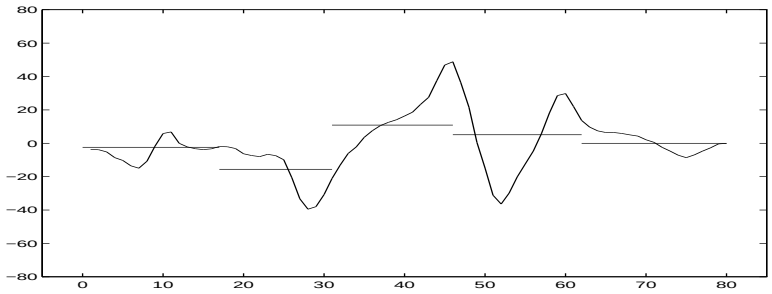

Figure 3.2: Time derivative of the sixth cepstral coefficient of the acoustic vector and its HMM approximation.

responds to an HMM state. As can be seen from Figure 3.1 and Figure 3.2, a piecewise constant function is usually a poor approximation to the mean trajectory. Another problem of this model arises from the fact that we try to model simultaneously a static feature vector (e.g. the cepstrum function) and its time derivative. To observe the disadvantage associated with this model, consider Figure 3.3, that presents a feature vector that consists of two components. The first component is a sine wave; the second is the time derivative of the first, hence a $\pi/2$ phase shifted sine wave. Figure 3.3 also presents the optimal partitioning into states of these components, and the resulting piecewise constant approximation. As can be seen, the state partitioning that is required for optimally approximating the trajectory of the first component is different from the one required for approximating the trajectory of the second. However, since the same state partitioning should be used for both components, this results in an approximation of reduced quality.

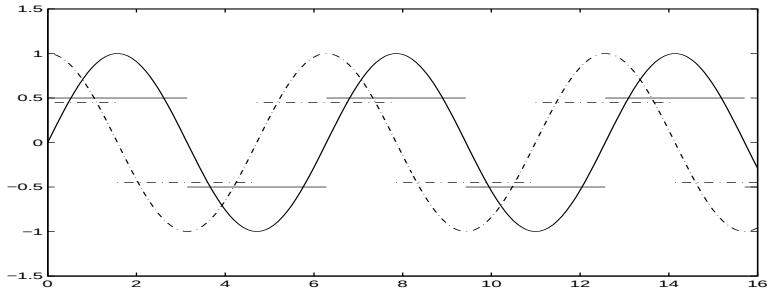

Figure 3.3: Piecewise constant approximation of sin and its derivative cos.

To gain further insight, consider a speaker independent, mixture of Gaussians, HMM system. The improved performance of this system, compared to a single Gaussian HMM is usually attributed to the fact that mixtures help to improve the modeling of the true state distribution which is clearly non-Gaussian, and in this way to improve the modeling of the variation between different speech styles.

We now suggest an alternative explanation for this phenomenon. Essentially, we assert that mixtures help to describe the non-stationary behavior of the feature vectors within a state. Assume that the trajectory of the mean in a state changes from a_0 at the beginning of the state to a_1 at the end. By setting the mean values of three mixture components to a_0, $(a_0 + a_1)/2$ and a_1 we may obtain improved modeling of the mean trajectory by segmenting the state into three parts. These mixture values can now be associated with the beginning, middle and final periods of this state respectively.

In Figure 3.4 we present the sixth cepstral coefficient of the acoustic feature vector, and its HMM parameters, for a word-based HMM with 5 states and 3 mixtures (the word shown is 'seven'). In Figure 3.5 we present the same data for the time derivative of the sixth cepstral coefficient. These figures demonstrate how the model employs these mixtures to track the trajectory of the mean. For example, consider the second state in Figure 3.4. The three mixture components refine the HMM approximation of the dynamic behavior of the mean trajectory.

To assess further the validity of the proposed explanation, we recorded the statistics of the transition between mixtures, using the sequence of most likely mixture

at each frame. Such a sequence is obtained using supervised Viterbi segmentation. The transition matrix for three mixtures at the second state of the word 'seven' is shown in Table 3.1. As can be seen, the IID within a state assumption is not valid in practice. In fact, there is a clear trend to choose the mixtures in a fixed order $(3 \rightarrow 2 \rightarrow 1)$. Similar data is shown in Table 3.2 except that a consecutive self transition is counted only once. This observation validates the explanation that we set above. Mixtures of Gaussians may thus be considered a tool for refining the HMM approximation but still there is no direct reference in this model to the continuous nature of the local dynamics within a state.

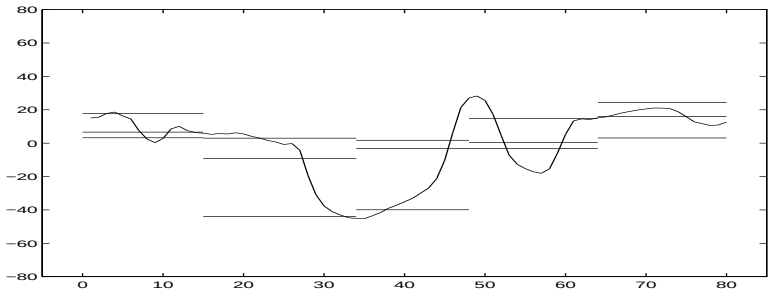

Figure 3.4: HMM with 5 states and 3 mixtures that models the sixth cepstral coefficient.

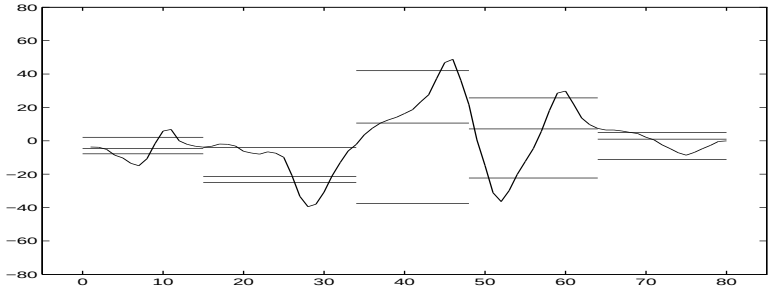

Figure 3.5: HMM with 5 states and 3 mixtures that models the derivative of the sixth cepstral coefficient.

3.2 The Concept of Segmental Modeling

The local IID assumption of standard HMM implies that once we have decided on the state boundaries, we model the data on a frame base level and ignore the continuous dynamic nature of the speech signal within the state. An alternative approach is segmental modeling, where the basic modeled unit is not a frame, but a segment. A segment is a variable duration part of the speech waveform that corresponds to a phonetic unit. A segment is commonly taken to correspond to a phone, but it may, however, correspond to a sub-phonetic event (one phone may be represented as a sequence of several segments). In the segmental approach the basic observation is not just a single frame from the sequence comprising the utterance, but rather a complete acoustic event. This enables us to model explicitly the correlation between successive frames, the dynamics along the segment and the statistical behavior of the segment duration. We can, therefore, reduce the portion of the data which is left unexplained in the standard HMM and considered as white noise.

Apart from the advantage of segmental modeling there are also some difficulties. The dimension of an acoustic unit is much larger than the dimension of a single frame. Therefore, the problems in parameter estimation become more serious. Another difficulty is that the length of a segment varies from phone to phone and among different realizations of the same phone. This is in contrast to HMM where we model each frame separately, and the feature vector is of a constant length. Another problem with a segmental model is the high complexity involved with computing the density of a given utterance. In HMM during the dynamic programming we have to

	mix1	mix2	mix3
mix1	394	0	0
mix2	202	582	6
mix3	2	216	1540

Table 3.1: Transition matrix of the three mixtures of the second state presented in figure 3.4.

	mix1	mix2	mix3
mix1	196	0	0
mix2	202	212	6
mix3	2	216	212

Table 3.2: Same as Table 3.1, consecutive self transitions is counted only once.

remember for each frame just the possible states associated with it. In a segmental model, however, every possible beginning and ending time should be considered for each segment model in the dynamic programming procedure. Therefore, with each possible beginning and ending time pair we must compute the segmental score. No recursion formula could be used even if there is only one frame difference.

The complexity problem is caused by the need to check all possible segmentations of a given utterance. A possible solution is to use an HMM system or any other simpler model to provide a set of sentence hypotheses that will be re-scored by the segmental model in a post-processing step. The sentence hypotheses can be described as a N-best list or a word lattice. In both cases the segmental model is not involved with the segmentation process. In updated large vocabulary recognition systems [51] [56], rescoring serves as a useful mechanism for combining additional knowledge sources (e.g. detailed linguistic model) to the acoustic model. The segmental model can be integrated into a speech recognition system as one of these knowledge sources. The scores from different sources are linearly combined, and using optimization procedure [39], we can find the optimized weight of the segmental model contribution.

3.3 Previous Segmental Models

Recently there have been a number of segment based approaches to the phonetic recognition problem. A comprehensive survey on segment models can be found in [49]. In this section we describe in some detail those segmental models that are

closely related to our work.

Stochastic Segment Model

Ostendorf and Roucos [50] suggested a general framework for segment models termed Stochastic Segment Model (SSM). They handled the variable length of an observed segment by assuming that the observation is a partially observed samples of a fixed length trajectory [57]. Another way to solve the variable length problem is by re-sampling the observation into a fixed length sequence using linear interpolation [49]. The grand vector which is the concatenation of the re-sampled frames, is modeled by a multivariate Gaussian distribution. The full covariance matrix allows explicit modeling of the time correlation. The general formulation of the SSM suffers from the estimation problem caused by the large number of parameters in the covariance matrix. To avoid this problem, a Gauss-Markov (GM) structure can be imposed on the covariance matrix of the SSM [4] [65] [41]. A Discrete observation Markov assumption was explored by Paliwal [52]. The GM model did not achieve significant improvement over the standard HMM [41]. A possible explanation for this is that in real situations of speech recognition there are large differences between speakers, dialects and recording equipment. This difference can be modeled as a noise signal added to the outputs of the Gauss-Markov process. Digalakis et al. [12] formulated this idea in the introduction of linear Dynamical System(DS) method into speech recognition. A fixed length observation sequence y associated with a phonetic state-segment s is generated according to the DS model as follows :

$$\begin{aligned} x_{t+1} &= F_t(s)x_t + w_t \\ y_{t+1} &= x_t + v_t \end{aligned} \tag{3.1}$$

The sequences w and v are realizations of uncorrelated Gaussian white noises with

model dependent variances. The hidden Gauss-Markov process is denoted by x. It is assumed that the initial state x_0 is Gaussian with model dependent mean and variance. $F_t(s)$ is a first order Markovian factor which depends on the segment s and the time index within the state t. The observation sequence y is obtained as a result of adding white noise to x. In order to reduce the number of free parameters the segment is divided into regions of equal length. It is assumed that the GM process is locally stationary within the region. This can be considered as a kind of parameter tying. The recognition is performed using a version of the Kalman predictor to compute the probability of the segment. Phoneme classification rate reported using a DS model was significantly better than the one obtained using independent frame HMM. A disadvantage of this model, however, is its high computational cost in both recognition and training. Some efficient yet suboptimal algorithms that reduce the number of segment evaluations during sentence recognition are described in [13].

Polynomial approximations

Deng et al. [7] dealt with the non-stationarity of the speech signal within a state by using a parametric function of time to model the mean trajectory in each state. Deng gave the name Trend HMM to this approach. In this model the observation sequence within a state $x = x_1, ..., x_n$ is generated according to :

$$x_t = \sum_k \mu_k t^k + \epsilon_t \quad t = 1, ..., n \tag{3.2}$$

where the first term is a state dependent polynomial function of t and the second term is the residual noise assumed to be the output of an IID zero-mean Gaussian source with state dependent covariance matrix. The time origin of the polynomial function in each state begins from the time the state is first entered. When the degree of the polynomial is zero, the model is reduced to the standard Gaussian

HMM.

In order to illustrate the strength of Deng's model, we compare the data fitting performance between the use of the trend HMM and the standard one. The data we used was a set of realizations of the phone 'l' in the triphone context aa-ih taken from the *Wall Street Journal* corpus. In Figure 3.6 we show the mean trajectory of the first cepstral coefficient and the relatively poor approximation of this trajectory with a 3 state HMM. In Figure 3.7 the same curve is approximated with a three state trended HMM, such that the regression polynomial is linear. From these figures it is evident that the trended HMM fits the data better than the standard HMM. Experimental results also show the superiority over standard HMM in recognition accuracy.

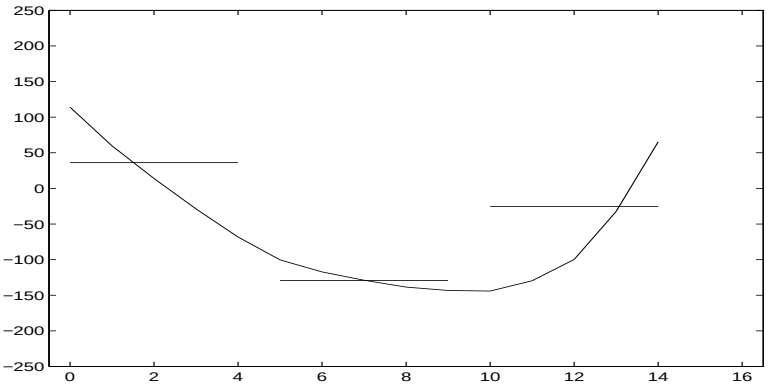

Figure 3.6: Modeling the mean trajectory of the phone 'l' with 3 states HMM.

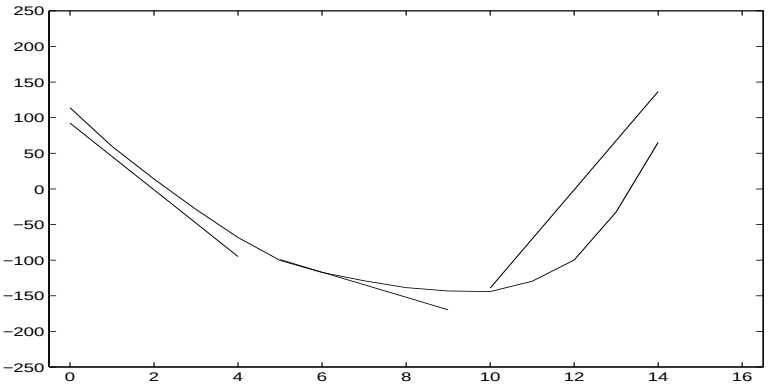

Figure 3.7: Modeling the mean trajectory of the phone 'l' with 3 states linear trended HMM.

When this model is implemented in the recognition step we can use Viterbi decoding to determine the division of the utterance into states according to the ML state sequence. This approach is computationally expensive. This is because during the dynamic programming we must remember not only the state associated with each frame, as we do in standard HMM, but also the time index of this frame. Another method is to implement this model as a post-processor. We can use this model to rescore the result obtained by a standard HMM system. Training of the model can be performed using an extension of the segmental k-means algorithm [37]. The algorithm is composed of two iterative steps. The first one is a segmentation step where we use Viterbi decoding in a way similar to the decoding algorithm we use during the recognition. After applying the Viterbi decoding we can split each utterance into separate segments. The next step of the segmental k-means algorithm is the maximization step. Given a list of segment realizations the problem of finding the polynomial coefficients is a linear one and can be solved using the least-squares method [8].

Deng and Rathinavelu [8] combined the models presented in the two last sections. They extended the polynomial model to the Gauss-Markov case. The model can be written as :

$$x_t = \sum_{k=1} a_k x_{t-k} + \sum_{l=0} b_l t^l + \epsilon_t$$

They reported that this model produces higher recognition accuracy than the simpler models from which it is built.

A model similar to Deng's was suggested by Gish and Ng [20] for keywords spotting task. They also consider a generalization where we allow a mixture of parametric trajectories to describe the variation among different realization of the modeled segment. The training of this model is slightly more complicated. We still use the Viterbi decoding to find the states alignment of the training data. Given the states alignments, the ML estimate of the model parameters can not be computed directly as was done in Deng model. Instead, we must use the EM algorithm as it happens in other mixture situations. A Parametric model that describes the

mean trajectory using an exponential function rather than a polynomial one, was suggested by Deng [10]. This model has the advantage of smooth movement between consecutive phoneme trajectories.

Non-Parametric Models

Another attempt to model explicitly the observations dependence on the time index was made by Ghitza and Sondhi [19]. They suggested that instead of imposing an a-priori constraint of parametric structure on the mean trajectory, we may choose a typical segment realization to capture the nonlinear nature of the dynamics within the segment. In this framework a phonetic unit is modeled as one HMM non-stationary state to which a template is associated. Non-parametric representation enables us to perform a non-linear warping for optimized matching between the model and the observation sequence. This can not be done if we use a polynomial regression function to represent the mean trajectory. The typical template is chosen from the ensemble of segments found in the training data-base for a selected phonetic unit. Ghitza and Sondhi defined a distance between sequences as the one used in the DTW method [60] in word recognition (the frame level distance is a Euclidean one, such that the vectors components are normalized to have an equal variance). The selected template is defined as the observation sequence whose cumulative distance from all other sequences in the ensemble is minimal. Once the template has been derived, each segment in the training set can be warped against it and the covariance matrix is estimated from the aligned segments. Computation of the probability of an unknown test segment is performed by dynamic programming. In this procedure, the test segment is non-linearly warped to the length of the template and then a Gaussian likelihood score is computed.

Another non-parametric approach was suggested by Goldenthal and Glass [24]. They defined the notion of a "track" which is a synthetic template prototype. The

tracks are computed from the training data by mapping the training segments of each phone to a sequence of fixed length. The mapping is done by linear interpolation. The track is obtained by averaging the fixed length segments. Once the tracks have been created they serve as the initial stage in evaluating hypothesized speech segments. To evaluate an unknown segment of length n, a synthetic segment of length n is generated from the track using a deterministic linear interpolation. Goldenthal and Glass proposed the following method to overcome the problem of high dimensionality of the full covariance matrix. The likelihood score is computed for the error vector obtained from the observations and the synthetic template. Their solution involved dividing the error sequence into sub-segments of equal duration and averaging the vectors within each sub-segment. The small sized averaged vector is modeled with a Gaussian distribution having a full covariance matrix.

Segmental Mixture Model

One of the main reasons for the undoubtable success of HMM in user independent large vocabulary recognition tasks is the use of mixtures of Gaussians to describe the distribution of a frame within a segment. The addition of the mixture component to the HMM formulation made it possible to present alternative acoustic models in parallel for the same phonetic unit. In this manner we can overcome the modeling problem caused by different realization of the same linguistic part. Much of recent progress in HMM research is related to questions of better estimation of the parameters of the mixture HMM. Given an observation sequence $x = x_1, ..., x_n$ and a mixture distribution consisting of k Gaussian models $m_1, ..., m_k$, the mixture model can be written as :

$$f(x) = \prod_{t=1}^{n} \sum_{i=1}^{k} c_i f(x_t|m_i)$$

where c_i are the mixture coefficients.

The segmental analogue of the mixture concept is the assumption that the step of choosing one of the models is not taken independently for each frame, but only once in a state transition. The segmental mixture model can be written as :

$$f(x) = \sum_{i=1}^{k} c_i f(x|m_i) = \sum_{i=1}^{k} c_i \prod_{t=1}^{n} f(x_t|m_i)$$

where c_i are the segmental mixture coefficients. The speech waveform is produced by a continuous physical system, hence there is no physical justification to select a different model for each new coming frame, as is done in frame level models. In contrast, frame level mixture can serve as a mathematical tool to approximate the real frame distribution which is certainly not Gaussian and not stationary. Kimball [42] showed that training and recognition algorithms of frame level mixture HMM can be generalized to the segment case. Especially the EM algorithm can be used for training the segmental mixture model. Kimball also reported that the segmental mixture model achieves a better recognition rate than the frame level mixture model in context-independent phone classification tasks.

The concept of mixture of distributions can be used as an extension of any segmental model. Instead of having only one segmental model to describe the acoustic behavior in a phonetic unit, we can have an ensemble of such distributions which are combined with mixtures weights. In this manner, the high variability in the speech waveforms can be handled by using a number of acoustic models for the same phonetic unit. For example, trended HMM, which consists of polynomial regression function to describe the mean trajectory, is generalized by Gish and Ng [20] as model which has a mixture of these functions. Given a training procedure for a segmental model, there is a general method for training the mixture version of this model. The training can be done using the EM algorithm. In the E-step, for each segment realization in the training set we compute the a-posteriori probability of each of the models in the mixture distribution. In the M-step we can use the training

procedure of the original segmental model to solve the optimization problem.

Another approach to segmental mixture modeling is to assume a prior distribution on the model parameters. For example, consider a Gaussian HMM such that the mean is not a fixed parameter. Instead, it is a Gaussian random variable which is sampled once in a state transition. This model was suggested by Russell [58] and it belongs to the family of random segmental models. Random segmental models will be discussed in detail in the next chapter. It should be noted that on the frame level there is no meaning for random Gaussian HMM. If we would be allowed to sample the mean independently for each frame, as we are in discrete mixture HMM, we would obtain a model consisting of a sum of independent Gaussian random variables which is still Gaussian. Therefore, the random Gaussian HMM is equivalent to the deterministic one.

Segmental Neural Network

We conclude this review on segmental models with a short exposition of how segmental ideas are combined with neural networks. In recent years, Neural Networks (NN) have been used for tasks of speech recognition. Several attempts have been made to build a hybrid HMM/NN system that would take advantage of both HMM and NN abilities. HMM systems are better in efficiently handling the global dynamic programming to find the best sentence transcription. In contrast, neural networks are well suited for local acoustic scoring, as they can accept continuous valued inputs without making any assumption as to the parametric shape of the density function.

A Segmental Neural Net (SNN) is a neural network that attempts to recognize a complete phonetic unit rather than a sequence of conditionally independent frames. This allows the network to take advantage of the global acoustic structure of the phone and also makes it easier to incorporate segmental information such as dura-

tion. There are two main problems with implementing segmental neural nets. First, the speech must be segmented before the neural net can evaluate the segment. The second problem is that the segment length varies with different segment realizations. However, the topology of the NN forces the input to have a fixed length representation.

Zavaliagkos et al. [66] [1] implemented SNN for continuous speech recognition. They developed a hybrid HMM/NN system that uses the HMM to find the N-best list of sentence hypotheses. The SNN is used to rescore the hypotheses and the scores of the systems are combined. In order to overcome the variable length problem, they converted each segment to a fixed number of frames using time re-sampling in a manner similar to how it is done in stochastic segmental models [50].

Another segmental approach was suggested by Konig and Morgan [43] [44]. They proposed a segmental phone classification where the input to the NN include a time index that describes the relative location of the frame in the phonetic unit in addition to the acoustic vector. In this manner the NN incorporates a non-parametric representation for the trajectory of the acoustic signal associated with the phonetic unit.

Chapter 4

Scaled Random Trajectory Segmental Models

In this chapter we present the concept of random trajectory segmental modeling. We review specific random models that were proposed in the past and suggest an alternative training algorithm for those models. We discuss the modeling problem created by the fact that segment realizations of the same phone differ in length and suggest an improved random segmental model that solves this problem. We conclude with experimental results that demonstrate the difference in performance among the models that are presented in this chapter.

4.1 Introduction

Over the past decade a number of studies have proposed a framework of stochastic description of the mean trajectory, as an alternative to the multi description of the mean trajectory, that is provided by an HMM whose state distributions are mixtures of Gaussians.

This concept of random trajectory segmental modeling (RTSM) was first suggested by Russell [58] who named this approach 'segmental HMM'. We prefer to use the more specific term RTSM because it better reflects the particular characteristics of the model that distinguish it from other segmental models. Random segmental modeling can be thought of a generalization of the Gaussian HMM formalism. The main difference is that the mean trajectory of the acoustic feature vector in a state is not a fixed parameter. Instead, it is a random variable sampled once for each state transition. The acoustic motivation for this framework is that we wish to separately model two distinct types of variability: long term variations, such as speaker identity, and short term variations which occur within a given state as a result of random fluctuations. The long term variability is modeled by a probability density function (PDF) used to select the sampled mean. The short term variability within a state is modeled by the deviation of the feature vectors from the sampled mean. In standard HMM these two effects are modeled implicitly by a single PDF.

We first provide a formal description of the random trajectory segmental modeling approach. Let $f_s(\gamma)$ be a PDF defined on some family of valid trajectories, γ, at a given state, s. On arrival at state s, a trajectory is chosen according to this PDF. Once γ is determined, we can model the within-segment variation at each frame independently. Denote by $f_s(x_t|\gamma, t)$ the PDF of the frame x_t given the chosen mean trajectory and the time index t. The PDF of the segment data realization $x = (x_0, ..., x_{n-1})$ is given by

$$P_s(x) = \int_\gamma f_s(\gamma) \prod_t f_s(x_t|\gamma, t) d\gamma$$

According to Russell's terminology [58], $f_s(\gamma)$ accounts for extra-segmental variation which would lead to different trajectories for the same phonetic unit, while $f(x_t|\gamma, t)$ hopefully accounts for much smaller intra-segmental variations in the realization of a particular trajectory.

4.2 Static Random Segmental Model

In this section we concentrate on the case where the trajectories PDF is Gaussian and the mean trajectory is constant over time. This static RTSM was originally presented by Russell [58]. More precisely, a static RTSM assumes that the observations within a state $x = (x_0, ..., x_{n-1})$ are generated according to

$$x_t = \mu + a + \epsilon_t \qquad t = 0, ..., n-1 \tag{4.1}$$

where μ is a fixed parameter, associated with the state, that describes the grand mean trajectory. The random variable a is a shift of the mean trajectory that is global to the entire segment realization. It is assumed that $a \sim N(0, \sigma_a^2)$ (i.e., a is a Gaussian random variable with mean 0 and variance σ_a^2). The short term variability is represented by ϵ_t, which is a zero mean Gaussian random variable with state dependent variance, $\epsilon_t \sim N(0, \sigma^2)$. To simplify notation, it will be assumed that the observations are one dimensional. Generalization to the multi-dimensional case is straight-forward. The PDF of the segment data realization, x, is given by

$$f(x) = \int_a f(x, a) da \tag{4.2}$$

$$f(x, a) = f(a) \prod_t f(x_t|a) \tag{4.3}$$

where

$$f(a) = \frac{1}{\sqrt{2\pi}\sigma_a} e^{-\frac{a^2}{2\sigma_a^2}} \qquad f(x_t|a) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2\sigma^2}(x_t - \mu - a)^2} \tag{4.4}$$

In this presentation of the static RTSM, the probability expressions are formulated in terms of the shift of the trajectory from the model mean, rather than the location of the trajectory itself as in Russell [58], and Gales and Young [17]. We are adopting this convention because it explicitly reflects the linear behavior of the model

(equation 4.1).

Two methods for computing the likelihood score have been proposed. The first one [58] is a maximum a posteriori (MAP) approach. The MAP method uses $f(x, \hat{a})$ instead of $f(x)$ as the target function, where $\hat{a}$ is the shift which maximizes the joint PDF $f(x, a)$. That is to say, segment identification is made based on the segment hypothesis for which $f(x, \hat{a})$ is maximized. A closed form for $\hat{a}$ is obtained by setting the derivative, $\partial \log f(x, a)/\partial a$ equal to zero, thus yielding,

$$-\frac{a}{\sigma_a^2} + \frac{1}{\sigma^2}\sum_{t=0}^{n-1}(x_t - \mu - a) = 0$$

Hence,

$$\hat{a} = \arg\max_a f(x, a) = \frac{\frac{1}{\sigma^2}\sum_{t=0}^{n-1}(x_t - \mu)}{\frac{1}{\sigma_a^2} + \frac{n}{\sigma^2}} \tag{4.5}$$

We use the following notation

$$E_x = \frac{1}{n}\sum_{t=0}^{n-1} x_t \quad , \quad V_x = \frac{1}{n}\sum_{t=0}^{n-1} x_t^2 - E_x^2 \quad , \quad V_n = \left(\frac{1}{\sigma_a^2} + \frac{n}{\sigma^2}\right) \tag{4.6}$$

Using this notation, we have,

$$\hat{a} = \frac{n}{\sigma^2 V_n}(E_x - \mu) \tag{4.7}$$

The last expression will be used to help evaluate $f(x)$.

Following Gales and Young [17], we now derive a closed form expression for $f(x)$. By (4.3) and (4.4),

$$f(x, a) = \frac{1}{\sqrt{2\pi}\sigma_a}\left(\frac{1}{\sqrt{2\pi}\sigma}\right)^n e^{-\frac{1}{2}g(x,a)} \tag{4.8}$$

where

$$\begin{aligned} g(x,a) &= \frac{a^2}{\sigma_a^2} + \frac{1}{\sigma^2}\sum_t (x_t - \mu - a)^2 \\ &= \left(\frac{1}{\sigma_a^2} + \frac{n}{\sigma^2}\right) a^2 - 2a\frac{1}{\sigma^2}\sum (x_t - \mu) + \frac{1}{\sigma^2}\sum (x_t - \mu)^2 \\ &= V_n a^2 - 2a\frac{n}{\sigma^2}(E_x - \mu) + \frac{1}{\sigma^2}\sum_t (x_t - \mu)^2 \\ &= V_n(a - \frac{n}{\sigma^2 V_n}(E_x - \mu))^2 + \frac{1}{\sigma^2}\sum_t (x_t - \mu)^2 - \frac{n^2}{\sigma^4 V_n}(E_x - \mu)^2 \\ &= V_n(a - \hat{a})^2 + \frac{n}{\sigma^2}\left(V_x + \frac{1}{\sigma_a^2 V_n}(E_x - \mu)^2\right) \end{aligned}$$

Hence,

$$g(x,a) = \left(\frac{1}{\sigma_a^2} + \frac{n}{\sigma^2}\right)(a - \hat{a})^2 + g(x,\hat{a}) \tag{4.9}$$

$$g(x,\hat{a}) = \frac{n}{\sigma^2}\left(V_x + \frac{\sigma^2}{\sigma^2 + n\sigma_a^2}(E_x - \mu)^2\right) \tag{4.10}$$

The conditional distribution of the shift a given the segment data x, is therefore Gaussian with mean and variance values given by,

$$E(a|x) = \hat{a} \quad , \quad Var(a|x) = V_n^{-1} \tag{4.11}$$

Substituting (4.8) and (4.9), (4.10) in (4.2) and carrying out the integration operation results in

$$f(x) = \left(\frac{\sigma^2}{\sigma^2 + n\sigma_a^2}\right)^{\frac{1}{2}} \left(\frac{1}{\sqrt{2\pi}\sigma}\right)^n e^{-\frac{1}{2}g(x,\hat{a})} \tag{4.12}$$

Comparison of (4.8) and (4.12) reveals the following relationship

$$f(x) = \left(\frac{1}{\sigma_a^2} + \frac{n}{\sigma^2}\right)^{-\frac{1}{2}} \sqrt{2\pi}\, f(x,\hat{a}) \tag{4.13}$$

As can be seen from (4.13), the likelihood score provided by the approximated MAP method is identical to the true likelihood, except for a term which depends on the length of the segment and not on the segment data.

When $n\sigma_a^2 \ll \sigma^2$, we have $\sigma_a^2 V_n \approx 1$, so that $f(x)$ is reduced to

$$f(x) \approx \left(\frac{1}{\sqrt{2\pi}\sigma}\right)^n \exp\left\{-\frac{1}{2\sigma^2}\sum_t (x_t - \mu)^2\right\}$$

Hence, in that case, the RTSM degenerates to the deterministic model $x_t \sim N(\mu, \sigma^2)$. In the deterministic model, the PDF assigns equal weight to the empirical variance of the samples and to the distance of the samples empirical average from the grand mean. On the other hand, from (4.10) it can be seen that the RTSM assigns larger weight to the empirical variance.

We now discuss the problem of parameter estimation for the static RTSM. We discuss two general estimation schemes. The first is a generalized Baum-Welch scheme which is a special case of the expectation - maximization (EM) algorithm [5]. Each iteration is composed of two steps. In the first step (E-step), the conditional expectation of the likelihood of some 'complete data', given the observed data is evaluated. In the second step (M-step), an updated parameters set is obtained, such that the conditional expectation at the new set attains its maximal value. The resulting Baum-Welch algorithm consists of an objective function, in which each state sequence realization, that is consistent with the given measurements, is weighted by its probability of occurrence. At each iteration, the algorithm attempts to bring the objective function to a maximum. The other estimation scheme is an approximation of the Baum-Welch method, based on an extension of the segmental k-means algorithm [37]. Each iteration is composed of two steps. In the first stage, a Viterbi decoding algorithm is applied to obtain the most likely state sequence (using the current values of the estimated parameters). Once the segments boundaries are determined, in the second step of the iterative algorithm, the segmental model parameters are re-estimated. Following Ostendorf, Digalakis and Kimball [49], we have

chosen to adopt segmental k-means notation in order to simplify the presentation. We note however, that all the algorithms that will be presented in the sequel can be plugged into the re-estimation step of the Baum-Welch scheme. Note also that the various segmental k-means algorithms that will be presented, iteratively derive optimal unit segmentation as a by-product of the algorithm. Since the Viterbi stage of the segmental k-means algorithm is standard, we only discuss the re-estimation stage (the second phase of each iteration).

Let $x = (x_1, ..., x_k)$ be k segment realizations associated with the state s. Denote the length of the sequence x_i by n_i. The frames of the segment x_i are denoted by $x_{i,0}, \ldots, x_{i,n_i-1}$. We wish to obtain the ML estimates of the model parameters $\mu, \sigma_a^2, \sigma^2$. Setting the derivative of $\log f(x)$ in (4.12), with respect to μ to zero, we have,

$$\frac{\partial \log f(x)}{\partial \mu} = \sum_{i=1}^{k} -\frac{n_i}{2\sigma^2\sigma_a^2 V_{n_i}} \frac{\partial}{\partial \mu}(E_{x_i} - \mu)^2 = \sum_{i=1}^{k} \frac{n_i}{\sigma^2 + n_i\sigma_a^2}(E_{x_i} - \mu) = 0$$

Hence,

$$\hat{\mu} = \frac{\sum_{i=1}^{k} \frac{1}{\sigma^2 + n_i\sigma_a^2} \sum_{t=0}^{n_i-1} x_{i,t}}{\sum_{i=1}^{k} \frac{n_i}{\sigma^2 + n_i\sigma_a^2}}$$

When all segments have the same length, denoted by n, the expression above is reduced to

$$\hat{\mu} = \frac{1}{nk} \sum_{i=1}^{k} \sum_{t=0}^{n-1} x_{i,t} \tag{4.14}$$

In this case we can also obtain the following closed form expressions for $\hat{\sigma}^2$ and $\hat{\sigma}_a^2$:

$$\hat{\sigma}^2 = \frac{n}{(n-1)k} \sum_{i=1}^{k} V_{x_i} \tag{4.15}$$

$$\hat{\sigma}_a^2 = \frac{1}{k} \sum_{i=1}^{k} (E_{x_i} - \hat{\mu})^2 - \frac{\hat{\sigma}^2}{n} \tag{4.16}$$

However, in the general case, where the segment realizations differ in length, we

cannot obtain a closed form expression for the ML estimators of σ_a^2 and σ^2, unless some approximation, such as $n_i\sigma_a^2 \gg \sigma^2$ is used [17].

Russell [58] used the joint probability of the observations and the optimal trajectory as the target function for the maximization problem. Setting the first partial derivatives of $f(x, \hat{a})$ with respect to the estimated parameters to zero yields,

$$\begin{aligned}
\hat{\mu} &= \frac{\sum_i \frac{1}{\sigma^2+n_i\sigma_a^2} \sum_t x_{i,t}}{\sum_i \frac{n_i}{\sigma^2+n_i\sigma_a^2}} \\
\hat{\sigma}_a^2 &= \frac{1}{k}\sum_i (\hat{a}_i)^2 \\
\hat{\sigma}^2 &= \frac{1}{\sum_i n_i}\sum_{i,t}(x_{i,t} - \hat{a}_i - \mu)^2
\end{aligned}$$

Note that $\hat{a}$, which was obtained in (4.5), is a function of the unknown parameters. Hence, the unknown parameters appear on both sides of each equation. Russell suggested a heuristic iterative solution, where the parameters estimated in the previous iteration of the segmental k-means algorithm are substituted back in the right-hand sides of the equations.

Digalakis, Rohlicek and Ostendorf [12] considered the static RTSM (which they referred to as a 'target state' segment model) as a special case of the dynamical system model. They suggested to utilize the EM algorithm as an iterative procedure for solving the maximization problem required in the second stage of the segmental k-means algorithm. The hidden values of the random shifts, which are sampled for each segment realization, are the missing data of the EM. A detailed derivation of the EM algorithm for a more general RTSM can be found in Appendix 4.B. Note that if a Baum-Welch algorithm is used, instead of segmental k-means, then the resulting algorithm would involve two levels of EM. In that case, the algorithm in Appendix 4.B, would be in the inner level of the iterations.

4.3 Scaled Random Trajectory Segment Models

Holmes and Russell [29] have pointed out that there is a balancing problem between the extra- and the intra-segmental components of the RTSM. Different explanations of an utterance using different number of segments will use different number of extra-segmental probabilities. Therefore, interpretations of the data which involve a large number of short segments require more probability terms than ones which use a small number of long segments. This phenomenon, which does not exist in the standard HMM formalism, is caused by the random segmental element of the RTSM. Holmes and Russell [29] observed that including self loop transitions in the segment model improves recognition performance. Self loops allow freedom in representing each occurrence of a basic phonetic unit using an optimal number of segments. In this manner the two model components can be automatically balanced. Their preferred solution is, however, to model the intra-segmental variability more accurately. They have found that using a Richter instead of a Gaussian distribution can greatly improve the performance. In this section we present yet another solution to this balancing problem.

The RTSM can be analyzed from another point of view. As was noted in the previous section, unlike standard HMM, RTSMs assign different weights in the PDF to the empirical variance of the samples and to the distance of the empirical mean of the samples from the grand mean. Recalling (4.12) and (4.10), this weighting ratio is given by

$$1 + \frac{\sigma_a^2}{\sigma^2} n$$

This ratio reflects the relative contribution of the extra- and the intra-segmental components in the likelihood function. Note that this ratio depends on the segment duration, n. Therefore, balancing problems can exist even in cases of interpretations of the data which involve the same number of segments but with different segment lengths.

We now suggest a modification of the static RTSM which aims to solve this

balancing problem. In this model the ratio between the empirical variance of the samples and the distance of the empirical mean of the samples from the grand mean is independent of the segment duration. This model, which we have termed scaled RTSM [23], is similar to Russell's model [58] that was presented in the previous section, except that

$$a \sim N\left(0, \frac{\sigma_a^2}{n}\right)$$

where n is the segment length. The scaled RTSM asserts that the variance is inversely proportional to the segment length. To assess this assumption, triphone realizations, that were extracted from the *Wall Street Journal* data base, using the SRI DECIPHER system [14], were considered. The various segment realizations were clustered into groups based on duration, such that all elements within a group had the same duration. As was mentioned in the previous section, in cases where all the segments have the same length, there is no need for using the methods we have described. Instead, the closed form ML solution (4.14) - (4.16) can be used. Hence, we were able to estimate the variance of the random trajectory σ_a^2, and the variance of the samples given the trajectory σ^2, separately for each group. Therefore, there was no pre-imposed assumption on the dependence of the parameters on the segment length. This experiment enables us to obtain an empirical dependence of σ_a^2 on the duration, n. We note that the number of segment realizations for each segment was large enough to obtain a reliable estimate to σ_a^2.

Figure 4.1 shows the inverse of the variance of the sampled trajectory σ_a^2, as a function of segment duration, for the first seven mel-cepstrum features, based on 750 realizations of the phoneme 't' in the triphone context 's-t-ih'. It can be seen from Figure 4.1 that the variance of the sampled trajectory σ_a^2 is inversely proportional to the segment length.

The PDF of an observation segment x in the scaled model is obtained by sub-

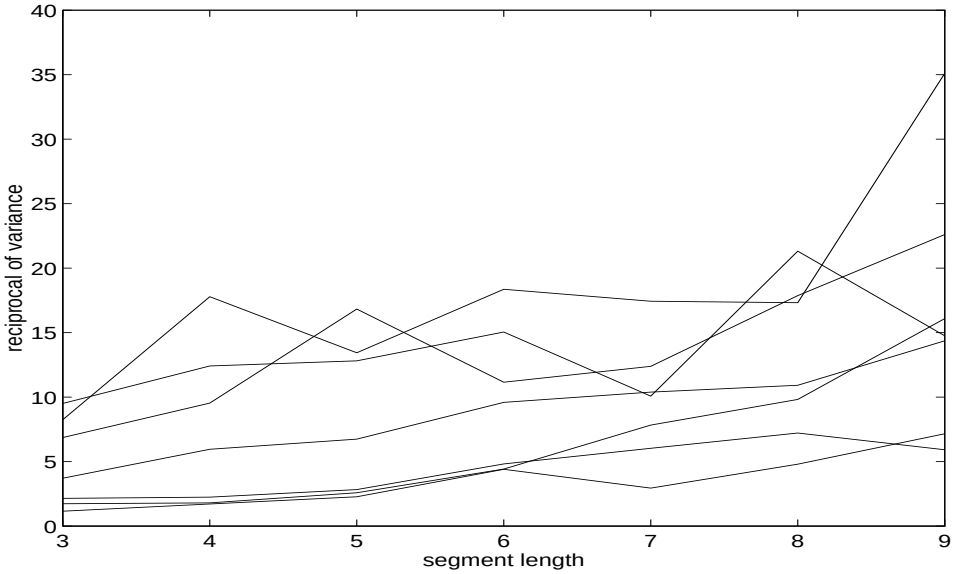

Figure 4.1: The inverse of the variances of the sampled trajectory as a function of the segment length.

stituting $\frac{\sigma_a^2}{n}$ in place of σ_a^2 in (4.12), thus yielding

$$f(x) = \left(\frac{\sigma^2}{\sigma_a^2 + \sigma^2}\right)^{\frac{1}{2}} \left(\frac{1}{\sqrt{2\pi}\sigma}\right)^n \exp\left\{-\frac{n}{2\sigma^2}\left(V_x + \frac{\sigma^2}{\sigma_a^2 + \sigma^2}(E_x - \mu)^2\right)\right\} \qquad (4.17)$$

We now discuss the parameter estimation problem. A distinct advantage of the scaled model is that we can solve the likelihood equations analytically and do not need to use an iterative algorithm (e.g. EM) for that purpose. Let $x = (x_1, ..., x_k)$ be k segment realizations associated with the state s. Denote the length of the sequence x_i by n_i. To obtain the ML estimators for the model parameters, we set the partial derivatives of $f(x)$, (4.17) to zero as follows.

$$\frac{\partial \log f(x)}{\partial \mu} = \frac{1}{\sigma_a^2 + \sigma^2} \sum_i n_i (E_{x_i} - \mu) = 0$$

$$\frac{\partial \log f(x)}{\partial \sigma_a^2} = -\frac{1}{2(\sigma_a^2+\sigma^2)} \sum_i \left(1 - \frac{1}{\sigma_a^2+\sigma^2} n_i (E_{x_i} - \mu)^2\right) = 0$$

Hence,

$$\sigma_a^2 + \sigma^2 = \frac{1}{k} \sum_i n_i (E_{x_i} - \mu)^2 \tag{4.18}$$

$$\begin{aligned} \frac{\partial \log f(x)}{\partial \sigma^2} &= -\frac{1}{2\sigma^2} \sum_i (n_i - 1) + \frac{1}{2\sigma^4} \sum_i n_i V_{x_i} - \\ &\quad \frac{1}{2(\sigma_a^2+\sigma^2)} \left(k - \frac{1}{\sigma_a^2+\sigma^2} \sum_i n_i (E_{x_i} - \mu)^2\right) \end{aligned} \tag{4.19}$$

Substituting (4.18) in (4.19) yields closed form solutions to the likelihood equations :

$$\begin{aligned} \hat{\mu} &= \frac{\sum_{i,t} x_{i,t}}{\sum_i n_i} \\ \hat{\sigma}^2 &= \frac{\sum_i n_i V_{x_i}}{\sum_i (n_i - 1)} \\ \hat{\sigma}_a^2 &= \frac{1}{k} \sum_i n_i (E_{x_i} - \hat{\mu})^2 \quad - \quad \hat{\sigma}^2 \end{aligned}$$

It should be noted that the ML estimator for σ_a^2 can be negative. It can be seen from the density expression (4.17) that the actual parameter is not σ_a^2 but $\sigma^2 + \sigma_a^2$, and the ML estimation of $\sigma^2 + \sigma_a^2$ is always non-negative. Negative values for the estimator of σ_a^2 do occur in real situations. Later in this chapter we describe experiments that have been conducted with real speech data. In these experiments we trained static random models. In some experiments the estimated variance was found to be negative. This fact does not coincide with the way we presented the static RTSM. The model was stated as a two-step experiment where we first sample a random mean trajectory and then add a Gaussian white noise to the sampled trajectory. The normal distribution associated with the random mean trajectory

has, of course, a non-negative variance. In order to understand what the meaning of negative variance is, we must consider another interpretation for the random model. The static RTSM assumes that the segment $x = x_0, ..., x_{n-1}$ has a non-diagonal multi-normal distribution, such that the covariance matrix has a certain structure. We have defined the scaled static RTSM as :

$$x_t = \mu + a + \epsilon_t \qquad a \sim N(0, \frac{\sigma_a^2}{n}) \quad , \quad \epsilon_t \sim N(0, \sigma^2)$$

The independence assumption between a and ϵ implies that the first two moments of the distribution of x are :

$$E(x) = \mu \cdot 1 \quad , \quad V(x) = \sigma^2 \cdot I + \frac{\sigma_a^2}{n} 1 \cdot 1^T$$

where 1 is an all ones column vector. Therefore, the scaled random model is no more than the assumption that the observation sequence x has the following distribution :

$$x \sim N\left(\mu \cdot 1 \ , \ \sigma^2 \cdot I + \frac{\sigma_a^2}{n} 1 \cdot 1^T\right)$$

The determinant of the covariance matrix is $(\sigma^2)^n \left(\frac{\sigma_a^2 + \sigma^2}{\sigma^2}\right)$. The matrix is positive definite if $\sigma^2 > 0$ and $\sigma_a^2 + \sigma^2 > 0$. Hence, σ_a^2 can be negative. The condition $\sigma_a^2 > -\sigma^2$ ensures that the matrix $\sigma^2 \cdot I + \frac{\sigma_a^2}{n} 1 \cdot 1^T$ is a valid covariance matrix. The two-step model serves, when σ_a^2 has positive value, as an intuitive interpretation for this distribution.

For purpose of comparison we rewrite the re-estimation formulas of the mean in the scaled and non-scaled RTSMs :

$$\text{non-scaled model :} \quad \hat{\mu} = \frac{\sum_i \frac{1}{\sigma^2 + n_i \sigma_a^2} \sum_t x_{i,t}}{\sum_i \frac{n_i}{\sigma^2 + n_i \sigma_a^2}}$$

$$\text{scaled model}: \quad \hat{\mu} = \frac{\sum_{i,t} x_{i,t}}{\sum n_i}$$

As can be seen, the re-estimation equation of the non-scaled model assigns smaller weight to frames that correspond to segments with longer duration. On the other hand, the scaled RTSM assigns equal weight to each frame, independently of the duration of the segment that corresponds to that frame. Hence, the re-estimation equation of the scaled model coincides with our intuition that each data sample encapsulates the same amount of information about the mean trajectory.

The scaled model also possesses a computational advantage over the non-scaled model. In order to compute the likelihood of a given utterance, the log-PDF values of the segments in that utterance need to be summed up. Now, (4.12) shows that in the non-scaled model, it is required to compute the logarithm of the term $\sigma^2/(\sigma^2 + n\sigma_a^2)$, which depends on n. On the other hand, in the scaled model, the corresponding term, $\sigma^2/(\sigma^2 + \sigma_a^2)$, is independent on n, and may therefore be computed in advance. By assuming a plausible range of segment durations, the duration-dependent variance terms obtained in the unscaled model can be still computed in advance, but there are more terms to compute and therefore also more to store.

4.4 Scaled Linear Random Trajectory Segmental Models

The assumption that the mean trajectory within a state is constant over time, is shared both by the Gaussian HMM and by the static RTSM. In practice, most states violate this assumption. A simple parametric extension of static models is obtained by representing the mean trajectory as a linear function of time.

Deng et al. [7] proposed a segment model which generalized the standard Gaussian HMM. In their model the mean trajectory is a deterministic linear function

of time. In this linear HMM an observation sequence within a state is generated according to :

$$x_t = \mu_a + \mu_b(\frac{t}{n-1} - \frac{1}{2}) + \epsilon_t \qquad t = 0, ..., n-1$$

such that the time index t is initialized to zero at the beginning of the state and then incremented with each new incoming data frame. The linear trajectory is represented here via the line mid point μ_a and the slope μ_b which are state dependent parameters.

Deng and Aksmanovic [9] extended this linear model by allowing a discrete mixture of linear functions. Holmes and Russell [28] presented a continuous stochastic variant of a linear HMM. In their model, the linear mean trajectory is a random variable which is sampled on each arrival at the state. The long term variation in this model is represented by an ensemble of linear mean trajectories. The short term variation is considered to be a result of random fluctuation as it is in the static case. In this model the segmented data is generated according to :

$$x_t = \mu_a + a + (\mu_b + b)(\frac{t}{n-1} - \frac{1}{2}) + \epsilon_t \qquad t = 0, ..., n-1$$

where $x_0, ..., x_{n-1}$ is the observation sequence, μ_a and μ_b are fixed parameters , a and b are independent normal random variables :

$$a \sim N(0, \sigma_a^2) \quad , \quad b \sim N(0, \sigma_b^2)$$

and ϵ_t is a Gaussian white noise term, $\epsilon_t \sim N(0, \sigma^2)$. The line $\mu_a + \mu_b(\frac{t}{n-1} - \frac{1}{2})$ is the average trajectory over all segment realizations and σ_a and σ_b define a distribution function over all linear trajectories. Denote,

$$F_a(n) = n \quad , \quad F_b(n) = \frac{n(n+1)}{12(n-1)} = \sum_{t=0}^{n-1}(\frac{t}{n-1} - \frac{1}{2}) \tag{4.20}$$

$$E_{a,x} = \frac{1}{F_a(n)} \sum_t x_t \quad , \quad E_{b,x} = \frac{1}{F_b(n)} \sum_t x_t (\frac{t}{n-1} - \frac{1}{2}) \tag{4.21}$$

$$(\hat{a}, \hat{b}) = \arg\max_{a,b} f(x, a, b)$$

In Appendix 4.A we compute explicit expressions for $\hat{a}$ and $\hat{b}$ and prove that the true PDF $f(x)$ and the MAP approximation $f(x, \hat{a}, \hat{b})$ are related via

$$f(x) = \left(\frac{1}{\sigma_a^2} + \frac{F_a(n)}{\sigma^2} \right)^{-\frac{1}{2}} \left(\frac{1}{\sigma_b^2} + \frac{F_b(n)}{\sigma^2} \right)^{-\frac{1}{2}} 2\pi f(x, \hat{a}, \hat{b})$$

This relation corresponds to (4.13) in the static case.

As in the static case, there is no closed form expression for the ML estimation of the linear model parameters. Holmes and Russell [28] proposed an approximated solution which is an extension of their solution for the static RTSM. Alternatively, although the linear RTSM is out of the scope of dynamical system models, the EM approach for static RTSM, suggested by Digalakis [11], can be generalized to the linear case. The missing data in the EM algorithm are the hidden values of the random variables a and b, which are sampled for each segment realization. The proposed EM algorithm is developed in Appendix 4.B. The balancing problem, discussed in the previous section, also exists in the linear model. The approach of using a Richter distribution for better modeling the intra-segmental variability, was applied to the linear RTSM by Holmes and Russell [30].

We now present the scaled version for the linear random segmental model. The motivation for this model is similar to that for the static case. The scaled model spreads the information on the hidden linear trajectory uniformly along the time axis. The segment x is generated in the scaled model according to :

$$x_t = \mu_a + a + (\mu_b + b)(\frac{t}{n-1} - \frac{1}{2}) + \epsilon_t \qquad t = 0, ..., n-1$$

The difference from the unscaled linear model is that now the variances of a and b

are dependent on the segment duration as follows :

$$a \sim N\left(0, \frac{\sigma_a^2}{F_a(n)}\right) \quad , \quad b \sim N\left(0, \frac{\sigma_b^2}{F_b(n)}\right)$$

The joint PDF of x, a and b is :

$$f(x,a,b) = \frac{\sqrt{F_a(n)}}{\sqrt{2\pi}\sigma_a}\frac{\sqrt{F_b(n)}}{\sqrt{2\pi}\sigma_b}\left(\frac{1}{\sqrt{2\pi}\sigma}\right)^n e^{-\frac{1}{2}g(x,a,b)}$$

where $\quad g(x,a,b) = \dfrac{F_a(n)a^2}{\sigma_a^2} + \dfrac{F_b(n)b^2}{\sigma_b^2} + \dfrac{1}{\sigma^2}\sum_t (x_t - (\mu_a + a) - (\mu_b + b)(\dfrac{t}{n-1} - \dfrac{1}{2}))^2$

Algebraic manipulation of $g(x,a,b)$ reveals :

$$\begin{aligned}(\hat{a},\hat{b}) &= \arg\max_{a,b} f(x,a,b) = E((a,b)|x) \qquad (4.22)\\ &= \left(\frac{\frac{1}{\sigma^2}\sum(x_t - \mu_a)}{F_a(n)(\frac{1}{\sigma_a^2} + \frac{1}{\sigma^2})} \quad , \quad \frac{\frac{1}{\sigma^2}\sum(x_t - \mu_b(\frac{t}{n-1} - \frac{1}{2}))(\frac{t}{n-1} - \frac{1}{2})}{F_b(n)(\frac{1}{\sigma_b^2} + \frac{1}{\sigma^2})}\right)\end{aligned}$$

To gain further insight to the probabilistic behavior of the model we derive an explicit expression for $f(x)$. In Appendix 4.A we compute the following equivalent expression for g(x,a,b) :

$$g(x,a,b) = \left(\frac{F_a(n)}{\sigma_a^2} + \frac{F_a(n)}{\sigma^2}\right)(a-\hat{a})^2 + \left(\frac{F_b(n)}{\sigma_b^2} + \frac{F_b(n)}{\sigma^2}\right)(b-\hat{b})^2 + g(x,\hat{a},\hat{b}) \quad (4.23)$$

and $g(x,\hat{a},\hat{b})$ may be written as :

$$\begin{aligned}g(x,\hat{a},\hat{b}) = \frac{1}{\sigma^2}(\sum x_t^2 &+ F_a(n)(\frac{\sigma^2}{\sigma_a^2+\sigma^2}(E_{a,x} - \mu_a)^2 - (E_{a,x})^2) \qquad (4.24)\\ &+ F_b(n)(\frac{\sigma^2}{\sigma_b^2+\sigma^2}(E_{b,x} - \mu_b)^2 - (E_{b,x})^2))\end{aligned}$$

The following explicit expression for $f(x)$ results,

$$f(x) = \left(\frac{\sigma^2}{\sigma_a^2+\sigma^2}\right)^{\frac{1}{2}} \left(\frac{\sigma^2}{\sigma_b^2+\sigma^2}\right)^{\frac{1}{2}} \left(\frac{1}{\sqrt{2\pi}\sigma}\right)^n e^{-\frac{1}{2}g(x,\hat{a},\hat{b})} \tag{4.25}$$

We shall now discuss the parameter estimation problem given the segmented data. We use the modified segmental k-means approach in the same manner outlined in the previous section. In spite of the fact that the linear model is more complicated than the static one, using a scaled model enables us to derive closed form expressions for the estimated parameters. Let $x = (x_1, ..., x_k)$ be k segment realizations associated with the state s. Denote the length of the sequence x_i by n_i. The ML equations are :

$$\frac{\partial \log f(x)}{\partial \sigma_a^2} = \frac{1}{\sigma_a^2+\sigma^2} \sum_i \left(1 - \frac{1}{\sigma_a^2+\sigma^2} F_a(n_i)(E_{a,x_i} - \mu_a)^2\right) = 0$$

Hence,

$$\sigma_a^2 + \sigma^2 = \frac{1}{k} \sum_i F_a(n_i)(E_{a,x_i} - \mu_a)^2 \tag{4.26}$$

$$\frac{\partial \log f(x)}{\partial \sigma_b^2} = \frac{1}{\sigma_b^2+\sigma^2} \sum_i \left(1 - \frac{1}{\sigma_b^2+\sigma^2} F_b(n_i)(E_{b,x_i} - \mu_b)^2\right) = 0$$

Hence,

$$\sigma_b^2 + \sigma^2 = \frac{1}{k} \sum_i F_b(n_i)(E_{b,x_i} - \mu_b)^2 \tag{4.27}$$

$$\begin{aligned} \frac{\partial \log f(x)}{\partial \sigma^2} &= \frac{1}{\sigma^2} \sum_i (n_i - 2) \\ &\quad - \frac{1}{\sigma^4} \sum_i \left(\sum_t x_{i,t}^2 - F_a(n_i)(E_{a,x_i})^2 - F_b(n_i)(E_{b,x_i})^2\right) \\ &\quad + \frac{1}{\sigma_a^2+\sigma^2} \left(k - \frac{1}{\sigma_a^2+\sigma^2} \sum_i F_a(n_i)(E_{a,x_i} - \mu_a)^2\right) \end{aligned} \tag{4.28}$$

$$+\frac{1}{\sigma_b^2+\sigma^2}\left(k-\frac{1}{\sigma_b^2+\sigma^2}\sum_i F_b(n_i)(E_{b,x_i}-\mu_b)^2\right)$$

Substituting (4.26) and (4.27) in (4.28) enables us to solve the likelihood equations. Therefore the ML estimators are :

$$\begin{aligned}
\hat{\mu}_a &= \frac{\sum F_a(n_i)E_{a,x_i}}{\sum F_a(n_i)} = \frac{\sum_{i,t} x_{i,t}}{\sum F_a(n_i)} \\
\hat{\mu}_b &= \frac{\sum F_b(n_i)E_{b,x_i}}{\sum F_b(n_i)} = \frac{\sum_{i,t} x_{i,t}(\frac{t}{n_i-1}-\frac{1}{2})}{\sum F_b(n_i)} \\
\hat{\sigma}^2 &= \frac{\sum_i(\sum_t x_{i,t}^2 - F_a(n_i)(E_{a,x_i})^2 - F_b(n_i)(E_{b,x_i})^2)}{\sum_i(n_i-2)} \\
\hat{\sigma}_a^2 &= \frac{1}{k}\sum F_a(n_i)(E_{a,x_i}-\hat{\mu}_a)^2 \quad - \quad \hat{\sigma}^2 \\
\hat{\sigma}_b^2 &= \frac{1}{k}\sum F_b(n_i)(E_{b,x_i}-\hat{\mu}_b)^2 \quad - \quad \hat{\sigma}^2
\end{aligned}$$

As in the static case the ML estimators for σ_a^2 and σ_b^2 can be negative. Equation (4.25), which present the density function of a given segment for the scaled linear model, reveals that the actual parameters are $\sigma_a^2+\sigma^2$ and $\sigma_b^2+\sigma^2$. The ML estimation of these expressions is always non negative.

4.5 Baum-Welch re-Estimation

In previous sections we have described a training procedure for scaled random models that is an extension of the segmental k-means algorithm. In this section we present a Baum-Welch type re-estimation procedure for the scaled segmental random model. Using a scaled model enables us to perform the exact E- and M-steps of the Baum-Welch algorithm. This can not be done in the unscaled model [58]. The E-step

can be performed even in the unscaled model. However, the optimization procedure during the M-step does not yield a closed form for the estimated parameters.

Consider a left to right segmental HMM consisting of m states. Denote the segmental models associated with the states by $M_1, ..., M_m$. Only one segment from each utterance is associated with a particular state. In other words, the utterance is divided into m segments that correspond to the m segmental models. While this assumption simplifies notation, it is not necessary. Other HMM topologies can be analyzed in a similar manner.

According to this model, a state sequence is associated with each utterance. During the generation of the utterance, first a state sequence is chosen and then the observations sequence is sampled according to that state sequence. Denote by $ts(s, j)$ the time index of the transition into the state j in the sequence s (for example, if s=11122333 then $ts(s, 2) = 4$). Denote by $te(s, j)$ the time index of the transition out from the state j in the sequence s. We denote the time interval of the visit in the state j in the sequence s by :

$$t(s, j) = ts(s, j), ts(s, j)+1, ..., te(s, j)$$

Using this notation we can write the joint density of a given utterance x and a state sequence s as follows :

$$f(x, s) = \prod_{j=1}^{m} f(x_{t(s,j)}|M_j)$$

The probability of the utterance x is obtained by summing over all possible state sequences :

$$f(x) = \sum_{s} f(x, s)$$

Dynamic programming must be applied in order to efficiently compute this expression. Denote by $\alpha(j, t)$ the conditional probability of $x_1, ..., x_t$, given that the last visited state is j, and this sojourn is ended at time index t. A recursion formula for

$\alpha(j,t)$ can be stated in the following way :

$$\alpha(1,t) = f(x_1,...,x_t|M_1) \qquad\qquad t = 1,...,n$$

$$\alpha(j,t) = \sum_{u<t} \alpha(j-1,u) f(x_{u+1},...,x_t|M_j) \quad j = 2,...,m \quad , \quad t = 1,...,n$$

where f is the density function of a entire segment according to a segmental model. The likelihood score of the utterance is $f(x) = \alpha(m,n)$, where n is the utterance length and m is the index of the state at the end of the word. This recursion formula reflects the topology of the HMM. Other topologies (e.g. addition of a self loop to permit multiple entry to the same state) imply modification of the recursion step. Similar dynamic programming is performed in order to compute the likelihood score in the standard HMM [53]. However, in segmental models computation of the density function is far more complex. This is due to the fact that the probability of a frame does not depend only on the state but also on the location of this frame within the segment sampled during the visit in the state. In a segmental model we can not compute the probability of a single frame in a state. We must compute the probability of the entire segment. The complexity of the algorithm can be reduced by assuming a maximal state duration.

In the Viterbi decoding approach we choose the best suited state sequence. For the Baum-Welch algorithm we must consider all the possible state sequences. Each state sequence is considered according to its relative weight. Denote by $w(j,t_1,t_2)$ the a-posteriori probability that the portion of the utterance x sampled at state j is $x_{t_1},...,x_{t_2}$. Applying Bayes rule yields :

$$w(j,t_1,t_2) = \frac{\sum_{\{s|ts(s,j)=t_1,te(s,j)=t_2\}} f(x,s)}{f(x)}$$

An extension of the Forward-Backward algorithm can be applied for efficient computation of $w(j,t_1,t_2)$. Denote by $\beta(j,t)$ the conditional probability of $x_t,...,x_n$ given

that at time index t there was a transition into the state j. A recursion formula for $\beta(t,j)$ is :

$$\beta(m,t) = f(x_t,...,x_n|M_m) \qquad\qquad t=1,...,n$$

$$\beta(j,t) = \sum_{u>t} f(x_t,...,x_{u-1}|M_j)\beta(j+1,u) \qquad t=1,...,n \quad , \quad j=1,...,m-1$$

Using this notation we derive the following expression for $w(j,t_1,t_2)$:

$$w(j,t_1,t_2) = \frac{\alpha(j-1,t_1-1)f(x_{t_1},...,x_{t_2}|M_j)\beta(j+1,t_2+1)}{f(x)}$$

Computing the expression $w(j,t_1,t_2)$ is the main step in performing the EM iteration in the Baum-Welch framework. In this chapter it is assumed that the observations are scalar. In case of multi-dimensional observations and diagonal matrix covariances, the weight $w(j,t_1,t_2)$ is computed for all the observations components together. Once the weight is computed, the estimation can be done for each component separately.

Until now we have discussed segmental models in general. Assume that the segmental model associated with the states is the scaled linear RTSM. Denote the parameters of the model M_j by $\{\mu_{j,a},\sigma^2_{j,a},\mu_{j,b},\sigma^2_{j,b},\sigma^2_j\}$. According to the definition of the scaled linear RTSM, given in the previous section, the model M_j can be written as :

$$x_t = \mu_{j,a} + a + (\mu_{j,b}+b)(\frac{t-t_1}{t_2-t_1} - \frac{1}{2}) + \epsilon_t$$

where $t_1,...,t_2$ is the time interval of the sojourn in the state j, and a and b are sampled at the transition into the state j. Assume that the training data-base consists of the k utterances $x_1,...,x_k$. Denote the length of x_i by n_i. Denote by $w_i(j,t_1,t_2)$ the a-posteriori probability that the portion of x_i sampled at state j is $x_{i,t_1},...,x_{i,t_2}$. The EM auxiliary function is :

$$Q(\theta,\theta_0) \quad = \quad E(\log f(x,s,\theta)|x,\theta_0)$$

$$= \sum_i \sum_s f(s|x_i, \theta_0) \log f(x_i, s, \theta)$$

Differentiating the auxiliary function with respect to $\mu_{j,a}$ yields :

$$\begin{aligned}
\frac{\partial Q(\theta, \theta_0)}{\partial \mu_{j,a}} &= \sum_i \sum_s f(s|x_i, \theta_0) \frac{\partial}{\partial \mu_{j,a}} \log f(x_i, s, \theta_0) \\
&= \sum_i \sum_s f(s|x_i, \theta_0) \frac{\partial}{\partial \mu_{j,a}} \log f(x_{i,t(s,j)}|M_j) \\
&= \sum_i \sum_{t_1<t_2} w_i(j, t_1, t_2) \frac{\partial}{\partial \mu_{j,a}} \log f(x_{i,t_1}, ..., x_{i,t_2}|M_j) \\
&= \sum_i \sum_{t_1<t_2} w_i(j, t_1, t_2) \frac{1}{\sigma_{j,a}^2 + \sigma_j^2} \sum_{t=t_1}^{t_2} (x_{i,t} - \mu_{j,a})
\end{aligned}$$

Setting this partial derivative to zero yields the re-estimation formula for μ_a. In a similar manner we can obtain re-estimation formulae for the other parameters. The algebraic manipulation involved here is similar to one presented in Appendix 4.B for the segmental k-means training procedure. In order to simplify the presentation of the re-estimation formulae, we defined the following notation :

$$\Delta t = t_2 - t_1 + 1$$

$$E_{a,x_i}(t_1, t_2) = \frac{1}{F_a(\Delta t)} \sum_{t=t_1}^{t_2} x_{i,t}$$

$$E_{b,x_i}(t_1, t_2) = \frac{1}{F_b(\Delta t)} \sum_{t=t_1}^{t_2} x_{i,t} (\frac{t - t_1}{t_2 - t_1} - \frac{1}{2})$$

We can now state the re-estimation equations of Baum-Welch algorithm for training the scaled linear RTSM :

$$\hat{\mu}_{j,a} = \frac{\sum_i \sum_{t_1<t_2} w_i(j, t_1, t_2) \sum_{t=t_1}^{t_2} x_{i,t}}{\sum_i \sum_{t_1<t_2} w_i(j, t_1, t_2) F_a(\Delta t)}$$

$$\hat{\mu}_{j,b} = \frac{\sum_i \sum_{t_1<t_2} w_i(j,t_1,t_2) \sum_{t=t_1}^{t_2} x_{i,t}(\frac{t-t_1}{t_2-t_1} - \frac{1}{2})}{\sum_i \sum_{t_1<t_2} w_i(j,t_1,t_2) F_b(\Delta t)}$$

$$\hat{\sigma}_j^2 = \frac{\sum_i \sum_{t_1<t_2} w_i(j,t_1,t_2)(\sum_t x_{i,t}^2 - F_a(\Delta t)(E_{a,x_i}(t_1,t_2))^2 - F_b(\Delta t)(E_{b,x_i}(t_1,t_2))^2)}{\sum_i \sum_{t_1<t_2} w_i(j,t_1,t_2)(\Delta t - 2)}$$

$$\hat{\sigma}_{j,a}^2 = \frac{1}{k} \sum_i \sum_{t_1<t_2} w_i(j,t_1,t_2) F_a(\Delta t)(E_{a,x_i}(t_1,t_2) - \hat{\mu}_{j,a})^2 - \hat{\sigma}_j^2$$

$$\hat{\sigma}_{j,b}^2 = \frac{1}{k} \sum_i \sum_{t_1<t_2} w_i(j,t_1,t_2) F_b(\Delta t)(E_{b,x_i}(t_1,t_2) - \hat{\mu}_{j,b})^2 - \hat{\sigma}_j^2$$

The random model is more informative than the deterministic one. Therefore, it is more sensitive to the initial values of the model parameters. Initialization can be done in the following manner [28]. First, train a deterministic linear model or even a standard HMM. Then given the deterministic model, we can apply Viterbi decoding to divide each utterance into segments. The segmented data can be used as an input for the initialization process described in Appendix 4.B.

4.6 Experimental Results

We evaluated the model presented in the previous section using the ARPA, large vocabulary, speaker independent, continuous speech, *Wall Street Journal* (WSJ) corpus [15]. Experiments were conducted with DECIPHER, SRI's continuous speech recognition system [14]. A detailed description of the WSJ data-base and the signal processing performed in the DECIPHER system can be found in Appendix A. The automatic segmentation is not perfect and is not comparable with that of a professional phonetician. However, this segmentation enables us to perform a fair and extensive comparison between different acoustic models.

The task we choose for evaluation is phonetic classification. In classification the

correct segmentation (phone beginning and ending time) of the input observation sequence is given. Our objective is to assign correct phone labels to each segment. The DECIPHER system was used to determine automatically the phone segmentation for each sentence in the database. Having obtained phonetically aligned test data, the actual classification process is just a matter of finding the most likely phone label for a speech segment according to the models being evaluated. The training set consists of 100 realizations in various contexts for each phone. The testing set consists of another 100 realizations for each phoneme.

The goal of the experiments we have conducted is to compare between the performance of the scaled and unscaled random mean trajectory models. In previous sections we have discussed both the cases of constant and linear mean trajectories. The experiments were performed for static as well as linear models. This enable us to find how much the assumption of linear mean trajectory improves the performance. We also add results for models where the mean trajectory is a deterministic parameter (i.e. standard Gaussian HMM and Deng's linear model) as a reference. It should be noted that a deterministic model has less parameters than the corresponding random model. The acoustic models we implemented for evaluation were:

1. Standard Gaussian HMM.
2. Static RTSM (Russell [58]).
3. Scaled static RTSM (presented in section 3).
4. Linear mean trajectory segment model (Deng et al. [7]).
5. Linear RTSM (Holmes and Russell [28]).
6. Scaled linear RTSM (presented in section 4).

Several alternatives for model topologies have been employed in order to analyze the balancing problem in different situations. The first analyzed topology assigns one segment to the entire phone. The second one still associates a single segment

	model	one state	one state self-loop	three states	three states with skips
static	deterministic	52.1		61.7	61.9
	non-scaled	50.7	52.4	61.1	61.7
	scaled	51.5	55.1	62.3	62.8
linear	deterministic	57.2	57.8	61.8	62.3
	non-scaled	58.0	57.0	63.3	63.9
	scaled	59.1	58.1	62.2	64.1

Table 4.1: Phoneme classification rate results

model with each phone, but includes self loops. In other words, the phone can be modeled by a number of segments which are all corresponding to the same segmental model. The third topology, models each phone using three states. The last examined topology is also a three state model but a skip over a state is allowed. In this manner a phone can be explained using at most three states. Note that in the second and the forth topologies the number of segments per phone is not fixed. Balancing the model components in those cases is critical.

Training the models that includes more than one segment per phone was done using the segmental k-means algorithm. The parameters of the unscaled random models were computed using the iterative inner EM algorithm that is presented in Appendix 4.B. We preferred to use this training method because, although it is an iterative procedure, it computes the true likelihood function. The scaled model was trained using the closed form formulae that were developed in previous sections. The random models, both scaled and unscaled, were initialized using the deterministic version of the model for the first iteration of the segmental k-means algorithm. Duration was not modeled explicitly. Therefore, all durations were assigned equal probability.

As can be seen from Table 4.1, the scaled model usually outperforms the previously suggested non-scaled model, both for the static case and for the linear case. These results also reassess the significant performance improvement caused by using a linear model instead of a static one. It is noteworthy to mention that by com-

	model	m-?-n	p-?-t	s-?-ae	t-?-s	k-?-t	average
static	non-scaled	49.4	61.0	80.3	69.3	82.9	79.6
	scaled	52.7	65.1	79.6	71.4	84.2	80.3
linear	non-scaled	59.3	64.0	88.4	72.1	87.5	82.0
	scaled	61.9	67.7	88.9	74.8	90.0	82.4

Table 4.2: Triphone classification rate results

paring the parameters of the scaled and the corresponding unscaled model, we have found that the values of the mean parameters and the values of the intra-segmental variances are similar.

Another task we evaluated is triphone classification. Given a triphone context (the phones before and after the current one), the goal is to determine the label of the current phone based on the acoustics. Context dependent classification was chosen because, in that case there are fewer discrepancies between utterances. Hence, in practice, this is usually the case of interest when using segment models. The topology we have used for this task is the simplest one. Each phoneme is modeled by a single segmental model. This is the first topology from the topologies list of the previous experiment.

In Table 4.2 we present recognition results for some frequently occurring triphone contexts. The first column in Table 4.2 summarizes a classification experiment given the triphone context m-n. The segmented WSJ database was used to extract the phones that appear between the phones m and n. The classification was done among those phonemes that have a significant number of occurrences (at least 60) in the m-n context. In this context these phonemes are (using ARPABET notation) : aa, ae, ah, aw, ax, ay, eh, ey and iy. Half of the data was used to train the triphone models. The other half was used for the actual classification task. The subsequent four columns present results for similar classification tasks in other triphone contexts. The final column presents the classification performance averaged over the 120 most frequently occurring triphone contexts. As can be seen, the scaled model outperforms the previously suggested non scaled model.

4.7 Conclusions

In this chapter we have proposed, implemented and evaluated a new type of random trajectory segment model where the variance of the mean trajectory is inversely proportional to the segment duration. In this model the division of the acoustic information in an utterance does not depend on a specific segmentation. Instead, we extract the same amount of information about the mean trajectory from each data frame. We have named this approach a scaled RTSM. One desirable attribute of the scaled model is that it leads to a simple training algorithm. More precisely, given a training set, consisting of a list of segment realizations, the ML estimation of the scaled model can be solved analytically. On the other hand, in the non-scaled model, an iterative algorithm, (e.g., EM) is required. Such iterative algorithm, is not guaranteed to reach the global maximum.

Appendix 4.A

We derive an explicit expression for the PDF of the linear RTSM. Both the scaled and the non-scaled versions are discussed. We begin by analyzing the scaled model. The definition of the scaled linear model implies that the joint PDF of the observed segment x of length n and the linear function coefficients a and b is :

$$f(x,a,b) = \frac{\sqrt{F_a(n)}}{\sqrt{2\pi}\sigma_a}\frac{\sqrt{F_b(n)}}{\sqrt{2\pi}\sigma_b}\left(\frac{1}{\sqrt{2\pi}\sigma}\right)^n e^{-\frac{1}{2}g(x,a,b)}$$

where

$$g(x,a,b) = \frac{F_a(n)a^2}{\sigma_a^2} + \frac{F_b(n)b^2}{\sigma_b^2} + \frac{1}{\sigma^2}\sum_t (x_t - (\mu_a + a) - (\mu_b + b)(\frac{t}{n-1} - \frac{1}{2}))^2$$

We show that $g(x,a,b)$ may be written as :

$$\begin{aligned} g(x,a,b) &= \left(\frac{F_a(n)}{\sigma_a^2}+\frac{F_a(n)}{\sigma^2}\right)(a-\hat{a})^2+\left(\frac{F_b(n)}{\sigma_b^2}+\frac{F_b(n)}{\sigma^2}\right)(b-\hat{b})^2+ \\ &\quad \frac{1}{\sigma^2}(F_a(n)(\frac{\sigma^2}{\sigma_a^2+\sigma^2}(E_{a,x}-\mu_a)^2-(E_{a,x})^2)+ \\ &\quad\quad F_b(n)(\frac{\sigma^2}{\sigma_b^2+\sigma^2}(E_{b,x}-\mu_b)^2-(E_{b,x})^2)+\sum_t x_t^2) \end{aligned}$$

The terms $\hat{a}$, $\hat{b}$, $F_a(n)$, $F_b(n)$, $E_{a,x}$ and $E_{b,x}$ are defined in (4.20), (4.21).

$$\begin{aligned} g(x,a,b) &= \frac{F_a(n)a^2}{\sigma_a^2}+\frac{F_b(n)b^2}{\sigma_b^2}+\frac{1}{\sigma^2}\sum_t(x_t-(\mu_a+a)-(\mu_b+b)(\frac{t}{n-1}-\frac{1}{2}))^2 \\ &= \left(\frac{F_a(n)}{\sigma_a^2}+\frac{F_a(n)}{\sigma^2}\right)a^2-2a\frac{1}{\sigma^2}\sum_t(x_t-\mu_a)+ \\ &\quad \left(\frac{F_b(n)}{\sigma_b^2}+\frac{F_b(n)}{\sigma^2}\right)b^2-2b\frac{1}{\sigma^2}\sum_t(x_t-\mu_b(\frac{t}{n-1}-\frac{1}{2}))(\frac{t}{n-1}-\frac{1}{2})+ \\ &\quad\quad \frac{1}{\sigma^2}\sum_t(x_t-\mu_a-\mu_b(\frac{t}{n-1}-\frac{1}{2}))^2 \\ &= \left(\frac{F_a(n)}{\sigma_a^2}+\frac{F_a(n)}{\sigma^2}\right)\left(a-\frac{\sigma_a^2}{\sigma_a^2+\sigma^2}(E_{a,x}-\mu_a)\right)^2+ \\ &\quad \left(\frac{F_b(n)}{\sigma_b^2}+\frac{F_b(n)}{\sigma^2}\right)\left(b-\frac{\sigma_b^2}{\sigma_b^2+\sigma^2}(E_{b,x}-\mu_b)\right)^2 \\ &\quad -\frac{F_a(n)}{\sigma^2}\frac{\sigma_a^2}{\sigma_a^2+\sigma^2}(E_{a,x}-\mu_a)^2-\frac{F_b(n)}{\sigma^2}\frac{\sigma_b^2}{\sigma_b^2+\sigma^2}(E_{b,x}-\mu_b)^2+ \\ &\quad\quad \frac{1}{\sigma^2}\sum_t(x_t-\mu_a-\mu_b(\frac{t}{n-1}-\frac{1}{2}))^2 \end{aligned} \tag{4.29}$$

Given the observation sequence x, $g(x,a,b)$ is a quadratic form in a and b. Hence, we conclude that the conditional distribution of a and b given the segment x is Gaussian. Furthermore, a and b are conditionally independent. Now, the first moments may be read directly from (4.29):

$$E(a|x) = \hat{a} = \left(\frac{\sigma_a^2}{\sigma_a^2+\sigma^2}\right)(E_{a,x}-\mu_a)$$

$$
\begin{aligned}
Var(a|x) &= \left(\frac{F_a(n)}{\sigma_a^2} + \frac{F_a(n)}{\sigma^2}\right)^{-1} \\
E(b|x) &= \hat{b} = \left(\frac{\sigma_b^2}{\sigma_b^2 + \sigma^2}\right)(E_{b,x} - \mu_b) \\
Var(b|x) &= \left(\frac{F_b(n)}{\sigma_b^2} + \frac{F_b(n)}{\sigma^2}\right)^{-1}
\end{aligned}
$$

Direct algebraic manipulations reveal the following relation

$$
\sum_t (x_t - \mu_a - \mu_b(\frac{t}{n-1} - \frac{1}{2}))^2 =
$$

$$
F_a(n)((E_{a,x} - \mu_a)^2 - (E_{a,x})^2) + F_b(n)((E_{b,x} - \mu_b)^2 - (E_{b,x})^2) + \sum_t x_t^2
$$

Substituting this relation in (4.29) yields :

$$
\begin{aligned}
g(x,a,b) &= \left(\frac{F_a(n)}{\sigma_a^2} + \frac{F_a(n)}{\sigma^2}\right)(a - \hat{a})^2 + \left(\frac{F_b(n)}{\sigma_b^2} + \frac{F_b(n)}{\sigma^2}\right)(b - \hat{b})^2 + \\
&\quad \frac{1}{\sigma^2}(\sum x_t^2 + F_a(n)\left((1 - \frac{\sigma_a^2}{\sigma_a^2 + \sigma^2})(E_{a,x} - \mu_a)^2 - (E_{a,x})^2\right) \\
&\quad + F_b(n)\left((1 - \frac{\sigma_b^2}{\sigma_b^2 + \sigma^2})(E_{b,x} - \mu_b)^2 - (E_{b,x})^2\right))
\end{aligned}
$$

Hence,

$$
g(x,a,b) = \left(\frac{F_a(n)}{\sigma_a^2} + \frac{F_a(n)}{\sigma^2}\right)(a - \hat{a})^2 + \left(\frac{F_b(n)}{\sigma_b^2} + \frac{F_b(n)}{\sigma^2}\right)(b - \hat{b})^2 + g(x, \hat{a}, \hat{b})
$$

Now, $g(x, \hat{a}, \hat{b})$ may be written as :

$$
\begin{aligned}
g(x, \hat{a}, \hat{b}) = \frac{1}{\sigma^2}(\sum_t x_t^2 &+ F_a(n)(\frac{\sigma^2}{\sigma_a^2 + \sigma^2}(E_{a,x} - \mu_a)^2 - (E_{a,x})^2) \\
&+ F_b(n)(\frac{\sigma^2}{\sigma_b^2 + \sigma^2}(E_{b,x} - \mu_b)^2 - (E_{b,x})^2))
\end{aligned}
$$

Using this representation we may solve the double integral :

$$f(x) = \int_a \int_b f(x,a,b)da\,db$$

and obtain the following explicit expression for the PDF,

$$\begin{aligned} f(x) &= \left(\frac{F_a(n)}{\sigma_a^2} + \frac{F_a(n)}{\sigma^2}\right)^{-\frac{1}{2}} \left(\frac{F_b(n)}{\sigma_b^2} + \frac{F_b(n)}{\sigma^2}\right)^{-\frac{1}{2}} 2\pi f(x,\hat{a},\hat{b}) \\ &= \left(\frac{\sigma^2}{\sigma_a^2+\sigma^2}\right)^{\frac{1}{2}} \left(\frac{\sigma^2}{\sigma_b^2+\sigma^2}\right)^{\frac{1}{2}} \left(\frac{1}{\sqrt{2\pi}\sigma}\right)^n e^{-\frac{1}{2}g(x,\hat{a},\hat{b})} \end{aligned}$$

The expressions we derived for $f(x)$, $g(x,a,b)$ and the moments of the conditional distribution of a and b given x may be easily transformed to the non-scaled linear case. We need only substitute $F_a(n)\sigma_a^2$ and $F_b(n)\sigma_b^2$ for σ_a^2 and σ_b^2. For example, in the non-scaled case we have :

$$E(a|x) = \hat{a} = \left(\frac{F_a(n)\sigma_a^2}{F_a(n)\sigma_a^2+\sigma^2}\right)(E_{a,x} - \mu_a)$$

$$E(b|x) = \hat{b} = \left(\frac{F_b(n)\sigma_b^2}{F_b(n)\sigma_b^2+\sigma^2}\right)(E_{b,x} - \mu_b)$$

and the PDF of the observation sequence x of length n is :

$$f(x) = \left(\frac{1}{\sigma_a^2} + \frac{F_a(n)}{\sigma^2}\right)^{-\frac{1}{2}} \left(\frac{1}{\sigma_b^2} + \frac{F_b(n)}{\sigma^2}\right)^{-\frac{1}{2}} 2\pi f(x,\hat{a},\hat{b})$$

Appendix 4.B

The static RTSM can be seen as a special case of the dynamical system model [12] such that the state space is constant over time. The EM algorithm presented by Digalakis et al. can be applied to the static RTSM. In this appendix we derive the EM re-estimation equations for the unscaled linear RTSM which is out of the scope of dynamic segment systems. The re-estimation equations for static RTSM can be easily deduced from the equations developed here. Suppose we have k segment realizations $x_1, ..., x_k$. Denote the length of x_i by n_i. Denote by a_i and b_i the hidden coefficients of the line which was sampled for the segment x_i. Define $z_i = (x_i, a_i, b_i)$, $z_1, ..., z_k$ are the complete data for this EM framework. Denote the parameter set we want to estimate by $\theta = \{\mu_a, \mu_b, \sigma_a^2, \sigma_b^2, \sigma^2\}$. The current estimate at the beginning of the iteration is denoted by $\theta_0 = \{\mu_{a0}, \mu_{b0}, \sigma_{a0}^2, \sigma_{b0}^2, \sigma_0^2\}$.

$$\begin{aligned}\log f(z_i, \theta) &= \log \sigma_a^2 + \frac{a_i^2}{\sigma_a^2} + \log \sigma_b^2 + \frac{b_i^2}{\sigma_b^2} + n_i \log \sigma^2 + \\ &\quad \frac{1}{\sigma^2} \sum_t (x_{i,t} - (\mu_a - a_i) - (\mu_b - b_i)(\frac{t}{n-1} - \frac{1}{2}))^2 + C\end{aligned}$$

where C is a constant that is independent of the parameter vector θ. Hence,

$$E(\log f(z_i, \theta)|x, \theta_0) = n_i \log \sigma^2 +$$

$$\log \sigma_a^2 + \frac{1}{\sigma_a^2} E(a_i^2|x_i, \theta_0) + \log \sigma_b^2 + \frac{1}{\sigma_b^2} E(b_i^2|x_i, \theta_0) +$$

$$+ \frac{1}{\sigma^2} \sum_t E((x_{i,t} - (\mu_a - a_i) - (\mu_b - b_i)(\frac{t}{n_i - 1} - \frac{1}{2}))^2 | x_i, \theta_0)$$

Denote :

$$E_{a_i} = E(a_i|x_i, \theta_0) = \left(\frac{1}{\sigma_{a0}^2} + \frac{F_a(n_i)}{\sigma_0^2}\right)^{-1} \frac{F_a(n_i)}{\sigma_0^2} (E_{a,x_i} - \mu_{a0})$$

$$
\begin{aligned}
V_{a_i} &= Var(a_i|x_i,\theta_0) = \left(\frac{1}{\sigma_{a0}^2} + \frac{F_a(n_i)}{\sigma_0^2}\right)^{-1} \\
E_{b_i} &= E(b_i|x_i,\theta_0) = \left(\frac{1}{\sigma_{b0}^2} + \frac{F_b(n_i)}{\sigma_0^2}\right)^{-1} \frac{F_b(n_i)}{\sigma_0^2}(E_{b,x_i} - \mu_{b0}) \\
V_{b_i} &= Var(b_i|x_i,\theta_0) = \left(\frac{1}{\sigma_{b0}^2} + \frac{F_b(n_i)}{\sigma_0^2}\right)^{-1}
\end{aligned}
$$

where

$$
\begin{aligned}
F_a(n_i) &= n_i \quad , \quad E_{a,x_i} = \frac{1}{F_a(n_i)} \sum_t x_{i,t} \\
F_b(n_i) &= \frac{n_i(n_i+1)}{12(n_i-1)} \quad , \quad E_{b,x_i} = \frac{1}{F_b(n_i)} \sum_t x_{i,t}(\frac{t}{n_i-1} - \frac{1}{2})
\end{aligned}
$$

These relations are developed in Appendix 4.A. Direct algebraic manipulations reveal the following relation :

$$
E((x_{i,t} - (\mu_a - a_i) - (\mu_b - b_i)(\frac{t}{n_i-1} - \frac{1}{2}))^2|x_i,\theta_0) =
$$

$$
F_a(n_i)V_{a_i} + F_b(n_i)V_{b_i} + \sum_t (x_{i,t} - (\mu_a + E_{a_i}) - (\mu_b + E_{b_i})(\frac{t}{n_i-1} - \frac{1}{2}))^2
$$

We now define the EM auxiliary function :

$$
\begin{aligned}
Q(\theta,\theta_0) &= E(\log f(z,\theta)|x,\theta_0) = \sum_i E(\log f(z_i,\theta)|x_i,\theta_0) \\
&= k\log\sigma_a^2 + \frac{1}{\sigma_a^2}\sum_i E(a_i^2|x_i,\theta_0) + \frac{1}{\sigma^2}\sum_i F_a(n_i)V_{a_i} + \\
&\quad k\log\sigma_b^2 + \frac{1}{\sigma_b^2}\sum_i E(b_i^2|x_i,\theta_0) + \frac{1}{\sigma^2}\sum_i F_b(n_i)V_{b_i} + \\
&\quad \sum_i n_i\log\sigma^2 + \frac{1}{\sigma^2}\sum_{i,t}(x_{i,t} - (\mu_a + E_{a_i}) - (\mu_b + E_{b_i})(\frac{t}{n_i-1} - \frac{1}{2}))^2
\end{aligned}
$$

Optimization of the auxiliary function with respect to the model parameters yields the following new estimated parameter vector, $\hat{\theta} = \{\hat{\mu}_a, \hat{\mu}_b, \hat{\sigma}_a^2, \hat{\sigma}_b^2, \hat{\sigma}^2\}$

$$\hat{\mu}_a = \frac{\sum_i F_a(n_i)(E_{a,x_i} - E_{a_i})}{\sum_i F_a(n_i)}$$

$$\hat{\mu}_b = \frac{\sum_i F_b(n_i)(E_{b,x_i} - E_{b_i})}{\sum_i F_b(n_i)}$$

$$\hat{\sigma}_a^2 = \frac{1}{k}\sum_i (V_{a_i} + E_{a_i}^2) = \frac{1}{k}\sum_i E(a_i^2|x_i, \theta_0)$$

$$\hat{\sigma}_b^2 = \frac{1}{k}\sum_i (V_{b_i} + E_{b_i}^2) = \frac{1}{k}\sum_i E(b_i^2|x_i, \theta_0)$$

$$\hat{\sigma}^2 = \frac{1}{\sum_i n_i}\sum_i (F_a(n_i)V_{a_i} + F_b(n_i)V_{b_i} + \sum_t (x_{i,t} - (\hat{\mu}_a + E_{a_i}) - (\hat{\mu}_b + E_{b_i})(\frac{t}{n_i - 1} - \frac{1}{2}))^2)$$

Possible initialization values for the EM algorithm are :

$$\mu_{a0} = \frac{1}{\sum F_a(n_i)}\sum_i F_a(n_i)E_{a,x_i} \quad , \quad \sigma_{a0}^2 = \frac{1}{k}\sum_i (E_{a,x_i} - \mu_{a0})^2$$

$$\mu_{b0} = \frac{1}{\sum F_b(n_i)}\sum_i F_b(n_i)E_{b,x_i} \quad , \quad \sigma_{b0}^2 = \frac{1}{k}\sum_i (E_{b,x_i} - \mu_{b0})^2$$

$$\sigma_0^2 = \frac{1}{\sum n_i}\sum_{i,t} (x_{i,t} - \mu_{a0} - \mu_{b0}(\frac{t}{n_i - 1} - \frac{1}{2}))^2$$

Chapter 5

Continuous Mixture of Segmental Models

In this chapter we shall present a segmental modeling approach based on a non-parametric description of the mean trajectory during a sojourn in a phonetic unit. First, we provide a motivation for the model from examination of segment data realizations. Then we shall present the model itself, followed by a discussion of the training algorithm. Finally, we shall present experimental results which examine various aspects of the model.

5.1 Motivations

Our model was motivated by extensive examination of segment data realizations. In Figure 5.1, several realizations of the first cepstral coefficient in the triphone ih-s-ow are presented (The database used was the speaker independent, large vocabulary, *Wall Street Journal* (WSJ) corpus [15]). Figure 5.2 presents the same data after nonlinear time warping of the segment realizations, so as to achieve time alignment

between the various realizations. Figure 5.3 also presents time aligned segment realizations, but with an additional stage of global displacement removal. It can be clearly seen, that the variance of the trajectories in Figure 5.3 is smaller than the corresponding variance in Figure 5.2. Hence, in this case, the incorporation of the global displacement term yields improved modeling.

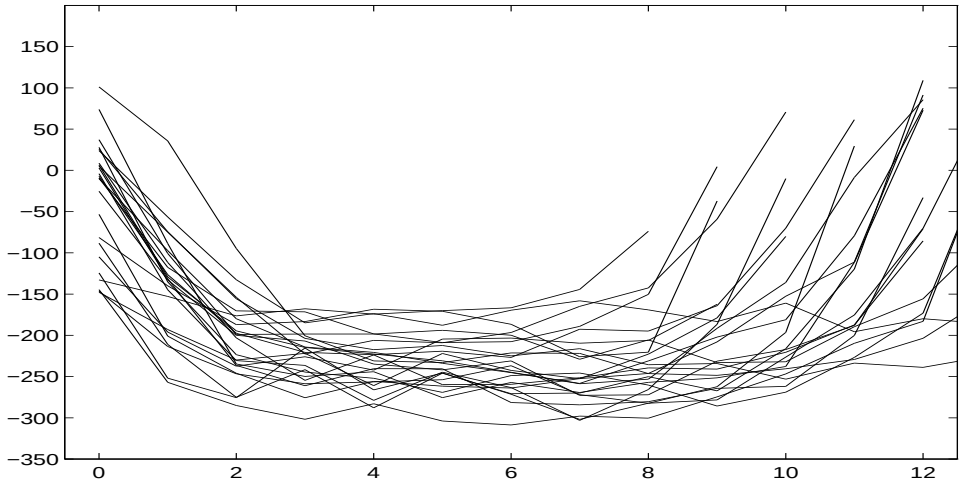

Figure 5.1: Realizations of first cepstral coefficient of the phone 's' in the triphone context ih-ow.

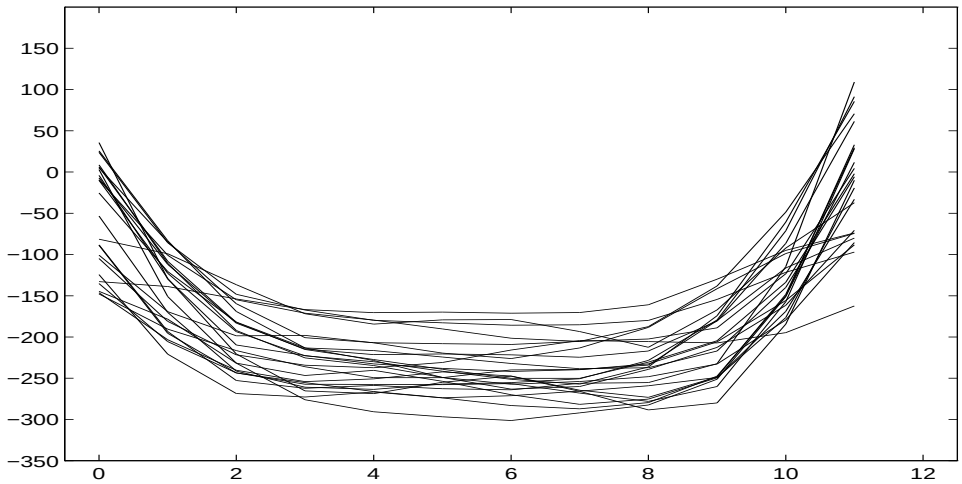

Figure 5.2: Data after nonlinear time warping.

These figures reflect the fact that the speech signal is produced by a continuously varying physical system (the vocal track). There is a smooth local dynamic along the sojourn in the phonetic unit. We can also observe from these figures that when

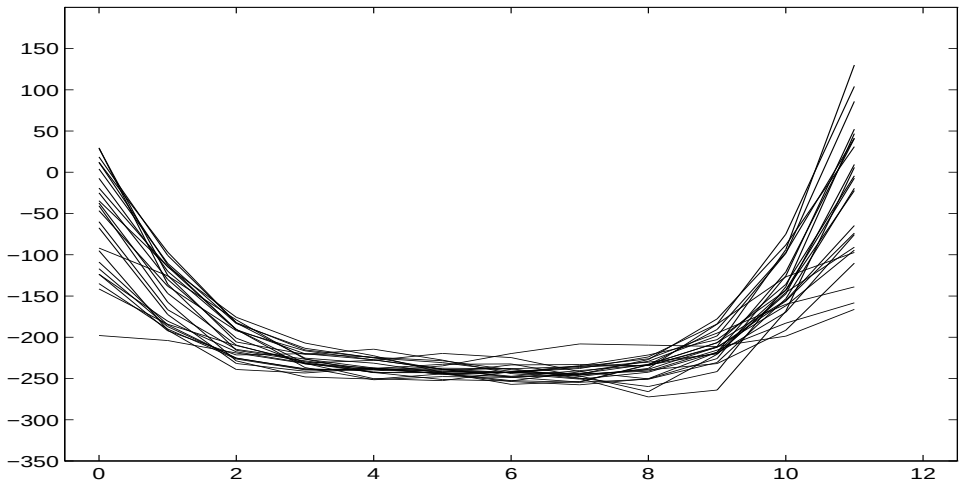

Figure 5.3: Data after nonlinear time warping and displacement elimination.

we fix a context there is not any significant variability among different realizations of a phonetic unit. In other words, the phone does not exhibit several distinct trajectories. The observed difference can be described as a shift from the mean trajectory that is global to the entire segment. This phenomenon can be explained on two levels. First, the shift reflects the speaker personal style which is dependent upon the shape of vocal organs, gender, age, dialect etc. From a more local point of view the size of the shift depends on the ending point of the previous phoneme and the starting point of the next one. This is one way of describing the co-articulation effect. The vocal articulators move from the position necessary for articulation of the previous phone towards the position required for the next phone, via the position needed for the current phone. During fluent speech when the context phones are not fully articulated there will be a shift from the mean trajectory due to the continuous nature of the speech signal. In the next section we shall formulate these intuitive observations into a probabilistic model.

5.2 Model Formulation

In this section we present a new segmental model which is composed of two elements. The first element is a non-parametric representation of the mean and variance tra-

jectories, and the second is a parameterized transformation (e.g. random shift) of the trajectory that is global to the entire segment [21], [22]. The mean trajectory curve is represented using a non-parametric description. That is to say, instead of using a polynomial or some other parametric description, as in [7], [8], [41], [11], [58], [27], [28], [29] and [18], the curve is represented by specifying a list of sampled points along it. More precisely, we assume that each segment may be represented by a left to right HMM structure, such that each HMM state is represented by a single Gaussian HMM. The sequence of mean values of the HMM state sequence constitutes a template of the mean trajectory. Likewise, the sequence of variances of the HMM state sequence constitutes a template of the variance trajectory. Time warping of the template trajectory is made possible by controlling the state sequence of the HMM (e.g., contracting may be realized by rapid transitions out of states). The second element of the model is a parameterized transformation of the trajectory, that is global to the entire segment. Let the state sequence of some given segment realization be denoted by $s = (s_1, s_2, \ldots, s_n)$, and let the corresponding observation sequence be denoted by $x = (x_1, x_2, \ldots, x_n)$. To simplify notation, it will be assumed during this chapter that the observations are one dimensional. Generalization to the multi-dimensional case is straight-forward. We assume the following model for the observation, x_t, at time t :

$$x_t = T_a\left(\mu(s_t), \sigma(s_t), t\right)$$

where $\mu(s_t)$ and $\sigma(s_t)$ are the mean and variance at state s_t, and $T_a(\cdot)$ is some random transformation indexed by a. a is a random variable that is chosen once for the entire segment realization. The transformation that we focus on in this chapter, is a random displacement of the mean trajectory. In that case, $T_a(\mu, \sigma, t) = \mu + a + \epsilon_t(\sigma)$. Hence,

$$x_t = \mu(s_t) + a + \epsilon_t(\sigma(s_t)) \tag{5.1}$$

Here, a is a zero mean, normal random variable, sampled once per segment, that represents the global displacement of the current segment realization. $\epsilon_t(\sigma)$ is a zero mean, Gaussian random variable.

$$a \sim N(0, \sigma_a^2) \quad , \quad \epsilon_t \sim N(0, \sigma^2(s_t))$$

The effect of the displacement variable may be interpreted as a continuous mixture of parallel curves that represent the mean trajectory along the segment. The distribution of a is the continuous segmental analog to the mixture coefficients in standard HMM. That is to say, in standard HMM, a discrete mixture component is chosen once per frame, i.e., it is a frame based approach, while in a random segmental model, a continuous mixture component is chosen once per segment realization.

The proposed model (5.1) is similar to the models suggested in [58], [27], [28], [29] and [18]. In these references, however, the approach is parametric, while our approach is non-parametric and allows time warping of the mean trajectory. The flexibility that is gained by allowing time warping could significantly improve the modeling capability of the template trajectory. The boundaries between phones are not well defined and there is no any consistent starting point. Non-parametric representation allows us flexibility in the starting point of the phone. Non-parametric description of the mean trajectory appears in [19] and [25], but we suggest a different estimation procedure which seems to be more robust. This shall be discussed further in the next section. The idea of non-unique mean trajectory can be originated in the traditional mixture HMM. Kimball and Ostendorf [42], [49] described a model that consists of discrete mixtures of trajectories. That approach assists us enrich the model family we use, but as we can see from Figure 5.1, it does not always reflect the actual behavior of the data.

5.3 Recognition and Training Algorithms

We now present recognition and training algorithms for the new proposed model. The input to the recognition algorithm is a segment realization. The output of the algorithm is the identity of the segment. The optimal maximum likelihood (ML) solution to this problem is to determine on the segment identity $\hat{p}$, based on the likelihood of the segment data x, i.e.

$$\hat{p} = \arg\max_{p \in \mathcal{P}} f(x) \tag{5.2}$$

where $\mathcal{P}$ is the set of candidate segments, and $f(x)$ is the density of x under the assumption that the segment identity is p. $f(x)$ is given by :

$$f(x) = \int_a \sum_s f(x, s, a) da$$

$$\text{where} \quad f(x, s, a) = f(s) f(a) \prod_t f(x_t - a | \mu(s_t), \sigma(s_t))$$

$$\text{denote also} \quad f(x, s) = \int_a f(x, s, a) da$$

A shortcoming of the ML approach is that the computation of $f(x)$ is extremely complicated. There is no efficient algorithm to obtain the integral as a closed from. This is in contrast with the parametric random model discussed in the previous chapter. In the parametric model a distinct shift value is attached to each state, and therefore there is no summation over all the state sequences inside the integral. The model presented in this chapter can be viewed as a random parametric model such that the shift operation is done jointly for several states.

As an alternative we propose the following segment recognition criterion :

$$\hat{p} = \arg\max_{p \in \mathcal{P}} \left\{ \max_s f(x, s) \right\} \tag{5.3}$$

The approximation of (5.2) by (5.3) is similar to the standard approximation of ML word estimation by ML sequence estimation (Viterbi decoding). Even for this simplified criterion there is no efficient algorithm to compute it. In order to solve this maximization problem we must explicitly check all the exponential amount of state sequences. In the next section we give an efficient algorithm, for finding the exact solution, for a simpler version of our model. We now present an iterative algorithm to evaluate $\max_s f(x,s) = f(x,\hat{s})$ numerically.

1. Initialization : $\hat{a} = 0$.

2. Compute $\hat{s} = \arg\max_s f(x,s,\hat{a})$ by applying standard Viterbi segmentation on the data after displacement elimination (i.e., $x_1 - \hat{a}, x_2 - \hat{a}, \ldots, x_n - \hat{a}$).

3. Compute $\hat{a} = \arg\max_a f(x,\hat{s},a)$. In Appendix 5.A we obtain the following expression for $\hat{a}$.
$$\hat{a} = \frac{\sum_{t=1}^{n} \frac{1}{\sigma^2(\hat{s}_t)} \left(x_t - \mu(\hat{s}_t)\right)}{\frac{1}{\sigma_a^2} + \sum_{t=1}^{n} \frac{1}{\sigma^2(\hat{s}_t)}} \tag{5.4}$$

4. Repeat 2 and 3 until convergence.

5. Compute :

$$f(x,\hat{s}) = \int_a f(x,\hat{s},a)da = \left(\frac{1}{\sqrt{2\pi}}\right)^n \frac{1}{\sigma_a \prod_{t=1}^{n} \sigma(\hat{s}_t)} \left(\frac{1}{\sigma_a^2} + \sum_{t=1}^{n} \frac{1}{\sigma^2(\hat{s}_t)}\right)^{-\frac{1}{2}} e^{-\frac{1}{2}k(x,\hat{s})} \tag{5.5}$$

where

$$k(x,s) = \sum_{t=1}^{n} \frac{1}{\sigma^2(s_t)} \left(x_t - \mu(s_t)\right)^2 - \frac{\left(\sum_{t=1}^{n} \frac{1}{\sigma^2(s_t)} \left(x_t - \mu(s_t)\right)\right)^2}{\frac{1}{\sigma_a^2} + \sum_{t=1}^{n} \frac{1}{\sigma^2(s_t)}}$$

Eq. (5.5) is derived in Appendix 5.A.

We have compared the performance of this scheme with the true solution obtained by performing exhaustive search. These tests ensure us that this heuristic algorithm indeed achieves the global maximum most of the time.

The proposed training algorithm is a combination of the algorithm above and the well-known Baum-Welch training procedure. Given a sequence of k segment data realizations $x_1, \ldots, x_k$, where $x_i = (x_{i,1}, \ldots, x_{i,n_i})$, denote by a_i, the segmental mixture coefficient of x_i. Training consists of the following iterative steps :

1. Initialization :

$$a_i = 0 \qquad i = 1, 2, \ldots, k \quad , \quad \sigma_a = \infty$$

2. Apply the Baum-Welch algorithm to

$$x_{i,1} - a_i, \ldots, x_{i,n_i} - a_i \qquad i = 1, \ldots, k$$

   in order to obtain a new set of segment template parameters (state means, variances, and transition probabilities). We then apply the Viterbi algorithm in order to determine the state segmentation $\hat{s}_i$.

3. Apply the previous iterative algorithm to obtain $a_i = \arg\max_a f(x_i, \hat{s}_i, a)$.

4. Given $a_1, a_2, \ldots, a_k$, update the variance of the random displacement :

$$\sigma_a^2 = \frac{1}{k} \sum_{i=1}^{k} (a_i)^2$$

5. Repeat 2-4 until convergence.

A major decision that needs to be made concerns the number of states that are used in our model. While trajectory descriptions with large number of states are more accurate on the one hand, when a large number of states are used, the training algorithm needs to estimate a large number of parameters on the other.

Hence, in that case, it is essential to properly initialize the training algorithm that was described above. Otherwise, the algorithm does not produce meaningful results. This problem is avoided in Kimball *et al.* [42], [49], since a relatively small number of states (typically five states per segment) is used.

The following initialization algorithm is proposed.

1. Given the segment data realizations, $x_1, x_2, \ldots, x_k$, an initial segment template is determined. The length, M, of this template is set equal to the average length of the given segment realizations. Then each segment realization is linearly time warped to size M. Finally, the initial segment template is set to the mean of these linearly time warped segment realizations.

2. A dynamic time warping (DTW) [60] routine is used to time align each segment realization x_i against the template segment.

3. The time aligned segment realizations are averaged together in order to obtain a new template.

4. Stages 2 and 3 are repeated as many times as necessary. Typically two iterations are sufficient.

5. Finally, M vectors of means and variances of the HMM states, that constitute the initial template, are obtained by averaging the last version of time aligned segment data realizations.

Note that the initialization routine does not employ random displacement modeling. The DTW routine that we used specifies the local constraint that no more than two adjacent template frames can be mapped to the same segment realization frame, and vice versa (no more than two adjacent segment realization frames can be mapped to the same template frame). In addition, the DTW routine specifies the standard global constraint that the grid region of matching frames is limited to a band diagonal region. These constraints limit the amount of permitted time contracting and expanding. Standard Viterbi decoding does not incorporate such

constraints, and thus does not produce reliable initialization. The method presented here ensures that we will get a relatively smooth and continuous curve for the trajectory. Other methods to estimate the mean trajectory appear in the literature. Ghitza [19] suggested choosing one of the realizations, which is the centroid of the segments ensemble, as the mean trajectory. Goldenthal [25] used the first step of our initialization i.e. averaging the segments without warping in order to find a trajectory model.

The recognition and training algorithms that were described above are useful for re-scoring an N-best list. We assume the phone boundaries are known. Note that due to the fact that mean trajectory time warping is allowed, segmentation inaccuracies at the previous stage can be tolerated.

Figure 5.4 presents the mean trajectory of the first cepstral coefficient of the triphone ih-s-ow produced by the training algorithm. The figure also shows the variance trajectory (we draw a line that is one standard deviation away from the mean trajectory).

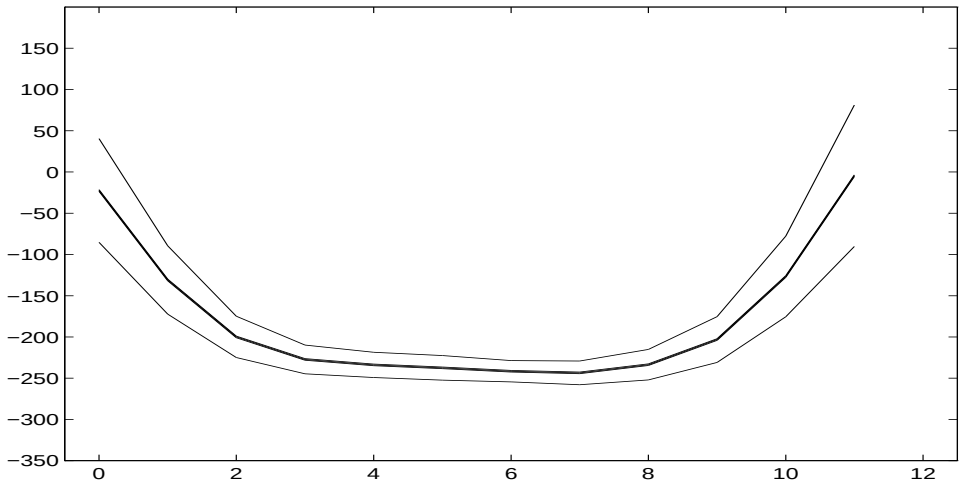

Figure 5.4: Non-parametric model of the mean trajectory.

5.4 A Simplified Version of the Model

In the previous section we have presented the segmental model :

$$x_t = \mu(s_t) + a + \epsilon_t(\sigma(s_t))$$

During the recognition stage we already know the model parameters and want to classify a given segment realization. Given a frame sequence $x = (x_1, ..., x_n)$ we want to compute the following Viterbi-like approximation for the likelihood score :

$$f(x, \hat{s}) = \max_s f(x, s) = \max_s \int_a f(x, s, a) da$$

$f(x, \hat{s})$ is the joint density of the data x and the best suited state sequence $\hat{s}$. We have suggested a heuristic iterative procedure to find $f(x, \hat{s})$. A natural question in this situation is the following. Is this procedure the best available? As a partial answer we shall discuss a very restricted version of the model. We will show that in this case we do have an exact algorithm to compute $f(x, \hat{s})$ which is neither heuristic nor iterative. In spite of this fact, even for this simplified model the solution is not straightforward.

We add the following assumptions to our framework :

1. The data frames are one-dimensional. In other words, we model each feature separately. The states are chosen independently for each feature. In the general case discussed at the previous sections we have enabled a common state segmentation for all the features.

2. There is no Markovian structure imposed on the state sequence. We assume instead IID behavior. In other words, the model is reduced to the standard mixture model.

3. The normal distributions attached to each of the states share a common vari-

ance denoted by σ^2.

The density function of a frame sequence according to this model is :

$$f(x) = \int_a \frac{1}{\sqrt{2\pi}\sigma_a} \exp(-\frac{a^2}{2\sigma_a^2}) \sum_s \prod_t \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2\sigma^2}(x_t - \mu(s_t) - a)^2\right) da$$

As before, we want to find $\max_s f(x,s)$. The joint density of x, s and a can be written as :

$$f(x,s,a) = (\frac{1}{\sqrt{2\pi}\sigma})^n \frac{1}{\sqrt{2\pi}\sigma_a} \exp(-\frac{1}{2} g(x,s,a))$$

$$\text{where} \quad g(x,s,a) = \frac{a^2}{\sigma_a^2} + \frac{1}{\sigma^2} \sum_t (x_t - \mu(s_t) - a)^2$$

It can be verified from Appendix 5.A that :

$$g(x,s,a) = \left(\frac{1}{\sigma_a^2} + \frac{n}{\sigma^2}\right)(a - \hat{a}_s)^2 + g(x,s,\hat{a}_s)$$

$$\text{where} \quad \hat{a}_s = \max_a f(x,s,a) = \frac{\frac{1}{\sigma^2}\sum_t (x_t - \mu(s_t))}{\frac{1}{\sigma_a^2} + \frac{n}{\sigma^2}}$$

From this we can conclude that :

$$f(x,s) = \int_a f(x,s,a) da = \left(\frac{1}{\sigma_a^2} + \frac{n}{\sigma^2}\right)^{-\frac{1}{2}} \sqrt{2\pi} f(x,s,\hat{a}_s)$$

This implies :

$$\begin{aligned} \arg\max_s f(x,s) &= \arg\max_s f(x,s,\hat{a}_s) = \arg\min_{s,a} g(x,s,a) \\ &= \arg\min_{s,a} \left(\frac{a^2}{\sigma_a^2} + \frac{1}{\sigma^2}\sum_t (x_t - \mu(s_t) - a)^2\right) \end{aligned}$$

In other words, in order to find the best state sequence $\hat{s}$, we do not have to perform the integral over the shift variable a. Instead, we can take for each s the best

associated shift :

$$\arg\max_s f(x,s) = \arg\max_s(\max_a f(x,s,a))$$

We can now reduce our problem to the following abstract formulation. Let $x_1, ..., x_n, \mu_1, ..., \mu_k, w$ be known real numbers. For each "state sequence" $s \in \{1, ..., J\}^n$ and each "shift" $a \in R$ define :

$$g(s,a) = wa^2 + \sum_{t=1}^{n}(x_t - \mu(s_t) - a)^2$$

We want to solve efficiently the minimization problem $\min_{s,a} g(s,a)$. We can give a probabilistic interpretation for this situation. Define a discrete random variable Y_s that accepts the n+1 values $0, x_1 - \mu(s_1), ..., x_n - \mu(s_n)$ with probabilities $\frac{w}{w+n}, \frac{1}{w+n} ... \frac{1}{w+n}$. Using this terminology one can easily observe that :

$$g(s,a) = (w+n)E(Y_s - a)^2$$

Denote :

$$a_s = E(Y_s) = \frac{1}{w+n}\sum_t (x_t - \mu(s_t))$$

The variance is the smallest second moment. It satisfies the following inequality for each real number a :

$$Var(Y_s) = E(Y_s - E(Y_s))^2 \leq E(Y_s - a)^2$$

Therefore we obtain :

$$g(s) \quad = \quad g(s,a_s) = \min_a g(s,a)$$

$$= \quad (w+n)Var(Y_s) = (w+n)(E(Y_s^2) - E^2 Y_s)$$

$$= \quad \sum_{t=1}^{n}(x_t - \mu(s_t))^2 - \frac{1}{w+n}\left(\sum_{t=1}^{n} x_t - \mu(s_t)\right)^2$$

Thus far we have shown that for each state sequence s we can easily find the optimal shift associated with it. We shall now explain how we can find the best state sequence.

Define :

$$\begin{aligned} A(t,i) = (x_t - \mu(i))^2 \quad &, \quad B(t,i) = x_t - \mu(i) \\ A_s = \sum_t A(t,s_t) \quad &, \quad B_s = \sum_t B(t,s_t) \end{aligned} \tag{5.6}$$

Using this notation we can write :

$$g(s) = A_s - \frac{1}{w+n}B_s^2$$

We can project each state sequence s onto a point $(A_s, B_s) \in R^2$. Let S be the convex hull of $\{(A_s, B_s)\}$. $A - \frac{1}{w+n}B^2$ is a concave function of A and B, therefore it obtains its minimum over the region S at the extreme points (which are a subset of $\{(A_s, B_s)\}$. From here we can conclude that if we want to find $\min_s g(s)$ there is no need to check all the state sequences. We need only check those state sequences which are projected to extreme points of S.

It will be explained now how we can efficiently check all the extreme points of S. We use a standard technique from computational geometry. An extreme point (A, B) is characterized by some real λ, such that :

$$A + \lambda B = \max_s(A_s + \lambda B_s) \quad \text{or} \quad -A + \lambda B = \max_s(-A_s + \lambda B_s)$$

From definition (5.6) we obtain

$$\max_s(A_s + \lambda B_s) = \max_s \sum_t (A(t, s_t) + \lambda B(t, s_t))$$

We can solve this maximization problem component wise due to the fact that we assume no Markovian structure on the state sequences s, i.e there are no prior restrictions on the state sequence. We need only solve a trivial maximization problem. The solution is :

$$\max_s(A_s + \lambda B_s) = \sum_t \max_j (A(t, j) + \lambda B(t, j))$$

In this manner, for each $\lambda \in R$ we find a point s_λ which is an extreme point of S and it is, therefore, a potential candidate for optimality. We cannot check each real number λ, but there is no need to do so. We can concentrate only on critical points λ where s changes, and search for the minimum point among them. The critical points are the solutions to the equations :

$$\pm A(t, i) + \lambda B(t, i) = \pm A(t, j) + \lambda B(t, j) \qquad 1 \leq t \leq n \quad , \quad 1 \leq i < j \leq k \quad (5.7)$$

Solving equation (5.7) we get the following λ values :

$$\pm(2x_t - \mu(i) - \mu(j)) \quad , \quad \pm\frac{(x_t - \mu(i))^2 + (x_t - \mu(j))^2}{\mu(i) - \mu(j)} \qquad (5.8)$$

This step concludes the presentation of the algorithm to compute $\min_{s,a} g(s, a)$.

We summarize the algorithm to find $\min_{s,a} g(s, a)$:

1. Compute the set of critical points using expression (5.8).

2. The critical points divide the real line into disjoint intervals. At each interval

choose a number λ. Find the state sequence s characterized by λ :

$$s_t = \arg\max_j (A(t,j) + \lambda B(t,j)) \qquad t = 1, ..., n$$

3. For each state sequence s that was found at the previous step compute explicitly

$$g(s) = \sum_{t=1}^{n} (x_t - \mu(s_t))^2 - \frac{1}{w+n} \left(\sum_{t=1}^{n} x_t - \mu(s_t) \right)^2$$

and choose the state sequence s where g accepts its minimum.

When the length of the data sequence is over 10, this algorithm is much more efficient than the exhaustive search over all the state sequences. A detailed analysis of an algorithm similar to the one presented in this section can be found in [26].

We can relax the first assumption mentioned at the beginning of the section. Suppose that the data frames are K-dimensional and the covariance matrices are diagonal. In a way similar to definition (5.6), we can define :

$$\begin{aligned} A_k(t,i) = (x_{t,k} - \mu_k(i))^2 \quad &, \quad B_k(t,i) = x_{t,k} - \mu_k(i) \\ A_s = \sum_{k=1}^{K} \sum_t A_k(t, s_t) \quad &, \quad B_{s,k} = \sum_t B_k(t, s_t) \end{aligned}$$

Using this definition we obtain :

$$\begin{aligned} g(s) &= \sum_k \left(\sum_{t=1}^{n} (x_{t,k} - \mu_k(s_t))^2 - \frac{1}{w+n} \left(\sum_{t=1}^{n} x_{t,k} - \mu_k(s_t) \right)^2 \right) \\ &= A_s - \frac{1}{w+n} \sum_k B_{s,k}^2 \end{aligned}$$

This is a function of more than two variables and therefore it is much more complicated to find its maximum.

The third assumption can be relaxed too. In case where each state has a different

variance associated with it, the resulting function is of the form $A-\frac{B^2}{C}$. Although this function is not concave, it is quasi-concave. Therefore, it still obtains its minimum at the extreme points of a convex set. This convex set is now a subset of a 3-dimensional space.

5.5 Experimental Results

We evaluated the model presented in the previous section using the (WSJ) corpus [15]. Experiments were conducted with the DECIPHER [14]. A detailed description of the WSJ data-base and the signal processing performed in the DECIPHER system can be found in Appendix A.

Our model was implemented using the N-best re-scoring paradigm, by re-scoring the list of the N-best sentence hypotheses generated by the DECIPHER. Context dependent phonetic models were used. A segmental model was constructed for each triphone that appears in the training data set. The test set consisted of 200 sentences. In Table 5.1 we present the word error rate of standard HMM, as implemented in the DECIPHER, and the word error rate after re-scoring the N-best list using the segmental model. In that case, language modeling was not incorporated. In Table 5.2 we show the decrease in the word error when we add the segmental model to the HMM as another knowledge source, and linearly combine the two scores. In that case, language modeling was incorporated.

Tables 5.1 and 5.2 show that the new model is comparable to state of the art HMM system, with sophisticated tying of parameters. To probe the new model further and to compare it to alternative models, we carried out several triphone recognition experiments. Context dependent phonetic units were chosen since in that case there is less variability between utterances. Hence, in practice, this is usually the case of interest.

In Table 5.3 we present recognition results for some frequently occurring triphone contexts. The first data row indicates the number of triphone occurrences for each

model	word error
HMM acoustics	22.1
segmental acoustics	21.4

Table 5.1: Word error rate results without language model.

model	word error
HMM acoustics + linguistics	8.1
HMM acoustics + segmental acoustics + linguistics	7.8

Table 5.2: Word error rate results with language model.

context. Half of the occurrences were used to train each model. The other half were used to test the models. There were six triphones in the first context (s[k]ih, s[l]ih, s[m]ih, s[p]ih, s[t]ih and s[w]ih), five triphones in the second context (n[ay]t, n[eh]t, n[ey]t, n[ih]t and n[ow]t), five triphones in the third context (aa[k]t, aa[n]t, aa[p]t, aa[r]t and aa[s]t), ten triphones in the forth context (ih[b]eh, ih[d]eh, ih[f]eh, ih[jh]eh, ih[l]eh, ih[m]eh, ih[p]eh, ih[r]eh, ih[s]eh and ih[v]eh), and seven triphones in the fifth context (g[aa]t, g[ae]t, g[ah]t, g[ax]t, g[eh]t, g[ey]t and g[ih]t).

The models examined were :

1. Mixture of Gaussians HMM. Such model, with s states and m mixtures is denoted by HMM(s,m).

2. A segmental polynomial model (3.2) with deterministic μ_k parameters. Such model with s states and a polynomial of order K describing the mean trajectory of each state is denoted by POLY(s,K).

3. A segmental polynomial model (3.2) with multi-normal μ_k parameters. Such model with s states and a polynomial of order K describing the mean trajectory of each state is denoted by POLYRND(s,K).

4. The new proposed model with random displacement modeling. Such model with s states is denoted by NPRMDISP(s).

5. The new proposed model without random displacement modeling, i.e. a standard non-parametric model. Such model with s states is denoted by NPRM(s).

To implement model (3.2) (both for the case where μ_k are deterministic parameters, and for the case where they are random variables), all possible state partitions were considered for each utterance that needs to be recognized.

	s[·]ih	n[·]t	aa[·]t	ih[·]eh	g[·]t
#	1088	740	2263	1619	662
HMM(3,3)	90.7	85.2	96.6	89.3	64.1
POLY(3,2)	89.0	82.7	95.9	87.5	66.8
POLYRND(3,1)	89.6	79.2	96.3	87.4	64.4
NPRM(9)	90.7	78.7	94.5	89.9	58.7
NPRMDISP(9)	91.6	85.4	96.5	87.9	67.1

Table 5.3: Triphone recognition rate results.

As can be seen, in four out of the five contexts presented, global random displacement, non-parametric modeling (NPRMDISP) is preferable to standard non-parametric segmental modeling (NPRM). The new model also compares favorably with the other models that were examined.

The experiments summarized in Table 5.3 were repeated for many other frequently occurring triphone contexts. For most triphone contexts examined, random displacement modeling improved the standard non-parametric model. Nevertheless, in many other cases, random displacement modeling decreased the recognition rate. Hence, for some of the triphones, a standard non-parametric model (i.e., a degenerated displacement model that employs fixed zero displacement) is expected to be preferable. On the other hand, we observed that a random displacement model always assigns higher likelihood values to previously unseen data, and hence has an improved prediction capability. The maximum likelihood criterion can not , therefore, be used in order to determine when the random displacement model should be degenerated.

5.6 Conclusions

We presented a new model, that is a continuous mixture of segment trajectories. This model is composed of two elements. The first element is a non-parametric representation of the mean and variance trajectories, and the second is some parameterized transformation of the trajectory that is global to the entire segment. This transformation adapts the general model to a specific segment realization, and may for example account for different speech styles. We then focused on a particular transformation that applies a random displacement to the mean trajectory. The model was compared to alternative segment models on a triphone recognition task. The model improves segment modeling, in the sense that it improves the prediction of previously unseen data. Our triphone recognition experiments show benefit to the new model for most of the contexts examined, compared to a standard non-parametric model without global displacement modeling.

Several avenues of future research seem appropriate. First, other global trajectory transformations need to be examined. One possibility is to consider transformations that control the sharpness of peaks and valleys of the mean trajectory. In that case, the model accounts for segment utterances with varying degrees of smoothness.

Second, we have seen that a global, random displacement transformation always improves the ability of a non-parametric model to predict previously unseen data. However, the new model was not always superior in the triphone recognition experiments. The maximum likelihood criterion cannot be used in order to determine for which triphones random displacement modeling should degenerate to a fixed zero displacement. Other criteria need to be investigated in order to successfully implement a combined model, for which some of the triphones employ such degenerated transformation.

Appendix 5.A

We derive Eqs. (5.4) and (5.5). Denote by $f(x,s,a)$ the joint density function of the data x, the states sequence s, and the shift a. Similarly $f(x,s)$ is the joint density of x and s. We first derive the optimal shift, denoted by $\hat{a}$, given x and s.

$$f(x,s,a) = \frac{1}{\sqrt{2\pi}\sigma_a}\exp\left\{-\frac{a^2}{2\sigma_a^2}\right\}\prod_{t=1}^{n}\frac{1}{\sqrt{2\pi}\sigma(s_t)}\exp\left\{-\frac{(x_t-a-\mu(s_t))^2}{2\sigma^2(s_t)}\right\}$$

$$\frac{\partial \log f(x,s,a)}{\partial a} = -\frac{a}{\sigma_a^2} + \sum_{t=1}^{n}\frac{1}{\sigma^2(s_t)}\left(x_t - a - \mu(s_t)\right) = 0$$

Therefore the maximum likelihood shift, given x and s, is :

$$\hat{a}_s = \arg\max_a f(x,s,a) = \frac{\sum_{t=1}^{n}\frac{1}{\sigma^2(s_t)}\left(x_t-\mu(s_t)\right)}{\frac{1}{\sigma_a^2}+\sum_{t=1}^{n}\frac{1}{\sigma^2(s_t)}}$$

Next, we derive a closed form expression for $f(x,s)$.

$$\begin{aligned} f(x,s) &= \int_a f(x,s,a)da \\ &= \left(\frac{1}{\sqrt{2\pi}}\right)^{(n+1)}\frac{1}{\sigma_a\prod_{t=1}^{n}\sigma(s_t)}\int_a \exp\left\{-\frac{1}{2}g(x,s,a)\right\}da \end{aligned}$$

$$\text{where} \qquad g(x,s,a) = \frac{a^2}{\sigma_a^2} + \sum_{t=1}^{n}\frac{1}{\sigma^2(s_t)}\left(x_t-a-\mu(s_t)\right)^2$$

$g(x,s,a)$ can be written as :

$$\begin{aligned} g(x,s,a) &= a^2\left(\frac{1}{\sigma_a^2}+\sum_{t=1}^{n}\frac{1}{\sigma^2(s_t)}\right) - 2a\sum_{t=1}^{n}\frac{1}{\sigma^2(s_t)}\left(x_t-\mu(s_t)\right) + \\ &\quad \sum_{t=1}^{n}\frac{1}{\sigma^2(s_t)}(x_t-\mu(s_t))^2 \\ &= \left(\frac{1}{\sigma_a^2}+\sum_{t=1}^{n}\frac{1}{\sigma^2(s_t)}\right)\left(a-\frac{\sum_{t=1}^{n}\frac{1}{\sigma^2(s_t)}\left(x_t-\mu(s_t)\right)}{\frac{1}{\sigma_a^2}+\sum_{t=1}^{n}\frac{1}{\sigma^2(s_t)}}\right)^2 + \end{aligned}$$

$$\begin{aligned} & \sum_{t=1}^{n} \frac{1}{\sigma^2(s_t)} \left(x_t - \mu(s_t)\right)^2 - \frac{\left(\sum_{t=1}^{n} \frac{1}{\sigma^2(s_t)} \left(x_t - \mu(s_t)\right)\right)^2}{\frac{1}{\sigma_a^2} + \sum_{t=1}^{n} \frac{1}{\sigma^2(s_t)}} \\ = & \left(\frac{1}{\sigma_a^2} + \sum_{t=1}^{n} \frac{1}{\sigma^2(s_t)}\right) (a - \hat{a}_s)^2 + g(x, s, \hat{a}_s) \end{aligned}$$

where

$$g(x, s, \hat{a}_s) = \sum_{t=1}^{n} \frac{1}{\sigma^2(s_t)} \left(x_t - \mu(s_t)\right)^2 - \frac{\left(\sum_{t=1}^{n} \frac{1}{\sigma^2(s_t)} \left(x_t - \mu(s_t)\right)\right)^2}{\frac{1}{\sigma_a^2} + \sum_{t=1}^{n} \frac{1}{\sigma^2(s_t)}}$$

Hence,

$$\begin{aligned} f(x, s) &= \left(\frac{1}{\sigma_a^2} + \sum_{t=1}^{n} \frac{1}{\sigma^2(s_t)}\right)^{-\frac{1}{2}} \sqrt{2\pi} f(x, s, \hat{a}_s) \\ &= \left(\frac{1}{\sqrt{2\pi}}\right)^n \frac{1}{\sigma_a \prod_{t=1}^{n} \sigma(s_t)} \left(\frac{1}{\sigma_a^2} + \sum_{t=1}^{n} \frac{1}{\sigma^2(s_t)}\right)^{-\frac{1}{2}} e^{-\frac{1}{2} g(x, s, \hat{a}_s)} \end{aligned}$$

Chapter 6

Synthetic Segmental Triphone Models

Acoustic models which are context dependent can improve the performance of large vocabulary speech recognition systems. Due to insufficient examples, however, it is not possible to train a distinct model for each triphone context. In this chapter we suggest a method of segmental model sharing based on phonetic knowledge of possible similar triphones. A synthetic description of the triphone mean trajectory is constructed from simpler elements that are related to the phones composing the triphone. Experimental results on phone classification task show that the suggested synthetic modeling significantly outperforms context independent models.

6.1 Introduction

Small vocabulary speech recognition systems are based on modeling of words which are the basic units of language. Many examples of each word are needed for robust modeling. This demand is not feasible when the size of the vocabulary is large.

In this case there is a need to use smaller phonetic units in order to allow data sharing across words. Most attempts of sub-word modeling are based on phones. A phone is the acoustic realization of a phoneme, and each word can be presented as a concatenated sequence of phonemes. The drawback of phone modeling is that the acoustic realization of a word is far from being simply a concatenated sequence of phones. Speech is a product of the vocal tract which is a continuous physical system. There is no sudden transition of the vocal articulator from one phone to the next one. Instead, there is a smooth and moderate flow between consecutive phones. This effect, known by the name co-articulation, implies that phone models that take the context of the phone realization into consideration, are more consistent. The coarticulation effect is demonstrated in Figure 6.1. Each curve was produced by averaging trajectories of the first cepstral coefficient of the phoneme 'l' in a particular triphone context. The averaging was done along with non-linear warping. The length of the curve is the average length of the phone 'l' in that context. The data was extracted from the WSJ data-base that was segmented by SRI's DECIPHER system into phone boundaries. A detailed description of the procedure that was employed to produce the figure can be found in chapter 5. From Figure 6.1 it can be seen that the phones before and after the current phone have a remarked influence on both the duration and the shape of the trajectory of the current phone. Figure 6.1 demonstrates the loss of information occurring when the context is not taken into consideration.

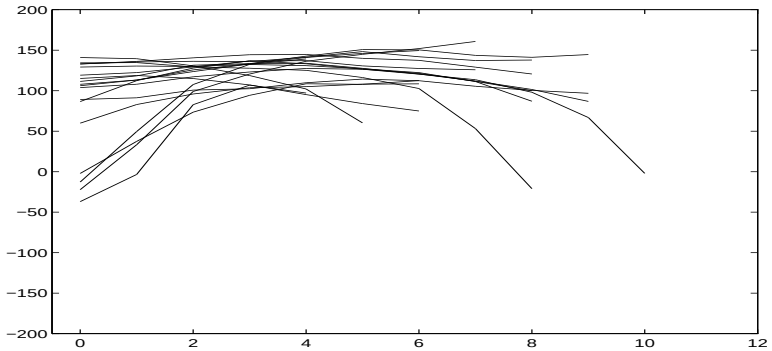

Figure 6.1: Mean trajectories of the phone 'l' for different triphone contexts.

The most commonly used context in acoustic modeling is a triphone which is a phone with its left and right phone contexts. Denote the phone b appearing after the phone a and before the phone c by $a-b-c$. Triphones suffer from the same modeling problem as do words. There are thousands of triphones in English and, therefore, there is a training problem caused by insufficient data. One solution is to back-off to biphone or monophone models in a rarely used triphone. However, the most popular way to increase the trainability of a triphone system is to share models or part of models between different triphones. Lee [46] proposed a sharing method based on merging triphone contexts that show acoustic similarity into a generalized triphone. The acoustic models are associated with the these generalized triphones. Hwang and Huang [33] [34] refined this method by introducing the term *senone* to denote a state in phonetic HMM considered as a basic subphonetic unit. They suggested a sharing between triphones in the senonic level. Digalakis et al. [11] suggested a sharing method for mixture of Gaussians HMM. They proposed that triphone HMM states that show acoustic similarity will share the same codebook of mixture components. In the studies reviewed here, the acoustic data related to each triphone was used to find triphone models (or part of models) which show a similar acoustic behavior in order to share some of the model parameters across the triphones. In this approach there is a modeling problem in triphones that appear only a few times in the training set or do not appear at all. In these cases acoustic information, if it exists, is misleading. The same problem of sharing triphone models exists in segmental models. In this chapter we concentrate on a segment model that describes in a non-parametric manner the triphone mean trajectory along the feature space. We shall show how phonetic information can be used to build a synthetic description of the triphone mean trajectory.

6.2 Synthetic Modeling of the Mean Trajectory

Relying on phonetic knowledge to determine similarity between triphones is usually preferable to acoustic knowledge gained from the training data. For example, there is no need to refer to the acoustic data to conclude that the last parts of the triphones $a-b-c$ and $d-b-c$ have the same acoustic structure. The phonetic approach was successfully applied to the triphone sharing problem by Young and Odell [48]. They used phonetic decision trees to check possible triphone state tyings. The decision itself on the preferred clustering was made according to information conveyed in the acoustic data. A phonetic decision tree for predicting unseen triphones was implemented in CMU's SPHINX system [35].

Here we take the phonetic approach one step further. Instead of phonetic analysis of the similarity between triphone models, we try to obtain a phonetic synthesis of a triphone. This is done by using the phonetic elements composing the triphone in order to build a synthetic triphone model. Goldenthal [25] proposed merging biphone tracks to create a triphone based synthetic model. Our method is motivated by the physical explanation for the coarticulation effect. After the vocal articulators have reached their target position for the current phone, there is a movement towards the position needed to articulate the next phone. This phenomenon is demonstrated in Figure 6.2. In this figure we present mean trajectories of the first cepstral coefficient of several triphone contexts of the phone 'ao' (the phone appears in 'bought'). The trajectory of each triphone was computed from averaging triphone realizations taken from the WSJ corpus. The length of each curve is the average length of the related triphone. In four of the triphones the next phone is 's' and in the rest of them the next phone is 'r'. It can be clearly seen how the next phone pulls the last part of the mean trajectory towards its starting articulation position.

We choose a modeling method that can easily incorporate the phonetic synthesis. The model associated with each phone is a non-parametric description of the mean trajectory. Each phone was modeled by a Gaussian HMM with n states. The HMM topology is a left to right, such that skipping over states is allowed. For simplicity

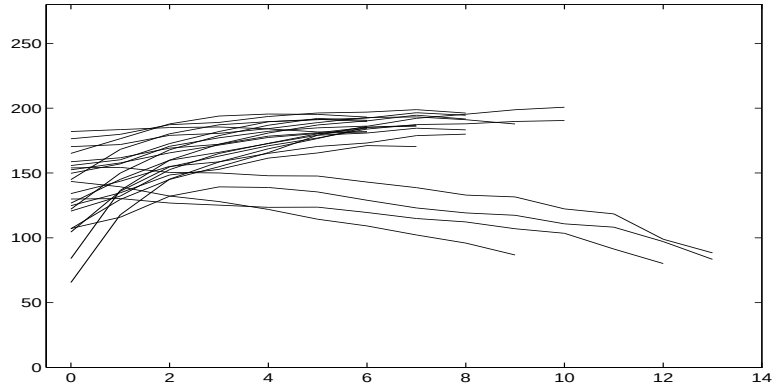

Figure 6.2: Mean trajectories of the phone 'ao' in various triphone contexts.

it is assumed that the acoustic data are scalars. Denote the Gaussian distributions associated with the HMM states by

$$N(\mu_i, \sigma_i^2) \qquad i = 1, ..., n$$

The sequence $\mu_1, \mu_2 ...$ can be considered as the mean trajectory of the phone along the feature space. The curves in Figure 6.2 are examples of modeling the mean trajectory. The model can be trained by using the well known Baum-Welch re-estimation procedure. The model claims to be a description of the mean trajectory and, therefore, a reasonable decision for the number of states is the average length of the phonetic unit. The assumption that the number of the HMM states is in the order of the number of frames can cause problems in initialization of the training algorithm. An initialization method for a phonetic model that describes the mean trajectory along the feature space is described in chapter 5.

Our target is to adapt this general phone model to a specific triphone context. In order to quantify the intuition presented above, regarding the influence of the context, we estimate the feature value at the transition point for each pair of consecutive phones. Denote the value at the transition point from phone a to phone b by $G(a, b)$. The synthetic triphone model for $a{-}b{-}c$ will be composed from $G(a, b)$,

$G(b,c)$ and the mean trajectory model for the phone b denoted by $\mu_i(b), i = 1, ..., n$. The context independent mean trajectory of the phone is adapted to the context in the following way :

$$\mu_i(a-b-c) = \begin{cases} \frac{n-2i}{n} G(a,b) + \frac{2i}{n}\mu_i(b) & i \leq \frac{n}{2} \\ \\ \frac{2i-n}{n} G(b,c) + \frac{2n-2i}{n}\mu_i(b) & i > \frac{n}{2} \end{cases}$$

The synthetic model, therefore, is based on the monophone model. This model is linearly warped along the feature axis such that the mean trajectory will begin at the transition point from the previous phone and end at the transition point to the next phone. This construction is demonstrated in Figures 6.3 and 6.4. Figure 6.3 shows the mean trajectory of the first cepstral coefficient of the triphone $s-ao-r$. Figure 6.4 shows the mean trajectory of the first cepstral coefficient of the phone 'ao' averaged over all contexts. The two short horizontal lines in Figure 6.4 are the average values of the first cepstrum at the transition from 's' to 'ao' and from 'so' to 'r'. The sloped curve in Figure 6.4 is the synthetic triphone trajectory. It can be seen from Figures 6.3 and 6.4 that the synthetic triphone model better approximates the exact triphone model than does the context independent phone model.

6.3 Experimental Results

We evaluated our model using the WSJ corpus. Experiments were conducted with DECIPHER system which was used for segmenting the data into phone boundaries (see Appendix A). The recognition task we performed is phone classification. For each phone, we have trained a Gaussian HMM with eight states. The training data includes 400 examples from each phone. In order to obtain the values of the function

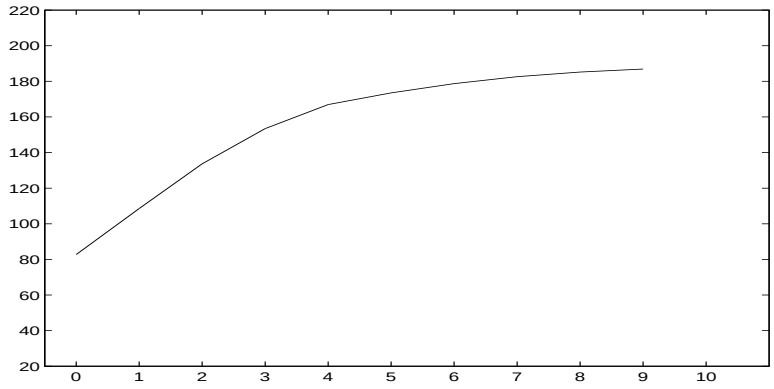

Figure 6.3: Mean trajectory of the triphone $s-ao-r$.

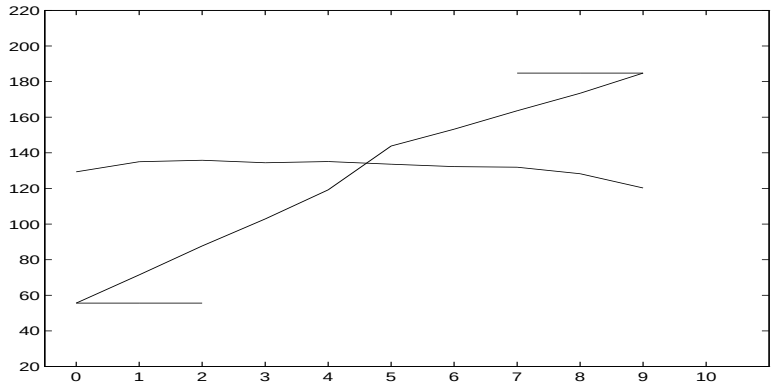

Figure 6.4: Synthetic construction of the mean trajectory of the triphone $s-ao-r$ from simpler elements.

$G(a,b)$, we estimated from the segmented data base the values of the features at the transition point from one phone to the next. The test set included an additional 400 examples in various contexts for each phone. Table 6.1 shows phone classification results of context independent models and synthetic triphone models built according to the method presented in the previous section.

model	recognition rate
phone models	56.1
synthetic triphone models	66.2

Table 6.1: Phone classification results.

Another experiment was performed to evaluate the quality of the approximated

triphone model as opposed to an exact triphone model that was trained from examples of a specific triphone. For each phone we have chosen one frequently used triphone context. We have trained a triphone model which consists of Gaussian HMM with eight states. The training set was composed of 400 examples of the triphone. The test set included another 400 examples of the triphone. The performance of the triphone models was compared to the performance of the two models trained in the previous experiment. Table 6.2 summarizes the results.

model	recognition rate
phone models	56.8
synthetic triphone models	67.3
triphone models	92.2

Table 6.2: Triphone classification results.

6.4 Conclusions

In this chapter we have presented a general framework for parameter sharing between triphone models. The method is based on constructing synthetic triphone models from simpler model elements related to the three phones composing the triphone. This method does not rely heavily on the acoustic realization of a given triphone. It can suggest models even for triphones that do not appear at all in the training set. This approach can, therefore, be applied in cases of small sized training sets.

The results presented in Table 6.1 indicate the potential of the synthetic approach. Table 6.2, however, indicates that there is still a room for improving the synthesis. Other decomposition of the triphone model into simpler elements can be considered. The acoustic behavior of triphones is much more consistent than phones, and, therefore, a triphone model is tighter. The context adaption may also include adaption of the variance of the Gaussian distributions. We have demonstrated the triphone synthesis approach using a simple segmental modeling method. There is

still a need to generalize the synthesis method in a manner that it can be applied to other models discussed in this thesis.

The importance of the method proposed in this chapter lies in the reasoning that a better performing synthesis will apply a better understanding of the mechanism that governs the acoustic implication of the coarticulation effect.

Chapter 7

Conclusions

This thesis has addressed a number of issues in segmental modeling. In particular it has investigated the question of incorporating into the acoustic model the temporal behavior of the speech signal along with effects that are global to the entire utterance. In this chapter we summarize the contribution of the thesis and suggest some directions for future work.

7.1 Thesis Contribution

In this thesis we first analyzed the gap existing between the actual behavior of the speech waveform and the assumptions of the HMM paradigm. The trajectory of the speech along the feature space is continuous, and there is most of the time smooth and moderate movement from one acoustic-phonetic event to the other. The HMM, however, assumes that there is no progress along time within the state boundaries. Instead, there is a sudden transition at the end of the state into the next one. It should be noted that the speech signal can be non-continuous. For example, in the release of a stop consonant such as 't', the movement of the tongue tip is continuous but the resulting acoustic pattern is not smooth and arguably not continuous.

The segmental modeling approach, which has been reviewed in chapter 3, tries to fill the aforementioned gap. In this study we have concentrated on random segmental models. In this approach the speech variability is due to two effects. First, the mean trajectory is considered a random process. This enables us to incorporate random effects, that are common to (at least) all the frames in the segment, into the acoustic model. The second random element models the local fluctuation from the mean trajectory as a white noise. It was reported by Holmes and Russell [30] that a parametric description of the random mean trajectory outperforms the deterministic model which, in the case of a static mean trajectory, coincides with the standard Gaussian HMM. In chapter 4 we analyzed the problems that exist in the random segmental model caused by the fact that different segment realizations of the same phone differ in length. This implies that the balance between the two random elements of the model changes as a function of the segment length. In recognition tasks where the phone boundaries are not known, it is important that the local score of the segment will be weighted according to segment length in order to avoid a tendency toward longer or shorter segments. We have shown the benefits of rescaling the model parameters according to segment length. On the rescaled model the variance of the random mean trajectory is inversely proportional to segment length. It is shown that, unlike previously suggested models, each frame contributes the same amount of information during the mean trajectory estimation process. A technical advantage of rescaling is the much simplified parameter estimation procedure. In a rescaled model there is no need to approximate the target function or to use an iterative procedure in order to find the ML estimate. Instead, a closed form can be obtained. Phonetic classification results support this approach. It was found that rescaling the model can improve recognition rate.

The parametric representation we have used in this thesis is actually a polynomial description of the mean trajectory. We have analyzed in detail the static and the linear cases. The acoustic trajectory during the articulation of a phonetic unit cannot always be well described by a linear function. The use of static or linear

functions results in a need for dividing the signal into small segments. This contradicts our attempt to find an acoustic model which refers to the complete phonetic event. To solve this problem, we have generalized in chapter 5 the concept of random segmental modeling to the case of non-deterministic representation of the mean trajectory. In this model there is no apriori assumption about the structure of the mean trajectory. Instead, it is represented by a synthetic template that describes the local dynamic within the segment. The random element associated with the model enables us to shift the template along the feature axis in order to achieve a better match with the data. The random non-parametric model can also be considered a random parametric model such that the same random shift is shared by a number of consecutive segments. We have proposed a heuristic procedure for evaluating the model parameters. We demonstrated the difficulty that exists with the parameter estimation by giving an exact solution for a very special case. It has been shown that even in that special case the exact solution for the problem of parameter estimation is not trivial. This difficulty however is theoretical. The proposed iterative procedure is a good solution for practical goals. The presentation of the random non-parametric approach is concluded with classification experiments that demonstrate the importance of modeling the local dynamical behavior.

The coarticulation effect approves the usage of triphone models. This can be accomplished by using model sharing across different contexts. Many sharing methods have been suggested for the standard HMM modeling. In order to achieve high recognition rate when using a segmental model, we must also construct in this case a model for each triphone context. We have concentrated on a segmental model which is based on a description of the mean trajectory along the feature space. We have presented a method for constructing a synthetic mean trajectory for each triphone context in order to enable model sharing across triphones. The results presented in Table 6.1 indicate the potential of the synthetic approach. The importance of the method proposed in this thesis lies in the reasoning that a better performing synthesis will apply a better understanding of the mechanism that governs the acoustic

implication of the coarticulation effect.

7.2 Future Work

The idea behind random segmental modeling is to distinguish between random effects which change every frame and other effects that last for a longer period which is at least the articulation of a basic phonetic unit. There are, however, effects such as speaker identity or recording equipment that are global to the entire sentence. In HMM and even in segmental models there is no mechanism to force joint acoustic decisions among consecutive segments. In order to solve this problem the segmental approach should be generalized beyond the segment level. The acoustic model may include a number of levels of random effects, such that higher levels will correspond to random effects that remain steady for a longer period.

We have considered in this thesis the issue of a smooth description of the mean trajectory which is implemented via a non-parametric representation. In the current model there is still a discontinuity at the transition point from one segment to the other. This transition point is artificial. In reality there is no distinct boundary between phonetic units. Hence, it is desired to extend the concept of a continuous description of the mean trajectory beyond the segment boundaries.

Many segmental models have been suggested in the past and in this thesis we have proposed several new ones. There are models that are adequate for vowels, other describe better transitions between phonemes etc. The information obtained from different models can be used to achieve a more accurate classification decision. This can be done either by using a simple linear combination of the contributions of each model or by a more complicated expert system that can automatically choose, according to the acoustic situation, a specific way for combining the models.

Today the HMM is still the most popular technique in speech recognition. In order to enable fair competition between HMM and segmental models there is still a need for further development in the segmental modeling area. For example, in order

to use triphone models, a segmental approach should include a method for parameter sharing across different contexts. Another drawback of segmental modeling is the high complexity needed for computing the likelihood score. A major advantage of HMM is its relative simplicity which has resulted in the development of fast decoding algorithms. Segment models can be used in real tasks only by rescoring the HMM decisions. Therefore there is a need for further research in order to find efficient algorithms to compute a satisfactory approximation of the segmental score.

Acoustic speech modeling can be implemented in two different ways. One option is a modeling approach that attempts to give a realistic description of how the speech waveform is actually generated. Another approach is to consider the model only as a probabilistic flexible mechanism that can adapt itself to different speech conditions. While HMM takes the second approach, segmental models pretend to approximate the mechanism of creating the acoustic sound. A target for further research is to try to develop models that attempt to describe the acoustic process and still have the flexibility of the HMM.

Appendix A

Corpus and Signal Processing

Our experiments on various types of segment models were conducted using the ARPA, large vocabulary, speaker independent, continuous speech, *Wall Street Journal* (WSJ) corpus [15]. The texts were drawn from articles appearing in the daily American financial newspaper. The database we used includes 18000 sentences from the 1994 version.

SRI's DECIPHER speech recognition system was used to produce the phone alignment of the speech waveform i.e. to determine the boundaries between the phones. The recognizer was configured with a front end that outputs a 39-dimensional vector. The first components of the vector consist of 12 cepstral coefficients and an energy term. The other components of the feature vector are the first and second time derivatives of the first 13 components. The cepstral coefficients were computed from an FFT filterbank. The acoustic modeling of the DECIPHER is based on continuous distribution, consisting of a mixture of Gaussian distributions, which models a triphone context. Parameter sharing is performed by using the same code-book (genone) of Gaussian distributions for similar triphones. A detailed description of the acoustical modeling component of the DECIPHER system can be found in [14].

Table A.1 presents the phone set we have used. The phones are written in

ARPABET notation.

phone	example	phone	example	phone	example
aa	father	em	bottom	ow	boat
ae	bat	en	button	oy	boy
ah	but	er	bird	p	pen
ao	bought	ey	bait	r	rose
aw	about	f	fun	s	sun
ax	about	g	green	sh	shine
axr	butter	hh	hat	t	ten
ay	bite	ih	bit	th	thick
b	bet	iy	beat	uh	book
br	bridge	jh	judge	uw	boot
ch	cheap	k	cat	v	very
d	dog	l	list	w	wet
dh	that	m	moon	y	yes
eh	bet	n	nut	z	zoo
el	bottle	ng	song	zh	measure

Table A.1: List of phone symbols and examples.

Bibliography

[1] S. Austin, J. Makhoul, R. Schwartz and G. Zavaliagkos, "Continuous speech recognition using segmental neural nets", *Proc. DARPA Workshop on Speech and Natural Language*, pp. 249-252, 1991.

[2] J. K. Baker, "Stochastic modeling as a means of automatic speech recognition", Ph.D Thesis, Computer science department, CMU, 1975.

[3] L. E. Baum, "An inequality and associated maximization technique in statistical estimation of probabilistic functions of Markov processes", *Inequalities* 3, pp. 1-8, 1972.

[4] P. F. Brown, "The acoustic modeling problem in automatic speech recognition", Ph.D Thesis, Computer science department, CMU, 1987.

[5] A. P. Dempster, N. M. Laird and D. B. Rubin, Maximum likelihood estimation from incomplete data, *Journal of the Royal Statistics Society*, vol 39, pp 1-38, 1977.

[6] L. Deng, "A generalized hidden Markov model with state conditioned trend functions of time of the speech signal", *Signal Processing*, vol 27, pp 65-78, 1992.

[7] L. Deng, M. Aksmanovic, D. Sun and J. Wu, "Speech recognition using hidden Markov models with polynomial regression functions as non stationary states", *IEEE Trans. Speech and Audio Processing*, vol. 2, pp. 507-520, 1994.

[8] L. Deng and C. Rathinavelu, "A Markov model containing state-conditioned second order non-stationary: application to speech recognition", *Computer Speech and Language*, vol. 9, pp. 63-86, 1995.

[9] L. Deng, M. Aksmanovic, "Speaker independent phonetic classification using hidden Markov models with state conditioned mixtures of trend functions", *IEEE Trans. Speech and Audio Processing*, vol 5, pp 319-324, 1997.

[10] L. Deng "A dynamic, feature-based approach to speech modeling and recognition", Proc. of the 1997 IEEE Workshop on Automatic Speech Recognition, pp 107-114, 1997.

[11] V. Digalakis, "Segment-based stochastic models of spectral dynamics for continuous speech recognition", Ph.D Thesis, Boston University, 1992.

[12] V. Digalakis, M. Ostendorf and J. R. Rohlicek, "A dynamical system approach to continuous speech recognition". *IEEE Trans. on Speech and Audio Processing*, vol 1, pp 431-442, 1993.

[13] V. Digalakis, J. R. Rohlicek and M. Ostendorf, "Fast search algorithms for phone classification and recognition using segment-based models", *IEEE Trans. Signal Processing*, vol. 40, pp. 2885-2896, 1992.

[14] V. Digalakis, P. Monaco and H. Murveit, "Genones: generalized mixture tying in continuous hidden Markov model-based speech recognizers", *IEEE Trans. Speech and Audio Processing*, vol. 4, pp. 281-289, 1996.

[15] G. Doddington, "CSR Corpus Development", *Proc. ARPA Workshop on Spoken Language Technology*, Feb. 1992.

[16] T. Fukada, Y. Sagisaka and K. Paliwal, "Model parameter estimation for mixture density polynomial segment models", *Proc. Int. Conf. Acoust., Speech and Signal Processing*, 1997.

[17] M. Gales and S. J. Young, "The theory of segmental hidden Markov models", Technical Report CUED/F-INFENG/TR 133, Cambridge, U.K., 1993.

[18] M. Gales and S. J. Young, "Segmental hidden Markov models", *Proc. Eurospeech*, pp. 1579-1582, 1995.

[19] O. Ghitza and M. Sondhi, "Hidden Markov models with templates as non-stationary states: an application to speech recognition", *Computer Speech and Language*, vol. 7, pp. 101-119, 1993.

[20] H. Gish and K. Ng, "A segmental speech model with applications to non-stationary states: an application to speech recognition", *Proc. Int. Conf. Acoust., Speech and Signal Processing*, pp. 447-450, 1993.

[21] J. Goldberger, D. Burshtein and H. Franco, "Segmental modeling using a continuous mixture of non-parametric models", *Proc. Eurospeech*, pp. 1195-1198, 1997.

[22] J. Goldberger, D. Burshtein and H. Franco, "Non-parametric random trajectory segmental models", *IEEE Trans. Speech and Audio Processing*, submitted.

[23] J. Goldberger and D. Burshtein, "Scaled random trajectory segmental models", *Computer Speech and Language*, submitted.

[24] W. D. Goldenthal and J. Glass, "Modeling spectral dynamics for vowel classification", *Proc. Eurospeech*, pp. 289-292, 1993.

[25] W. D. Goldenthal, "Statistical trajectory models for phonetic recognition", Ph.D thesis, MIT, 1994.

[26] R. Hassin and A. Tamir, "Maximizing classes of two-parameter objectives over matroids", *Mathematics of Operations Research*, vol. 14, pp. 362-375, 1989.

[27] W. Holmes and M. Russell, "Experimental evaluation of segmental HMMs", *Proc. Int. Conf. Acoust., Speech and Signal Processing*, pp. 536-539, 1995.

[28] W. Holmes and M. Russell, "Speech recognition using a linear dynamic segmental HMMs", *Proc. Eurospeech*, pp. 1611-1614, 1995.

[29] W. Holmes and M. Russell, "Modeling speech variability with segmental HMMs", *Proc. Int. Conf. Acoust., Speech and Signal Processing*, pp. 447-450, 1996.

[30] W. Holmes and M. Russell, "Linear dynamic segmental HMMS : Variability representation and training procedure", *Proc. Int. Conf. Acoust., Speech and Signal Processing*, pp. 1399-1402, 1997.

[31] X. Huang and M. Jack, "Semi-continuous Hidden Markov models for speech signals", *Computer Speech and Language*, vol. 3, pp. 239-252, 1989.

[32] X. Huang, Y. Ariki and M. Jack, "Hidden Markov models for speech recognition", Edinburgh University Press, Edinburgh U.K., 1990.

[33] M. Hwang and X. Huang, "Subphonetic modeling with Markov state - Senone", *Proc. Int. Conf. Acoust., Speech and Signal Processing*, pp. 33-36, 1992.

[34] M. Hwang and X. D. Huang, "Shared distribution Hidden Markov models for speech recognition" *IEEE Trans. Speech and Audio Processing*, vol. 1, pp. 414-420, 1993.

[35] M. Hwang, X. D. Huang and A. Alleva, "Predicting unseen triphones with senones", *IEEE Trans. Speech and Audio Processing*, vol. 4, pp. 412-419, 1996.

[36] F. Jelinek, "Continuous speech recognition by statistical methods", *Proc. of the IEEE*, vol. 64, pp. 532-556, 1976.

[37] B. H. Juang and L. R. Rabiner, "The segmental k-means algorithm for estimating parameters of hidden Markov models", *IEEE Trans. Acoust., Speech and Signal Processing*, vol. 38, pp. 1639-1641, 1990.

[38] B. H. Juang and L. R. Rabiner, "Mixture Autoregressive hidden Markov models for speech signals", *IEEE Trans. Acoust., Speech and Signal Processing*, vol. 33, pp. 1404-1413, 1985.

[39] A. Kannan, M. Ostendorf, and J. R. Rohlicek, "Weight estimation for N-best rescoring", *Proc. DARPA Workshop on Speech and Natural Language*, pp. 455-456, 1992.

[40] A. Kannan and M. Ostendorf, "A comparison of trajectory and mixture modeling in segment-based word recognition", *Proc. Int. Conf. Acoust., Speech and Signal Processing*, pp. 327-330, 1993.

[41] P. Kenny, M. Lennig and P. Mermelstein, "A linear predictive HMM for vector valued observation with application to speech recognition", *IEEE Trans. Acoust., Speech and Signal Processing*, vol. 38, pp. 220-225, 1990.

[42] O. Kimball, "Segment modeling alternatives for continuous speech recognition", Ph.D thesis, Boston University, 1994.

[43] Y. Konig and N. Morgan, "Modeling dynamics in connectionist speech recognition - the time index model", *Proc. Int. Conf. Spoken Language Processing*, pp. 1523-1526, 1993.

[44] Y. Konig, "REMAP : recursive estimation and maximization of a posteriori probabilities in transition-based speech recognition", Ph.D thesis, University of California at Berkeley, 1996.

[45] R. G. Leonard, "A database for speaker independent digit recognition", *Proc. Int. Conf. Acoust., Speech and Signal Processing*, pp. 42.11.1-4, 1984.

[46] K. Lee, "Automatic speech recognition : the development of the SPHINX system", Kluwer Academic Publishers, Boston, 1989.

[47] K. Lee, "Context-dependent phonetic Hidden Markov models for continuous speech recognition", *IEEE Trans. Acoust., Speech and Signal Processing*, vol. 38, pp. 599-609, 1990.

[48] J. Odell, "The use of context in large vocabulary speech recognition", PhD thesis, Cambridge University, Engineering Dept, 1995.

[49] M. Ostendorf, V. Digalakis and O. A. Kimball, "From HMMs to segmental models: a unified view of stochastic modeling for speech recognition", *IEEE Trans. Speech and Audio Processing*, vol. 4, pp. 360-377, 1996.

[50] M. Ostendorf and S. Roucos, "A stochastic segment model for phoneme-based continuous speech recognition", *IEEE Trans. Acoust., Speech and Signal Processing*, vol. 37, pp. 1857-1869, 1989.

[51] M. Ostendorf, A. Kannan, S. Austin, O. Kimball, R. Schwartz and J. R. Rohlicek, "Integration of diverse recognition methodologies through reevaluation of N-best sentence hypotheses" *Proc. DARPA Workshop on Speech and Natural Language*, pp. 83-87, 1991.

[52] K. K. Paliwal, "Use of temporal correlation between successive frames in a hidden Markov model based speech recognizer", *Proc. IEEE Int. Conf. Acoust., Speech and Signal Processing*, pp. 215-218, 1993.

[53] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition", *Proc. IEEE*, vol. 77, pp. 257-285, 1989.

[54] L. R. Rabiner, J. Wilpon and F. Soong, "High performance connected digit recognition using hidden Markov models", *IEEE Trans. Acoust., Speech and Signal Processing*, vol. 37, pp. 1214-1225, 1989.

[55] L. R. Rabiner and B. H. Juang, "Fundamentals of Speech Recognition", Prentice-Hall, 1993.

[56] M. Rayner, D. Carter, V. Digalakis and P. Price, "Combining knowledge sources to reorder N-best speech hypothesis lists", *Proc. DARPA Workshop on Human Language Technology*, pp. 217-221, 1994.

[57] S. Roucos, M. Ostendorf, H. Gish and A. Derr, "Stochastic segment modeling using the Estimate-Maximize algorithm", *Proc. Int. Conf. Acoust., Speech and Signal Processing*, pp. 127-130, 1988.

[58] M. Russell, "A segmental HMM for speech pattern modeling", *Proc. Int. Conf. Acoust., Speech and Signal Processing*, pp. 499-502, 1993.

[59] M. Russell and W. Holmes, "Linear trajectory segmental HMM's", *IEEE Signal Processing Letters*, vol 4, pp. 72-74, 1997.

[60] H. Sakoe and S. Chiba, "Dynamic programming optimization for spoken word recognition", *IEEE Trans. Acoust., Speech and Signal Processing*, vol. 26, pp. 43-49, 1978.

[61] R. Schwartz and Y. Chow, "A comparison of several approximate algorithms for finding multiple (N-best) sentence hypotheses", *Proc. Int. Conf. Acoust., Speech and Signal Processing*, pp. 701-704, 1991.

[62] X. D. Sun, L. Deng and C. F. J. Wu, "State-dependent time warping in the trended hidden Markov model", *Signal Processing*, vol 39, pp 263-274, 1994.

[63] S. Takahashi, T. Matsuoka, Y. Minami and K. Shikano, "Phoneme HMMS constrained by frame correlations", *Proc. IEEE Int. Conf. Acoust., Speech and Signal Processing*, pp. 219-222, 1993.

[64] A. J. Viterbi, "Error Bounds for convolutional codes and an asymptotically optimum decoding algorithm", *IEEE Trans. on information theory*, vol. 13, pp 260-269, 1967.

[65] C. J. Wellekens, "Explicit correlation in hidden Markov models for speech recognition", *Proc. IEEE Int. Conf. Acoust., Speech and Signal Processing*, pp. 384-387, 1987.

[66] G. Zavaliagkos, Y. Zhao, R. Schwartz and J. Makhoul, "A hybrid segmental NN/HMM system for continuous speech recognition", *IEEE Trans. Speech and Audio Processing*, vol. 2, pp. 151-160, 1994.