BOSTON UNIVERSITY
COLLEGE OF ENGINEERING

DISSERTATION

SEGMENT MODELING ALTERNATIVES FOR CONTINUOUS SPEECH RECOGNITION

BY
Owen Ashley Kimball
B.A., University of Rochester, 1982
M.S., Northeastern University, 1988

Submitted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy
1995

dvi

Approved by

First Reader

Dr. Mari Ostendorf, Associate Professor,
Department of Electrical, Computer and Systems Engineering
Boston University

Second Reader

Dr. J. Robin Rohlicek, Manager, Research and Development,
BBN Hark Systems Corp.
Research Associate,
Department of Electrical, Computer and Systems Engineering
Boston University

Third Reader

Dr. David Castanon, Associate Professor,
Department of Electrical, Computer and Systems Engineering
Boston University

Fourth Reader

Dr. Carol Espy-Wilson, Assistant Professor
Department of Electrical, Computer and Systems Engineering
Boston University

Acknowledgments

I am indebted to a number of people who helped me in various ways during this work. I would first like to thank my adviser, Mari Ostendorf, for her guidance and support throughout my time at Boston University. Her technical ideas, thoughtful criticism, and constant encouragement were all invaluable in the process of completing this research. I also wish to thank Robin Rohlicek who, in his role as "second adviser," brought creative thinking and fresh perspectives to many of the issues in this work. My discussions with Mari and Robin formed the basis for a number of the ideas presented here and always helped me sharpen the focus of the research. I also wish to thank my other readers, David Castañon and Carol Espy-Wilson, for their careful reading of the dissertation and numerous helpful suggestions. I'm grateful to my fellow students and researchers at SPILAB, both for the technical discussions that contributed directly to this thesis and for the comradeship that made the lab a stimulating, fun place to be. Finally, I wish to thank my wife, Allison, for her support and good humor through the long hours and inevitable ups and downs that accompanied this work.

This research was jointly supported by NSF and ARPA, under NSF grant number IRI-8902124 and by ARPA and ONR, under ONR grant number N0001492-J-1778.

SEGMENT MODELING ALTERNATIVES
FOR CONTINUOUS SPEECH RECOGNITION

(Order No.)
Owen Ashley Kimball
Boston University, College of Engineering, 1994
Major Professor: Mari Ostendorf Professor of: Electrical Engineering

Abstract

This dissertation presents alternative parametric statistical models of phonetically-based segments for use in continuous speech recognition (CSR). A categorization of segment modeling approaches is proposed according to two characteristics: the assumed form of the probability distribution and the representation chosen for segment observations. The question of distribution form divides models into two groups: those based on conditional probability densities of feature given label and those using *a posteriori* probabilities of label given feature. The second characteristic concerns whether a model uses a variable or fixed-length representation of observed speech segments. The choices for both characteristics have important implications, particularly for context modeling and score normalization. In this work, specific segment models are developed in order to understand the benefits and limitations that follow from these choices.

Mixture distributions are a particular type of conditional density with appealing modeling properties. Under a special case of segment models using variable-length representations and conditional densities, various forms of Gaussian mixture mod-

els are examined for the individual samples of the feature sequence. Within this framework, a systematic comparison of both existing and novel mixture modeling techniques is conducted. Parameter-tying alternatives for frame-level mixtures are explored and good performance is demonstrated with this approach.

Within the conditional-density variable-length framework, a generalization of mixture distributions that captures properties of the complete segment is proposed in the form of a segment-level mixture model. This approach models intra-segment correlation indirectly using a mixture of segment-length models, each of which uses conditionally independent time samples. Parameter estimation formulae are derived and the model is explored experimentally.

The alternative assumption of modeling based on *a posteriori* probabilities is examined through the development of a recognition formalism using classification and segmentation scoring. Posterior distributions have been less well studied than conditional densities in the context of CSR, and this work introduces a theoretically consistent, segment-level posterior distribution model using context-dependent models. Issues concerning fixed versus variable-length representations and segmentation scoring are explored experimentally. Finally, some general conclusions are drawn concerning the practical and theoretical trade-offs for the models examined.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The primary goal of research in automatic speech recognition is to develop a device that transcribes human speech into written text at the same level of accuracy as, or higher than, that exhibited by humans. The potential benefits of such technology are enormous, both as a means of entering text and data to existing computer applications more efficiently than by typing, and as an integral part of future intelligent devices that will use speech recognition and speech synthesis as a natural means of communicating with humans.

Today, speech recognition systems find use in a number of tasks, allowing humans to communicate with machines where typing is either cumbersome or impossible. For example, small vocabulary recognition has proven very useful for environments where data must be entered by a worker whose hands or eyes are otherwise occupied, such as in manufacturing control applications. Speech recognition also enables the use of computers for many individuals, who because of injury or other disability, cannot operate a keyboard. Recently, there has been a large increase in the number of recognition applications for use over the telephone, including automated dialing, operator assistance, and remote data access services, such as financial services. Limited voice dictation systems have also been introduced, both for general topics and for

specialized domains, such as medical transcription applications.

In all of the above examples, current recognition accuracy typically limits the type of speech accommodated to either words spoken in isolation or to speech from a restricted domain. As the technology improves and users grow accustomed to voice interaction with machines, the uses of speech recognition will expand dramatically to include a broad range of applications.

Most obviously, speech recognition will find use in fast, automatic dictation systems that allow the production of written text at the same speed as natural talking. A very high performance version of such a system could be adapted for use as an aid for the hearing impaired, translating general speech sources to text. Such a device would have the advantage of being useable in situations where lipreading is not possible or sign language translation is unavailable. An accurate recognition system may eventually find use in voice communications as a very low rate coding device, in which the transmitted information for a voice line is just the text of a spoken sentence, which can then be resynthesized at the receiving end. In speech-to-speech translation systems, where speakers of one language communicate with those of another through a computer intermediary, the "front end" processing rests primarily on speech recognition technology. Finally, speech recognition will have a critical role in future communication interfaces between humans and computers. Ultimately, general man-machine communication will involve not just speech transcription, but understanding the meaning of utterances as well as the generation of intelligent actions and responses from the computer. High performance speech recognition will be crucial to the development of all of the above systems.

The past decade has seen a dramatic improvement in recognition performance; measurements on comparable test sets have shown an 80% reduction in error rate just in the period from 1987 to 1991 [81]. In large part this improvement can be attributed to the exploitation of statistical models of the speech process. Of the many advantages

that all statistical methods share, perhaps the most important is the existence of well-defined criteria for automatic optimization in training and recognition, in contrast with the generally *ad hoc* procedures that were the basis of many early recognition systems. Automatic training algorithms allow the use of large amounts of data, which in turn supports robust modeling of the varied acoustic phenomena that occur in real speech. The most widely used statistical model in speech recognition research today is the hidden-Markov model (HMM). For the HMM, not only are the questions of training and recognition clearly posed, but there are particularly efficient algorithms for solving the resulting optimization problems.

Although progress has been substantial, and current systems are approaching levels of performance that are useful for limited tasks, it is clear that the state of the art today is far from the the performance required for future sophisticated recognition applications, and even further from the plausible upper bound of human performance. What is required to achieve these future performance levels? Although we can expect further progress as improvements are made within the HMM framework, more dramatic improvements may be possible if we can directly target and overcome known limitations of these models.

In reviewing the strengths and weaknesses of statistical models, some directions for improvement in this area become evident. There is great advantage to having a clear mathematical framework that admits definite answers to the issues of training and recognition optimization, and any new model would do well to retain this advantage. On the other hand, the most obvious weakness in current statistical models stems from particular simplifying assumptions that are inaccurate.

For HMMs, perhaps the most important assumption, both for the advantages and disadvantages that stem from it, is the assumption of conditional independence of acoustic feature vectors given the underlying state sequence. This assumption yields very efficient training and recognition algorithms, but it has the drawback that

it disagrees with what is known of the actual speech process and prevents us from effectively modeling the correlation of features across time.

It is the purpose of this thesis to propose alternative statistical models that can incorporate more realistic assumptions and to evaluate those models in a common test environment. The models we will investigate fall into the general class known as segment models. Broadly speaking, segment modeling can be described as an approach in which the characteristics of complete speech segments are modeled together, in contrast with HMMs and similar models in which the distributions of observation vectors that represent the speech signal over short time intervals are effectively modeled independently. Through the modeling of larger time-scale observations, segment models attempt to capture the correlation of observations within phonetic segments and/or make use of acoustic-phonetic features that span segments.

The notion of segment modeling is not new and a number of important models of this type have preceded this work, dating as far back as the knowledge-based, segmental approaches of the 1970's. Although many of these have shown promising results in applications of limited domain, none of them has yet shown a significant improvement in performance on large vocabulary continuous speech recognition. From this perspective, the potential of segment modeling has not yet been fulfilled. It is the goal of this thesis to explore alternative segment models, both to increase our understanding of segment modeling issues, as well as to achieve higher accuracy recognition.

Among segment models of a statistical nature, the stochastic segment model (SSM) [57] has played a prominent role and represents a general framework that accommodates a number of modeling alternatives [19, 56, 69]. The models developed in this thesis are posed within this general framework.

Our work proposes a characterization of segmental methods according to two issues: the representation of segment observations and the type of distribution used in computing the likelihood of phone sequences. For the first issue, the choice is

whether to use distributions of variable-length observations versus modeling a fixed-length transformation of the observation sequence. The second issue concerns whether the likelihood for the sequence of phones comprising an utterance is computed based on *a posteriori* probabilities of phones given observations versus class conditional densities of observations given phones. For practical and historical reasons, conditional densities are typically applied using variable-length distributions, and *a posteriori* distributions most often with with fixed-length observations, although these particular associations are not mandatory. There are advantages and disadvantages for each of these choices, and we elucidate some of the tradeoffs in the discussion of the specific models developed in the thesis.

Under the category of conditional, "variable-length" models, we can, as a special case, make conditional independence assumptions similar to those found in hidden-Markov models. In this thesis, we explore the issues of Gaussian mixture modeling, using a model based on such assumptions. Although conditionally independent models do not exploit the segmental properties of our general approach, they allow us to establish baseline performance for segment models under conditions similar to HMMs. Using Gaussian mixture frame-level densities in this context, we demonstrate performance comparable to that found in high-accuracy HMM systems. Additionally, we are able to apply insights gained in this domain to a second, more general mixture model developed in this work, the segmental mixture model.

The segmental mixture model, which also falls under the category of variable-length methods, captures distinct patterns of correlation of the feature sequence within phonetic segments by using mixtures of segment-length distributions. Although the mixture components still assume conditional independence of successive observation vectors, because separate components model distinct patterns, the overall model has the ability to capture within-segment correlation effects. Our results indicate that this approach is effective in capturing correlation within a segment, achieving

context-independent results significantly better than comparable non-segmental models, although we were unable to show a similar improvement in the context-dependent case due to the high dimensional parameter space of the model and limited training data.

We also explored properties of segment models based on the alternative distribution of *a posteriori* probabilities of phones given speech and the issues surrounding the use of a fixed-length representation for the speech segment. We examined theoretical issues in using this type of model and obtained experimental evidence about the impact of different modeling assumptions in the context of a specific posterior model, the *classification-in-recognition* model. Models with a similar framework to this have been proposed by others and our work clarifies some of the common issues that arise from the simplifying assumptions required when using *a posteriori* distributions in recognition. Comparing this model with the segment mixture model, we draw some general conclusions about the relative merits of *a posteriori* versus conditional models and fixed- versus variable-length representations. In particular, we find that conditional models can more easily incorporate phonetic context, while *a posteriori* models have a natural formulation for including a window of observation context. The recognition performance of the specific posterior model developed in this work was found to be lower than conditional model performance, although we did not fully exploit some of the potential advantages of this approach.

The rest of this thesis is organized as follows. In Chapter 2, the speech recognition problem is described more fully and a review of previous work is presented. Chapter 3 describes the conditions used in the experiments presented throughout the rest of the thesis. In Chapter 4, we present work on high performance, Gaussian mixture distributions using a version of the SSM that makes conditional independence assumptions similar to those in HMMs. Chapter 5 describes a segmental approach based on mixtures of segment length distributions. Chapter 6 presents work on an

alternative, fixed-length segmental formalism that is based on posterior distributions of phones given observations. Finally, Chapter 7 discusses contributions of the thesis and some possible future research directions.

Chapter 2

Background

In this chapter, we present a review of the speech recognition problem as well as the segment modeling approach. We first describe the recognition problem in general and then the statistical approach to recognition, with emphasis on the role of segment models and how they contrast with hidden-Markov models. We then review a number of previous segmental methods including earlier versions of the stochastic segment model and some recent segmental methods based on artificial neural networks.

2.1 Speech Recognition

The goal of speech recognition is the accurate transcription of human speech by computer. As noted in the introduction, there are a variety of uses for a successful speech recognition system, both by itself, as in dictation machines and aids for the handicapped, and as a component in other systems, as in the case of spoken language systems and speech-to-speech translation systems. The enormous potential benefit of these and other applications has led to active research in speech recognition for a number of years.

Knowledge Based Systems

Early efforts in speech recognition were dominated by so called "knowledge-based" approaches. In this methodology, researchers typically attempted to model the speech process by writing programs with explicit rules that directly described that process. Most of these early efforts were separate from concurrent research in statistical pattern recognition, and thus did not borrow from the mathematical frameworks developed in that work. While many of these efforts used essentially *ad hoc* programming techniques for the inclusion of more rules (e.g., the HWIM system of BBN [76]), and others developed a fairly cohesive formal structure for this purpose (e.g., CMU's Hearsay project [44]), the fundamental approach common to most of this work was the use of a broad knowledge base of "if-then" type rules describing the acoustic-phonetic knowledge of the system's developers. Part of the appeal that this approach offered was the hope that it might bring the considerable body of knowledge developed by linguists about the acoustic-phonetic properties of speech to bear directly on the understanding of the speech signal by the computer. Accordingly, it was hoped that as the systems were developed, many of the errors made by the programs could be analyzed in terms of a lack of knowledge on the program's part, and the solution would be simply to add the appropriate acoustic-phonetic rules until the errors disappeared. Moreover, the typically *ad hoc* framework allowed researchers to include whatever measures and heuristics seemed most useful in a fairly unconstrained manner. In particular, there were no real obstacles in such systems to the use of segmental measurements and acoustic-phonetic "features" that spanned phones and even across phone boundaries, in addition to the short-time spectral analysis features commonly used in statistical methods [76].

Unfortunately a number of disadvantages accompanied the rule-based approach as well. It was generally found that the rule-based framework became more and more

cumbersome as a system's knowledge base grew. First, the sheer magnitude of the task often became overwhelming: the process of describing all the subtle variations in acoustic-phonetics, when such effects as coarticulation were accounted for, required an unprecedented amount of linguistic analysis, not to mention programming time to put the results of that analysis into the computer. Moreover, as the knowledge base grew, the interactions between rules became larger in number and subtler in effect. As a result, the process of adding knowledge became progressively harder as more knowledge was added to the system. Often developers found themselves with a system whose detailed rules were precisely understood but whose global behavior was extremely hard to analyze or predict because of the many subtle interactions of the independent rules.

Despite the significant efforts of a number of research sites in this area, systems of this type never achieved distinguished success in large vocabulary speech recognition and have generally fallen into disfavor in current research. Unfortunately, with the wide-spread abandonment of the knowledge-based approach, the question of the possibility and usefulness of incorporating acoustic-phonetic knowledge in speech recognition systems has received much less attention as well. It remains an open question as to how and if such knowledge can actually improve speech recognition systems today.

Statistical Modeling

A second broad trend in speech recognition can be identified as the "statistical" approach. In the past ten years, statistical methods have dominated the research in recognition, both in the number of researchers employing them and by measures of comparative system performance. The common characteristic of this work is the use of an explicit stochastic model of the speech process. Such a framework provides answers to questions about crucial issues such as the combination of knowledge sources

and meaningful optimization criteria for training and recognition algorithms. The methods that will be investigated in this thesis can be broadly characterized as being of the statistical type. We will describe characteristics of this approach in section 2.2.

Artificial Neural Networks

A third area of research that has recently gained significant attention is the use of artificial neural networks (ANNs) for speech recognition. Much of the interest in the general area of neural networks was stimulated in the early 1980's by the introduction of the "back-propagation" algorithm, which permitted the automatic training of multi-layer networks [71]. Such networks, often called multi-layer perceptrons (MLPs), have been shown to have very general classification properties: they can be trained to approximate arbitrary functions given a sufficient number of hidden units [14]. There has also been considerable interest in the use of artificial neural networks for the particular task of speech recognition. Initial work in phonetic classification, i.e., identifying phones when the segmentation boundaries are given, produced promising results [80] and excited considerable interest in the possibilities for this approach. More recent efforts area have focused on extending the use of neural networks to the more general problem of speech recognition in which the segmentation is unknown, e.g., [78, 68].

Some of the recent research in ANN's has particular relevance for the segmental, statistical approach to recognition we take in this thesis, for the following reasons. First, under certain training conditions, MLPs can be shown to approximate posterior classification probabilities [25, 54, 59, 9] and, can thus be be integrated into a statistical approach to speech recognition (e.g., [8]). Second, in recent ANN research, attempts have been made to use neural networks to model segmental information in the speech signal. As will become apparent later, there are a number of parallels between some of these segmental neural network approaches and the posterior-distribution

method described in Chapter 6.

In the remainder of this chapter, we present a general overview of statistical modeling for speech recognition, highlighting the fundamental differences between the widely used hidden-Markov model based approaches, and segment-based systems, and emphasizing the issues relevant to successful segment modeling. We conclude with a brief review of previous segment modeling work.

2.2 Statistical Approach

In the statistical framework, the goal of recognition is simply to find the most likely sequence of words given the spoken utterance. More formally, we wish to find the maximum *a posteriori* (MAP) label sequence (where the labels are either phones or words) given the acoustic observations (speech representation), i.e., to find

$$A^* = \underset{A}{\operatorname{argmax}}\, p(A \mid Y), \tag{2.1}$$

where $A = a_1, ..., a_N$ is a sequence of labels of variable length N, and Y is the sequence of features representing the acoustic input. If the distribution $p(A|Y)$ is known for every possible input, Y, this rule yields the minimum probability of error. Since the probability of observation $p(Y)$ is common to all hypotheses in (2.1), we can ignore this factor and instead use

$$A^* = \underset{A}{\operatorname{argmax}}\, p(A, Y). \tag{2.2}$$

Typically, the input speech waveform is converted into a sequence of feature vectors, called frames, each of which represents the spectral properties of the speech signal over a short (e.g., 10 to 20 millisecond) fixed-length window of the signal. In this case, if the input speech observation corresponds to T frames, Y is written as a sequence of frame vectors: $Y = y_1, ..., y_T$.

In general, the boundaries of the elements of the label sequence (beginning and end times for each of the labels a_i) are unknown, and the segmentation of the signal can be modeled probabilistically as well, yielding

$$A^* = \underset{A}{\operatorname{argmax}} \sum_S p(A, Y, S), \tag{2.3}$$

where the summation is over all possible segmentations of the input. Each segmentation, S, consists of a sequence of segments $S = s_1, ..., s_N$, where s_i represents the begin and end time for the corresponding label, a_i, in A.

Most speech recognition systems produce word sequences rather than phones as their output, since this is typically more useful to the humans ultimately using the systems. Typically, however, each of the allowable words in a system is represented as a network of phone models. For this reason, several of the models presented below will be described in terms of phone models and the label sequence A will denote a phone sequence.

As described in the next section, for the particular case of HMMs we are not usually concerned with an explicit segmentation. However, an HMM does have an underlying state sequence. We shall see that a maximization of a similar form as (2.3) arises for HMMs, but with the alternate interpretation that S represents a state sequence of the model, and the marginal probability of phones and speech is written as the sum across all such state sequences.

Under either interpretation of S, instead of summing as in (2.3), it is common to perform Viterbi decoding [79] in recognition and compute

$$A^* = \underset{A}{\operatorname{argmax}} \max_S p(A, Y, S) \tag{2.4}$$

under the assumption that the most probable segmentation (or state sequence) dominates the sum.

The above recognition criteria are quite general and simplifying assumptions must be made to make the tasks of parameter estimation and recognition tractable. The

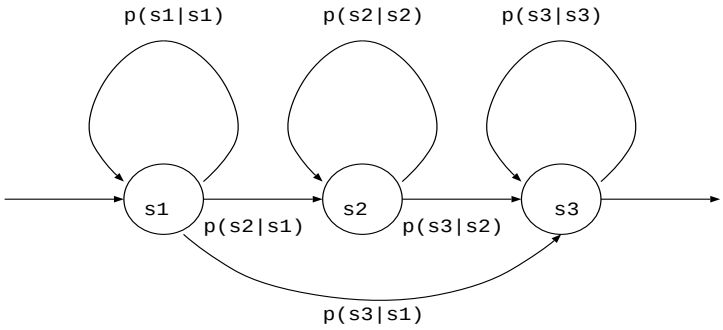

Figure 2.1: Illustration of the Markov chain of a three-state hidden-Markov model. Circles represent states of the model; arrows indicate allowable transitions between states.

different methods in statistical modeling that we review next can essentially be characterized by the particular assumptions they make to simplify the above equations.

2.2.1 Hidden-Markov Models

Much of the success of the statistical approach in improving recognition performance was achieved in the framework of hidden-Markov models [13, 42] and these models continue to dominate recognition research efforts today [30, 4, 23, 17, 62, 72]. In HMMs [5, 3, 67] the speech process is characterized by an unobserved state sequence, with the observed speech feature vectors produced according to the output probability distributions of the states of the model.

Specifically, with HMMs, we assume that each phone is represented by a Markov chain of states. A 3-state HMM for a phoneme is depicted in Figure 2.1, where the circles represent states, and arrows indicate allowable transitions between them. Each state in the chain is associated with a distribution giving the probability of observations conditioned on that state, and a second set of probabilities gives the probability of transition between states of the model, $p(s_j|s_i)$. The Markov assumption implies that the probability of a complete state sequence is just a product of these transition

probabilities.

Given a model, we find the joint probability of speech, Y, and a phone sequence, A, as the marginal over all possible state sequences, S, consistent with A (we think of composing the Markov chains of individual phones in the phone sequence into one big Markov chain):

$$p(Y, A) = \sum_S p(Y, A, S), \tag{2.5}$$

but as before, this can be approximated by using only the most probable state sequence:

$$p(Y, A) \cong \max_S p(Y, A, S). \tag{2.6}$$

We can rewrite the probability in (2.6) using the fact that the state sequence, S, uniquely determines the phone sequence, A:

$$\begin{aligned} p(Y, A, S) &= p(Y|A, S)\, p(S, A) \\ &= p(Y|S)\, p(S). \end{aligned} \tag{2.7}$$

In addition to the Markov property, the HMM assumes that individual observations comprising the sequence Y are conditionally independent given the state of the model, analogous to a memoryless channel in communications. Incorporating these assumptions, (2.7) becomes

$$p(Y, A, S) = \prod_t p(y_t|s_t)\, p(s_t|s_{t-1}), \tag{2.8}$$

where as before y_t is a single frame in the sequence Y. These assumptions establish the basis for computationally efficient automatic training and recognition algorithms [5].

One of the important innovations introduced for HMMs was the use of context-dependent phonetic models [2, 74, 43]. In context modeling, the statistics of the model, including both the state transition probabilities and observation distributions, are conditioned not just on the particular phoneme in which they occur, but also on

the surrounding phonetic context. For instance, in triphone models, probabilities are conditioned on the preceding and following phone in the phone sequence, in addition to the current phone. Context models essentially expand the state space of the HMM, and by doing this, capture more specific, detailed information about the statistics of the speech process, leading to substantially improved recognition performance.

Another important innovation was the incorporation of derivative features [21] in the observation sequence. The use of derivatives of spectral features has enabled HMMs to model more of the dynamic behavior of the speech process, and thus partially compensate for the inaccuracy of assuming frames are conditionally independent.

In early HMM systems, the observation probabilities, $p(y_t|s_t)$, were typically modeled as a discrete distribution of vector quantized features [3, 13], or using a single multi-variate Gaussian density [60]. Recently, the introduction of mixture densities to model these probabilities has led to improved recognition performance. This approach includes both "semi-continuous" or tied mixture density modeling [6, 28] as well as untied or "continuous-density"[1] mixture models [41, 55, 23, 82]. These approaches have allowed the use of models that are highly detailed, yet which retain the smoothness characteristic of continuous parametric densities. The application of mixture densities is not restricted to HMMs, and in subsequent chapters we describe the use of mixtures in both a simple version of our segment model that shares much in common with HMMs and in a more sophisticated model in which a generalization of the Gaussian mixture density serves to capture segmental information.

The advantages of the statistical framework and the effectiveness of the innovations described above have led to very good performance that have helped make

[1] We put "semi-continuous" and "continuous density" in quotes because tied mixtures and single mode Gaussian densities are also continuous densities, and the popular mixture terminology is therefore somewhat misleading.

HMMs the dominant approach to recognition in recent years. However, known weaknesses of the HMM framework leave open the possibility of better performance using alternative models.

The assumptions that HMMs rely on – that state sequences are Markovian and that observations are independent given the states – are not motivated so much by what is known of the speech process, as by the need for efficient training and recognition algorithms. These assumptions provide only a weak model of the correlation of the speech signal across time, contrary to linguistic and statistical evidence that the acoustic observations within a segment are highly correlated. Segment models, which are the focus of this thesis, relax the HMM's assumptions and thus have the potential to model the speech process more accurately.

2.2.2 Segment Models: General Considerations

Segment models can be broadly defined as models of the speech process that in some way attempt to capture directly the correlation of features across segments (phonetic units) in the speech signal. As such, segment models seek to avoid the limiting assumptions fundamental to the HMM formulation, and model complete phonetic events as a whole. Under this broad definition fall a large number of approaches, including knowledge-based, statistical, and artificial neural network methods.

When we view segmental approaches from the statistical framework, the observations we model in the recognition process are fundamentally different from those in "independent-frame" models like the HMM. In a segmental approach, the basic observation is not just a single frame from the sequence comprising the utterance, but rather the complete acoustic event spanning the range from beginning to end of a particular putative phone occurrence. Two aspects of this observation space are immediately apparent. First, the dimension of the space, since it generally corresponds

to longer acoustic observations, is much greater than that for the distribution of a single frame. We can therefore expect that the usual problems in parameter estimation for high-dimensional spaces will be particularly acute for segment models. Second, since segment durations vary from phone to phone and from instance to instance of a single phone, the dimension of this observation space varies too. This is in contrast to HMMs, in which the observations are just the speech frames and each is a vector of constant length.

Segment Representation: Fixed versus Variable-Length

The variable dimensionality of the observation space constrains our choice of recognition methodology for segments. For instance, we cannot simply take the approach of modeling complete segments as observations from a single, simple density, such as a Gaussian distribution, since Gaussians are well-defined only for fixed-dimensional spaces. The methods of dealing with this issue can essentially be divided into two categories, depending on whether we use some fixed-length representation of each segment or whether we instead use the variable-length segment observation without first transforming it.

In the first category, some fixed-length representation of the (inherently variable-length) segmental observation is computed first, and statistical modeling techniques are then applied to this representation. The fixed-length representation allows the use of statistical methods that can model the complete observation with a single distribution and thus directly capture the correlation across a segment. However, this approach introduces new difficulties as well. By applying a variable-to-fixed-length function to individual segments, we essentially change the observation dimension for the complete *utterance* (sequence of segments) to be proportional to the number of phones hypothesized for the utterance. That is, a fixed-length representation of segments changes the the original observation sequence, Y, to Y', a concatenation of

fixed-length segments. The difficulty arises in trying to apply the MAP criterion of (2.1) using the altered observations, i.e. choosing,

$$A^* = \underset{A}{\operatorname{argmax}}\, p(A \,|\, Y'). \tag{2.9}$$

This maximization is not well-defined since Y' varies from sentence to sentence. Consequently, when a recognition system of this type is allowed to hypothesize different numbers of phones for an utterance, some sort of (possibly *ad hoc*) normalization of the resulting scores that accommodates this transformation of the space must be introduced.

In the second category, the representation of each segment is not transformed before the statistical modeling stage and observations are thus proportional to the duration of the putative phone being scored. The most obvious advantage of this method is that it requires no normalization of scores, which in practice can prove to be a very difficult task. On the other hand, unlike fixed-length methods, we are unable simply to apply the standard vector pattern recognition techniques, so modeling segment correlation may require the development of novel statistical methods.

Distribution Alternatives: Conditional versus Posterior Probabilities

In addition to the choice between fixed- and variable-length representations, segment models can also be grossly characterized by the type of probability distribution used for modeling a segment. As stated in (2.4), the goal in the statistical approach to recognition is to choose the word sequence that maximizes the joint probability $p(A, Y, S)$. One approach for segment modeling is to rewrite this probability using conditional observation densities, i.e.,

$$\begin{aligned} A^* &= \underset{A}{\operatorname{argmax}} \max_{S} p(A, Y, S) \\ &= \underset{A}{\operatorname{argmax}} \max_{S} p(Y \,|\, S, A)\, p(S \,|\, A)\, p(A) \end{aligned} \tag{2.10}$$

$$= \quad \underset{A}{\text{argmax}} \max_{S} \prod_{j} p(Y_j \mid s_j, a_j)\, p(S|A)\, p(A). \tag{2.11}$$

where, s_j is the segmentation for phone j, as before, and Y_j is defined to be the segment observation for the j^{th} hypothesized phone (note that the segment observations, Y_j, differ from the frame observations y_i described earlier). The first two of the equations above are similar to the decomposition of the joint probability used for HMMs, but where HMMs further assume individual frames are conditionally independent, in the segment case, we need assume only that complete segments are independent given the label sequence. The general approach characterized by (2.11) will be called "conditional" segment modeling for later reference.

Within the conditional framework, we can examine more specifically the issues raised in the previous section concerning segment observation representation. If we represent each segment, Y_j, by some appropriately defined, fixed-length function, $f(Y_j)$, it may be possible to capture segment correlation simply by modeling this representation with a single joint density of the segment, but the resulting sequence of scores must be normalized in order to make $\prod_j p(Y_j \mid S, A)$ comparable across different label hypotheses. Alternatively, we can use a variable-length representation of segments, such as would be the case with a fixed-rate frame-based analysis of the speech (in which case, Y_j would simply consist of the subsequence of frames spanning the hypothesized begin and end times for the segment). In this case, since the dimension of the complete observation sequence does not depend on the label sequence, the conditional probability of the observations will have the same dimension for different hypothesized sentences, and scores for these sentences will be comparable without any score normalization. As mentioned before though, the drawback of this approach is that it is more difficult to model correlation across a segment.

Note that as an obvious special case of the conditional approach, segment models can use frame-based analysis and assume conditional independence of frames within

a segment (given the segment length and within-segment distribution sequence), thus adopting the essential characteristics of HMMs. In Chapter 4, we investigate the use of Gaussian mixture densities in such an "independent-frame" segment model and achieve performance comparable to analogous HMM systems.

In addition to conditional models, we consider a second broad class of probabilistic segment models based on posterior distributions. Using posterior distributions, the recognition equation (2.4) can be rewritten as

$$A^* = \underset{A}{\operatorname{argmax}} \max_{S} p(A|Y,S)\, p(S,Y). \tag{2.12}$$

The factors of (2.12) can be viewed as a "classification" probability, $p(A|Y,S)$, and a "segmentation" probability, $p(S,Y)$. This thesis introduces a specific segment model, called the *classification-in-recognition* (CIR) model, that follows this general approach. In recent segment modeling by others [46, 1], this type of approach has also been taken, with relevant probabilities being approximated with ANN's. We examine some properties of the general approach and present our specific CIR model with experimental results in Chapter 6.

The use of segmental models in a statistical framework is not a new goal and a number of different approaches have been taken towards this end. In the next section we review some of the previous work in this area.

2.3 Previous Segmental Models

In this section, we survey some previous efforts in segment modeling. These include the work of Bush and Kopec on segmental network-based digit recognition [12], several distinct variants of the SSM [57, 18], including the dynamical system segment model [16] and the microsegment model [16, 36], MIT's SUMMIT speech recognition system [64], as well as a number of artificial neural network approaches, including the

segmental neural network [1], and the multi-layer perceptron based work of Leung et al. [46]. Each of the models reviewed can be categorized according to the issues raised above: whether a fixed or variable-length segment representation is chosen and whether conditional or posterior distributions model a segment's probability. In addition to these choices we will see various approaches to the recurrent questions of correlation modeling and score normalization.

Bush and Kopec, 1987

The work of Bush and Kopec on network-based recognition had the explicit goal of developing a formalism that could score segments as a whole [12]. Their system, which was developed for the task of digit recognition, used frame-based measurements augmented by segmental features. Their approach had a probabilistic framework that explicitly segmented the input but did not directly account for the probability of segmentation. With extensive testing, only two of the segmental features, segment duration and the peak of low-frequency energy in a segment, were found to help system performance.

Stochastic Segment Models

The SSM is another formulation that uses segmental measurements in a statistical framework [70, 57]. This model, represents the probability of a phoneme based on the joint statistics of an entire segment of speech. Several variants of the SSM [18, 19, 69] have been developed since its introduction, and recent work has shown this model to be comparable in performance to hidden-Markov model systems for the task of word recognition [38]. Since the SSM is the basis for much of the proposed research, this model will be presented in some detail.

The SSM assumes that a phone a generates a random length sequence of obser-

vations $Y = (y_1, \ldots, y_L)$ according to the density

$$b_{a,L}(Y) = p(y_1, \ldots, y_L | L, a) p(L|a).$$

The model for a phone consists of a sequence of m model *regions*, each associated with the observation density for one part of the phone. The length-dependent density for the complete segment, $b_{a,L}(Y)$, is then characterized by 1) the set of region-dependent density functions (typically Gaussian) with assumptions about correlation within and across regions, 2) a collection of distribution mappings (or, time-warping transformations) that associate each frame y_i in the variable length observation Y with one of the m model regions, and 3) a duration distribution $p(L|a)$ that models the likelihood of the segment length. Together the mapping and the region-dependent distributions specify the particular distribution to be used in computing the probability $p(y_1, \ldots, y_L | L, a)$.

The training algorithm for the model consists of an iterative estimation procedure, similar to the "Viterbi training" algorithm sometimes used with HMMs. The input to the algorithm is the training speech and a set of word transcriptions for each sentence spoken. In the first step of the algorithm, the training data is segmented phonetically (i.e., phone boundaries and labels are assigned) using the current best estimate of the model parameters, the input word transcriptions and a dictionary that gives the phonetic pronunciation for each word in the vocabulary. The resulting segmentation determines the alignment of speech to model parameters, and the parameters are updated using maximum likelihood (ML) estimation. Starting from a plausible initial segmentation, the steps are repeated alternately until convergence.

In general, the distribution used to model an L-frame segment observation is an $L \times q$ dimensional multivariate Gaussian, where q is the dimension of the feature vector at each frame. One possible mechanism for specifying the distribution is to assume that the observed frames are sampled from an unobserved fixed-length frame

sequence, and the probability of the observed segment is just a marginal distribution of the complete, unobserved segment [69]. If the observations within a segment are not assumed to be conditionally independent, the ML phase of training in this case is accomplished using the Expectation Maximization (EM) algorithm.

The EM algorithm [15], is a general formulation for maximum likelihood estimation when some of the observations of a model are "hidden" or unobserved. There are a wide variety of applications for this algorithm. For instance, the Baum-Welch training algorithm for HMMs can be recast as EM training in which the state sequence is the unobserved component, and later in this thesis, we will encounter an application of the algorithm for mixture training in which the mode of the mixture is treated as unobserved. For the particular case of training the SSM variant discussed above, each phone is associated with an underlying "complete," fixed-length sequence of frames, so that for an actual observed segment from the training data, the unobserved components are the frames that are "missing" from the underlying fixed-length sequence. Details of this approach are found in [69].

A different mechanism for specifying the SSM's distribution, which has been used in several variants of the model, is to assume that the observed frames map to a fixed number of regions and that there are no "unobserved" regions. The linear mapping of observations to model regions is shown schematically for two different segments in Figure 2.2. In the segment at the top of the figure, there are more frames in the speech segment than there are regions in the phonetic model, and multiple frames are mapped to a single region. In the second case, there are fewer frames than regions, and the frames map to only a subset of the model regions.

Using this mapping and assuming the regions of the model are conditionally independent, the ML phase of training is simplified, since there is no unobserved component to the model. In recognition, the probability of the observed segment is just the product of the probabilities of the observed vectors using the density of their

respective model regions. This formulation is similar to an HMM with a fixed state trajectory, except that the segment model has an explicit duration model $p(L|a)$ whereas the HMM has an implicit geometric duration model. The assumptions allow the robust estimation of the model's statistics and reduce the computation of determining a segment's probability. However, the full potential of the segment model is not utilized.

A recent innovation in the development of the SSM has been the introduction of a dynamical system approach to recognition [19]. In this, phonemes are modeled as linear, time-varying, continuous state systems and the output of these systems is the observed speech. With this framework, recognition is performed using a version of the Kalman predictor to compute the probability of a segment in terms of its innovations. This formulation permits modeling of the time correlation of the observation sequence, and initial results for phoneme classification on the TIMIT database show an improvement with this approach over models that assume observations are independent. A disadvantage of this model is its high computational cost in both recognition and training.

The microsegment model is another variant that has been developed within the SSM framework [16, 36]. The general microsegment model posits an embedded, hierarchical model of the speech process that can potentially capture both short and long-range correlation effects. The particular versions of the model that have thus far been explored experimentally use a simplified, two-level model to capture correlation within a segment. The microsegment units at each level model the probability of observations, and, in principle, the observation features used in the probabilities may be level dependent too. A grammar component gives the probability of generating particular sequences of units at a level, and the grammar probabilities can be interpreted as mixture weights. In Chapter 5 of this thesis, we present a segmental mixture model which has a number of similarities to the microsegment model. We

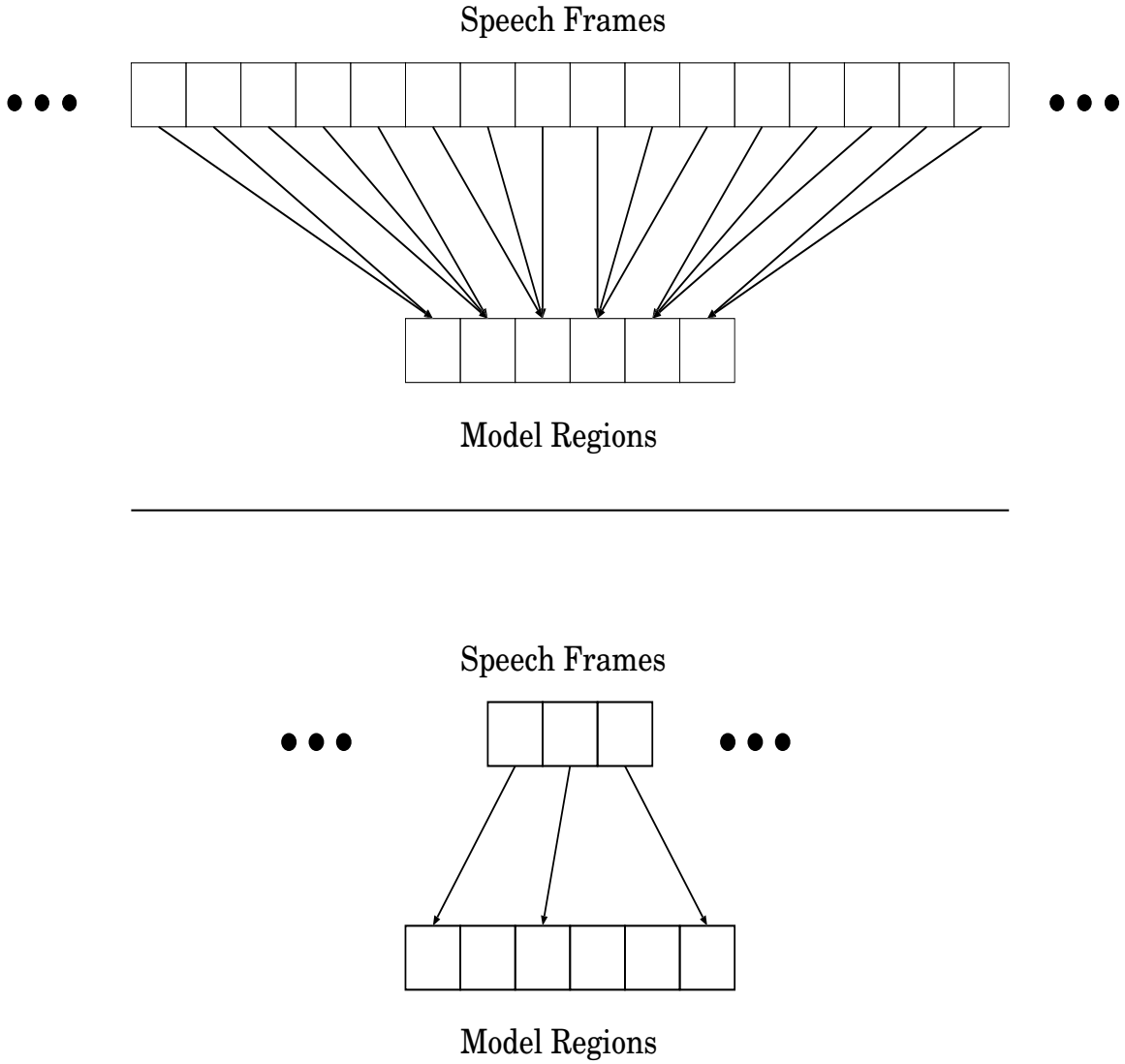

Figure 2.2: SSM warping of segment frames to model regions shown for two different length segments. Segments with more frames than the number of model regions are mapped many-to-one; those with fewer frames than model regions use a subset of the regions.

will explore the characteristics of the two models further in that chapter after the more general segment-mixture model has been introduced.

The SUMMIT System

In the SUMMIT recognition system [85, 64], a hierarchical segmentation (called a dendrogram) is produced and converted into a network of hypothesized segments, and measurements of relevant features are taken over these segments. A second network describes all possible word sequences in terms of their phonetic pronunciations. The network of acoustic segments and their relevant features is then matched against the word sequence network using dynamic programming to produce the best (or N-best) answers.

In this system, the approach taken to the problem of normalizing for the probability of segmentation is to combine boundary probabilities determined in the initial segmentation process into a segment "transition penalty" for the entire utterance. This transition penalty is one of the inputs to a linear discriminant classifier, along with segment acoustic scores and other measurements. The classifier is trained to find the combination of scores that best discriminates correct from incorrect answers in a list of N-best hypotheses.

Artificial Neural Networks

Artificial neural networks are well suited to a segmental approach in that a neural network can readily be constructed that decides the most likely phoneme by considering an entire segment of speech at a time. By examining the entire segment, or more accurately, a fixed-length function of the entire segment, instead of one frame at a time, the network can attempt to explicitly model the correlation of observation vectors through the segment.

Since the promising results of early classification research using Time Delay Neural Networks [80], numerous techniques have been proposed to apply neural networks to the more difficult task of recognition of continuous speech with unknown segmentation. For the recognition problem, many of the more recent approaches abandon the approach of using a single classification probability to score a complete phone, and instead use neural networks to score utterances on a frame by frame basis, for example [32, 78, 49, 68]. Several of these efforts also attempt to model some of the effects of correlation across frames, either by using a multi-frame window of input to the neural network, as in [49], or through the use of a recurrent network, as in [68].

Other recent work has focused on making use of neural networks as classifiers of complete segments within the recognition context. Among these, of particular relevance to this thesis is the neural network research at BBN and by Leung and colleagues. Both the BBN segmental neural network (SNN) [1, 84] and the neural network model pursued by Leung *et al* at MIT and NYNEX, apply neural network segmental classification properties to the recognition problem. These approaches have several similarities with the model proposed in Chapter 6, and we defer a more detailed review of these models until that chapter.

Summary

The segment models discussed above have yet to achieve the goal of using segmental properties to improve continuous speech recognition performance over the conditional independence case of HMMs. There is good evidence that such an improvement is possible. Versions of the SSM that make assumptions similar to HMMs have in recent benchmarks performed competitively with HMMs, indicating that the HMM assumptions can be viewed essentially as a special case of the segment formalism. Digalakis' dynamical system approach to recognition was the first to show increased accuracy using explicit information about the correlation between frames. This improved per-

formance demonstrated the existence of important information that is ignored by conditionally-independent models. However, his result was shown only for phonetic classification, and the approach involves a very computationally expensive algorithm. In this sense, the goal of improving large vocabulary continuous speech recognition by using segment models is still unfulfilled.

In this chapter, we have provided a brief overview of a number of approaches to acoustic modeling, with an emphasis on statistical models and segmental techniques in particular. Before turning to the acoustic modeling issues central to this thesis, in the next chapter we present the experimental conditions and recognition methodology used in the experiments described in later chapters.

Chapter 3

Experimental Approach

In this chapter, we describe the experimental conditions and recognition methodology used in the experiments described throughout the remainder of the thesis. We first describe the database and signal processing analysis used in our experiments. This is followed by a discussion of the methods of evaluation used – phonetic classification and continuous word recognition. The word recognition algorithm is based on N-best rescoring, an approach that greatly reduces the recognition search space, and thus allows the use of more computationally expensive acoustic models, like the segment models discussed in this thesis.

3.1 Corpus

The corpus used in this work is the 1000-word Resource Management continuous speech recognition task [66]. This task was designed to simulate queries to the spoken language interface of a hypothetical computer controlled naval database. Speech was collected for speaker-dependent and speaker-independent word recognition research; in this work, the speaker-independent portion of the database is used. The training data in this case consists of a total of 3977 sentences spoken by 109 speakers.

Numerous researchers have observed that for speaker-independent systems, gender-

dependent models (i.e., models trained separately on males and females), yield higher accuracy than when all speakers train a single model. Accordingly, the models described in this work are gender-dependent. The Resource Management database is not balanced for gender however; of the 3977 speaker independent training sentences, 2823 utterances are spoken by 78 male speakers and 1154 utterances are spoken by 31 females. As we will see later, this imbalance in training data can sometimes lead to different recognition strategies working better for the different sexes.

The database also has standard test sets for evaluation of recognition accuracy. These typically consist of 300 utterances spoken by 10 speakers. Testing material was released over a number of years to allow researchers to test their systems periodically on data that had not been seen before, and the test sets used in this thesis are the "Feb89", "Oct89" and "Sep92" sets.

The grammar used in recognition evaluation is the standard "word-pair" grammar. The function of a grammar for speech recognition is to reduce the set of word sequences that are possible or likely and thus reduce the search space of the recognizer [39]. The word-pair grammar was designed by examining a set of example sentences from the Resource Management domain to determine which pairs of words occurred in the set. The grammar allows only those sequences for which each pair of words was observed in the original set of sample sentences. The resulting grammar, although generally considered too restrictive to model a real computer interface, has a simple measure of complexity and has become a standard condition for reporting results on Resource Management.

The signal processing for our experiments consists of computing a sequence of vectors that capture the spectral characteristics of the speech over a small window, or

"frame", of the speech. Specifically, every 10 milliseconds, a vector of 14 Mel-warped [1] cepstral coefficients is computed using a 20 millisecond window of the speech. In addition, we use the first differences of the cepstral coefficients and energy in order to capture some of the dynamic characteristics of the signal.

The last official evaluation with the Resource Management database was in September 1992 [58]. At that time, all ARPA sponsored research systems were evaluated on the Sep92 set, and sites that chose to were given the chance to report performance on the earlier Feb89 and Oct89 sets using their most recent systems. For the Sep92 set using the word-pair grammar, the performance for different speaker-independent systems varied from 4.4% to 11.7% word error rate, with the average performance over the best system from each of the 10 sites of 7.2%. Four of the sites (including the two systems with lowest error rates for the Sep92 set) also submitted results for the Feb89 and Oct89 tests. For Feb89, the results ranged from 2.7% to 6.1%, and for Oct89, the results ranged from 2.7% to 6.4%. (There is no simple explanation for the generally higher error rates for the Sep92 test set compared with earlier tests, although informal listening suggests that the speakers for this set are "less careful" in their speech than speakers from earlier tests.)

---

[1] By Mel-warping we mean a non-uniform scaling of the frequency axis, f, by the function

$$m(f) = f_0 \log_\alpha (1 + (\alpha - 1)\frac{f}{f_0}),$$

where $\alpha = 2$ and $f_0 = 1000$Hz. Additionally, we consider only the band from $f_{min} = 125$ to $f_{max} = 6000$Hz and we linearly warp the Mel scale from $m(f_{min})$ to $m(f_{max})$ into the range 0 to f_n, where f_n is Nyquist frequency, using the function

$$\tilde{m}(f) = f_n \frac{m(f) - m(f_{min})}{m(f_{max}) - m(f_{min})}.$$

Cepstral analysis is performed on the resulting transformed frequency scale.

3.2 Recognition Methodology

The two general methods used to evaluate speech recognition systems are classification and recognition. In classification, the segmentation is given and the goal is simply to find the most likely label for each segment of speech given its begin and end time. In the more general problem of recognition, on the other hand, both labels and segmentation are unknown. In this thesis we looked at classification of phonemes for initial model development and recognition of words for their final evaluation.

3.2.1 Phonetic Classification

In phonetic classification, the correct segmentation (phoneme beginning and end times) of the input observation sequence is given, and the object is to assign correct phone labels to each segment. Because classification does not involve a search for the best segmentation, the computation in this process is much less than that for recognition, and it therefore makes a good initial test bed for model development.

We wished to run both word recognition and phonetic classification experiments on the Resource Management database, but classification requires that the test data be correctly labeled for both the correct phone sequence and the phone boundaries. The Resource Management database is not phonetically labeled and has just correct word transcriptions, so an automatic labeling scheme was used to determine the reference phoneme sequence and segmentation for each sentence in the database. The labeler, based on a context-dependent SSM, took the correct orthographic transcription, a pronunciation dictionary, and the speech for a sentence and used a dynamic programming algorithm to find the best phonetic alignment. The procedure used an initial labeling produced by the BBN BYBLOS system [40] as a guide, but allowed some variation in pronunciations, according to the dictionary, as well as in segmentation. The dictionary used here and for other aspects of recognition gives phonetic

pronunciations for each of the 1000 words in the Resource Management task. The average number of pronunciations per word is 1.1, with most words having just a single allowable phone string. The alignment that results from this process is flawed in comparison with carefully hand transcribed speech, like that found in the TIMIT database, but it is good enough for simple comparisons.

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 model being evaluated. Performance is measured as the percent correct classification averaged across the database. Our early experiments showed that using comparable models and analysis, there is only about a 4-6% loss in classification performance between the TIMIT and Resource Management databases for models with performance in the range of about 70% correct (e.g., from 72% to 68% correct for context-independent models). This indicates that the automatic labeling procedure produces transcriptions that are accurate enough to make preliminary comparisons of classification algorithms.

3.2.2 Continuous Word Recognition

In continuous word recognition the segmentation is unknown and the object is to determine the most likely word sequence from all possible segmentations. In recognition we use a phonetic dictionary that determines the allowable phoneme pronunciations for words, so finding the most likely word sequence is equivalent to finding the most likely phone sequence using the constraints of the dictionary.

Many of the models developed in this thesis require substantially more computation for word recognition than is usual for other statistical methods, like the HMM; it is not surprising that models that relax restrictive assumptions incur a penalty of larger search spaces and correspondingly greater computation. For continuous word

recognition, to be able to use these less restrictive models and still be able to run experiments in a timely manner, we can greatly reduce the search space by adopting the N-best word recognition strategy [73, 56].

The N-best mechanism was originally developed for use in a spoken language system in which a speech recognizer generates a list of the N highest scoring candidate hypotheses for a single utterance, which are then passed to a natural language component that applies language constraints to determine the most likely of these candidates [73]. As developed in [56], this mechanism can be used to rescore hypotheses with a variety of knowledge sources, not just a natural language component.

The particular configuration of the N-best formalism used for this work is as follows. The hypotheses to be rescored are generated by the BBN BYBLOS N-best recognition system [73], a very high performance HMM recognition system. In this work, the BYBLOS system used the word-pair grammar to constrain the possible word sequences that were scored, and for each utterance it generated $N = 20$ hypotheses, where each hypothesis corresponds to a different possible sentence. This list contains the correct answer about 97% of the time.

After the hypotheses are created, they are rescored with the SSM. The SSM receives from BYBLOS the words in each hypothesis as well as the begin and end times and labels for the highest scoring phoneme transcription corresponding to the words. The SSM uses the phoneme boundary times as a guide for conducting a constrained dynamic programming search that considers alternative boundaries within a small window around the times found by the HMM ($\pm$ 100 milliseconds). Additionally, the SSM considers not just the phonemes given by the HMM transcription, but all pronunciations that are listed for a word from a pronunciation dictionary (the same dictionary used in the alignment process described in the previous section).

To assign a single score to a hypothesis, the system takes a linear combination of the following scores: the SSM log score, the number of words in the sentence

(effectively, a word insertion penalty), the number of phonemes in the sentence, and optionally the HMM log score. Note that although the HMM provides the number of phonemes from its labeling of the hypothesis, since the SSM considers different phonetic pronunciations, its alignment may have a different number of phonemes, and to be consistent we use the number of phonemes as computed by the SSM. The set of weights used in the linear combination of scores is estimated to minimize the word error rate over a set of sentences from a development test set [37]. The speech in this set is not part of the training used for the SSM model parameters. For most of the experiments described later in the thesis, the principle focus is on performance of the SSM alone. In this case, we do not use the N-best system to combine the SSM with the HMM. However, even in this simpler case, we make use of the mechanism to combine scores in order to weight the SSM score, the number of words, and the number of phones, since weight estimation provides an automatic method for determining good "insertion penalties" (weights for the number of words and phones).

Ideally, the weights for combining the different knowledge sources should be trained on a set of data independent from the set that is being evaluated, since the weights may become overtuned to the particular set they are trained on, making evaluation less meaningful. Previous work, however, suggests that this effect is negligible, probably because the number of parameters being estimated is small (typically 3 or 4) and the amount of data in a test set is relatively large (300 sentences or about 2500 words). Therefore, in order to save computation in experiments, unless otherwise noted, the results presented in this thesis generally use weights trained on the same evaluation set.

In evaluating performance using the N-best formalism, the word error is computed from the highest-ranked of the rescored hypotheses. The percent word error is defined to be the sum of the number of substitutions, deletions and insertions, divided by the

number of words in the correct sentence. The overall performance is the word error averaged across all utterances in the database.

Chapter 4

Frame-Level Mixture Models

4.1 Introduction

In recent non-segmental work based on HMM's, researchers have obtained very good recognition performance using mixture densities to model the probability of individual frames (e.g. [23, 82]). As noted in Chapter 2, the stochastic segment model, which is the basis for the models in this thesis, has as a special case, an independent-frame formulation similar to HMM's, so we can expect to achieve comparable performance by adopting similar techniques. Although this approach does not exploit the segmental characteristics that make our models potentially more powerful than HMM's, it is useful to have a high-performance, baseline system with which to compare any improvements due to segmental modeling algorithms. Additionally, our experience with the independent-frame mixture algorithms described in this chapter is directly useful in the development of the more general segment-mixture model described in Chapter 5.

In HMM-based speech recognition systems, mixture distributions have recently achieved widespread acceptance as a high accuracy acoustic modeling technique, first in the form of tied-mixture, or semi-continuous systems [28, 6] and, more recently,

with the adoption of untied mixtures (also called "continuous-density" mixtures) [23, 82, 41, 55]. Both types of mixture have the ability to model a wide variety of data distributions, and the different types of tying allow the researcher to trade off smoothness versus model detail in a controlled manner.

In this chapter, we first describe Gaussian mixture modeling and review previous work in this area. We then describe the training procedure for mixtures in the independent-frame SSM framework, including algorithm modifications introduced for computational efficiency. We next discuss some of the parameterization issues and simplifying assumptions that can affect performance with mixtures, and conclude with results of experiments exploring these issues.

4.2 Previous Work

A central problem in speech recognition for both the HMM and SSM is finding a good model for the probability of acoustic observations conditioned on the state in a hidden-Markov model (HMM) or, for the case of the independent-frame SSM, conditioned on a region of the model. For these models, the acoustic observations of interest are typically some frame-based representation of the spectral properties of the speech, such as a vector of cepstral coefficients, computed over a short time window. Some of the options that have been investigated for modeling these observations include discrete distributions based on vector quantization of the feature space, as well as continuous distributions such as Gaussians, Gaussian mixtures and tied-Gaussian mixtures.

In discrete distribution HMM's, the successive feature vectors (frames) of the speech signal are vector quantized and the observation probabilities become non-parametric discrete distributions of the quantized codebook indexes. Examples of this type of model include the early systems developed at IBM [3], BBN [13], SRI

[52], and CMU [42]. This method suffers from a loss of information introduced by quantization of the feature vector, but has the advantage of making few assumptions about the shape of the distribution. Estimates of discrete distributions may also require large amounts of training data and special smoothing techniques to ensure robustness of the estimates [77, 75].

An alternative approach, is to use simple parametric distributions, such as multivariate Gaussian distributions. In this case, we denote the probability density of frame y conditioned on model region r (alternatively, HMM state r) as $p(y|r) \sim N(\mu_r, \Sigma_r)$, where μ_r and Σ_r are the distribution mean and covariance matrix, respectively. (For the purposes of this discussion, the region index r denotes a region within a particular phonetic model, i.e., every region of every model has a different index.) With Gaussian distributions, the problems of quantization are eliminated and robust estimates can generally be found with moderate amounts of data, but the assumed parametric form may be a poor fit to the actual distribution of the speech data. Systems of this form have been explored in [60] and [69].

A more general parametric model that overcomes some of these limitations is a Gaussian mixture density. In this case, the probability of an observation vector conditioned on model region r is written as

$$p(y|r) = \sum_k w_{rk}\, p_{rk}(y). \tag{4.1}$$

Here the set of mixture "component densities", $p_{rk}(y)$, are again assumed Gaussian, $p_{rk}(y) \sim N(\mu_{rk}, \Sigma_{rk})$, for $k = 1, 2, \ldots$, and the mixture weights, w_{rk}, give the probability of each mixture component for the region, i.e., $w_{rk} = p(k|r)$. Mixture models allow more general types of distribution "shapes" (e.g., multimodal and/or asymmetric distributions) and, in principle, can approximate arbitrary distributions given enough component Gaussians. They also retain the characteristic smoothness of simpler parametric densities. Since the number of parameters is directly related to the

number of mixture components in the densities, it is important to avoid introducing so many components that the model cannot be well estimated. Some examples of this type of system can be found in [34, 65, 53, 41, 55, 23].

A crucial issue in the development of these models has been the trade-off between modeling power and "trainability": the ideal density should be powerful enough to model arbitrary distributions, yet have a representation sufficiently parsimonious to allow robust parameter estimation with limited training data.

In tied-mixture modeling, distributions are again modeled as a mixture of continuous densities, but rather than estimating the component Gaussians separately, every mixture density in the system is constrained to share the same underlying Gaussians with only the weights differing. The probability density for observation vector y conditioned on being in region r is thus

$$p(y \mid r) = \sum_k w_{rk} p_k(y). \tag{4.2}$$

Note that here the component Gaussian densities, $p_k(y) \sim N(\mu_k, \Sigma_k)$, are not indexed by the region, r. This approach can be seen as a particular example of the general technique of tying to reduce the number of model parameters that must be trained [33].

"Tied mixtures" and "semi-continuous HMMs" are used in the literature to refer to HMM distributions of the form given in Equation (4.2). The term "semi-continuous HMMs" was coined by Huang and Jack, who first proposed their use in continuous speech recognition [28]. The semi-continuous terminology highlights the relationship of this method to discrete and continuous density HMMs, where the mixture component means are analogous to the vector quantization codewords of a discrete HMM and the weights to the discrete observation probabilities, but, as in continuous density HMMs, actual quantization with its attendant distortion is avoided. Bellegarda and Nahamoo independently developed the same technique which they termed "tied

mixtures" [6]. For simplicity, we use only one name in this work, and choose the term tied mixtures, to highlight the relationship to other types of mixture distributions.

Although untied mixtures are slightly simpler conceptually and were introduced earlier [34, 65, 53], the use of mixture modeling first became widespread for large vocabulary recognition with the introduction of tied mixtures. Tied mixture systems offer two advantages. First, they provide smooth, robust estimates for moderate sized training databases and second, in their basic form, the computation for tied-mixture systems is substantially smaller than for untied systems.

More recently, untied mixture densities, with both different weights and different Gaussians for each distribution in a system, have been found to offer the possibility of more detailed, "higher resolution" modeling of observation densities when there is adequate training data. Performance on the Wall Street Journal dictation task [63], with its relatively large training database, indicates that untied mixture modeling may yield generally higher performance than tied mixtures in this situation.

In another recent trend, researchers have recognized that a continuum of tying options exists, and it is possible to choose a level of tying appropriate to the training data and computation allotted for a task. Accordingly, a number of variants of mixtures have been proposed. One possibility is to reduce the tying from a single "codebook" of underlying Gaussian densities, to one codebook for each phone [61]. This type of system allows higher resolution models while retaining some of the smoothness and speed of tied systems. SRI has recently introduced the so called "genonic" mixture model [17]. In this model, multiple tied-mixture codebooks are also used, but in this case, the group of models that share a set of Gaussians in a mixture is determined automatically via clustering, rather than being prechosen according to linguistic class. Error rate reductions over tied-mixture systems of 25% and more have been reported with this approach.

In large-vocabulary recognition, the number of context-dependent states rises very

quickly with the number of allowed words. In order to reduce the number of free parameters in the system and to be able to use the training data of triphones that occur infrequently, in addition to tying the underlying Gaussians of a mixture, it is possible to tie the mixture *weights* across states that have similar densities, so that they share complete distributions. Equivalence classes of context-dependent HMM states can be automatically derived, using either agglomerative or divisive clustering techniques. The clustering criteria can include information theoretic measures of density similarity or linguistic information about the phonetic context of a state. The semi-continuous senone model developed at CMU [31] was one of the first to explore the possibilities of this approach. In this case, starting from a tied-mixture system, all states grouped in an equivalence class share mixture weights as well as the underlying Gaussians. Similar state-tying has also proven effective for "untied" mixture systems [82] (untied in that each equivalence class has separate Gaussians and weights, but tied in that states in the same class share complete distributions), as well as uni-modal Gaussians [35].

In addition to the degree of tying, there are several other important parameter choices that lead to mixture models with different performance characteristics. One choice involves the feature correlation structure assumed for the component densities of the mixture. In some systems, input features such as cepstra and their derivatives are treated as distinct observation "streams", where each block of features is assumed independent and is modeled by a separate set of mixtures. Alternatively, the blocks can be concatenated and modeled by a single density. Within an observation stream, different assumptions about feature correlation are possible, leading to either full covariance, diagonal, or block diagonal Gaussians. For tied mixture systems, Gaussian components are modeled using either full covariance matrices [30, 40] or diagonal [62, 50, 4], whereas, recent untied mixture systems have all used diagonal covariances to limit both computation and the number of parameters that must be trained.

Parameter initialization is another issue that can be important, since the EM algorithm used for training mixtures is an iterative hill-climbing technique that guarantees convergence only to a local optimum. Many researchers initialize mixture component Gaussians with parameters estimated from data subsets determined by K-means clustering, e.g., [30], although Paul describes a different, bootstrapping initialization [62]. Often a large number of mixture components are used and, since the parameters can be overtrained, contradictory results are reported on the benefits of parameter re-estimation. For example, while many researchers find it useful to re-estimate all parameters of the mixture models in training (e.g., [6, 30]), BBN reports no benefit for updating means and covariances after the initialization from clustered data [40].

In addition to the work described above, other related methods have also informed the research concerning mixture distributions for speech recognition. First, mixture modeling does not require the use of Gaussian distributions; good results have also been obtained using Laplacian distributions [55, 4], and presumably other component densities would perform well too. Ney [54] examines the neural network technique of radial basis functions and finds that it is essentially the same as tied mixtures of diagonal Gaussian densities. One version of the BBN segmental neural network makes use of elliptical basis functions, which are analogous to tied mixtures of full-covariance Gaussians [83].

Of course, the best choice of model depends on the nature of the observation vectors and the amount of training data. As stated before, it is likely that the amount of tying in a system can be adjusted across a continuum to fit the particular task and amount of training data. Later in this chapter, we describe experiments with various mixture models for speaker-independent recognition in the context of the independent-frame SSM system. A number of results reported previously in the literature are difficult to compare due to differing experimental paradigms; our work, in addition to improving overall recognition accuracy for the SSM, attempted to shed

light on the performance implications of some of these parameterization issues in a controlled setting.

4.3 Training Algorithms

In this section we first describe the training algorithm used for mixtures in the independent-frame SSM. Next, we describe a parallel implementation of the trainer that greatly reduces experimentation time. Lastly, we present a training modification that improves efficiency when estimating context-dependent models in the tied-mixture case.

4.3.1 The SSM and "Viterbi" Mixture Training

In this chapter, we use the independent-frame SSM described in Chapter 2. Specifically, a linear time warping function maps each observed frame to one of m regions of the segment model. Each region is characterized by a (tied or untied) Gaussian mixture distribution, and the frames are assumed conditionally independent given the length-dependent warping. The conditional independence assumption allows robust estimation of the model's statistics and reduces the computation required to determine a segment's probability, but the potential of the segment model is not fully utilized. Under this formulation, the SSM is similar to a mixture HMM with a phone-length-dependent, constrained state trajectory. Thus, many of the experiments reported here are comparable with analogous HMM systems.

As described before, the SSM training algorithm [57] alternates between segmentation and maximum likelihood (ML) parameter estimation, so that during each parameter estimation phase of the algorithm, the segmentation computed in the preceding phase gives a set of known phonetic boundaries. Additionally, for a given phonetic segmentation, the assignment of observations to regions of the model is uniquely de-

termined by the model's deterministic warping. SSM training is thus similar to HMM "Viterbi training", in which training data is segmented using the most likely state sequence and model parameters are updated using this segmentation. Although it is possible to define an SSM training algorithm equivalent to the Baum-Welch algorithm for HMMs (i.e. an ML estimation algorithm that treats the segmentation as an unobserved component), the computation is prohibitive for the SSM because of the large effective state space.

Estimation of the mixture distributions is greatly simplified by the constrained segmentation in SSM training; the mapping of training observations to model regions is known, so the training of each region's distribution becomes an independent problem of simple mixture estimation. In this case, the parameter estimation step of the SSM's iterative segmentation/estimation algorithm involves the use of the expectation-maximization (EM) algorithm, itself an iterative algorithm, to estimate the parameters of a mixture distribution [15]. In applying the EM algorithm, we assume that each training observation is generated by a single mode (Gaussian distribution) of the mixture density for a model region, but the identity of the particular mode is unobserved. In contrast to SSM mixture training, the full EM algorithm for mixtures in an HMM handles both the unobserved state in the Markov chain and the unobserved mixture mode [6]. For completeness, we summarize the re-estimation formulae for the SSM below.

Untied Mixture Re-estimation Formulae

We will develop the untied mixture case first, and then indicate the changes required for tied-mixture re-estimation. We wish to estimate the parameters of each mixture distribution, specifically, the mixture weights, $w_{rk} = p(k|r)$, and the means and covariances of the component Gaussians, μ_{rk} and Σ_{rk}. For convenience, we can consider the complete training data as a single sequence of frames, $Y = Y^1, \ldots, Y^T$. In the

EM algorithm, we compute a set of "counts" based on the training data, and then use these to update our estimates of the model parameters. The count for a mixture component given a specific frame from the training set, Y^t, is just the estimate of the probability of observing that particular component given the frame. The motivation for the terminology "count" may be understood by considering the case where the mode of the mixture is somehow known for every training sample. In this case, the probabilities in question would always be 1 for some particular component of the mixture and 0 for all others. The sum of these probabilities over the database would then just be the count of the number of times each mode occurred in training in each region of the model. In the real case, when the mixture mode is unknown, we are accumulating instead sums of probabilities that each mode occurred over the database, and the sums therefore give probabilistic "counts" of the number of occurrences for a region.

Specifically, the count ξ_{krt} is defined to be the estimate of the probability that the k^{th} component of the Gaussian mixture distribution for model region r is the unobserved component, given that the observation is frame Y^t:

$$\begin{aligned} \xi_{krt} &= p^p(k|r, Y^t) \\ &= \frac{p^p(Y^t|k, r)\, p^p(k|r)}{p^p(Y^t|r)}, \end{aligned} \tag{4.3}$$

where the notation $p^p(\cdot)$ indicates a probability based on model parameters from the p^{th} iteration of the EM algorithm. That is, $p^p(Y^t|k, r)$ is the estimate of the Gaussian distribution for component k of the mixture with mean and covariance estimated from iteration p of the algorithm, and $p^p(k|r)$ is the estimate of the mixture weight for component k in region r from iteration p. The denominator of (4.3) is computed as the marginal sum over mixture components:

$$p^p(Y^t|r) = \sum_k p^p(Y^t|k, r)\, p^p(k|r). \tag{4.4}$$

Let $t(r)$ denote the set of frames from the training sequence that are mapped to a particular region r of a model (i.e., those frames assigned to region r based on the current segmentation of the training data and the model's deterministic warping within a segment). As shown in a later appendix, the update formula for the mixture weight for region r, Gaussian component k, can then be expressed as:

$$\hat{p}\,(k|r) = \frac{\sum\limits_{t \in t(r)} \xi_{krt}}{\sum\limits_{k} \sum\limits_{t \in t(r)} \xi_{krt}}. \tag{4.5}$$

The mean and covariance estimates are

$$\hat{\mu}_{kr} = \frac{\sum\limits_{t \in t(r)} \xi_{krt}\, Y^t}{\sum\limits_{t \in t(r)} \xi_{krt}}, \tag{4.6}$$

and

$$\hat{\Sigma}_{kr} = \frac{\sum\limits_{t \in t(r)} \xi_{krt}\, (Y^t - \hat{\mu}_{kr})(Y^t - \hat{\mu}_{kr})^T}{\sum\limits_{t \in t(r)} \xi_{krt}}. \tag{4.7}$$

Tied Mixture Re-estimation Formulae

The modifications to the above equations for re-estimating the tied mixtures of (4.2) are straightforward. The counts, ξ_{krt}, are computed in the same way, as are the estimates for the weights, $\hat{p}(k|r)$. However, the re-estimation of the parameters of the tied Gaussians, $\hat{\mu_k}$ and $\hat{\Sigma}_k$ is different. Since the Gaussians are now shared by every region of every phonetic model in the system, we sum the counts across all these regions in estimating the new parameter values. In this case, the update formula for the Gaussian means is

$$\hat{\mu}_k = \frac{\sum\limits_{r} \sum\limits_{t \in t(r)} \xi_{krt}\, Y^t}{\sum\limits_{r} \sum\limits_{t \in t(r)} \xi_{krt}}, \tag{4.8}$$

where the summation over r denotes a summation across all the regions of the model. The corresponding equation for covariances is

$$\hat{\Sigma}_k = \frac{\sum_r \sum_{t \in t(r)} \xi_{krt} \, (Y^t - \hat{\mu}_k)(Y^t - \hat{\mu}_k)^T}{\sum_r \sum_{t \in t(r)} \xi_{krt}}. \tag{4.9}$$

In Chapter 5, we will introduce a segmental mixture model that has this chapter's independent-frame model as a special case. The full derivation of the EM algorithm for that more general case will be given in an appendix there.

Parameter Initialization

Since the training algorithm described above is an iterative hill climbing technique, the type of initialization used can be important if we are to avoid convergence to a poor solution. We investigated two different methods for choosing initial estimates for context-independent models.

In the first method, we cluster the training data based on the *k-means* algorithm (called the isodata algorithm in [20] and referred to as the LBG algorithm in information theory literature [48]) using a Euclidean distance metric. A mean and covariance are then estimated from the data corresponding to each *k-means* cluster and these are used as the parameters of the component Gaussian densities for the initial mixture [29]. The mixture weights are initially equal for all components.

In the second method, we obtain initial Gaussians from models trained in a non-mixture version of the SSM. The initialization procedure is slightly different for the tied and untied case. In tied-mixture initialization, we need to choose just a single set of Gaussians that will be shared among all regions of all phones. In this case the Gaussian means are assigned from a subset of the non-mixture triphone Gaussians using the following procedure: we cycle through all phones in the phone alphabet, choosing from all the context-dependent triphone models of the particular phone,

the one that has the highest frequency of occurrence in training. One of the regions of the triphone is chosen (arbitrarily, region 1, since Gaussians will be shared across regions) and assigned as the mean of the next Gaussian in the model being initialized. We continue in this manner going through the set of phones (repeatedly, if there are more mixture components than phones), choosing the next highest frequently observed triphone for each phone, until we have the required number of Gaussians for the mixture. When the mean of a mixture component is assigned from a non-mixture triphone model, the covariance is initialized to be the non-mixture covariance from the corresponding context-independent model (i.e. using the same phone and region as the mean). Context-independent covariances are used in order to give a broad initial distribution and thus avoid premature convergence in training. In general, this initialization procedure attempts to cover the space of phones while using robustly estimated models. For tied mixtures, it has the possible advantage over the *k-means* initialization of broader coverage of the phone set, which is important for phone discrimination and not necessarily accomplished by minimizing Euclidean distance as in the *k-means* algorithm.

In the untied case, in which each region of each phone has a separate set of Gaussians, the initial Gaussian means for a phone's mixture are chosen only from the corresponding region in the associated non-mixture triphone. We cycle through each phone's triphones from most to least observed models, and as before, we initialize the covariance to be the context-independent covariance from the corresponding region in the non-mixture model.

Given one of the above initial estimates, mixture training for context-independent models proceeds for several iterations of the EM algorithm. Training of context-dependent models begins after convergence of the context-independent models. Each context-dependent allophone is initialized to have the same mixture distributions as its corresponding context-independent phone model, and several more iterations of

the EM algorithm are applied to the combined context-independent and context-dependent models. In the early iterations of the EM algorithm, we add a small padding value to the weight estimates and then renormalize, in order to avoid premature parameter convergence. The padding value for the first iteration is typically about $1/2N$, where N is the number of mixture components. This value is decreased by a factor of 10 at subsequent iterations and not used at all after three or four iterations.

Robust Variance Estimates

The EM algorithm described above seeks to maximize the likelihood of training data, but under certain conditions this approach can lead to degenerate parameter estimates. In particular, if the mean of one of the Gaussian mixture components takes on the value of one of the training samples and that component's variance becomes zero, the likelihood of the training data becomes unbounded. Since an unbounded likelihood is optimal for ML training, the EM algorithm will in fact seek such a solution given the right initial conditions. For tied mixtures, this behavior is unlikely for the number of mixture components typically used, since the Gaussians are shared by all densities in the model and the covariance estimates will therefore almost always be broadened by having non-zero contributions from many training samples. The problem can arise, however, in the case of untied mixtures, where there are usually many more Gaussian components to be trained, and some of the components can become "dedicated" to just one or two outlying training samples.

A number of schemes are possible to prevent singular covariance estimates in this case, such as "clipping" variance estimates that fall below some threshold [41] or using Bayesian estimation techniques that make use of a prior distribution over variances [24]. Both of these approaches serve to eliminate very narrow variance estimates and have similar performance characteristics [22]. In our work, we found problems with

degenerate variances only in the case of untied mixtures, for which we had assumed a diagonal covariance structure. We adopted a variance clipping approach patterned after that described in [41]. Specifically, the clipping threshold for each feature of the Gaussian was estimated as a fixed percentile of the overall variance for that feature estimated across all training observations. All variance estimates below this were replaced with the threshold value. In experiments, we found that the specific choice of percentile was not critical – anything in the range from about to 15% to 50% gave similar performance – as long as improbably small variances were removed.

4.3.2 Parallel Training

To reduce computation, our system prunes low probability observations, using the approach described by Paul in [62], and for tied mixtures, uses the marginal training algorithm described in the next subsection. However, even with these savings, mixture training – particularly tied mixture training – involves substantial computation, making experimentation potentially cumbersome. When the available computing resources consist of a network of moderately powerful workstations, as was the case for this work, we would like to make use of many machines at once to speed training. At the highest level, mixture training is inherently a sequential process, since the model estimates from each pass are used in obtaining the estimates for the next pass. However, the bulk of the training computation involves estimating the counts defined in (4.3) over a database, and these counts can be computed independently of each other. We can therefore speed training by letting machines estimate the counts for different parts of the database in parallel and combine and normalize their results at the end of each pass.

To implement this approach, we use a simple "bakery" algorithm to assign tasks: as each machine becomes ready to do work, it reads and increments the value of a

counter from a common location indicating the sentences in the database it should work on next. Mutual exclusion must be ensured for operations on the task counter in order to prevent two machines from reading the same value and estimating the same counts twice. The use of a task counter, rather than simply assigning a portion of the database to each of the available machines at the beginning of training, provides simple dynamic load balancing, allowing us to make efficient use of machines that may differ in speed over the course of training. Finally, one processor is distinguished as the "master" processor and it is assigned the task of collating and normalizing the counts at the end of each pass. This processor signals the others to proceed to the next iteration of training by setting a synchronization flag (also protected by mutual exclusion).

For parallel algorithms that require that all processors obtain exclusive access to a resource (such as the task counter and synchronization flag in this algorithm), there is a danger that processors will have to wait idly while others access the resource, greatly reducing the overall efficiency of the algorithm. For this reason, in many cases it is important to have very fast mutual exclusion operations. However, because of the relatively coarse grain of parallelism in our case (one task can consist of processing 10 utterances, and we typically use less than 5 machines at once), we can take the simple approach of storing the counter and synchronization flags in files on the network-accessible file system and use the mechanism of file locking for mutual exclusion. Although network disk operations are relatively slow (in comparison with processor instruction speed, for example), a much greater amount of time is spent estimating the counts than accessing the counter, and there is therefore no noticeable penalty in terms of efficiency.

With the use of this algorithm, we obtain a speedup in training linear with the number of machines, providing a much faster environment for experimentation.

4.3.3 Mixture Context Modeling

We have investigated two methods for training context-dependent mixture densities. The first is useful for both tied and untied systems, and the second, though applicable only to tied mixtures, offers a significant speed improvement.

In the first method, weights are used to combine the probability of different types of context. These weights are typically a function of both the density's position in the model (i.e., whether the density's region is near the left or right end of a phonetic model) and the number of training samples observed for the particular context. Essentially, this method estimates the combined context model as a mixture of individual context models. This mixture is treated differently than the frame-level mixtures that are the focus of this chapter. In particular, the context mixture weights cannot be reliably estimated using the ML techniques of this chapter without overtuning to the training data. Specifically, triphone models are able to match the training data so well that if the context weights are trained using ML, the weights for other contexts will be estimated as essentially zero. This form of overtuning leads to very poor performance on independent test data. The most common methods of avoiding this are to choose weights manually [74] or derive them automatically using a deleted-interpolation algorithm [33]. Paul evaluated both automatic and manually-derived weights for tied-mixture context modeling and found no significant performance difference between the two [62]. In our experiments with this method, we used manually-picked weights.

For tied mixture systems, we developed a second method that takes advantage of the fact that all densities in the system are weighted combinations of the same underlying Gaussians. Accordingly, in this method, only models of the most detailed context (in our case triphones) are estimated directly from the data, and simpler context models (left, right, and context-independent models) are computed as marginals

of the triphone distributions. For example, the estimate for the probability density of observation y given region r in the context-independent model for phone a, $p(y \mid r, a)$, is just computed as the prior-weighted sum of the corresponding triphone context models of phone a. If we use the shorthand notation $b[a]c$ to denote the triphone "phone a, when it is preceded by phone b and followed by phone c," we can write this marginal as

$$p(y \mid r, a) = \sum_b \sum_c p(y \mid r, b[a]c)\, p(b[a]c \mid a). \tag{4.10}$$

The probability $p(b[a]c \mid a)$ is the relative frequency estimate of observing the context-dependent allophone given phone a.

This approach is feasible because the Gaussian components are shared. If they were not, the marginal in (4.10) would result in a density with as many Gaussian components as are in all the context models of the allophone, and recognition would involve a prohibitively large computation. With tied mixtures though, the probability of an observation given a model is expressed as

$$p(y \mid r, a) = \sum_k p(y \mid k)\, p(k \mid r, a) \tag{4.11}$$

where $p(y \mid k)$ is the k^{th} tied Gaussian density value and $p(k \mid r, a)$ is a model-dependent mixture weight. Expanding the context-dependent probability $p(y \mid r, b[a]c)$ in the marginal sum (4.10) in this way, we get

$$\begin{aligned} p(y \mid r, a) &= \sum_b \sum_c p(y \mid r, b[a]c)\, p(b[a]c \mid a), \\ &= \sum_b \sum_c \sum_k p(y \mid k)\, p(k \mid r, b[a]c)\, p(b[a]c \mid a), \\ &= \sum_k p(y \mid k)\, (\sum_b \sum_c p(k \mid r, b[a]c)\, p(b[a]c \mid a)) \end{aligned} \tag{4.12}$$

and (4.12) is the same as (4.11) with

$$p(k \mid r, a) = \sum_b \sum_c p(k \mid r, b[a]c)\, p(b[a]c \mid a). \tag{4.13}$$

The computation of marginals in training is negligible since it involves just the summing of mixture weights as in (4.13) at the end of each training pass over the database. This method reduces the number of model updates in training in proportion to the number of context types used, although the computation of observation probabilities conditioned on the mixture component densities remains the same.

In recognition with marginal models, it is still necessary to combine the different context types, and we use the same hand-picked weights as before for this purpose.

We compared the two training methods for tied mixtures and found that performance on an independent test set was essentially the same for both methods (marginal training produced 2 fewer errors on the Feb89 test set) and the marginal trainer required 20 to 35% less time, depending on the model size and machine memory.

4.4 Experiments

We conducted a series of experiments to explore issues associated with parameter allocation and training in mixture distributions. In this section, we discuss experiments involving tied-mixture Gaussian distributions, followed by results using untied mixtures. The results are compared to a baseline, non-mixture SSM that uses a full covariance Gaussian distribution at each region of each phonetic model.

4.4.1 Tied-Mixture Densities

Under tied-mixture modeling, the first set of experiments examined the number of component densities in the mixture, together with the choice of full- or diagonal-covariance matrices for the mixture component densities. Although the full covariance assumption provides an explicit description of the correlation between features, diagonal covariance models require substantially less computation and it may be possible to obtain the same correlation information using a larger number of diagonal

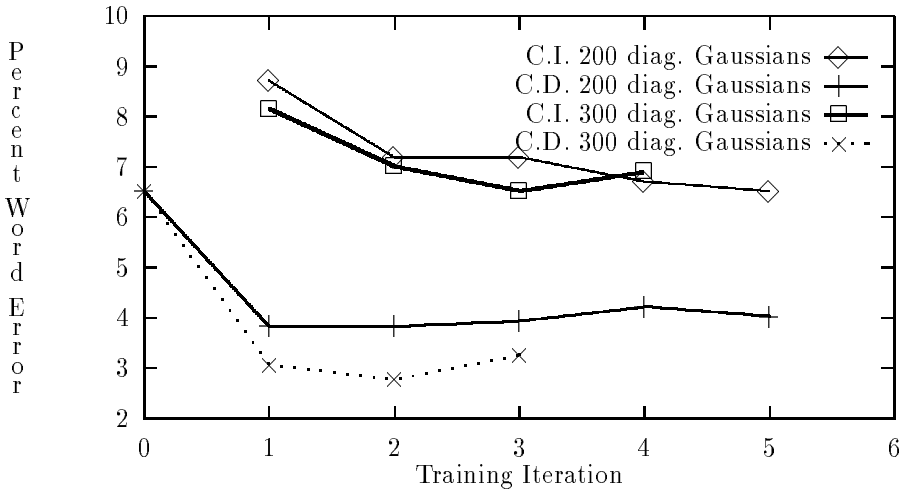

Figure 4.1: Diagonal covariance tied-mixture results for Feb89 females

models at a lower cost than the full covariance assumption.

In initial experiments with just female speakers, we used diagonal covariance Gaussians and compared 200- versus 300-density mixture models, exploring the range typically reported by other researchers. With context-independent models, after several training passes, both systems got 6.5% word error on the Feb89 test set. For context-dependent models, the 300-density system performed substantially better, with a 2.8% error rate, compared with 4.2% for the 200 density system. These results compare favorably with the baseline SSM, which has an error rate on the Feb89 female speakers of 7.7% for context-independent models and 4.8% for context-dependent models.

A summary of these results is shown in Figure 4.1. In this figure, the word error rate for female models from successive iterations of the EM algorithm are plotted for context independent and context dependent models, for both 200 and 300 mixture systems. The plots show effects typical of iterative training procedures: error

rates decrease up to a point and then begin to increase as the algorithm overtunes to the training data. Note also the rapid convergence of the context-dependent models from iteration zero to iteration one (the model at iteration zero is simply the context-dependent model after initialization but before any training; its performance is necessarily identical to that of the context-independent model it is initialized from).

The overtuning effect exhibited in Figure 4.1 is commonly encountered in the EM algorithm. On successive iterations, the algorithm essentially narrows the variance of Gaussian densities, leading to models that give a higher likelihood on the training data, but that perform poorly on independent test data. Rather than attempting to find the best model that is not overtuned based on a training likelihood measure, we take the simple approach of running successive models on an independent test set and choosing the model with the highest recognition accuracy. This approach, in turn, runs the possible risk of overtuning to the test set if the choice of iteration is somehow test-set dependent. This does not seem to be the case in practice, however, since we have observed that improvements on the set used to choose the model iteration generalize well to independent test sets. Additionally, the final evaluation of a model chosen in this way is to conducted on an independent test set.

For male speakers, we again tried systems of 200 and 300 diagonal covariance density systems with context independent models, obtaining error rates of 10.9% and 9.1% for each, respectively. Unlike the females, however, this was only slightly better than the result for the baseline SSM, which achieves 9.5% for this condition. We tried a system of 500 diagonal covariance densities, which gave only a small improvement in performance to 8.8% error. Finally, we tried using full-covariance Gaussians for the 300 component system and obtained a significant improvement to 8.0% error rate. The context-independent performance for males with the above models for different iterations of training is summarized in Figure 4.2. The context-dependent performance for males using the full covariance configuration showed similar improve-

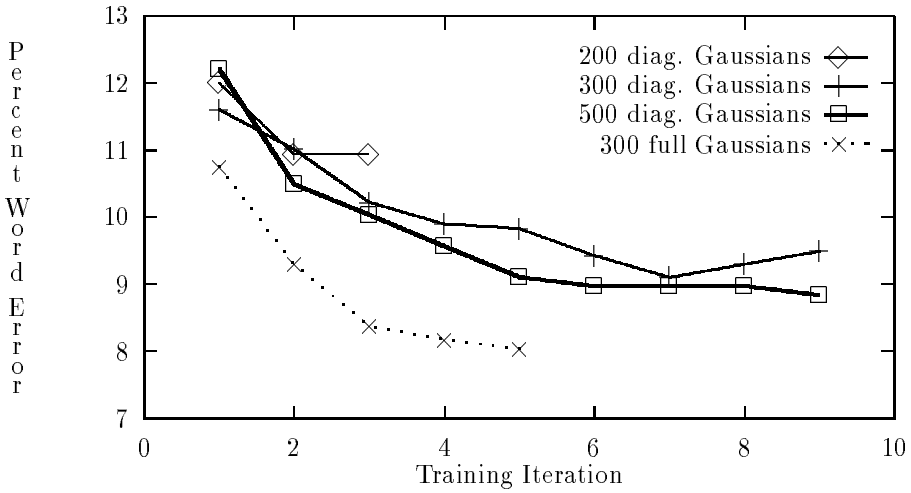

Figure 4.2: Context-independent, tied-mixture results for Feb89 males.

ment over the non-mixture SSM, with an error rate of 3.8% for the mixture system compared with 4.7% for the baseline.

Returning to the females, we found that using full-covariance densities gave the same performance as diagonal. We adopted the use of full-covariance models for both genders for uniformity, obtaining a combined word error rate of 3.3% on the Feb89 test set. We plot the results for these models in Figure 4.3. In the RM SI-109 training corpus, the training data for males is roughly 2.5 times that for females, so it is unsurprising that the optimal parameter allocation for each may differ slightly.

Most other reported tied-mixture systems have treated cepstral parameters and their derivatives as independent "streams" of observations. Specifically, let c represent the vector of cepstral coefficients for a frame, dc the frame's cepstral difference vector, and $y = (c, dc)$ the complete frame vector consisting of the concatenation of cepstral

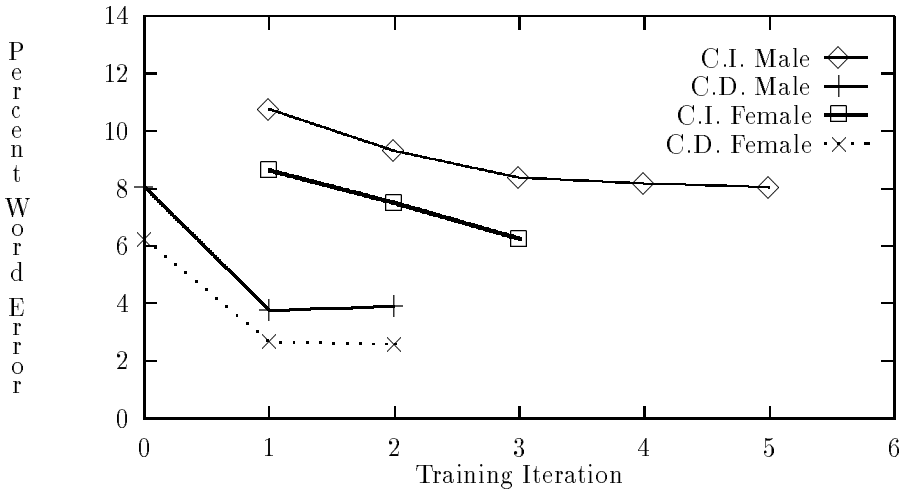

Figure 4.3: Best case, tied-mixture results for Feb89 males and females.

features and differences. Then, in the independent stream case we assume

$$p(y) = (\sum_k w_k^0 \, p(c|k)) \, (\sum_k w_k^1 \, p(dc|k)), \tag{4.14}$$

where the densities for each observation stream have their own mixtures and w_k^0 and w_k^1 are the weights for the distributions. When we use a single observation stream, as is the case for most of the experiments in this thesis, a single mixture gives the probability for the complete vector, i.e.

$$p(y) = \sum_k w_k \, p(y|k). \tag{4.15}$$

The use of independent streams has the effect of reducing the dimensionality of the Gaussians that are estimated, but may be a weaker model if the streams are actually dependent. In earlier non-mixture work with the SSM, it was found to be more effective to model cepstral parameters and their differences jointly rather than independently. In this work, we compared the two approaches and found that for tied

mixtures as well, concatenating the cepstra and their differences into a single observation vector performed better than treating each independently. The improvement for joint modeling was an 18% reduction in error over the independent case.

A more recent trend has been to use second derivatives of cepstra in addition to cepstra and their first derivatives. We have not incorporated this into our mixture density system, but we expect we would find an improvement similar to that reported by others. However, in recent work with a non-mixture version of the SSM that uses single multivariate, Gaussian densities, it was found that simply using second derivatives failed to improve performance. In this case, the second derivatives were added to the cepstral coefficients and their first derivatives to form a single vector that is modeled with a single, full covariance Gaussian density. The poor performance is presumably due to a lack of sufficient training data for the greatly increased number of free parameters. We would expect similar problems for the case of mixtures of full covariance Gaussians as well. It might be possible to avoid the strong assumption of independent observation streams, while reducing the number of parameters, by modeling all parameters in a single vector, but imposing a block diagonal structure on the covariance matrix so that correlation is modeled within each of the blocks (cepstra, derivatives, and second derivatives) but not between blocks.

As described earlier, we investigated two methods of initialization for mixture training, one based on *k-means* clustering and the other using Gaussians estimated in a non-mixture model. We found that the *k-means* initialized models converged slower and had significantly worse performance on independent test data than that of the second method. Although it is possible that with a larger padding value added to the weight estimates and more training passes, the *K-means* models might have "caught up" with the other models, we did not investigate this further.

The various elements of the mixtures (means, covariances, and weights) can each be either updated in training, or assumed to have fixed values. In our experiments,

Table 4.1: *Word error rate on the Oct89 and Sep92 test sets for the baseline non-mixture SSM, the tied-mixture SSM alone and the SSM in combination with the BYBLOS HMM system.*

System	Test set	
	Oct 89	Sep 92
Baseline SSM	4.8	8.5
T.M. SSM	3.6	7.3
T.M. SSM + HMM	3.2	6.1

we have consistently found better performance when all parameters of the models are updated.

Table 4.1 gives the performance on the RM Oct89 and Sept92 test set for the baseline SSM, the tied-mixture SSM system, and the tied-mixture system combined in N-best rescoring with the BBN BYBLOS HMM system. The mixture SSM's performance is comparable to results reported for many other systems on these sets. We note that it may be possible to improve SSM performance by incorporating second difference cepstral parameters as most HMM systems do.

4.4.2 Untied and "Partially Tied" Mixture Densities

As mentioned in the introduction of this chapter, there are a number of possibilities for the degree of tying with Gaussian mixtures. We explored some of these variants for frame-level density modeling in the SSM, although in somewhat less detail than the tied mixture experiments described above. We examined untied, or "continuous-density", mixture models in which each region of each phonetic model has its own set of Gaussians and mixture weights. We also looked at two other approaches,

phonetically-tied mixtures and covariance tying, that each lie somewhere between the tied and untied models in terms of the degree of parameter sharing.

The initialization procedure for untied mixtures used the second of the context-independent methods (the "non-*k-means*" approach) described before, in which a context-independent mixture is obtained by choosing Gaussian components from a context-dependent, non-mixture SSM. As in the tied case, context-dependent mixtures were in turn initialized from a well-estimated context-independent mixture model. The covariance structure for the untied mixtures was assumed to be diagonal to reduce the number of parameters in the model. In experiments, the number of Gaussians in a mixture varied from 8 to 64. In general, context-independent models were able to use larger numbers of Gaussians to improve performance, while triphone models typically had insufficient amounts of data to train the mixture parameters as the number of Gaussians increased; we found an effective limit of about 32 Gaussians per model for this case.

For the context-independent condition, performance was significantly better for untied models than for tied, whereas for context-dependent models, tied mixtures were somewhat better than untied. On the Feb89 test set, the best context-independent system performance was 5.7%, compared with a 7.3% error rate for the tied-mixture SSM on this set. For context-dependent models, the best untied system used 32 diagonal Gaussian components and a training threshold on the models of 50 occurrences (i.e., triphones that occurred fewer times than this were neither trained nor used in recognition). The performance in this case was worse than the best tied-mixture system, with a combined error rate for both sexes of 3.7% compared with 3.3% for the tied-mixture case. The difference between untied and tied-mixture performance was somewhat closer for females on this set (2.8% versus 2.6% for untied versus tied mixtures), than it was for males (4.3% versus 3.7% for untied and tied, respectively).

As discussed earlier, it is also possible to use a more limited form of tying in

Gaussian mixture models. We investigated phone-tied mixtures, in which all of the context-dependent models for a given phone shared the same set of Gaussians densities, with just the mixture weights varying between models. The best configuration with this approach was to use a 32 component, diagonal Gaussian mixture, as in the untied model, but in this case a threshold of 1 occurrence on the context-dependent models was used in training (using a threshold of 50 results in 445 triphones being selected for the male subset; a threshold of 1 results in about 7000). On the male subset of the Feb89 test, this approach led to an error rate of 4.2% – slightly better than the completely untied mixture model. Because of the reduced number of Gaussian probabilities that must be computed at each frame, however, the overall computation for this model is significantly less than for untied models.

A final method of parameter tying was investigated, this time letting context-dependent allophones of a phone share just the covariances, which were again assumed to be diagonal. In this case, both the means and the mixture weights had context-dependent values, but the covariance matrices were assumed to be common to all the context-dependent allophones of each phone. On the male subset of the Feb89 set, the error rate was reduced to 3.6%, essentially the same as that for the tied mixture model on this set. Table 4.2 summarizes the context-dependent results for male speakers of the Feb89 test set under the various tying approaches discussed in this chapter.

For context-independent models, the different versions of untied mixtures performed better than tied mixtures. The Gaussian components in a tied-mixture system tend to be broader than in untied systems, since each Gaussian is trained with data from all of the phones. As a result, tied models may be "smoother" than necessary for the amount of training available in the context-independent case. Additionally, there were about 1.5 times as many free parameters in the best untied model as in the best tied model. Although this difference is not great, it is possible that the untied

Table 4.2: *Word error rate on the Feb89 male speakers for different tying approaches with frame-level, diagonal-covariance, Gaussian mixture densities and context-dependent models.*

Mixture Type	Word Error
Untied	4.3%
Phone-tied Gaussians	4.2%
Phone-tied covariances	3.6%
Tied Gaussians	3.7%

models, by adding more detail, essentially capture some of the context-dependent phenomena in different mixture components.

The context-dependent untied model performs essentially the same as (for the covariance-tied model) or slightly worse than (for completely untied models) the tied-mixture model. The latter result seems to contrast with results reported recently by other research sites in which untied-mixture models have been found to perform better than tied. However, there are some differences that may account for this anomaly. First, most of the tied mixture systems that have been reported so far model the cepstra and their derivatives as independent streams, while most untied systems model them in a single observation vector. Some of the benefit others have noticed going from tied to untied may be attributable to dropping the assumption of independence in the observation streams. Our tied-mixture results use a single observation stream for all the features, with 300 full-covariance mixture components representing this feature space, where most other sites use 200 or 256 Gaussians per observation stream. Some other systems also use untied distributions, but apply them in modeling distributions of clustered states [82], thus ensuring a sufficient amount of

data for training every distribution in the system. Finally, most recent results have been reported on the Wall Street Journal (WSJ) task, for which there is significantly more training data than there is for the RM task. It seems likely that as training data increases, untied models, which generally have more free parameters than tied systems, will continue to improve as tied-mixtures systems reach an asymptote of performance. However, to limit the scope of our work we did not explore performance on the WSJ task and conclude that there is no real performance advantage for untied mixtures on the RM task.

4.5 Summary

This chapter described the use of frame-level mixtures in the SSM as well as several innovations in the mixture training algorithm, including the marginal distribution estimation for context-dependent models and a parallelized version of the training algorithm. Experiments comparing performance for different parameter allocation choices using tied-mixtures were presented, and mixture models were shown to be useful in improving the baseline recognition performance of the SSM under the assumption of conditional independence. The performance of the best tied-mixture SSM is comparable to HMM systems that use similar input features. We also looked at the questions of the degree of parameter sharing through different types of tying. In general, when sufficient training data is available, we find improved performance for models as more free parameters are introduced, assuming those parameters capture relevant information. We conjecture that for the RM task with its somewhat limited training data, when using the feature set of cepstra and derivatives, there is little benefit for the use of untied mixtures, which introduce substantially more free parameters than tied mixtures have. In the next chapter, we generalize the notion of mixture to a segment-mixture density that captures intra-segmental correlation via

the modeling of independent time-frequency trajectories in separate segment-length mixture components.

Chapter 5

Segmental Mixture Models

In this chapter, we present a segmental modeling approach based on the use of mixtures of segment-length distributions. First, we briefly provide background and review previous work relevant to this model, then we present the model itself, followed by a discussion of the training algorithm. Finally, we present experimental results that examine various aspects of the model.

5.1 Background

The goal of segment modeling is to capture segmental properties of the phone that are ignored by simpler models and that improve phonetic discriminability. When the representation of a phone's observation is the sequence of frames spanning it, an important aspect we wish to include in our formulation is the statistical correlation of frames across a phone. One possibility for capturing this "time correlation" in our models is to explicitly model the statistical dependence between different frames of a phone. This approach has the difficulty that the number of parameters to estimate increases dramatically as our models incorporate the correlation effects across larger windows of the phone, and measuring the correlation between all frames of a phone is infeasible given realistic amounts of training data. On the other hand, modeling cor-

relation in a very restricted manner, by assuming the frames of a phone are generated by a Gauss-Markov process, has been shown to give little performance improvement [11]. More recently, Digalakis introduced a successful model of correlation for context independent models with the dynamical system model [16], but this approach is quite expensive computationally.

An alternative approach is to look at the correlation information implicitly present in the "trajectory constraints" for a phone. To the extent that frames are correlated across time, the frame sequence constituting a phone's observation will have a predictable energy trajectory across time and frequency. If we can model these trajectories with sufficient accuracy, we will effectively model the correlation between frames as well, in the sense that we can predict a frame from the latter portion of a phone given the value of an earlier frame and the trajectory constraints.

In the independent-frame SSM formulation described in the previous chapter, in which observations are conditionally independent given the model regions they warp to, the observation trajectory across time is modeled by the assumption of time-dependent stationary regions in combination with the constrained warping of observations to regions. The left-to-right nature of the model and the fact that successive model regions have distinct observation densities provide a constraint on the permissible sequences of observations. In that formulation, the assumption is made that each model has a single distinct time-frequency trajectory, corresponding to the single sequence of observation densities in the model's regions. In this way, all trajectories for a given phone will be grouped together and modeled in the single segment distribution given by the model's sole density sequence.

However, if the phone we are modeling in fact exhibits more than one distinct trajectory, such a model may be inaccurate. To see this, we can imagine the effect of modeling a "multiple-trajectory" phone using one of the independent-frame models discussed earlier. For the baseline SSM that uses a single multivariate Gaussian at

each region, the variances of the Gaussians would widen to accommodate the different observed trajectories, while for the Gaussian mixture model discussed in the previous chapter, there might be multiple modes in each region's density, each corresponding to the most likely observation for a different trajectory in that region. In either case however, although the elements of the trajectories at each region may be modeled more or less well by the different frame densities, we will have lost any information about what observations go with each other across time. For instance, if a particular phone we are modeling actually has two distinct trajectories, the independent-frame model would score true observations of the phone no higher than an artificial segment consisting of a sequence of frames that are each picked randomly from one of the true instances of the phone, no matter how incongruous the resulting trajectories might be. Essentially this model allows the elements from different trajectory sequences to intermix without penalty.

The model introduced in this chapter attempts to address this deficiency by generalizing the trajectory modeling capability of the basic independent-frame formulation. The approach taken is simply to permit multiple trajectories for a phone, where the individual trajectories are each modeled by a segment-length distribution. The overall distribution for a phone, in turn, becomes a mixture distribution of these segment-length components, and the mixture weights are estimates of the probability of observing a particular trajectory in a phone. With sufficient training, it is hoped that the model avoids the "intermixing" of different trajectory elements found in simpler models, and each component will represent a separate time-frequency path of the phone.

The model described in this chapter has been preceded by earlier models with several important similarities. Digalakis introduced the microsegment model [16], a general formalism that allows modeling speech at multiple levels of resolution. This model consists of abstract units that describe the speech process at different resolu-

tions, where the units have densities for the observations at their respective levels. A grammar describes the probabilities of particular unit sequences as well as the probability of a unit at one level generating units at a lower level. A phone segment then consists of a sequence of microsegment units that model observations of duration greater than a frame but less than the complete phonetic segment. As a special case, using a two level model and treating the grammar probabilities as mixture weights, the model can be interpreted as a segment model composed of a sequence of mixtures at the microsegment level. Digalakis experimented with a two level version of the model and found encouraging results for phoneme classification on the TIMIT database.

Kannan extended work on the two-level microsegment mixture to include algorithms for context-dependent microsegments in continuous word recognition [36]. He examined the tradeoffs for using different initialization procedures and for different allocations of free parameters, in terms of microsegment length, number of mixture components and trajectory assumptions. The results of using this model on the Resource Management task indicated that the microsegment model improved context-independent performance significantly, but failed to improve context-dependent results, perhaps due to an inadequate number of free parameters in the model.

The model introduced in this chapter is similar to the above models in that it uses mixtures of longer-than-frame distributions. It is different, however, in that it focuses on phone-length segmental mixtures rather than the sub-phonetic microsegment units. As described later, this chapter's model also generalizes the type of density for the regions of a segmental component, allowing either single Gaussian or mixture Gaussian densities. Additionally, in experiments, we have chosen to estimate parameters for the model using the EM algorithm, as opposed to the Viterbi-style training used in the microsegment work. Other similarities and differences between the models will be noted in the following sections.

In the next section we give a more complete description of the segmental mixture model.

5.2 Segmental Mixture Formalism

As stated above, our goal is to model the distinct trajectories of a phone using a mixture of segment-length distributions. More specifically, for the i^{th} observed speech segment, Y_i, in the complete utterance Y, we model the probability of Y_i given phonetic unit α, as

$$p(Y_i \mid \alpha) = \sum_{j=1}^{N_C} p(c_j|\alpha)\; p(Y_i|c_j, \alpha). \tag{5.1}$$

Here, Y_i consists of the sequence of L_i frames $Y_i = (Y_i^1, Y_i^2, \ldots, Y_i^{L_i})$ spanning the hypothesized phone. For each of the N_C mixture components, c_j, $p(Y_i|c_j, \alpha)$ gives the probability of the complete segment conditioned on that component, and the probabilities $p(c_j|\alpha)$ are the mixture weights. (More formally, we view the mixture component identity of segment i as a random variable, C_i, that takes on values c_j, $j = 1, \ldots, N_C$, and we let $p(c_j|\alpha)$ denote the probability that C_i takes value c_j, i.e., $p(c_j|\alpha) \doteq Pr(C_i = c_j|\alpha)$.) To compute $p(Y_i|c_j, \alpha)$, we assume, as in the independent-frame model of Chapter 4, that frames are independent given the segment length. Expanding (5.1) in terms of frame-level probabilities, we then have

$$p(Y_i \mid \alpha) = \sum_{j=1}^{N_C} p(c_j|\alpha) \prod_{t=1}^{L} p(Y_i^t|c_j, r_t, \alpha), \tag{5.2}$$

where r_t is the model region that frame t maps to, and $p(Y_i^t|c_j, r_t, \alpha)$ is the probability of frame t conditioned on that density. As before, the mapping of an observed frame to a stationary region of a segmental mixture component is computed using a deterministic linear warping, and we use the same warping for all segmental components. Initially, we assume that the region density for a frame is normally distributed:

$$p(Y_i^t|c_j, r_t, \alpha) \sim N(\mu_{jr}, \Sigma_{jr}). \tag{5.3}$$

From this basic model, we introduce two generalizations, one to allow more general distributions at the regions, the other to reduce the dimension of the densities we are estimating.

In the previous chapter, performance was significantly improved in the independent-frame case by changing the distribution for frames from Gaussian to mixture Gaussian. This same generalization may be useful for this model if the region-dependent frame distributions in the segmental components are poorly represented by a simple Gaussian. In this case, the resulting model has mixture distributions at two levels, the segment and frame level. Equation (5.2), in turn becomes

$$p(Y_i \mid \alpha) = \sum_{j=1}^{N_C} p(c_j|\alpha) \prod_{t=1}^{L} \sum_{k=1}^{N_D} p(Y_i^t|d_k, c_j, r_t, \alpha)\, p(d_k|c_j, r_t, \alpha)., \tag{5.4}$$

where the inner summation is the frame-level mixture, the components of which are assumed Gaussian, $p(Y_i^t|d_k, c_j, r, \alpha) \sim N(\mu_{jkr}, \Sigma_{jkr}), \quad k = 1, \ldots, N_D$. (Formally, we again assume a random variable D_t for the identity of the mixture component at frame t, and denote the probability of it taking the value d_k as $p(d_k|c_j, r_t, \alpha) \doteq Pr(D_t = d_k|C_i = c_j, r_t, \alpha).$) This model allows both the independent-frame and previous segmental mixture model as special cases, by setting either N_C or N_D to 1 in (5.4), respectively. The ability of the model to tradeoff the number of segmental and frame-level mixture components will allow us to compare experimentally the contributions of these two parts of the model.

The second type of modification we investigated was concerned with reducing the dimension of the model's elements. Although our objective is to let the different mixture components model complete segmental trajectories, in practice, each such trajectory represents a point in a very high-dimensional space, and it may be necessary to reduce the dimension in order to train the models effectively. We have looked at two ways to do this, corresponding to two different independence assumptions that can be imposed.

First, we can model the trajectories within smaller, subphonetic units, as in the microsegment model described in [16, 36]. Taking this approach and assuming microsegments are independent, the probability for a segment becomes

$$\mathrm{p}(Y_i \mid \alpha) = \prod_j \sum_k w_{jk}\, \mathrm{p}(Y_{ij} \mid \alpha_{jk}), \tag{5.5}$$

where α_{jk} is the k^{th} mixture component of microsegment j and Y_{ij} is the subset of frames in Y_i that map to microsegment j. The independent-frame mixture SSM discussed in Chapter 4 can be viewed as a special case of this model, in which we restrict each microsegment to have just one stationary region and a corresponding mixture distribution.

A second way to reduce the parameter dimension is to continue to model the trajectory across a complete segment, but assume independence between subsets of the features of a frame. This case can be expressed in the general form of (5.5) if we reinterpret the Y_{ij} as vectors that have the same number of frames as the complete segment, but for each frame, only a specific subset of the original features. We give experimental results comparing these methods of dimensionality reduction later in the chapter.

Another possibility to reduce the number of parameters is to parameterize the trajectories, as suggested in work by Gish and Ng [26]. They have explored the use of least square estimates of linear and quadratic segmental trajectories of cepstral coefficients across phone segments. This trajectory modeling is embedded in a word spotting system that finds segments based on discontinuities in the speech signal, rather than using only phonetic segments, as in our approach. Within a segment, their approach differs from ours in that they estimate a smooth approximation of the trajectory, whereas our mixtures sample the trajectory with no smoothness constraints. Additionally, their model assumes that the error of observed data from the estimated trajectory is Gaussian and independent and identically distributed at each

frame, so that a single covariance matrix models the error statistics for all frames of a segment. In contrast, the segmental mixture of this chapter assumes the deviation from the trajectory mean is time-varying and models it with a separate covariance matrix for each region of the segment, motivated by differences we have observed in covariance determinants as a function of the region of a phone (e.g. there is more variation at the beginning and end of a phone than in the middle).

5.3 Training

In this section, we describe the training algorithm for the segment-mixture model. As discussed above, there are a number of variants of the proposed model, including the "two-level" mixture, the microsegment mixture and the independent feature-block model. The specific algorithm presented here is for the segment-mixture model with Gaussian mixture frame distributions. The modifications to the algorithm required when further independence assumptions are introduced, as in the case of microsegments or independent feature blocks, are straightforward. Essentially, the deterministic warping and known segmentation that are present in the estimation phase of training allow us to decompose the estimation of the distributions of such independent units into separate mixture estimation problems. Similarly, training for simpler versions of the model, such as using a single Gaussian for the frame densities, or the special case of the independent-frame model of Chapter 4, (i.e., a segmental mixture with just one mixture component), follow as special cases of the algorithm presented, as will be seen later.

Parameter estimation for segment-level mixtures follows a similar approach to the previous chapter's frame-level mixture training algorithm. As before, we use the iterative, two phase algorithm consisting of segmentation of the training database followed by maximum-likelihood (ML) parameter estimation based on that segmen-

tation, and the ML phase again uses the EM algorithm. As before, in the ML phase, the known segmentation of the database and the SSM's deterministic mapping of observations to model regions allows us to estimate the parameters for each of the mixture distributions of the model independently. Accordingly, for simplicity we will drop the explicit conditioning on phone α in the equations that follow.

There are four sets of parameters to be estimated in training this model: the weights for components of the segmental mixtures, $p(c_j)$, the weights for the Gaussian mixtures of the segment's stationary regions, $p(d_k|c_j, r)$, and the means and covariances, μ_{jkr}, Σ_{jkr}, of these Gaussians.

In training, for each segmental mixture distribution we accumulate two sets of counts, one for the segmental components and one for the region-dependent frame densities of the segmental components. At iteration p of the EM algorithm, the segmental count, η_{ij}, gives the estimate of the probability of segmental component c_j given observation i (observation i is now the complete phonetic segment $Y_i = (Y_i^1, Y_i^2, \ldots, Y_i^{L_i})$) :

$$\eta_{ij} = p^p(c_j|Y_i) = \frac{p^p(Y_i|c_j)\ p^p(c_j)}{p^p(Y_i)}, \tag{5.6}$$

where, once again, we use the superscript p to denote probabilities based on iteration p of the EM algorithm, and

$$p^p(Y_i) = \sum_j p^p(Y_i|c_j)\ p^p(c_j). \tag{5.7}$$

The "frame-based" count, ξ_{ijkrt}, gives the estimate of the probability $p(d_k|c_j, r_t, Y_i^t)$ that the k^{th} component of the Gaussian mixture distribution is used at region r_t, given that the segmental mixture component is c_j and the observation is frame t of Y_i :

$$\xi_{ijkrt} = p^p(d_k|c_j, r_t, Y_i^t) \tag{5.8}$$

$$= \frac{p^p(Y_i^t|d_k, r_t, c_j)\ p^p(d_k|c_j, r_t)}{p^p(Y_i^t|c_j, r_t)}. \tag{5.9}$$

The complete EM algorithm consists of iteratively computing these two types of counts for each observed segment in the training database and then re-estimating the parameters of the model using the formulas given below. The iteration is repeated until adequate convergence is observed.

The update formula for the segment mixture component weight based on the database counts is

$$\hat{p}\left(c_j\right) = \frac{\sum\limits_i \eta_{ij}}{\sum\limits_i \sum\limits_j \eta_{ij}}. \tag{5.10}$$

The frame-based mixture weight for region r, Gaussian component k, segmental component j is:

$$\hat{p}\left(d_k|c_j, r_t\right) = \frac{\sum\limits_i \eta_{ij} \sum\limits_{t(r)} \xi_{ijkrt}}{\sum\limits_i \eta_{ij} \sum\limits_k \sum\limits_{t(r)} \xi_{ijkrt}}, \tag{5.11}$$

where $t(r)$ represents the subset of frames in $1, \ldots, L_i$, that map to the region r under consideration.

The mean and covariance estimates are

$$\hat{\mu}_{jkr} = \frac{\sum\limits_i \eta_{ij} \sum\limits_{t(r)} \xi_{ijkrt}\, Y_i^t}{\sum\limits_i \eta_{ij} \sum\limits_{t(r)} \xi_{ijkrt}}, \tag{5.12}$$

and

$$\hat{\Sigma}_{jkr} = \frac{\sum\limits_i \eta_{ij} \sum\limits_{t(r)} \xi_{ijkrt}\, (Y_i^t - \hat{\mu}_{jkr})(Y_i^t - \hat{\mu}_{jkr})^T}{\sum\limits_i \eta_{ij} \sum\limits_{t(r)} \xi_{ijkrt}}. \tag{5.13}$$

The complete derivation of these equations is given in Appendix 5.A.

In the implementation of this algorithm, we must compute the segment observation probabilities, $p(Y_i|c_j)$. Since these are computed as a product of frame probabilities over all the frames of an observed segment, the resulting quantities can have a very wide dynamic range, yet because we use them in a mixture, we must be able to add them. Using a floating point representation – even double precision floating

point – quickly leads to problems of overflow and underflow in the algorithm. To avoid these problems, we use a log representation of probabilities. We have used the particular approach presented by Brown [11] that represents positive numbers over a very large dynamic range and allows for fast addition and multiplication of these quantities.

5.4 Experiments

For the experiments concerning this chapter's models, we used the speaker-independent Resource Management database described in Chapter 3. Given the large number of free parameters in the segment-mixture model, it was desirable to have enough training data to ensure robustness of the estimated models. For the Resource Management database, there are only half as many utterances for female speakers as there are for males, and for this reason, these experiments use just the male speakers.

Context-Independent Models

In the first context-independent experiment, we addressed the questions of how the model performed with varying numbers of segmental components and whether full covariance Gaussians were better than diagonal. Figure 5.1 shows the word error rate on the Feb89 test set for context-independent segmental mixture models using full versus diagonal covariance matrix Gaussians. This model uses segmental mixture components with single multi-variate Gaussians at each region (as opposed to Gaussian mixtures). The figure shows performance as a function of the number of segmental mixture components, plotted on a log scale, and it can be seen that performance improves as the number of components increases. We have plotted the full covariance performance only up to 32 components. In fact, with just 4 components, the full covariance estimates become singular, and the points in the plot for 8 com-

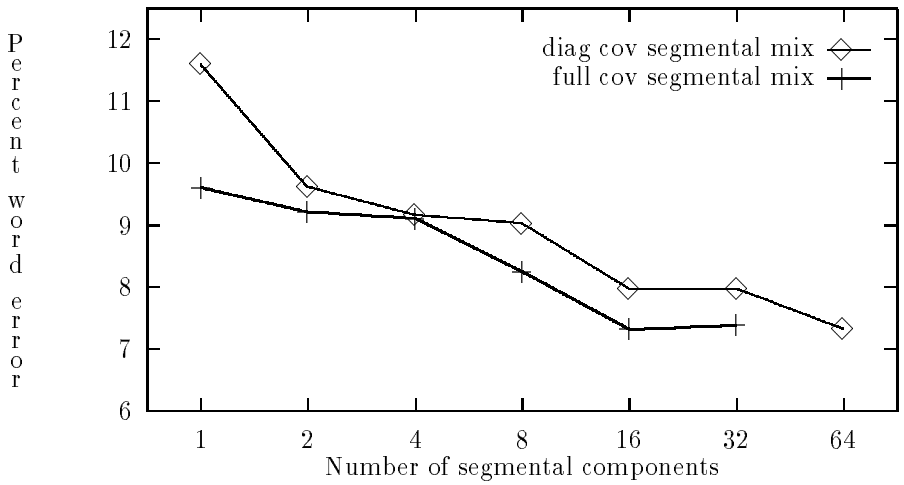

Figure 5.1: Word error for segmental mixtures as a function of the number of components, shown for full versus diagonal covariance models.

ponents and above actually use covariances that are tied within a model in order to make their estimates robust. This phenomenon is not too surprising given the much larger number of free parameters in the full covariance models: each full covariance matrix for a 30 dimensional vector has 465 free parameters that must be estimated compared with a diagonal matrix that has just 30. Although tying the full covariance matrices results in non-singular estimates, the computation using full covariance matrices in all cases is significantly larger than for diagonal models. Since we were able to match or exceed full covariance performance by using diagonal models with more mixture components, and to do so with less computation, we abandoned the use of full covariance models – all subsequent experiments used only diagonal covariance Gaussians.

Figure 5.2 gives the performance of context-independent segmental mixtures with three-region segments versus eight-region segments, plotted for varying numbers of

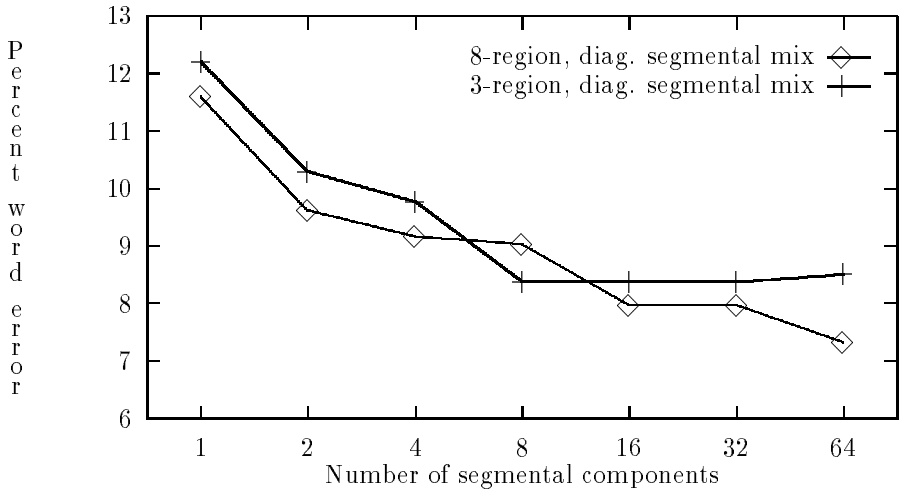

Figure 5.2: Word error rates for 3-region versus 8-region segmental mixture models

segment-level mixture components. As before, the frame densities for each region of a mixture component are single Gaussian. We can see that the three-region model has converged when there are eight segmental mixture components in the model, while the eight-region model continues to improve up to 64 mixture components. This may be due to the fact that the 8-region model, with more resolution across time, is better able to capture differences in segment trajectories that are necessarily "blurred" together in the 3-region model.

Figure 5.3 gives the performance of context-independent segmental mixtures when single Gaussians are replaced by Gaussian mixtures for the frame densities of the segmental mixture components. In this case, we kept the number of segmental mixtures fixed at 16 and increased the number of Gaussian mixtures from 1 to 16. The plot shows a substantial improvement for combining the two types of mixtures. The performance of the best case is better than both independent-frame versions of the SSM discussed previously, i.e., the baseline single Gaussian model and the non-segmental

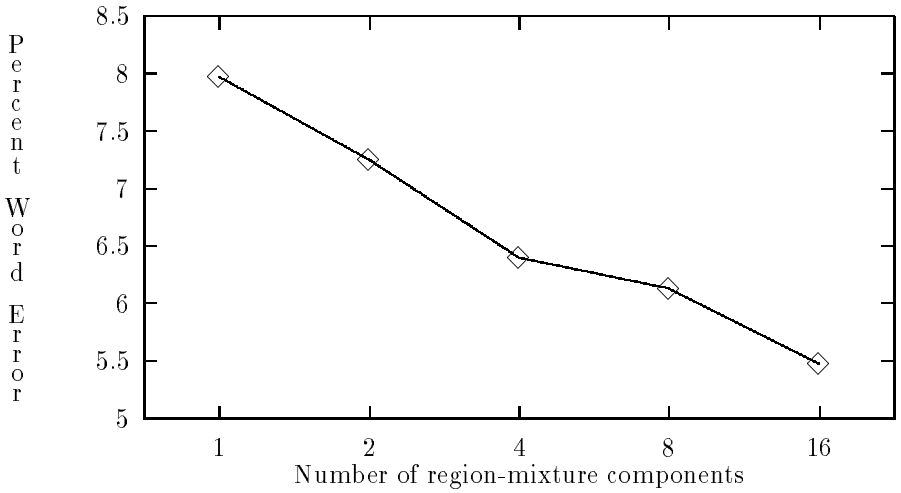

Figure 5.3: Performance of 16-component segmental mixture model with different numbers of Gaussian mixtures per model region.

Gaussian mixture model.

Context-Dependent Models

In initial experiments with context-dependent models, we attempted to use similar numbers of mixture components as in the context-independent case. However, this led to very poor performance in comparison with the best independent-frame models described in the previous chapter. One obvious problem in this case is simply the large number of free parameters that must be trained, i.e., with the same number of parameters per model, but many more context-dependent models, there is much less training per parameter. Accordingly we attempted to limit the number of parameters by reducing both the number of segmental components and the number of regions per segmental component.

We first looked at using segmental mixture models with the assumption that the

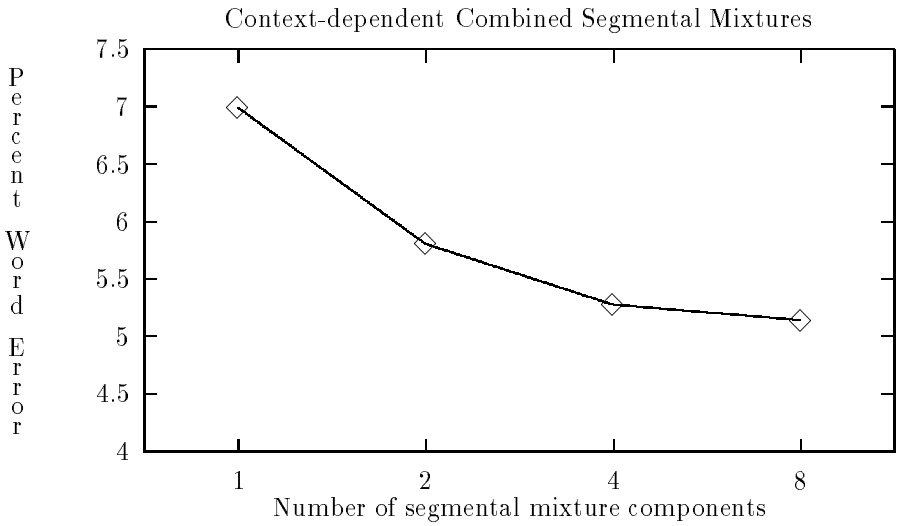

Figure 5.4: Performance of context-dependent segmental mixture model with a single Gaussian distribution per model region as a function of the number of segmental components.

density at each region of the model was a single, non-mixture Gaussian. Figure 5.4 shows the performance for the Feb89 male speakers as a function of the number of segmental components. To reduce the free parameters in the model, in addition to limiting the number of segmental components, the number of regions per segmental component was set to five as opposed to the eight region model used in most of the context independent experiments. Here we see that performance improves as the number of segmental components increases, but the rate of increase is small and seems nearly level at eight components. Furthermore, the performance of the best of these models is substantially worse than that of the best independent-frame Gaussian mixture models (e.g. 4.3% word error rate for the untied independent-frame model, or, 3.6% for the best case, covariance-tied model).

Although the results displayed for eight components are not especially good, they

are actually optimistic in that this model was tested several times to find conditions that avoided very poor performance. In particular, when run with no modifications, the mixture training algorithm quickly converged to parameters near the initial estimate that gave poor performance on the independent test set. Simply padding the mixture weights proved inadequate to prevent this rapid convergence, since the mixture weights are of much smaller dynamic range than the mixture density values, and the weights consequently play only a minor role in determining the updated parameter values in training.

When the estimates converge too quickly, as in this case, it is because the variances become narrow enough that training samples that are not near the current estimate of a mixture mean have virtually no effect on the next training iteration's estimate of that mean. Therefore, to delay convergence, we forced the variances to remain wide for the first few training iterations. Specifically, for the results shown in the figure for eight components, we multiplied the estimated covariances from the first two iterations by 1.4 and 1.2, respectively, and held the covariance estimates fixed after the 2nd iteration. The unimpressive performance despite these measures indicates the difficulty of robustly estimating models in such a high dimensional space.

Another contributor to the poor performance in this model is the assumption that the region densities within a segmental component are single Gaussians In the next set of experiments, we used a segmental mixture model with a small number of segmental components to try to control the number of free parameters, but instead of a single Gaussian at each region, we used mixture Gaussians. Figure 5.5 shows the word error rate on the Feb89 male speakers for models using two segmental components with the number of Gaussians in the region mixtures varying from one to eight. As in the previous experiment, we again used five regions per segmental component and the variance estimates were prevented from becoming too narrow. In the best case, for region mixture densities with four Gaussians, performance is nearly

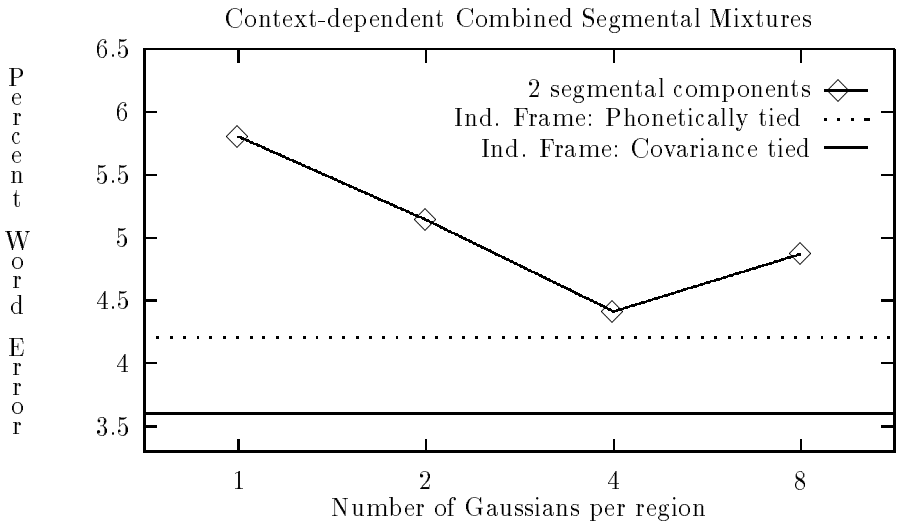

Figure 5.5: Performance of context-dependent segmental mixture model with two segmental components per model region as a function of the number of Gaussian mixture components per model region.

the same as for the phone-tied, independent-frame, mixture SSM, indicated in the figure by the dashed line at 4.2% word error, though it is still substantially worse than the covariance-tied independent-frame result of 3.6% error. For eight mixtures per region, performance degrades despite an increase in the training likelihood. This drop is again evidence of an overtuning effect when there are too many parameters to estimate robustly.

It is interesting to note that the difficulties in obtaining good estimates with this model are not caused solely by the raw number of free parameters. In fact, both of the segmental mixture models discussed above actually have fewer parameters than do the independent-frame models of the previous chapter. Although the experiments we ran are not completely definitive on this, the explanation for this phenomenon is most probably related to the modeling of trajectories by separate mixture components and

the resulting split of training data for common acoustic regions.

In particular, linguistic analysis of the acoustic properties of phones indicates that there is often a "target" in the acoustic space of a phone that is typically reached (or at least approached most closely) near the center of a phonetic segment. The edges of the segment diverge from the target due to coarticulatory effects with neighboring phones, and the acoustic trajectories discussed earlier can be described in terms of the movement towards and away from such targets. To the extent that the segmental mixture is successful in modeling these trajectories, however, a new problem arises. In this case the training for the common target region is no longer shared across the separate segmental mixtures, possibly resulting in a poor model of this part of the phone. This phenomenon would be particularly bad for the context-dependent case where there is less training data per model, in general. In contrast, models that assume complete independence of frames will pool the data from every training sample to estimate parameters of every region of the model. Thus, the segmental components of the segmental mixture model make it more discriminating by requiring that all frames of a trajectory match the model well, but the price for this accuracy is difficulty modeling the regions of a phone that are less variable in the context-dependent case.

There are two obvious approaches to solving this problem. The first is to introduce a parameter sharing mechanism, such that regions of the model corresponding to target regions of the phone share training data. The second is simply to use more training data. For example, the Wall Street Journal database, which has been collected relatively recently, has significantly more training than Resource Management. Due to time limitations, we were unable to investigate these possible solutions in this work, though we expect it to be a fruitful area for future research.

5.A EM Algorithm for Segmental Mixtures

In this appendix, we derive the re-estimation formulae for segmental mixtures. Although no tying is assumed here, in practice, there may be situations where parameter tying is advantageous. In this case, the derivation given below is changed only slightly, essentially summing together counts of tied parameters.

The EM algorithm [15] requires that we iteratively update parameter estimates based on maximizing the expected value of the complete data conditioned on the observed data and the model in the current iteration. For mixture distributions, the complete data, X, consists of the observed features, Y, combined with the unobserved identity of the "underlying" mixture component that generated Y. In our case, Y is a sequence of segment observations, Y_i, and each of these in turn is a variable-length sequence $Y_i = Y_i^1, \ldots, Y_i^{L_i}$, where each frame in the segment, Y_i^t, consists of a q-dimensional feature vector. For the segmental mixture model under consideration, the unobserved component consists of the identity of the segmental mixture component, C_i, and a vector of frame-level mixture component identities, $\bar{D}_i = (D_1, \ldots, D_{L_i})$. $\bar{D}_i$ specifies for each frame of the observed segment, a particular Gaussian from the mixture distribution of the region that frame is mapped to. The complete data for a single segment is thus the vector $X_i = (Y_i, C_i, \bar{D}_i)$.

Let Θ denote the parameters of the model and $\Theta^{(p)}$ the estimate of Θ from the previous iteration of the EM algorithm. Then at each iteration of the algorithm, the expected value that we seek to maximize is

$$l(\Theta) = E_X\{\log p\,(X|\Theta) \mid Y, \Theta^{(p)}\} \tag{5.14}$$

Assuming independence of the observed segments, Y_i, and the corresponding complete

data elements, X_i,

$$l(\Theta) = \sum_i E_{X_i}\{\log p\,(X_i|\Theta) \mid Y_i, \Theta^{(p)}\} \tag{5.15}$$

$$= \sum_i E_{X_i}\{\log p\,(Y_i, C_i, \bar{D}_i|\Theta) \mid Y_i, \Theta^{(p)}\} \tag{5.16}$$

If we drop the explicit conditioning on Θ, and use the assumed independence of regions within the segment, the probability $p\,(Y_i, C_i, \bar{D}_i|\Theta)$ can be written as

$$\begin{aligned} p(Y_i, C, \bar{D}_i|\Theta) &= p(Y_i, C_i, \bar{D}_i) \\ &= p\,(C_i) \prod_{t=1}^{L_i} p\,(D_t|C_i, r_t)\; p\,(Y_i^t|D_t, C_i, r_t). \end{aligned} \tag{5.17}$$

In (5.17) we have explicitly conditioned on the region r_t that frame t is mapped to by the model's deterministic warping. Using the above, we write $l(\Theta)$ as

$$\begin{aligned} l(\Theta) = & \sum_i E\{\log p\,(C_i) \mid Y_i, \Theta^{(p)}\} \\ & + \sum_i E\{\sum_{t=1}^{L_i} \log p\,(D_t|C_i, r_t) \mid Y_i, \Theta^{(p)}\} \\ & + \sum_i E\{\sum_{t=1}^{L_i} \log p\,(Y_i|D_t, C_i, r_t) \mid Y_i, \Theta^{(p)}\}, \end{aligned} \tag{5.18}$$

We can expand each of the three summations in (5.18) separately, approximating the expected values using probability estimates from the current model, $\Theta^{(p)}$, and introducing Lagrange multipliers to ensure normalization of the estimated distributions. In the sequel, for random variables C_i and D_t, we use lowercase to denote the event that a random variable takes on a particular value, e.g., $p(c_j|Y_i) \doteq Pr(C_i = c_j|Y_i)$ and $p(d_k|c_j, r, Y_i) \doteq Pr(D_t = d_k|C_i = c_j, r, Y_i)$, and we use superscript p to denote probabilities based on model $\Theta^{(p)}$, e.g., $p^{(p)}\,(c_j|Y_i) = p\,(c_j|Y_i, \Theta^{(p)})$. With this, the first summation in (5.18) becomes

$$\begin{aligned} \sum_i E\{\log p\,(C_i) \mid Y_i, \Theta^{(p)}\} &= \sum_i \sum_j p^{(p)}\,(c_j|Y_i) \log p\,(c_j) \\ &= \sum_i \sum_j \eta_{ij} \log p\,(c_j) - \lambda\,[\,p(c_j) - 1] \end{aligned} \tag{5.19}$$

where η_{ij} is as defined in (5.6), and we have introduced the Lagrange multiplier λ.

In the equations that follow, we decompose summations that range over the frames in the i^{th} segment, $t = 1, \ldots, L_i$, into two summations, the first over the regions of the model, $r = 1, \ldots, N_R$, and the second over the set $t(r)$, defined to be the set of frames that map to region r in the segment, i.e., $t(r) = \{t : r_t = r\}$. With this, we expand the second summation of (5.18) as

$$\begin{aligned}
&\sum_i \sum_{t=1}^{L_i} E\{\log p\,(D_t|C_i, r_t) \mid Y_i, \Theta^{(p)}\} \qquad (5.20)\\
&= \sum_i \sum_{r=1}^{N_R} \sum_{t(r)} E\{\log p\,(D_t|C_i, r) \mid Y_i, \Theta^{(p)}\}\\
&= \sum_i \sum_{r=1}^{N_R} \sum_{t(r)} \sum_j \sum_k p^{(p)}\,(d_k, c_j|r, Y_i) \log p\,(d_k|c_j, r)\\
&= \sum_i \sum_j p^{(p)}\,(c_j|Y_i) \sum_k \sum_{r=1}^{N_R} \sum_{t(r)} p^{(p)}\,(d_k|c_j, r, Y_i^t) \log p\,(d_k|c_j, r)\\
&= \sum_i \sum_j \eta_{ij} \sum_k \sum_{r=1}^{N_R} \sum_{t(r)} \xi_{ijkrt} \log p\,(d_k|c_j, r)\\
&\quad - \sum_j \sum_{r=1}^{N_R} \rho_{jr}\,[\sum_k p\,(d_k|c_j, r) - 1],
\end{aligned}$$

where ξ_{ijkrt} is as defined in (5.9), and in the last line we have introduced the Lagrange multipliers ρ_{jr}.

Similarly, the third summation in (5.18) can be written

$$\begin{aligned}
&\sum_i E\{\sum_{t=1}^{L_i} \log p\,(Y_i|D_t, C_i, r_t) \mid Y_i, \Theta^{(p)}\}\\
&= \sum_i \sum_j p^{(p)}\,(c_j|Y_i) \sum_k \sum_{r=1}^{N_R} \sum_{t(r)} p^{(p)}\,(d_k|c_j, r, Y_i) \log p\,(Y_i^t|c_j, k, r)\\
&= \sum_i \sum_j \eta_{ij} \sum_k \sum_{r=1}^{N_R} \sum_{t(r)} \xi_{ijkrt} \log p\,(Y_i^t|c_j, k, r). \qquad (5.21)
\end{aligned}$$

Recombining the expanded elements of $l(\Theta)$, we have

$$l(\Theta) = \sum_i \sum_j \eta_{ij} \log p\,(c_j) \qquad (5.22)$$

$$
\begin{aligned}
&+\sum_i \sum_j \eta_{ij} \sum_k \sum_{r=1}^{N_R} \sum_{t(r)} \xi_{ijkrt} \log p\left(d_k|c_j, r\right) \\
&+\sum_i \sum_j \eta_{ij} \sum_k \sum_{r=1}^{N_R} \sum_{t(r)} \xi_{ijkrt} \log p\left(Y_i^t|c_j, k, r\right) \\
&-\lambda\left[\, p(c_j) \;-\; 1\right] \\
&-\sum_j \sum_{r=1}^{N_R} \rho_{jr}\left[\sum_k p\left(d_k|c_j, r\right) \;-\; 1\right].
\end{aligned}
$$

In order to choose parameters that maximize (5.22), we set its derivatives with respect to those parameters to zero and solve the resulting equations. For the segmental mixture weights we have

$$\frac{\partial\, l(\Theta)}{\partial\, \hat{p}(c_j)} \;=\; \sum_i \frac{\eta_{ij}}{\hat{p}(c_j)} - \lambda \;=\; 0. \tag{5.23}$$

Rearranging yields

$$\hat{p}\left(c_j\right) = \frac{\sum_i \eta_{ij}}{\lambda} \tag{5.24}$$

and applying the normalization constraint $\sum_j p(c_j) = 1$ to solve for λ, we obtain

$$\hat{p}\left(c_j\right) = \frac{\sum\limits_i \eta_{ij}}{\sum\limits_i \sum\limits_j \eta_{ij}}. \tag{5.25}$$

Similarly, for the Gaussian mixture weights of the segments' region-dependent distributions,

$$\frac{\partial\, l(\Theta)}{\partial\, \hat{p}(d_k|c_j, r)} \;=\; \sum_i \eta_{ij} \sum_{t(r)} \frac{\xi_{ijkrt}}{p(d_k|c_j, r)} - \rho_j \;=\; 0. \tag{5.26}$$

Rearranging and using the constraint $\sum_k p(d_k|c_j, r) = 1$ to solve for ρ_j we find,

$$\hat{p}\left(d_k|c_j, r\right) \;=\; \frac{1}{\rho_j}\left[\sum_i \eta_{ij} \sum_{t(r)} \xi_{ijkrt}\right] \tag{5.27}$$

$$\;=\; \frac{\sum\limits_i \eta_{ij} \sum\limits_{t(r)} \xi_{ijkrt}}{\sum\limits_i \eta_{ij} \sum\limits_k \sum\limits_{t(r)} \xi_{ijkrt}}. \tag{5.28}$$

We are assuming $p(Y^t|c_j,k,r) \sim N(\mu_{jkr},\Sigma_{jkr})$, i.e., components of the mixture distributions for each region have a q-dimensional Gaussian distribution with mean μ_{jkr} and covariance matrix Σ_{jkr}. In this case, the conditional log probability of frame Y_i^t, used in equation (5.22), is expressed as

$$\log p(Y_i^t|c_j,k,r) = \log\left[\frac{1}{(2\pi)^{q/2}|\Sigma_{jkr}|^{1/2}}\cdot\exp\left[-\frac{1}{2}(Y_i^t-\mu_{jkr})^T\,\Sigma_{jkr}^{-1}(Y_i^t-\mu_{jkr})\right]\right].$$

In the following, we use partial derivatives of vectors and matrices and use the notation $\frac{\partial}{\partial u}$ (or $\frac{\partial}{\partial A}$) to mean the vector (or matrix) whose respective elements are partial derivatives, $\frac{\partial}{\partial u_i}$ (or $\frac{\partial}{\partial A_{ij}}$). Using the identity

$$\frac{\partial}{\partial u}\,u^T A\,u = 2A\,u$$

we find

$$\frac{\partial l(\Theta)}{\partial\hat{\mu}_{jkr}} \;=\; \sum_i \eta_{ij}\sum_{t(r)}\xi_{ijkrt}\Sigma_{jkr}^{-1}(Y_i^t-\hat{\mu}_{jkr}) \;=\; \underline{0}, \tag{5.29}$$

where $\underline{0}$ is a q-dimensional zero vector. Note that the covariance matrix Σ_{jkr} is necessarily non-singular, and can thus be eliminated from (5.29). Solving for $\hat{\mu}_{jkr}$ produces

$$\hat{\mu}_{jkr} = \frac{\sum_i \eta_{ij}\sum_{t(r)}\xi_{ijkrt}\,Y_i^t}{\sum_i \eta_{ij}\sum_{t(r)}\xi_{ijkrt}}. \tag{5.30}$$

To solve for the covariance matrix estimate, we make use of the equalities

$$\frac{\partial}{\partial A}\,\log|A| = A^{-1} \tag{5.31}$$

and

$$\frac{\partial}{\partial A}\,u^T A\,u = uu^T. \tag{5.32}$$

Differentiation of l with respect to Σ_{jkr}^{-1} then yields

$$\frac{\partial l(\Theta)}{\partial\hat{\Sigma}_{jkr}^{-1}} \;=\; \sum_i \eta_{ij}\sum_{t(r)}\xi_{ijkrt}\,[\,\hat{\Sigma}_{jkr}-(Y_i^t-\hat{\mu}_{jkr})(Y_i^t-\hat{\mu}_{jkr})^T] \;=\; \underline{0}, \tag{5.33}$$

where $\underline{0}$ is a $q \times q$ matrix of zeroes. Rearranging this we obtain

$$\hat{\Sigma}_{jkr} = \frac{\sum_i \eta_{ij} \sum_{t(r)} \xi_{ijkrt} \, (Y_i^t - \hat{\mu}_{jkr})(Y_i^t - \hat{\mu}_{jkr})^T}{\sum_i \eta_{ij} \sum_{t(r)} \xi_{ijkrt}}. \tag{5.34}$$

Chapter 6

The Classification-in-Recognition Framework

In this chapter, we discuss a segmental approach to recognition based on the use of classification probabilities. We first present background on the topic, followed by the formulation of the model for context-independent recognition. The issues that arise in context-dependent modeling using classification probabilities are more complex than in more typical recognition approaches, and we discuss these in the third section of the chapter. We then present experiments evaluating the model followed by a discussion of some unresolved issues concerning this approach.

6.1 Background

6.1.1 Motivation and Overview

In this chapter we consider a segment modeling approach based on the use of posterior probabilities of phonetic labels conditioned on observed segments. Recalling the discussion in Chapter 2, the general criterion for statistical speech recognition methods is to choose the label sequence that maximizes the joint probability of labels, A,

observed speech, Y, and segmentation, S, i.e., choose labels A^* such that

$$A^* = \underset{A}{\text{argmax}} \max_{S} p(A, Y, S). \tag{6.1}$$

In the more typical statistical approach, including the methods presented in Chapters 4 and 5, this joint probability is estimated as in (2.10), as a product of the conditional probability of observations given label sequence multiplied by a language model score. In this chapter we consider the alternative decomposition of the joint probability,

$$A^* = \underset{A}{\text{argmax}} \max_{S} p(A|Y, S)\, p(S, Y). \tag{6.2}$$

Here, the joint probability is written as the product of a "classification score" – the probability of label sequence conditioned on the observation and segmentation – and a "segmentation score" – the probability of observation and segmentation. We will refer to this general approach as *classification in recognition* (CIR). Since the probability of observation sequence, $p(Y)$, does not affect the maximization over A and S, we can alternatively maximize the probability of phones and segmentation *conditioned* on observations, i.e., an equivalent expression for (6.2) is

$$A^* = \underset{A}{\text{argmax}} \max_{S} p(A|Y, S)\, p(S \mid Y). \tag{6.3}$$

Accordingly, we will refer to both $p(S \mid Y)$ and $p(S, Y)$ as segmentation scores.

To be able to use this model in practice, assumptions must be introduced to the above general equations for the formalism. As developed later in the chapter, there are several different possibilities in this regard, but in each case the assumptions we make will lead to an expression of the label sequence posterior probability, $p(A|Y, S)$, as a product of segment posterior probabilities. For example, under the simple context independent assumptions of Section 6.2, we will write this probability as a product of individual phone classifications, each of which is conditioned only on its hypothesized

segmentation, s_i, and the corresponding speech segment, Y_i, i.e.,

$$p(A|Y,S) = \prod_{i=1}^{N} p(a_i|Y_i, s_i). \tag{6.4}$$

We will provide a more detailed derivation of this case and context-dependent cases of the model in later sections of the chapter.

There are several potential benefits for using posterior probabilities in acoustic modeling. First, there are a number of classifiers with interesting properties that naturally produce such probabilities. Two examples that have received considerable recent attention are classification and regression trees [10] and multi-layer perceptron (MLP) neural networks (e.g., [71]). Since these classifiers do not normally produce estimates of the conditional densities typically used in recognition, one method for exploring their potential is to reformulate our acoustic modeling approach to use posterior distributions instead.

In addition to allowing the use of specific classification techniques, posterior distributions may have generally useful properties for modeling the statistics of complete segments. One potential benefit may be their ability to condition on functions of the observed speech segment, rather than on the complete segment itself, i.e., we may be able to assume that

$$p(a_i \mid Y_i, s_i) = p(a_i \mid f(Y_i), s_i), \tag{6.5}$$

where $f(Y_i)$ is some representation of Y_i that retains the essential information about the phone's acoustic properties, but may have some other desired characteristics as well.

Representations of the segment that may be of interest in this regard include linguistically motivated features, reduced dimensional representations, as well as fixed-length representations. In general, reduced dimensional representations, if they do not remove important information, help alleviate the problems of parameter estimation when less training data than desired is available. This inadequacy of available

training is always a problem in large vocabulary recognition, but particularly so for segment models where we would like to estimate distributions for complete segments, rather than individual frames.

The specific class of fixed-length representations has similar desirable properties. These representations are typically a transformation of the segment to reduce its dimension, with the distinction that all observed segments are transformed to a space of the same dimension. There are two related advantages to this approach. First, some of the classifiers mentioned earlier, including MLPs and classification trees, generally require their input to be of fixed length. In order to classify a complete segment with these classifiers, the segment observation must first be transformed to some fixed-length representation. The second, more general advantage is that we can easily formulate a statistical model to conform to a fixed dimensional feature set. In this case, when training, every instance of a particular phone can be used to estimate a single model, regardless of the observed token length, i.e., we do not need to try to estimate a separate model for every possible observed length or address the problem of a hidden trajectory.

Although we have so far focused on the potentially useful properties of posterior probabilities, it should be noted that the segmentation score component is also critical to making this approach work. In particular, when speech is recognized, the scores for many different possible phone sequences are compared. If we were to use just the posterior probabilities, $p(A \mid Y, S)$, to compare these sequences, the probabilities would vary roughly in proportion to the number of phones in each. In order to make sequences comparable, the scores must be normalized, and we can think of the segmentation score as serving this purpose.

The ultimate goal of the CIR approach is to improve recognition performance through the inclusion of segmental information. However, because this area is much newer than the more usual conditional approach, there are a number of outstanding

issues that must be resolved before it can be applied successfully. In particular, less attention has been paid in the past to the careful derivation of the criterion for posterior-based recognition in the context-dependent case. For this reason, much of the theoretical and experimental work in this chapter focuses on the development of a framework using posterior distributions that is probabilistically well-defined and that allows the use of context-dependent phonetic models.

Before proceeding to a more detailed description of our model, we first present a review of work by others in this area.

6.1.2 Related Work

Interest in using posterior probabilities for speech recognition has recently increased greatly, in large part due to the increased attention paid to MLP neural networks. Since these networks can be trained to approximate classification probabilities of a label given input, it is not surprising that they have been applied in statistical approaches to speech recognition. Some researchers have used MLPs to estimate the output probabilities of HMMs and in the process looked at the more general question of normalizing posterior probabilities for use in a conditional distribution setting (e.g., [49]). There have also been two recent models proposed, the BBN segmental neural network (SNN) [1, 83] and the NYNEX/MIT stochastic explicit segment model (SESM) [46, 47], that attempt to use segmental measures directly in neural networks. These methods have a number of similarities to the model presented in this chapter. We briefly describe them next.

The SNN is one example of a model that retains segmental classification properties for use in the recognition problem. In this approach, an N-best algorithm is used in which an HMM computes the top N scoring hypotheses in the same manner as described earlier, and the SNN rescores these hypotheses. For each phoneme in the

hypothesized sequence, the corresponding segment of speech is scored by a neural network trained to respond to instances of the phoneme with a "1" output. The MLP neural networks used in this work are trained with variants of the back propagation algorithm [71]. As noted by several authors [25, 9, 54, 59], neural networks trained with a mean square error criterion, as in back propagation, will approximate the posterior class probabilities of the targets for which they are trained. In the original SNN, the score for the complete hypothesis is the sum of the logarithms of the segment scores, and using the above approximation, this score can be interpreted as

$$\log p(A \mid Y, S) = \sum_i \log p(a_i \mid f(Y_i), s_i).$$

In this formulation the phonemes are assumed to be conditionally independent. Comparing this to (6.2), it can be seen that there is no explicit term for the probability of segmentation, $p(Y, S)$. This version of the SNN assumes that the HMM will have eliminated very poor segmentations from consideration, so the joint probability of the segmentation and observation used in rescoring is implicitly assumed to be uniform over the N hypotheses. In more recent formulations of the SNN, an explicit normalization has been introduced that serves the normalization purpose. We will describe this further when we return to the question of normalization in context-dependent models.

Another neural network approach that incorporates segmental features is the work on the SESM at NYNEX and MIT, which builds on Leung's earlier studies of MLPs and their properties for phonetic classification [45]. In the SESM, the MLPs used for classification may be trained to classify phonemes using a variety of inputs, including cepstral coefficients or linguistically motivated acoustic-phonetic features. In each case, the input to the MLPs are vectors of fixed dimension.

In the SESM, the recognition problem is explicitly decomposed into finding the probability of classification times the segmentation probability. Using the property

that MLPs will converge to approximations of the posterior distribution, the posterior probability of the jth phoneme of the sequence given the jth segment of the observation sequence, $p(a_j \mid s_j, Y_j)$, is estimated by an MLP. To compute the probability of segmentation, $p(s_j \mid Y_j)$, the system uses a different specially trained MLP. At each frame, this MLP attempts to compute the probability of a segment boundary given the input observations and the possible broad phonetic classes on either side of the hypothesized boundary. The segmentation probability, $p(s_j \mid Y_j)$, is then assumed to be the product of probabilities of no boundaries at any internal frame of the segment, times the probability of a boundary at the left and right end frames of the segment.

Another area of recent research that has relevance to the model described in this chapter, though in a less direct manner, is the effort to use MLPs directly in HMM's. One form this has taken is to estimate the class conditional observation densities of an HMM, using posterior probabilities produced by an MLP that are normalized by class prior probabilities [9, 8]. In recent work [8], the correlation within a segment is modeled using a window of several frames around the current frame as input to the MLP. The repeated use of the same frame of input implicit in an approach using overlapping windows violates the HMM assumption of conditional independence of observations given state sequence, though in practice it may be useful. Although the model we describe in this chapter takes a quite different approach than this, both methods must contend with the issue of using posterior probabilities where conditional probabilities are typically used. The normalization used in the HMM/MLP approach is similar to the normalization used in recent BBN SNN triphone work, and is, in general, a useful reference point for methods that are based on posterior probabilities.

In addition to the above work concerning posterior distributions, the recent work of Goldenthal and Glass [27] has addressed another issue important to the model described in this chapter, namely the problem of capturing the correlation of complete segments using a fixed-dimensional model. Their method classifies vowels by

measuring the error from an instance of a vowel to its hypothesized trajectory model and modeling the error using a full-covariance Gaussian that includes the correlation across the length of the segment. They use a fixed-length feature representation in order to simplify the model, but they do not normalize for dimensionality differences among different segmentations.

6.2 Context-Independent CIR Formulation

As stated earlier, in the CIR approach, we compute the joint probability of phones, observation, and segmentation, $p(A, Y, S)$, as the product of a classification score, $p(A \mid Y, S)$, and a segmentation score, $p(S, Y)$. Although there are a wide variety of possible formulations for the two components, the central focus of this chapter will be on the development of models to investigate the general characteristics of posterior-based recognition, and to directly compare their performance with more traditional conditional models, like the independent-frame SSM. Toward this end, the models we describe next do not fully exploit the segmental modeling capabilities of the CIR approach, but instead are designed to minimize non-essential differences from independent-frame models, so that we may better understand the effects due to the methods' differing assumptions. Results of experiments comparing the models will be given later in the chapter.

6.2.1 Classification component

To obtain the context-independent classification probability, we first assume that the phones in the sequence A are independent given the observations and segmentation:

$$p(A \mid Y, S) = \prod_{i=1}^{N} p(a_i \mid Y, S). \tag{6.6}$$

The segmentation S was defined earlier to be the sequence $S = s_1, \ldots, s_N$, where

component s_i represents the begin and end time of speech segment i. S can equivalently be defined as the sequence of segment durations, $S = \tau_1, \ldots, \tau_N$, where $\tau_i = |s_i|$ is the duration of segment i. This formulation is more convenient mathematically and we will therefore use it in the sequel.

With this change, we further simplify equation (6.6) by assuming that the posterior probabilities for individual phones depend only on our representation of the corresponding speech segment, $f(Y_i)$, and the segment's duration, τ_i. This leads to the following expression for the context-independent classification probability:

$$p(A \mid Y, S) = \prod_{i=1}^{N} p(a_i \mid f(Y_i), \tau_i). \tag{6.7}$$

As mentioned before, posterior phone probabilities can be estimated using a number of techniques, including MLPs and classification trees. However, in order to compare the general characteristics of the CIR framework with that of earlier models, we estimated the posterior probabilities in the CIR approach using a modified version of the SSM.

In this approach, we chose to represent a speech segment as a fixed-length sequence of frames obtained by linearly time sampling the frames present in the original segment. Sampling to a fixed dimensional sequence in this way preserves the basic frame-based format common to earlier SSM models while allowing the possibility of modeling segmental characteristics with alternative classification techniques that use fixed dimensional inputs. With this representation, it is a simple change to the independent-frame SSM to compute $p(f(Y_i) \mid \tau_i, a_i)$ instead of the usual $p(Y_i \mid \tau_i, a_i)$: rather than taking the product of probabilities of all frames observed in a segment, we use only those chosen by the deterministic sampling. We can then apply Bayes' rule to obtain the desired posterior probabilities from the SSM density estimates:

$$p(a_i \mid f(Y_i), \tau_i) = \frac{p(f(Y_i) \mid a_i, \tau_i)\, p(a_i, \tau_i)}{\sum_{a_i} p(f(Y_i) \mid a_i, \tau_i)\, p(a_i, \tau_i)}. \tag{6.8}$$

For this work, we used the simplest formulation of the independent-frame SSM, in which the density for each frame of the model is assumed to be a single multivariate Gaussian. As in earlier models, we can compute $p(a_i, \tau_i)$ as a product of duration probability and phone prior:

$$p(a_i, \tau_i) = p(\tau_i | a_i)\, p(a_i). \tag{6.9}$$

The use of Bayes' rule in (6.8) is certainly not the most efficient method for computing posterior probabilities, nor does it take advantage of some of the potentially interesting features of classifiers such as neural networks or classification trees. However, for our initial exploration of the CIR approach, it allows the direct comparison of results using this method with those from independent-frame models based on similar underlying density estimates and assumptions. Additionally, it allows the straightforward use of a number of powerful frame-based distributions, such as Digalakis' dynamical system model [16] or the mixture density models described earlier. A related approach that is possible with classification probabilities is to use *fixed-length* segmental mixtures. Unlike the segmental mixtures of Chapter 5, these could capture both segmental trajectories and explicit inter-frame correlation (for example, by using a full covariance model that captures correlation across complete segment-length observations). Although such a model would introduce many more free parameters to estimate, it might be possible to use subsets of the features for dimensionality reduction and still capture segmental effects.

6.2.2 Segmentation Component

As noted earlier, in some other research systems that use classification probabilities, an explicit segmentation score is not always included. These systems depend on a first pass component of the system to give the classifier only "reasonable" segmentations of the input, and the lack of any explicit segmentation score is equivalent to the

assumption of a uniform probability distribution over the segmentations. In contrast, the model we have explored explicitly includes a segmentation probability in order to normalize the classification component in a probabilistically well-defined manner. There are a number of possible ways to estimate this probability.

For the context-independent case, we assume that the individual phone durations that define the segmentation are independent,

$$p(S) = \prod_i p(\tau_i), \tag{6.10}$$

and that the speech segments, Y_i are conditionally independent given their durations,

$$p(Y \mid S) = \prod_i p(Y_i \mid \tau_i). \tag{6.11}$$

Combining these assumptions, the segmentation probability is written

$$p(S, Y) = \prod_i p(Y_i, \tau_i).$$

A simple approach to computing the i^{th} segment probability is to use the SSM to find marginal probabilities,

$$\begin{aligned} p(Y_i, \tau_i) &= \sum_{c \in \Omega} p(Y_i, \tau_i, c) && (6.12) \\ &= \sum_{c \in \Omega} p(Y_i \mid \tau_i, c)\, p(\tau_i \mid c)\, p(c). && (6.13) \end{aligned}$$

The summation is over all the elements of a label set, Ω, for which (6.12) is well defined. In general, Ω can be any set that defines a valid mixture probability for $p(Y_i, \tau_i)$, for instance, a set of broad phonetic classes, the set of all phonemes, or a set of clustered context-dependent units. In the experiments described in a later section, we take Ω to be simply the set of phonemes.

A second method we investigated for computing the segmentation probability, similar to that presented in [46], is to find the posterior probability $p(S \mid Y)$ using

distributions that model the presence or absence of a segment boundary at each frame based on local features. In this approach, the segmentation probability is written as

$$p(S \mid Y) = \prod_i p(s_i \mid Y_i) \tag{6.14}$$

and the probability of an individual segment of length L is

$$p(s_i \mid Y_i) = p(b_L \mid Y_i) \prod_{j=1}^{L-1} p(\overline{b_j} \mid Y_i), \tag{6.15}$$

where b_L is the event that there is a boundary after frame L and $\overline{b_j}$ is the event that there is not a boundary after the jth frame of the segment. In estimating the frame boundary probabilities, we assume that the probability of a boundary at frame j depends only on frames near the boundary, y_j and y_{j+1}, not the complete segment. We can then estimate the frame boundary probabilities using Bayes' rule as

$$\begin{aligned} p(b_j \mid Y_i) &= p(b_j \mid y_j, y_{j+1}) && (6.16)\\ &= \frac{p(y_j, y_{j+1}, b_j)}{p(y_j, y_{j+1})} && (6.17)\\ &= \frac{p(y_j, y_{j+1} \mid b_j)\, p(b_j)}{p(y_j, y_{j+1} \mid b_j)\, p(b_j) + p(y_j, y_{j+1} \mid \overline{b_j})\, p(\overline{b_j})}. && (6.18) \end{aligned}$$

Dividing numerator and denominator by $p(y_j, y_{j+1} \mid b_j)p(b_j)$ we have

$$p(b_j \mid Y_i) = \frac{1}{1+L}$$

where L is the ratio

$$L = \frac{p(y_j, y_{j+1} \mid \overline{b_j})p(\overline{b_j})}{p(y_j, y_{j+1} \mid b_j)p(b_j)}.$$

In this, $p(b_j)$ and $p(\overline{b_j})$ are just the prior probabilities of observing a boundary or no boundary at a frame. The conditional probability of observation given no boundary is computed as

$$p(y_j, y_{j+1} \mid \overline{b_j}) = \sum_\beta p(y_j, y_{j+1} \mid \beta)\, p(\beta) \tag{6.19}$$

where β ranges over five manner-of-articulation phoneme classes (stops, nasals, fricatives and affricates, liquids and glides, and vowels) and the separate class, silence. The probability $p(y_j, y_{j+1} | \beta)$ is computed assuming a Gaussian density over the single vector composed of the concatenated cepstral coefficients from the two frames, and the density is estimated from the set of non-boundary events in the training data. Similarly, the probability of observation given boundary is estimated as

$$p(y_j, y_{j+1} \mid b_j) = \sum_{\beta_1} \sum_{\beta_2} p(y_j, y_{j+1} \mid \beta_1, \beta_2)\, p(\beta_1, \beta_2), \tag{6.20}$$

where $p(y_j, y_{j+1} | \beta_1, \beta_2)$ is estimated from occurrences of boundaries between classes β_1 and β_2 in the training data.

The two segmentation models presented above each have different advantages. The first method, based on marginal distributions, makes use of the complete set of SSM phone models in determining likely boundaries for each segment and hence may have a more complete model of the speech process. On the other hand, the second approach uses models explicitly trained to differentiate between boundary and non-boundary acoustic events and it makes explicit use of information outside the current segment observation in determining the likelihood of a boundary. When we compare these two models experimentally, it should be noted that the explicit boundary model has far fewer free parameters than the marginal model (about a factor of 10), and this may affect performance too.

If we combine the marginal segmentation approach, as given in (6.13), with the classification probability of (6.8), we get

$$A^* = \underset{A}{\text{argmax}} \max_{S} \prod_i \frac{p(f(Y_i) \mid a_i, \tau_i) p(a_i, \tau_i)}{p(f(Y_i), \tau_i)} p(Y_i, \tau_i) \tag{6.21}$$

It is clear that with the simplifying assumptions we have made and using the SSM model with Bayes' rule, the CIR model of this section is similar to the conditional density, context-independent SSM. In particular, if the function $f(Y_i)$ represents the

segment well, we expect the components $p(Y_i, \tau_i)$ in the numerator and $p(f(Y_i), \tau_i)$ in the denominator to have very similar properties, with differences essentially proportional to the dimensionality of the two observation vectors. So we expect their ratio to essentially serve as a scaling factor for the rest of the product, roughly L_i/M, where L_i is the segment length and M is the representation dimension. The remaining factor, $\prod_i p(f(Y_i) \mid a_i, \tau_i)\ p(a_i, \tau_i)$, is like the ordinary SSM except for its use of the representation $f(Y_i)$ in place of the original observation Y_i. Essentially then, with the given assumptions, this model tests the adequacy of the sampled representation of the speech.

6.3 Context-Dependent Models

The independence assumptions made in the previous section lead to a straightforward version of the general formalism. However, the inaccuracies of these assumptions are known to limit recognition performance, so we now consider less restrictive forms of the CIR model.

In general, to the extent that there is adequate training data to estimate models, a recognizer becomes more accurate as it accounts for the effects of neighboring labels and observations. One example of this is the use of context-dependent models in conditional densities: the assumption that the acoustic observation depends on not just the phoneme but the complete phonetic context reflects the actual speech process more faithfully and leads to higher performance models. Another example is the improvement in recognizer performance due to modeling the dependence of the elements a_i of the label sequence via a language model [3]. This principle can be extended in other ways as well.

We consider first the classification component, $p(A|Y, S)$ under less restrictive assumptions. As was the case with independent segments, here we can generalize the

type of classifiers that we may use by allowing probabilities conditioned on functions of the observation sequence, not just the observations themselves, i.e., $p(A|Y,S) = p(A|f(Y),S)$. As before, if we are interested in classifiers that use fixed-length inputs, we can consider functions that have appropriate fixed-dimensional ranges. Note that in the context of reduced independence assumptions, the functions can, in principle, take into account the acoustic observations from the entire utterance, not just the segment for a hypothesized phoneme.

The posterior probabilities can be computed using the same classification methods discussed earlier, including tree-based classifiers, neural networks, and Gaussians. For the Gaussian case, it is again possible to use class conditional densities in the classifiers, analogous to (6.8):

$$p(A|f(Y),S) = \frac{p(f(Y)|S,A)\,p(S,A)}{p(f(Y),S)}. \tag{6.22}$$

To simplify the following discussion, for the label sequence $A = a_1, ..., a_N$, let A_{L_i} be the "left subsequence" preceding a_i :

$$A_{L_i} = \begin{cases} a_1, ..., a_{i-1} & \text{if } i \neq 1 \\ \phi & \text{if } i = 1 \end{cases} \tag{6.23}$$

Similarly, let A_{R_i} be the "right subsequence" following a_i:

$$A_{R_i} = \begin{cases} a_{i+1}, ..., a_N & \text{if } i \neq N \\ \phi & \text{if } i = N \end{cases} \tag{6.24}$$

The "classification" expression can then be written using the chain rule of probability, as

$$p(A|Y,S) = \prod_{i=1}^{N} p(a_i|A_{L_i},Y,S) \tag{6.25}$$

$$= \prod_{i=1}^{N} p(a_i|A_{R_i},Y,S) \tag{6.26}$$

Note that in contrast to the conditional model, both of the above forms of the CIR model readily allow the use of complete "observation context;" that is, we can condition the probability of each phone on the entire observation and segmentation sequences, Y and S.

It is clear from the last equation that the CIR formalism can be used with either left- or right-context-dependent models. In particular, we can use left-phone or right-phone-dependent models or we can condition on a more general function of the left or right subsequence of labels. What is not obvious at this point is how to condition simultaneously on both left and right phonetic contexts, as in conditional density triphone models. In a later section, we will examine some of the issues concerning the use of this type of joint context modeling with posterior probabilities. First, though we present the details of a left-context model that was used in experiments.

6.3.1 Left-Context Model

The left-context model described in this section allows us to explore characteristics of context modeling in the CIR approach without having to completely resolve some of the issues posed by higher order models (i.e. those that use left *and* right context jointly) that we discuss in a later section. The choice of a left-context model instead of right-context for this purpose was essentially arbitrary. Although linguistic analysis indicates that many coarticulatory effects are anticipatory in nature, suggesting that right-context models might be more powerful, our results with other context-dependent systems indicate that there is no real difference in performance between left and right context models.

Classification Probability

In the first version of the left-context CIR model we examined, the posterior probability was simplified from (6.25) as:

$$p(A|Y,S) = \prod_{i=1}^{N} p(a_i|f(Y_i),\tau_i,a_{i-1}), \tag{6.27}$$

where, as before, $f(Y_i)$ is the time-sampled representation of the i^{th} segment Y_i. In (6.27) as in the context-independent case, we have assumed the probability of a phone depends on the corresponding segment, $f(Y_i)$, and its duration, τ_i, but in addition we assume dependence on the identity of the previous hypothesized phone, a_{i-1}.

The above assumptions for the classification probability, although apparently "natural" for left-context modeling, lead to a formulation that is not identical to the left-context version found in conditional models like the independent-frame SSM. In those models, the left-context assumptions for the joint probability of observation, segmentation, and label sequences can be expressed as

$$\begin{aligned} p(Y,S,A) &= p(Y|S,A)\, p(S|A)\, p(A) && (6.28)\\ &= \prod_i p(Y_i|\tau_i,a_i,a_{i-1}) \prod_i p(\tau_i|a_i,a_{i-1}) \prod_i p(a_i|a_{i-1}) && (6.29)\\ &= \prod_i p(Y_i,\tau_i,a_i \mid a_{i-1}) && (6.30) \end{aligned}$$

where, in addition to the assumptions made in going from (6.28) to (6.29), we assume that $p(Y_1,\tau_1,a_1 \mid a_0) = p(Y_1,\tau_1,a_1)$.

Clearly, the use of a function of the segment observation, $f(Y_i)$, rather than the segment itself is a basic difference between the two. Ignoring this assumption for the moment, it is interesting to ask if the methods are otherwise identical. Using the classification score in (6.27) modified to use the complete observation, and deferring for the moment the question of how best to compute the segmentation score, we can

write the joint probability for an utterance in the CIR formalism as

$$\begin{aligned} p(Y,S,A) &= p(Y,S)\; p(A|Y,S) \\ &= p(Y,S) \prod_i p(a_i|Y_i,\tau_i,a_{i-1}) \qquad (6.31) \\ &= p(Y,S) \prod_i \frac{p(Y_i,\tau_i,a_i \mid a_{i-1})}{p(Y_i,\tau_i \mid a_{i-1})}. \qquad (6.32) \end{aligned}$$

Since the numerator terms in the latter expression's product are identical to expression (6.30) for the non-CIR version of the complete probability $p(Y,S,A)$, it is clear that the different ways of introducing left-context assumptions result in final forms that are not strictly equivalent. They are equivalent if $p(Y,S) = \prod_i p(Y_i,\tau_i \mid a_{i-1})$, which will not be true for the left-context case developed here. We will return to this issue later in the chapter when we discuss results comparing the two models.

Although the assumptions for the formalisms differ in this case, there is no particular reason to believe one set of assumptions is correct and the other incorrect. Since they are both approximations, and neither violates the rules of probability, the evaluation of their relative merit becomes an empirical question of which approach gives the more faithful approximation to the process being modeled. In fact, our initial assumption was that the difference between them should be insignificant in terms of system performance. Later, we will present results of the CIR and conditional left-context models, and offer some interpretations of how the different assumptions affect performance.

Segmentation Probability

There are several options for computing the segmentation component of the left-context model. The simplest possibility is just to continue to use either of the methods described for the context-independent model. Combining this with the left-context posterior probability would give some of the benefit of left-context models.

A somewhat more powerful approach is to modify the segmentation probability of (6.12) to compute the marginal over all left-context models instead of over context-independent models, i.e.,

$$p(Y,S) = \prod_i \sum_{\{\omega_i\}} \sum_{\{\omega_{i-1}\}} p(Y_i, \tau_i, \omega_i, \omega_{i-1}), \tag{6.33}$$

where the summations are each over the phone alphabet, Ω. Since left-context models generally have better acoustic resolution than context-independent models, the resulting marginal probability should be more accurate too.

Note that although this formulation uses more accurate models, it treats the marginal computed for the i^{th} segment, $p(Y_i, \tau_i)$, as independent from all other segments. A more accurate approach is to use the marginal of the sequence probability, rather than a product of individual segment marginal probabilities. In general we can sum over sequences of phones:

$$p(Y,S) = \sum_{\{B \in G\}} p(Y,S,B), \tag{6.34}$$

where $\{B \in G\}$ is the set of all valid label sequences in some grammar, G. If we make the same assumptions as in the non-CIR left-context case, $p(Y,S,B)$ has the same simplification as in (6.30), so

$$\sum_{\{B \in G\}} p(Y,S,B) = \sum_{\{B \in G\}} \prod_i p(Y_i, \tau_i, b_i \mid b_{i-1}). \tag{6.35}$$

This equation has a dynamic programming (DP) solution:

$$\begin{aligned} \alpha_j(k) &= p(Y_1, \ldots, Y_j, \tau_1, \ldots, \tau_j, b_j = \omega_k) \\ &= \sum_{i=1}^{|\Omega|} p(Y_j, \tau_j, b_j = \omega_k | b_{j-1} = \omega_i)\ \alpha_{j-1}(i) \end{aligned} \tag{6.36}$$

where ω_i is an element of the label alphabet, Ω, and the summation is over the indexes of the alphabet. The number of states that must be maintained at each step of the algorithm for $|\Omega|$ possible labels is $|\Omega|^2$ and the computation to sum all

sequences of length N is thus $O(N|\Omega|^2)$. Although this algorithm is very efficient in comparison with the exponential computation associated with the naive algorithm to compute (6.35) directly, it represents much more computation than any of the previous segmentation scoring algorithms described.

6.3.2 Joint Left and Right Context

Having considered the left-context case, we examine next the question of conditioning classification probabilities jointly on both the left and right neighbors of a phone.

In comparing the CIR framework with the conditional formalism that follows from (2.10), we can discern a tradeoff in modeling power that arises from the way that simplifying assumptions are typically made in the two approaches. In the abstract, if no independence assumptions are made, the CIR and conditional approaches produce exactly identical results since both compute the same joint probability. In practice though, we can never hope to estimate the complete statistics of the sequences involved, and we must introduce assumptions to make estimation feasible. The differences between the methods arise in the ways we can make assumptions while ensuring that our estimates do not violate the rules of probability.

For conditional statistical models, like the HMM and the conditional segment models presented elsewhere in this thesis, the probability $p(Y|A,S)$ expresses a general dependence of the observation Y and each of its components, Y_i, on the complete label sequence, A. This dependence may then be assumed to extend only to the immediate phone, or to the triphone encompassing the phone, etc. In general, the probability of Y_i may be conditioned on any reasonable function of the complete label sequence that can be adequately trained.

However, when using conditional models, conditioning on joint "observation context", i.e., using the information in the acoustic signal on both sides of the immediate

segment Y_i, is problematic. Specifically, it is possible to condition segment probabilities on observations either to the left or right of the segment, e.g., using $p(Y_i|Y_{i-1}, A, S)$ or $p(Y_i|Y_{i+1}, A, S)$, but not on both, i.e., not $p(Y_i|Y_{i-1}, Y_{i+1}, A, S)$. Note that although this may be a limitation for some forms of utterance-dependent normalization, most current statistical methods would have trouble effectively modeling any more information than is present within a single segment, simply due to limitations in training data for more complex models.

For the classification-in-recognition formulation of this chapter, the situation is reversed. The term $p(A|Y, S)$ expresses the general dependence of the label sequence (and each label in it) on the complete observation sequence, Y. This general dependence can readily be simplified to having labels depend on various functions of the observation sequence that are deemed most relevant. These functions may include acoustic-phonetic features, or features measured from the entire utterance that are useful for classifying a particular phonetic segment.

However, unlike the conditional approach, if we wish to condition on the neighboring phonetic context of the label sequence, in general the problem is somewhat complex. For instance, to model triphone context, we cannot simply use $\prod_i p(a_i|a_{i-1}, a_{i+1}, Y, S)$ since this is not equal to $p(A|Y, S)$. It is not clear if this restriction for the CIR framework is practically more limiting than the prohibition of joint observation context in the conditional model case. There is clear evidence from triphone-based HMMs that information about the phonetic context from both sides of a segment is a powerful constraint on the acoustic realization of a segment, whereas there is little experience yet with the use of "observation context" in posterior models. It may be possible that the acoustic information on both sides of a hypothesized segment adds little to the information from within a segment about the identity of the segment's phone label, but this is a question that requires further research.

Because of the evident importance of using joint phonetic context in acoustic

models, we describe next some alternatives for incorporating context in the CIR model.

Returning to (6.26), if $p(a_i|A_{L_i}, Y, S)$ and $p(a_i|A_{R_i}, Y, S)$ are mutually consistent estimates, that is if both of them can be expressed as marginal distributions of a common distribution, then we can write

$$p(A|Y,S) = \prod_{i=1}^{N} (p(a_i|A_{L_i}, Y, S) \;\; p(a_i|A_{R_i}, Y, S))^{1/2} \tag{6.37}$$

In this equation, the sequence probability is conditioned on both the left and the right phonetic contexts of each phoneme. Note however that the equality is strictly valid only if both the left and right conditioned probabilities are marginals of a common distribution. In this case, unfortunately, they will both produce identical probabilities for the complete label sequence. Since there is no point in computing the same quantity twice, such a decomposition is not directly useful as a practical means of computing the probability, except possibly in the case where less likely hypotheses are pruned from the search. If on the other hand, the component probabilities are not mutually consistent, as would be the case if they were estimated with typical training procedures, simply combining them by multiplying would not result in the desired posterior probability, and in fact, would not be a valid probability.

Other Triphone Models

In recent work done separately at both NYNEX [47] and BBN [84], researchers have incorporated triphone context in neural network posterior segment models. The approaches taken in both models are similar, and for brevity, we will examine just the BBN SNN model closely.

In this approach, each phone posterior probability (approximated by an MLP neural net) is conditioned on the full phonetic context as well as the segment observation, and this probability is divided by the conditional prior of phone given

context. For example, for triphone context, the MLP score for a segment is computed as $p(a_i|Y_i, a_{i-1}, a_{i+1})$ and this is divided by the probability $p(a_i|a_{i-1}, a_{i+1})$. The product of these terms is shown to approximate the *conditional* probability of label sequence given observations, $p(Y|A)$, divided by a normalizing factor approximating $p(Y)$. The two terms, $p(Y|A)$ and $p(Y)$, are weighted using the N-best mechanism to give the optimal combination for recognition performance (in the process, altering their interpretation as strict probabilities). Using this approach, the SNN alone gets results nearly as good as a baseline context-dependent HMM, and when the two models are combined, a significant improvement over the performance of the HMM alone is observed.

Despite the very good performance achieved so far, some of the assumptions made in this approach are questionable, and may prove to be obstacles to further improvement. Specifically, in the model derivation, it is assumed that the segments comprising the observation sequence are independent, $p(Y) = \prod_i p(Y_i)$, as opposed to the more usual conditional independence assumption, $p(Y) = \prod_i p(Y_i|A)$. This would be objectionable, except that this term is used only as a normalization factor common to all scored hypotheses, and should thus be irrelevant to the recognition maximization; it can be thought of as not the probability of observations, but some function of the observations, $g(Y)$, used as a scaling factor. However, because the SNN uses a fixed-length representation of each segment in the observation sequence, this factor will differ for any two hypotheses that do not have the same segmentation. For example, sentence hypotheses with different numbers of phones will be normalized by very different values for $g(Y)$, and this may cause problems in comparing sentence hypotheses. A second issue in the derivation is that the context of a phone is assumed to be conditionally independent given the observed segment, e.g., for triphones $p(a_{i-1}, a_{i+1}|Y_i) = p(a_{i-1}, a_{i+1})$. This assumption seems counter to the motivation of using context models and may also limit performance. Ultimately, these objections

may be irrelevant if the various terms are regarded as non-probabilistic knowledge sources that the N-best formalism can combine optimally for recognition.

In the above approach, because the MLPs used in the SNN produce only posterior probabilities of the form $p(a_i|Y_i, a_{i-1}, a_{i+1})$, a normalization factor is required if these scores are to approximate conditional densities, $p(Y_i|a_i, a_{i-1}, a_{i+1})$. For the model in this chapter, it would be possible to perform a similar normalization, but this idea is less appealing in our case. The CIR model was largely motivated to explore the advantages that accrue from the use of posterior probabilities, and in fact, in the version developed here for initial experiments, we used Bayes' rule to convert conditional densities into posterior probabilities just for this purpose. Using a normalization to convert back to conditionals is probably not very sensible and would not tell us anything more about the properties of posterior models. For this reason, we chose to explore context modeling using just posterior distributions.

6.4 Experiments

6.4.1 Context-Independent Recognition

Preliminary experiments were designed to test the feasibility of using the time-sampled segment representation described earlier under the phoneme classification framework described in Chapter 3. In classification, the observation vectors in each phonetic segment were linearly sampled to obtain a fixed number of vectors per segment, $m = 5$ frames. For observed segments of length less than five frames, the transformation repeated some vectors more than once. The feature vector for each frame consisted of 14 Mel-warped cepstral coefficients and their first differences as well as differenced energy. Each of the m distributions of each segment were modeled as independent full covariance Gaussian distributions. Separate models were trained

for males and females by iteratively segmenting and estimating the models using the algorithm described in [57]. The testing material came from the standard "Feb89" and "Oct89" test sets. In classification experiments using the Feb89 test set, the percent correct is reported over the complete set of phoneme instances, 11,752 for our transcription. In the initial experiments, only context-independent models were estimated, and the labels and segments of the observation sequence were considered independent.

On the Feb89 test set the classification results were 65.8% correct when the entire observation sequence was used and 66.4% correct when a fixed number of observations was used for each segment. This result indicates that, in context-independent classification, fixed length measurements can work as well as using the entire observation.

With this result, the next step was to evaluate the use of fixed-length features in recognition with the CIR formalism, using the N-best formalism described in Chapter 3.

An experiment was run comparing the CIR word recognizer to a baseline SSM recognizer that uses all observations. For the classification component of the CIR model, we used the context-independent posterior estimate given in (6.8). The segmentation probability, $p(Y, S)$, was computed as in Equation (6.13), using a version of the SSM that considered the complete observation sequence for a segment. That is, not just m, but *all* observation vectors in a segment were mapped to the distributions and used in finding the segmentation score. The weights for combining scores in the N-best formalism were trained on the Feb89 test set. In this case the scores to be combined were simply the SSM score, the number of words and the number of phonemes in a sentence.

On the Feb89 test set, the word error rate for the CIR method was 8.9% (227 errors in a reference set of 2561 words), while the original approach had a 9.1% word error rate (233 errors). To determine if these results were biased due to training the

weights for combining scores on the same test data, this experiment was repeated on the Oct89 test set using the weights developed on the Feb89 test set. Here, the performance for both the CIR and original recognizer was 9.1% word error (244 errors in a set of 2684 reference words). The CIR formalism thus has the same performance as the original scheme.

6.4.2 Segmentation Probability

As mentioned previously, some current systems use a classification scheme with no explicit probability of segmentation. We attempted to simulate this effect with the classification recognizer by simply suppressing the score for the joint probability of segmentation and observations. This is equivalent to assuming that the segmentation probabilities are equally likely for all hypotheses considered. Scores were computed for the utterance with and without the $p(Y, S)$ term on the Feb89 test set. When just the classification scores were used, word error went from from 8.9% to 10.8%, an 18% degradation in performance. Evidently, the joint probability of segmentation and observations has a significant effect in normalizing the posterior probability.

Experiments were also run to compare the first two of the segmentation scoring methods described earlier in the chapter. In the first method, based on equation (6.13), the same speech analysis described earlier was used at each frame (cepstra plus differenced cepstra and differenced energy) and the marginal probability was computed by summing over the set of context-independent phones. In the second method, which computes $p(S \mid Y)$ using equations (6.14) - (6.20), we modeled each of the conditional densities in (6.19) and (6.20) as the joint, full covariance, Gaussian distribution of the cepstral parameters of the two frames adjoining the hypothesized boundary. In order to reduce the number of free parameters to estimate in this Gaussian model, we used only the cepstral coefficients (i.e., omitting differences)

as features for each frame. On the Feb89 test set the first method had 9.1% combined word error for male and female speakers, while the second method had 11.0% word error. Using the best weights for the N-best combination from this test set, the segmentation algorithms were also run on the Oct89 test set. In this case, the word error rates for the two methods were 9.4% and 11.9%, respectively. As noted before, in comparing these methods, certainly part of the reason the marginal model improves is simply due to its having a larger number of free parameters.

This result suggests that the boundary-based segmentation score yields performance that is worse than no segmentation score. However, the "no segmentation" case actually uses an implicit segmentation score, since the N hypotheses are assumed to have equally likely segmentations (while all other segmentations have probability zero). Additionally, since weighted phoneme and word counts are included in the combined score, we effectively use another simple grammatical constraint since this scheme penalizes hypotheses that are too long or short.

6.4.3 Left-Context Experiments

The first left-context experiment used a model with a left-context classification score (with a time-sampled segment representation), as in (6.27), together with a context-independent segmentation score computed using the marginal method given by (6.13). The motivation for this initial configuration was the hypothesis that it was more important to have a more discriminating model in the classification component, and any "reasonable" segmentation component, including the method that worked in the context-independent case, should do an adequate job of scoring the segmentations.

The performance of this initial model was quite poor: the CIR word error rate was 8.6% on the Feb 89 test set (77 errors for females, 142 for males) compared with the baseline rate of 6.4% for the corresponding non-CIR left-context model (64 female

and 100 male errors).

Following this, we conducted a number of experiments to try to determine why the CIR performance was lower than the non-CIR model's, exploring different segmentation models and eliminating differences in assumptions made by the two models. In these experiments, we chose to use just the female subset of the database for training and recognition to facilitate faster experimentation. For this subset, the error rate for females alone with the above CIR system was 7.4%, and the results for the baseline, non-CIR system were 6.0% and 5.7% for the 5 and 8-frame SSM, respectively. Some of the results we present below differ by only a small number of errors on this data set and the differences may not be statistically significant. However, we have generally found that this set is adequate for determining rough comparisons and eliminating truly bad approaches. Table 6.1 summarizes the results for the variants of the left-context CIR model that we discuss next.

As an alternative to the context-independent segmentation score, we again tested the "no-segmentation" paradigm, using the classification component score alone. As in the context-independent case, this approach resulted in much worse performance – 9.7% word error on the female subset. Although the performance of 7.4% word error obtained when using both classification and segmentation score is not as good as the baseline non-CIR performance, the substantially higher error rate when just the classification component alone is used confirms that both parts of the model are necessary to obtain reasonably normalized utterance probabilities.

To more fully exploit the power of context modeling, we changed the segmentation component from context-independent to the left-context model given in (6.33). However, this change made only a small improvement, reducing the error from 7.4% to 7.2%. In addition to the disappointing accuracy improvement, the computation for the left-context segmentation score is significantly larger than that for the context-independent segmentation score.

To rule out the possibility that our N-best weight optimization was simply badly mistuned for the left-context case, we tried using separate weights for the two components $p(A|X,S)$ and $p(X,S)$. This gave only a slight improvement from 7.2% to 7.0% error. Additionally, because we are in danger of overtuning to the test data in this case, even this small improvement is not reliable and we did not continue this practice in other experiments.

We next dropped the assumption that the sampled representation was equivalent to the original. We conducted an experiment that used the CIR framework, but calculated the classification score using the complete observation sequence. For a given segmentation, the numerator of this score would be computed identically to the non-CIR method of computing the complete score (i.e., the standard left-context benchmark we were attempting to replicate).

The result using the complete observation CIR model was 6.6% on the Feb89 females. This is an improvement from the 7.2% for the previous best CIR left-context system, but still worse than the 6.0% error rate of the non-CIR left-context experiment. It is interesting that the assumption that the particular fixed-length function was an adequate representation of the complete observation was acceptable for the context-independent CIR recognizer but caused problems for the left-context case.

We also looked at the recognition results when using just the classification numerator from the complete CIR score in this version of the system, since, when the observation consists of all frames in a segment, the numerator of the CIR score is computed with the same procedure as the non-CIR model. In this case the word error improved from 6.6% in the complete CIR model to 6.4% for just the numerator. The discrepancy between the numerator result and the non-CIR performance of 6.0%, despite their identical scoring algorithms, is due to the fact that the CIR score was computed over a segmentation determined by the combined classification

Table 6.1: *Word error rates for the left-context CIR system using different segmentation scoring methods evaluated on the female speakers of the Feb89 test set. Results are given for classification scores based on both sampled and complete (All Frame) segment observations and for the 5- and 8-distribution SSM. Comparable non-CIR scores are 6.0% and 5.7% for the 5- and 8- frame models, respectively.*

Classification Type	Segmentation Type	# Distributions	
		5	8
Sampled	No segmentation	9.7	–
	"Explicit" boundary	9.0	–
	Context Ind	7.4	–
	Left. Cont.	7.2	7.6
All Frame	Left. Cont.	6.6	7.0
	DP Left. Cont.	6.0	6.5

and segmentation scores. This result indicates that the dynamic programming search over possible segmentations was sufficiently misguided by the CIR score that the numerator probabilities were measured over a poor segmentation.

The above results were obtained using a five frame SSM model. When using an eight frame model, performance for the non-CIR model improves slightly (from 6.0% to 5.7% for Feb89 females) whereas the CIR model (with a sampled, fixed-length segment representation) actually degrades (from 7.2% to 7.6%). Similarly, for the case of using the complete observation sequence in the CIR classification component, using 8 frames instead of 5 gives a word error rate of 7.0% instead of 6.6%. The increase in error rate is somewhat surprising, and we do not have a good explanation for it.

To further investigate the effect of the segmentation score on the overall CIR score, the "explicit boundary" segmentation score, as given in (6.15), was also tried. This score was combined with the left-context classification score. As in the context-independent case, the performance for this model was significantly worse than using the context-independent marginal segmentation score: the error rate increased from 7.4% to 9.0%.

We also compared the use of Viterbi decoding in the CIR approach (the default in the system) versus "Baum-Welch" decoding. In the usual Viterbi criterion, the score for a sentence hypothesis is found as the score for the single most likely path through different possible word pronunciations and segmentations. This type of approximation to the complete probability has been found by researchers to give essentially identical performance for HMM systems, but we wanted to verify that it was not hurting performance in the CIR case. Viterbi decoding was compared with the more exact method in which the score for an utterance was summed over all segmentations allowed in the N-best rescoring (i.e., all segmentations with boundaries within ± 10 frames of the initial HMM segmentation) and over all alternate pronunciations for the words in the sentence. For both the context-independent and context-dependent CIR models, summing and using the Viterbi approach produced identical error rates.

Finally, we examined the effect of using the left-context, dynamic programming segmentation score given by (6.36). This method uses assumptions that are most similar to the non-CIR assumptions. We again used the complete observation for the classification component of the model in order to have the best chance of replicating the non-CIR performance. We compared this method with the CIR model that uses the simpler left-context segmentation score (from Equation (6.33)) for both the 5 and 8 frame CIR SSM. Using the DP segmentation improved performance from 7.0% to 6.5%, for the 8-frame case, and from 6.6% to 6.0% for the 5-frame case.

The 6.0% error rate for the 5-frame version of this CIR model is actually the same

as that for the non-CIR 5 frame model, although this is still worse than the best non-CIR of 5.7% for the 8-frame model. Furthermore, the computational and memory requirements for the DP segmentation score are quite large and probably infeasible even for research purposes, despite the reduced search space of the N-best approach.

Additionally, to get this performance, we were forced to drop the assumption that the fixed-length representation was equivalent to the original unsampled representation. With this, we gave up the ability to measure correlation in a very simple fashion by estimating the joint density of segment-length observations using a single classifier. This negative result, that the sampled representation loses information that is crucial for phonetic discrimination, stands in contrast to a number of earlier segmental studies that have used a sampled representation without carefully examining its effects [84, 57]. However, most of these other studies have dealt mainly with context-independent models, for which our experiments indicate that sampling may be entirely adequate.

Segmentation Distance Metrics

As part of the effort to understand how different segmentation scoring methods affected the CIR models in comparison with the non-CIR model, we measured the distances between the segmentations produced by different models. The segmentation for an utterance was defined to be the set of boundaries along the highest scoring path for the utterance produced by a model in recognition. We compared the segmentations produced by different versions of the CIR model, using as a reference, the segmentation produced by the non-CIR left-context model. The distance between a reference and a "test" segmentation was computed using the following procedure:

- Find the RMS distance from test boundaries to reference boundaries and save the index of the reference boundary that each test boundary maps to. If more

Table 6.2: *Distance metrics for different CIR segmentations from a reference, left-context non-CIR segmentation.*

Segmentation Type	Avg. dist.	# insertions	# deletions
Left-context, all frames	.765	26	36
Left-context, "explicit boundary"	1.612	78	80
Cont. Ind.	1.527	29	35

than one test boundary maps to a reference boundary, take the closest of the set and call the others insertions.

- Any reference boundaries that are not mapped to are labeled as deletions.

The error measures we examined were then the average RMS, in frames, to a correct boundary, the number of deleted boundaries, and the number of inserted boundaries. Using this definition, we compared the distance between CIR models and the left-context non-CIR model, for several CIR variants. The results are given in Table 6.2.

We can see that the direct segmentation has a significantly larger distance than the other methods and the left-context segmentation is closer than the context-independent segmentation.

Feature sampling and context-dependent models

The left-context results given above seem to indicate that the time-sampling representation chosen for segment observations may destroy information important for recognition with context models. To help determine if these problems were inherent in the representation rather than an artifact of the CIR formalism, we conducted an experiment to measure the effect of sampling using an approach similar to that taken in the original SSM work [57], in which a sampled representation was found to be

adequate for context-independent models.

In the earlier SSM work, the score for an utterance was the sum of the log probabilities of time-sampled segments and each segment score was scaled in proportion to the segment's original duration. The scaling normalization thus made each segment's score an approximation of what would be expected if all frames were used.

We conducted an experiment that replicated this scheme (with a slightly different normalization), but used left-context models instead of context-independent models. The normalization for scores of time-sampled segments consisted of a scaling factor of L_O/L_M where L_O is the length of the observed segment and L_M is the model length. The only difference between this experiment and the scoring in the original SSM work is the denominator term L_M, which could be accounted for by that system's phoneme insertion penalty. In our experiments, the model length was 8.

In an initial experiment, recognition was performed by scoring all segments using the above normalization. This resulted in a 4.7% word error rate for the male subset of the Feb89 test set. This compares with the 3.7% error rate when using the normal scoring for all segments. To rule out the possibility that the worse performance was due to errors introduced in sampling very long segments, we ran a second experiment with sampling done in a very limited context. In this experiment, sampling and normalization was done only for segments of length 9 and 10, with all other segments being scored normally, i.e., using the complete observation sequence. However, even sampling this much smaller subset of the segments led to significant performance degradation: the error rate in this experiment was 4.0% on the same test set.

To look more closely at the system's behavior in this experiment, we compared the scores for using the sampled/normalized scoring versus the original approach for each segment that was of length 9 or 10. We estimated the maximum and the mean square error of the differences between the two scores. The scores appear to be asymptotically unbiased – the average error and average squared error both approach 0 fairly rapidly

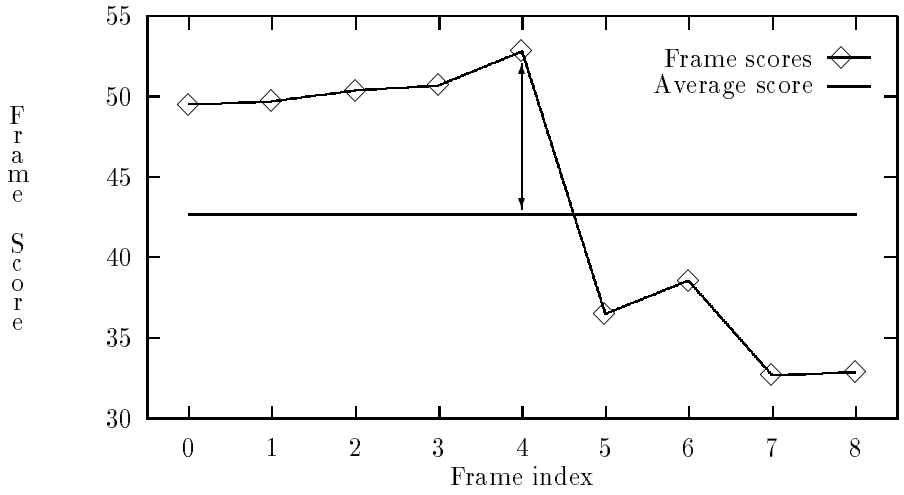

Figure 6.1: The effects of time sampling on the score for an example segment of nine frames. Approximation error is indicated by the double arrow at frame index four.

– but with considerable variance (as exhibited by fairly large maximum errors).

The large variance may account for the poor recognition performance. Since the dynamic programming algorithm used in recognition repeatedly chooses the maximum segment score, if the process includes some large errors, even if most errors are small, the large positive errors are likely to make their way into the final result. In fact, the recognition scores are consistent with this hypothesis; the sampled/normalized experiments produced larger utterance scores than those for the regular SSM experiment, with a larger increase when all segments are sampled than when just the 9 and 10-length segments were.

Essentially the normalization procedure assumes the missing frame scores should be the average of the sampled frames' scores. The fallacy of this assumption's generality becomes evident when looking at the frame scores for particular segments.

An example of the frame scores for a 9-length observation is shown in Figure 6.1.

Here the score for each frame (proportional to the log probability density for the frame's cepstral coefficient vector) is plotted for the 9 observed frames. When we use a sampled representation, the fourth frame is omitted, and the normalization essentially assigns to this frame a score equal to the average of the sampled frames. The average value is also plotted in this figure as a horizontal line and the error for this approximation is indicated by the double arrow below the fourth frame score. For this example, it is clear that using the average score as an approximation of *any* of the frame's true scores would result in a substantial estimation error, and the fourth frame happens to be particularly bad. This example highlights the difficulty in using any simple procedure to estimate the contribution of any missing frames.

The above experiment supports the interpretation that the problem with this representation is not solely an artifact of the CIR formalism, but an issue of missing information that is required by context-dependent models. The more discriminating left-context models seem to make use of information in the segment observation that context-independent models do not need and that is lost in sampling.

6.5 Discussion of Experiments

In the preceding experiments, the different versions of the CIR formalism performed worse than the conditional SSM model, and we were only able to come close to the conditional performance using a very computationally expensive model. One of the important factors contributing to the lower performance was the use of a time-sampled segment representation. The inaccuracy of this representation was demonstrated both in the CIR formalism and by the experiments that used the earlier SSM formalism that directly normalized the sampled segment scores. As mentioned, this stands in contrast to the results reported by other researchers in which time-sampled representations were found to be adequate.

One possible solution to the sampling problem that was explored was to decode by summing scores across segmentations for a phone sequence, instead of using Viterbi decoding, which uses only the maximum scoring path. Since different segmentation boundaries lead to different frames being sampled, if the errors due to sampling are not too large, summing across the hypothesized segmentations for a phone might have reduced the variance of the error and thus improved overall performance. However, our experiments showed no difference in performance between the two approaches, so it is likely that the magnitude of the sampling errors is too large for this strategy to have been effective.

Clearly, if we wish to retain the advantages of a fixed-length representation, there are other possibilities that may have more desirable characteristics than time sampling. For example, the BBN SNN system has recently made use of the discrete cosine transform across time of the cepstral features in a segment to capture segmental characteristics in a fixed-length representation [83]. By increasing the number of transform coefficients, in principle, it should be possible to reduce the information loss in this type of representation to a negligible amount. Of course as the representation accuracy improves, the number of parameters in our models that must be trained increases too, and we may not have a sufficient amount of data to estimate models using the most accurate representation. Although BBN reports that this method provides only a modest improvement over a time-sampled representation, some representation of this sort may have the desired fixed-length property without destroying important information.

The problems with the CIR formalism are not confined to the issue of segment representation, however. Even when no transformation of the segment observation was applied, the CIR performance still suffered in comparison with the conditional method. What are the possible causes for the remaining difference?

As noted earlier, the assumptions of the conditional and CIR left-context ap-

proaches, although both probabilistically consistent, are not identical. We saw before in (6.32), that the CIR expression for the joint probability of observations, segmentation and labels contains the non-CIR expression for that same probability in its numerator. That is, if we denote the expression for the joint probability under non-CIR assumptions as $p_N(Y, S, A)$, Equation (6.29) is

$$p_N(Y, S, A) = \prod_i p(Y_i, \tau_i, a_i \mid a_{i-1}) \tag{6.38}$$

and, denoting the CIR expression as $p_C(Y, S, A)$, (6.32) becomes

$$\begin{aligned} p_C(Y, S, A) &= p(Y, S) \prod_i \frac{p(Y_i, \tau_i, a_i \mid a_{i-1})}{p(Y_i, \tau_i \mid a_{i-1})} && (6.39) \\ &= p_N(Y, S, A)\, p(Y, S) \prod_i \frac{1}{p(Y_i, \tau_i \mid a_{i-1})}. && (6.40) \end{aligned}$$

This demonstrates that the two are not equivalent, and in fact the underlying sets of assumptions are incompatible. That is, if we make the CIR assumptions of (6.32), then the non-CIR case of (6.29) is impossible, and vice versa, unless we make the further assumption that

$$p(Y, S) = \prod_i p(Y_i, \tau_i \mid A_{i-1}), \tag{6.41}$$

which in turn implies

$$p(Y, S) = \prod_i p(Y_i, \tau_i), \tag{6.42}$$

i.e., that segment observations and durations are completely independent. However, this assumption, in turn, contradicts the assumption of left-context dependence. We therefore conclude that the two sets of assumptions cannot both hold in a single model.

From these experiments, we can draw some conclusions about aspects of segment modeling in general and approaches based on posterior-distributions in particular. As mentioned, we have seen that considerable care may be required in choosing a

representation of the observed segment that is used as the input feature vector to the classifier. This is particularly important as more discriminating, context-dependent models are introduced. Finding a representation that retains all the critical information about a segment, and yet is of low enough dimension to allow robust statistical modeling, remains an outstanding goal in this area.

Furthermore, our experiments indicate that the assumptions we make in context modeling when using a different general recognition formalism must also be considered carefully. We found for instance that the left-context assumptions for the CIR model, though apparently "natural," were fundamentally different from the assumptions associated with the left-context conditional model. Additionally, the question of joint left and right context is not yet completely resolved.

On the other hand, one of the strengths of the CIR formalism may be its ability to incorporate wider windows of "observation context," that is, conditioning a phone classification probability on functions of the entire utterance, rather than just the observation for a segment, in a probabilistically consistent fashion. Although we did not explore this aspect of the formalism in the preliminary work reported here, we believe that it may have useful application in contributing to solutions for some unresolved issues in acoustic modeling. By applying the information from a wider perspective to individual phone probability estimates, the CIR formalism may help to ensure consistency between different parts of a hypothesis in a manner that more traditional approaches, which only use local information in decisions, can not.

Chapter 7

Conclusions

This thesis has addressed some of the important issues in the use of segment models for acoustic modeling in speech recognition. We categorized segment models according to the key properties of observation representation and probability distribution type, and examined the characteristics of specific models in these categories. In this chapter we summarize the contributions of the thesis and suggest some directions for future work.

7.1 Contributions

In this thesis, we have examined segmental acoustic modeling for speech recognition from the perspective of two key characteristics: the representation of segment observations (fixed versus variable length) and the usefulness of different probability distribution assumptions in training and recognition.

Concerning the representation of the speech segment, we distinguished the various possibilities based on whether the dimension of the representation varied with phone duration or was a transformation from the inherently variable-dimensional observation space to a fixed-dimensional one. There are several key issues related to the choice of segment representation. They include the completeness of the repre-

sentation (i.e., how much of the information in the original signal is preserved in the representation), the suitability of the representation as input to a particular pattern recognition technique (for example, the desirability of fixed-dimensional inputs for some techniques, such as neural networks), and the normalization of probabilities that may be required if the representation is a transformation of the original input space.

Under distributional assumptions, we categorized segmental approaches based on whether they use class-conditional probability densities of observation given phone, versus posterior probabilities of phone given segment observation. We showed that the type of distribution chosen has important implications for the use of context-dependent models, with posterior models having a straightforward formalism for modeling the observation context outside a segment's boundaries, but requiring more complex methods for incorporating phonetic context. For class-conditional models, the opposite situation holds, with phonetic context arising as a natural extension of the basic approach, but observation context presenting some formal difficulties.

All four combinations of these characteristics – fixed versus variable-length representation and conditional versus posterior distributions – lead to valid segment models, although posterior distribution models are most typically associated with fixed-length representations and conditional models with variable-length representations. We focused our attention primarily on these two most common cases, investigating specific models from these categories. Through these models we were able to demonstrate some of the advantages and disadvantages of the general approaches, as described below.

As a special case of the conditional density / variable-length model, we looked at the the use of Gaussian mixture models for improved frame-level modeling in the independent-frame SSM. The introduction of mixtures allows much more general distributional modeling possibilities than under the unimodal Gaussian density assump-

tion made in some earlier versions of the SSM. Although mixture models offer the ability to model more complex densities, the resulting increase in the number of parameters requires that care be taken to ensure the parameters are robustly estimated. In general, recognition performance improves as the number of free parameters increases, so long as the parameters that are added represent useful information and there is adequate data to train them. Since not all parameters are equally relevant, a particular allocation of parameters may be more effective than another for a fixed amount of training data. We investigated several different methods of parameter allocation within the mixture formalism, using the mechanism of parameter tying to control the growth in the number of parameters. Specifically, we looked at so-called semi-continuous or tied-mixtures, untied or continuous density mixtures, covariance tying, and a method to tie the parameters of context allophones of each phone.

The experimental results we obtained with mixtures for the SSM using frame-independence assumptions, confirmed that this special case of the SSM performs comparably to state-of-the-art HMM systems. We also developed several innovations for the mixture training algorithm that greatly reduced training time, including the estimation of marginal distributions for context-dependent tied-mixture models and a parallel training algorithm that runs on a distributed network of workstations.

In order to take better advantage of the segmental framework, we developed a segmental mixture model that captures segment correlation indirectly through modeling distinct segment-length, time-cepstra trajectories. This approach also falls under the category of models based on conditional densities of the variable-length observed segments. With this method, we found very good performance for context-independent models and demonstrated the feasibility of capturing segmental characteristics via a detailed model-frame sequence. We did not observe improved performance for context-dependent segmental mixture, probably because more sophisticated parameter sharing is needed for the relatively small, Resource Management training set.

To investigate the characteristics of models based on posterior distributions, we developed and evaluated the CIR recognition formalism. Using context-independent models, we demonstrated the feasibility of the general posterior distribution approach. However, with more detailed, context-dependent models, we found that the assumptions required in this case did not readily lead to performance as good as with comparable class-conditional models. Although our posterior-based model was unable to improve performance over the comparable class-conditional density models, our work has improved the understanding of the theoretical issues associated with these models, and the questions of consistency in context-dependent posterior models in particular.

Of the models developed in this thesis, the best case improvement over the baseline SSM for the context-dependent case was using the frame-level tied-mixture model described in Chapter 4. In this case, the word error was reduced by 30% from 4.8% to 3.4%. For context-independent models the most significant improvement was with the segmental mixture model. On the male speakers of the Feb89 test set, this model reduced word error from 9.5% in the baseline to 5.5%.

7.2 Trade-Offs of Different Modeling Assumptions

In this section we present conclusions concerning specific modeling options that were considered in this work. We first review the comparative performance and number of parameters for a representative set of models from different parts of the thesis. Table 7.1 gives the number of free parameters and the word error rate on the female Feb89 speakers for left-context versions of several of the models discussed earlier. Although a larger test set is desirable for careful performance comparisons, the CIR models were evaluated only on this set, both because of the model's computational burden and because performance disadvantages of this approach relative to others were adequately

Table 7.1: *Word error rate and number of free parameters for several left-context models. The error is measured on the female speakers of the Feb89 test set.*

	Word Error Rate	# Free Parameters
Baseline SSM	5.7	370K
CIR Sampled Classification	7.2	320K
CIR DP Segmentation	6.0	320K
Independent Frame Covariance-tied Mixture	4.2	1.3M
Segmental Mixture	4.3	2.3M

highlighted on this set. Similarly, triphone models lead to better performance for the non-CIR models, as noted in earlier chapters, but the left-context case allows direct comparison of the different models for a common set of phonetic units.

The first row of the table gives the performance for a baseline SSM with 8 regions, each of which has a single full-covariance Gaussian distribution. Two CIR models discussed in Chapter 6 follow this; the first uses time sampling of the segment in its classification score and a marginal segmentation probability, the second has no sampling in the classification score and uses dynamic programming to compute the segmentation score. The table also includes the left-context version of the independent-frame, covariance-tied mixture model discussed in Chapter 4 and a segmental mixture model that uses 3 segmental mixture components, with 8 regions per segmental component and 4 diagonal Gaussians per region (this parameterization differs slightly from the triphone-dependent models described in Chapter 5).

In general, we can see that the higher performing models have significantly more free parameters than lower-performing models, but also that the number of parameters alone is not a good predictor of performance. At one extreme, for the case of the two CIR approaches where the input models are identical but the recognition algorithms differ, it is clear that performance differences are not at all due to the number of parameters, since these are identical. We note also that the segmental mixture model has nearly twice the number of parameters as the independent-frame mixture model, but performs no better than that model. It is likely that some of the parameters in the segmental case model irrelevant information, and in future work it may be possible either to achieve the same performance with fewer parameters or improve performance keeping the number of parameters fixed.

In the remainder of this section, we review four modeling choices that arose repeatedly in the development of the models discussed, and we attempt to draw some general conclusions concerning them. Specifically we look at the trade-offs in both performance and computation for: posterior-based recognition versus class-conditional methods, different types and amounts of parameter tying in a model, the type of correlation assumption to use (full versus diagonal-covariance) in Gaussian modeling, and segmental versus frame-based modeling. In looking at these issues, however, it is important to note that many of the aspects concerning performance are fundamentally empirical, and the conclusions that we reach are necessarily somewhat dependent on the characteristics of the particular corpus and task used in developing the models. A similar warning is in order concerning the evaluation of computational advantages of the methods. The primary goal of this work was to explore the recognition accuracy of the models, and considerations of speed were secondary. Obviously, questions of speed are critical to achieving near real time performance on inexpensive hardware, but in the research phase, software development time is also an important factor. Hence, the computation trade-offs presented below represent somewhat rough

estimates. Even with these caveats, however, there are still important general lessons that can be taken from the models developed in this thesis concerning recognition accuracy and computational efficiency.

Posterior versus Class-Conditional Densities

By far the most common approach to statistical speech recognition is to model the acoustic signal using class-conditional densities as is typical in HMMs. As mentioned before, the efficiency of HMM algorithms and their considerable success in improving accuracy make this an appealing approach. In general, class-conditional models can easily incorporate phonetic context models, although they are not well-suited for incorporating arbitrary windows of observation context in local probability distributions. Increasing the size of the observation window involves changing the modeling unit (e.g. from frame to segment) and a corresponding increase in the search space.

Posterior distributions are an alternative basis for statistical modeling that hold several possible advantages. Posterior models have the potential for directly incorporating segmental information by conditioning classification probabilities on complete segments, and unlike conditional models, there are no formal obstacles to incorporating observation context. The potential for segment modeling was the initial motivation for our investigation of these models. Our goal was to take advantage of the formalism by using a simple fixed-length representation of segments that would allow particularly straightforward measures of intra-segment correlation. This goal appeared reasonable given previous segment research that suggested fixed-length representations such as time-sampled segments are adequate for capturing the essential information in a speech segment. However, we found this to be true only in the context-independent case. Analysis of left-context results showed that time-sampling hurts performance significantly, due to a loss of information used by context models for phonetic discrimination.

Furthermore, our experience with left-context models showed that using the same underlying densities, our posterior-based model gave lower performance than a comparable class-conditional model. The question of posterior probability normalization seems to be difficult in general, and the most effective segmentation probability we found for the left context case was computationally very expensive and only marginally adequate. Other normalizations for triphone context models of posterior models proposed recently by researchers using similar approaches require problematic assumptions, and essentially lead to a poor approximation of conditional densities.

The CIR algorithms we explored are also significantly slower than any of the conditional models tested, primarily because the posterior distribution was implemented via the class-conditional distributions using Bayes' rule. In a more direct implementation, the classification component would have a similar cost to the class-conditional distributions. However, the segmentation score also adds a significant additional cost to the CIR algorithm. For instance, the most effective segmentation score – the left-context DP algorithm – is slower than the complete scoring algorithm for comparable class-conditional models by more than a factor of two.

In weighing the relative advantages of the two approaches for practical recognition today, the balance strongly favors conditional densities, both in terms of their accuracy and speed. However, the posterior-based approach has not been explored as thoroughly, and there may be solutions yet to be found for some of the problems we encountered (particularly with segmentation scoring) that will allow the exploitation of this formalism's advantages. Because we were interested in comparing models that were as similar as possible, we did not explore two options that are only possible with posterior models: direct scoring of class conditional probabilities (e.g., using MLP neural networks) and extended observation contexts.

Parameter Tying

In general, it is possible to reduce the number of free parameters in a model by defining sets of parameters that can be treated as equal. There are as many ways of tying parameters in a system as there are equivalence classes of the parameters, but the types of tying can be characterized by two factors. First, which element of the distribution is being tied (e.g. covariance, Gaussian, complete mixture distribution) and second, what group of models in the system shares the tied elements (context-dependent variants of a phone, clustered regions, all regions in the system, etc.). In this thesis, parameter tying arose in the investigation of mixture-density models. Despite the wide array of specific possibilities, many of the important characteristics of tying can be seen by comparing two extreme cases; on the one hand using completely untied mixture models, in which each region's density is characterized by separate Gaussians and mixture weights, and on the other, using tied-mixtures, where all mixtures use a single set of Gaussians.

For untied models, the fact that the sets of Gaussians are separate affords each region a great deal of modeling flexibility, but the number of Gaussians actually useable in the untied case is often limited by the amount of training data for a model. In particular, the data available for a context-dependent model is often too small to support the estimation of more than a few Gaussians in a mixture.

In a tied-mixture system, the regions each have distinct weights, but they share a common set of Gaussians. In this way each region density must be composed from the same set of "basis" Gaussians with a single set of means and covariances. Because of tying, the training from all models contributes to the estimation of one set of Gaussians, so a much larger number of Gaussian components can typically be used in the mixture. Another difference between tied and untied models is the relative "smoothness" of the models. The Gaussians in tied-mixture systems are typically broader since they must adequately represent the data associated with all phonetic

models using a single set of Gaussians, while the Gaussians of untied models may be narrower since they are modeling the relatively small amount of data associated with a single phonetic model. As a consequence of the generally smaller number of parameters per density in the tied-mixture case, it is usually possible to model a larger number of context-dependent phonetic units than with untied models. In this instance, the tradeoff is between having a smaller number of very specific untied models or a larger number of possibly smoothed models.

There are significant computational advantages for tied-mixtures in recognition. As each input feature vector is encountered, its probability is determined for every Gaussian in the system, but since there are many mixture densities that share the single set of Gaussians, the computation of the weighted sums of Gaussian probabilities dominates the computation.

For models with intermediate amounts of tying, their computational and distributional characteristics also usually lie somewhere between the extremes outlined above. For instance, in the mixture system of Chapter 4 that uses tied covariances but untied means, the re-estimation procedure leads to distributions whose smoothness falls somewhere between the untied and fully tied cases, and the number of context models that can be estimated for a given amount of training is usually less than in the tied case but more than in the untied.

We explored the issues of tying experimentally, looking at several mixture variants in Chapter 4. In general, with moderate amounts of training as in the case of Resource Management, tied-mixture systems performed better than untied systems for context-dependent modeling. For our system the inaccuracy inherent in sharing a common set of smoothed Gaussians with tied mixtures is offset by the advantage of having a larger number of mixture components per distribution. However, we achieved equally good results using the more limited form of tying of keeping means separate and sharing just the covariances between context models of the same phone. In context-independent

experiments, untied systems performed substantially better than tied-mixture models. This is not surprising since here there is enough data to train multiple Gaussian components for each model, whereas in the context-dependent case, the amount of data per model is typically much less.

Gaussian Correlation Assumptions

A basic choice that arises in using Gaussian models in general is whether to use full or diagonal-covariance matrices. In our experiments, in which the feature vector of cepstral parameters and their first differences contains 30 elements, a full-covariance structure leads to 465 free parameters per covariance whereas a diagonal assumption yields just 30. This difference has important implications for parameter estimation and computational speed.

We first encountered this issue experimentally in the case of tied mixtures, where we were able to use full-covariance matrices without incurring severe penalties in training or recognition speed. In this case, we were motivated to use full-covariance matrices since they significantly improved performance for the male speakers over the diagonal case, due to the larger amount of male speech in the training database. In general, the computation of a density value for a single, full-covariance Gaussian requires substantially more operations than for the diagonal covariance case. However, with tied-mixtures, as discussed above, computing Gaussian probabilities accounts for a relatively small amount of the computation, with the majority going to the computation of the different weighted sums of the shared probabilities. In this way, the cost of full-covariances is "amortized" over a large number of region densities.

The issue of training feasibility with full covariances is also aided by the parameter sharing in tied mixtures. In cases where the number of parameters in a system is proportional to the number in each Gaussian, using full instead of diagonal-covariance matrices increases the number of parameters tremendously, possibly leading to diffi-

culties in adequately estimating the models. However, for tied-mixtures the number of Gaussians is small compared to the number of mixture weights in the system, so using full-covariance matrices does not cause undue training problems. Thus, the special structure of tied-mixtures makes full-covariance Gaussians an appropriate choice despite the difficulties in training and computation generally associated with them.

The issue of full versus diagonal-Gaussian was revisited in the experiments for the segmental mixture model, which used untied mixtures. In this case, since the model's Gaussian parameters were not shared, the number of parameters in the system scaled with the size of the Gaussians, and the use of full-covariance matrices provoked the training and computational problems described above. In training, with relatively small numbers of Gaussians per mixture, the full-covariance matrix estimates were singular and unusable. Furthermore, the computation for the full-covariance case was nearly twice as slow as the diagonal covariances for the same number of mixture components. In figure 5.1, which compares the performance of full and diagonal-covariance matrices, we saw that the diagonal-covariance case has a higher error rate than the same number of full-covariance Gaussians, but about the same error rate when the number of Gaussians is increased sufficiently. For the same recognition performance, the diagonal Gaussian models ran slightly faster than the full covariance models. For this model, the diagonal case was chosen, since special measures were required to avoid singular estimates of full-covariance matrices and the resulting models held no advantage in terms of speed or recognition accuracy.

The pattern of these two cases can be readily generalized. Full covariance models are most appropriate when the number of parameters does not scale with the size of individual component Gaussians, as in the tied-mixture case. Diagonal models hold clear advantages for both trainability and recognition speed when the number of parameters in a system is proportional to the number of Gaussian elements.

Segmental versus Frame-based modeling

The segmental mixture allowed us to compare segmental modeling with frame-based models. The goal of the segmental mixture was to capture distinct segment-length trajectories in the acoustic space with different mixture components. This contrasts with independent-frame models in which there are no constraints to keep the inconsistent trajectories of distinct acoustic realizations of a phone from intermixing. In the context-independent case, the segmental mixture had significantly lower error rate than the independent-frame models we tested. We saw in particular that segmental models with larger numbers of regions across the model performed better than those with fewer regions. This is consistent with results reported in investigations of a microsegment mixture model [36], and with the notion that the increased time resolution afforded by more regions serves to capture details of the acoustic trajectories that are necessarily blurred together in models with fewer regions.

For context-dependent models however, segmental mixtures performed significantly worse than the independent-frame case. The exact cause of the problem is unclear, but there are two plausible contributing factors. We know that the amount of training per model is smaller in the context-dependent case and the number of parameters in the segmental mixture model may simply require more training data. There is a second factor, related to the shortage of training, but distinct to the segmental nature of the model. The allophones of a phone are often assumed to share a common acoustic "target" that is typically reached in the center of a phone with the variation from the target increasing away from the center. Since segmental mixtures attempt to model complete acoustic trajectories with separate components, the training for a target region may be split between the mixture components for the separate trajectories of a model. As a result, the overall model of the target region may be very poorly estimated in comparison with an independent-frame model for which the training for every region of the model is pooled together. For both reasons, some sort

of parameter sharing mechanism that overcomes the particular challenge posed by segment trajectory modeling is probably required for effective context modeling.

The computational aspects of segment models in general are significantly worse than for frame-based models. For segment models, the probability of a phone for a given speech segment may be computed in a completely different manner for each hypothesized begin and end time, and there may be no way to reuse partial scores of segments in the dynamic programming search for the maximum probability label sequence. In the particular case of the segmental mixture model however, the assumption of conditional independence of frames within a trajectory makes it possible to reuse the frame scores when a model is scored against two different speech segments that overlap in time. In our implementation, however, the score caching for this involved considerable bookkeeping overhead and as a result, recognition was never quite as efficient as in the frame-based case.

7.3 Future Directions

There are a number of possibilities for further extensions of the issues addressed in this work, some of which have been alluded to in the previous discussion. Improvements in frame-level density estimates seem to come from using models that have the appropriate amount of detail for the training data being used. Mixture densities represent a very general mechanism that allow us to control the tradeoff of representational detail versus smoothness through various parameter tying schemes. Additionally, because of the high-dimensional spaces typical of frame-based features, it is important to use robust training methods that avoid local optima. As new tasks are addressed in speech recognition, it will be useful to have a means of automatically determining the right degree of parameter tying rather than relying on extensive trial and error testing as is done now. Automatic distribution clustering addresses this

problem to some extent at the frame level, but does not incorporate all of the various parameter tying options. This is a difficult optimization problem since the number of parameters that may be tied is large and it is not possible to try them all.

Another issue that will be important to investigate is the computational cost of recognition algorithms based on mixture distributions. The algorithms used in this thesis, while perfectly acceptable for research purposes, would need to be significantly faster for use in practical applications that are likely to require real-time response. Some initial work in this area for tied mixtures includes [7, 51].

As noted earlier, as the amount of training increases, we are likely to see a limit in the improvements that can be made by simply increasing the resolution of frame-level densities. We expect that the highest performing systems of the future will have to better account for the effects of correlation within and across frames. Models of correlation, like that given by the segmental mixture, may play an increasingly important role.

The segmental mixture model we have described can be extended in a number of directions, in addition to the general improvements to the computational aspects of mixture algorithms mentioned above. As discussed earlier, performance may be significantly improved simply by increasing the amount of training data. Additionally, our results suggest the need for a selective tying mechanism that would still allow different mixture components to model distinct trajectories, but would also permit parameter sharing in those portions of a phoneme common to all trajectories. In this vein, it would also be interesting to compare the approach developed here with a method using parameterized segmental trajectories, similar to the techniques discussed by Gish and Ng [26]. Finally, the benefits of modeling time-frequency trajectories may be more pronounced for sequences longer than a single phoneme, such as for syllables or words. The extension of this approach to longer units may require a novel approach to the question of the frame-to-model warping, but it holds the

promise of greater performance improvement.

Posterior-based segment distributions, like the CIR formalism, are a relatively new research topic in acoustic modeling and there is clearly much more to be learned concerning the properties of these models. One of the particular issues that still needs to be addressed in this area is context modeling. We pointed out theoretical problems in some of the triphone-based formalisms that have been proposed by others, and presented a mathematically consistent alternative for the left context case. Further work is still needed to achieve high accuracy for context models, and in particular, the problem of finding an effective, computationally feasible segmentation score must still be resolved. One of the potentially very attractive aspects of posterior models that we did not have time to investigate and that may address the context modeling problem, is their ability to condition on larger windows of the observation sequence, potentially including an entire utterance. Unlike conditional density models, there are no real mathematical obstacles to using this type of observation with segmental posterior distributions. The practical issue of dealing with the increase in the number of free parameters accompanying a larger observation sequence must be dealt with, but this approach may allow modeling the segmental effects of larger units of speech in a single distribution.

Bibliography

[1] S. Austin, J. Makhoul, R. Schwartz and G. Zavaliagkos, "Continuous Speech Recognition using Segmental Neural Nets," *Proc. of the DARPA Workshop on Speech and Natural Language,* pp. 249–252, Feb. 1991.

[2] L. Bahl, R. Bakis, P. Cohen, A. Cole, F. Jelinek, B. Lewis and R. Mercer, "Further Results on the Recognition of a Continuously Read Natural Corpus," in *Proc. of the IEEE Int. Conf. Acoust., Speech, Signal Processing,* pp. 872–875, April 1980.

[3] L. Bahl, F. Jelinek and R. Mercer, "A Maximum Likelihood Approach to Continuous Speech Recognition," in *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. PAMI-5(2), pp. 179–190, March 1983.

[4] J. K. Baker *et al*, "Large Vocabulary Recognition of Wall Street Journal Sentences at Dragon Systems," *Proc. of the DARPA Workshop on Speech and Natural Language*, pp. 387–392, Feb. 1992.

[5] L. Baum, T. Petrie, G. Soules and N. Weiss, "A Maximization Technique in the Statistical Analysis of Probabilistic Functions of Finite State Markov Chains," in *Ann. Math. Stat.,* Vol. 41, pp. 164–171, 1970.

[6] J. Bellegarda and D. Nahamoo, "Tied Mixture Continuous Parameter Modeling for Speech Recognition," *IEEE Trans. on Acoust., Speech and Signal Processing,* Vol. 38., No. 12, pp. 2033–2045, Dec 1990.

[7] E. Bocchieri, "Vector Quantization for the Efficient Computation of Continuous Density Likelihoods," in *Proc. of the IEEE Int. Conf. Acoust., Speech, Signal Processing,* pp. 692–695, April 1993.

[8] H. Bourlard, N. Morgan, C. Wooters, and S. Renals, "CDNN: A Context Dependent Neural Network for Continuous Speech Recognition," in *Proc. of the IEEE Int. Conf. Acoust., Speech, Signal Processing,* pp. 349–352, March 1992.

[9] H. Bourlard and C. Wellekens, "Links Between Markov Models and Multilayer Perceptrons," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 12, No. 12, pp. 1167–1178, December 1990.

[10] L. Breiman, J. Friedman, R. Olshen and C. Stone, *Classification and Regression Trees,* Wadsworth and Brooks, 1984.

[11] P. Brown, *The Acoustic Modeling Problem in Automatic Speech Recognition*, Carnegie Mellon University Ph.D. Dissertation, 1987.

[12] M. Bush and G. Kopec, "Network-based Connected Digit Recognition," in *IEEE Trans. Acoust., Speech, Signal Processing*, Vol. ASSP-35(10), pp. 1401–1413, October 1987.

[13] Y.-L. Chow, M. Ostendorf, O. Kimball, M. Krasner, G. F. Kubala, J. Makhoul, P. Price, S. Roucos and R. Schwartz, "BYBLOS, The BBN Continuous Speech Recognition System," in *Proc. of the IEEE Int. Conf. Acoust., Speech, Signal Processing,* pp. 89–93, Dallas, April 1987.

[14] G. Cybenko, "Approximation by Superpositions of a Sigmoidal Function," *Tufts University*, Technical Report, 1988.

[15] A. Dempster, N. Laird and D. Rubin "Maximum Likelihood from Incomplete Data via the EM Algorithm," *J. Royal Statist. Soc. Ser. B*, Vol. 39, No. 1, pp. 1–22, 1977.

[16] V. Digalakis, *Segment-Based Stochastic Models of Spectral Dynamics for Continuous Speech Recognition*, Boston University Ph.D. Dissertation, 1992.

[17] V. Digalakis and H. Murveit, "High-accuracy Large-Vocabulary Speech Recognition using Mixture Tying and Consistency Modeling," *Proc. of the ARPA Workshop on Human Language Technology,* pp. 292–297, March 1994.

[18] V. Digalakis, M. Ostendorf and J. R. Rohlicek, "Improvements in the Stochastic Segment Model for Phoneme Recognition," *Proc. of the DARPA Workshop on Speech and Natural Language,* pp. 332–338, October 1989.

[19] V. Digalakis, J. R. Rohlicek and M. Ostendorf, "A Dynamical System Approach to Continuous Speech Recognition," in *Proc. of the IEEE Int. Conf. Acoust., Speech, Signal Processing,* pp. 289–292, Toronto, May 1991.

[20] R. Duda and R. Hart, *Pattern Classification and Scene Analysis,* New York, NY, 1973.

[21] S. Furui, "Speaker-Independent Isolated Word Recognition Based on Emphasized Spectral Dynamics," in *Proc. of the IEEE Int. Conf. Acoust., Speech, Signal Processing,* pp. 1991–1994, Tokyo, Japan, April 1986.

[22] J. Gauvain, *personal communication.*

[23] J. Gauvain, L. Lamel, G. Adda and M. Adda-Decker, "Speaker-Independent Continuous Speech Recognition," in *Proc. of Eurospeech,* pp. 125–128, Sept. 1993.

[24] J. Gauvain and C. Lee, "MAP Estimation of Continuous Density HMM: Theory and Applications," in *Proc. of the DARPA Workshop on Speech and Natural Language*, pp. 185–190, Feb. 1992.

[25] H. Gish, "A Probabilistic Approach to the Understanding and Training of Neural Network Classifiers," in *Proc. of the IEEE Int. Conf. Acoust., Speech, Signal Processing,* pp. 1361–1364, Albuquerque, NM, April 1990.

[26] H. Gish and K. Ng, "A Segmental Speech Model with Applications to Word Spotting," in *Proc. of the IEEE Int. Conf. Acoust., Speech, Signal Processing,* April 1993, pp. II-447–450.

[27] W. Goldenthal and J. Glass, "Modelling Spectral Dynamics for Vowel Classification," in *Proc. of Eurospeech,* pp. 289–292, Sept. 1993.

[28] X. D. Huang and M. Jack, "Performance Comparison between Semi-Continuous and Discrete Hidden Markov Models," *IEE Electronics Letters*, Vol. 24, no. 3, pp. 149–150.

[29] X. D. Huang, K.F. Lee and H. Hon, "On Semi-continuous Hidden Markov Modeling," in *Proc. of the IEEE Int. Conf. Acoust., Speech, Signal Processing,* pp. 689–692, April 1990.

[30] X. D. Huang, K. F. Lee, H. Hon and M.-Y. Hwang, "Improved Acoustic Modeling with the SPHINX Speech Recognition System," in *Proc. of the IEEE Int. Conf. Acoust., Speech, Signal Processing,* pp. 345–348, May 1991.

[31] M.-Y. Hwang and X. D. Huang, "Subphonetic Modeling with Markov States - Senone," in *Proc. of the IEEE Int. Conf. Acoust., Speech, Signal Processing,* pp. I-33–36, March 1992.

[32] K. Iso and T. Watanabe, "Large Vocabulary Speech Recognition Using Neural Prediction Model," in *Proc. of the IEEE Int. Conf. Acoust., Speech, Signal Processing,* pp. 57–60, May 1991.

[33] F. Jelinek and R. Mercer, "Interpolated Estimation of Markov Source Parameters from Sparse Data," in *Proc. Workshop Pattern Recognition in Practice* pp. 381-397, May 1980.

[34] B. H. Juang and L. Rabiner, "Mixture Autoregressive Hidden Markov Models for Speaker Independent Isolated Word Recognition," in *Proc. of the IEEE Int. Conf. Acoust., Speech, Signal Processing,* pp. 41–44, April 1986.

[35] A. Kannan, *Robust Estimation of Stochastic Segment Models for Word Recognition,* Boston University M.S. Thesis, December 1992.

[36] A. Kannan and M. Ostendorf, "A Comparison of Trajectory and Mixture Modeling in Segment-Based Word Recognition," in *Proc. of the IEEE Int. Conf. Acoust., Speech, Signal Processing,* pp. 327–330, April 1993.

[37] A. Kannan, M. Ostendorf and J. R. Rohlicek, "Weight Estimation for N-Best Rescoring", Proc. of the DARPA Workshop on Speech and Natural Language, pp. 455–456, Feb. 1992.

[38] O. Kimball, M. Ostendorf and I. Bechwati, "Context Modeling with the Stochastic Segment Model," *IEEE Trans. Signal Processing* Vol. 40, No. 6, pp. 1584–1587, June 1992.

[39] O. Kimball, P. Price, S. Roucos, R. Schwartz, F. Kubala, Y.-L. Chow, A. Haas, M. Krasner, J. Makhoul, "Recognition Performance and Grammatical Constraints," *Proc. of the DARPA Workshop on Speech Recognition,* pp. 53–59, Feb. 1986.

[40] F. Kubala, S. Austin, C. Barry, J. Makhoul, P. Placeway, R. Schwartz, "BYBLOS Speech Recognition Benchmark Results," *Proc. of the DARPA Workshop on Speech and Natural Language,* pp. 77–82, Feb. 1991.

[41] C. Lee., L. Rabiner, R. Pieraccini and J. Wilpon, "Acoustic Modeling for Large Vocabulary Speech Recognition," *Computer Speech and Language,* Vol. 4, pp. 127–165, April 1990.

[42] K.-F. Lee, "Large-Vocabulary Speaker-Independent Continuous Speech Recognition: The SPHINX System," *Ph. D. Thesis,* Computer Science Department, Carnegie Mellon University, April 1988.

[43] K.-F. Lee, "Context-Dependent Phonetic Hidden Markov Models for Speaker-Independent Continuous Speech Recognition," *IEEE Trans. Acoust., Speech and Signal Processing,* Vol. ASSP-38, No. 11, pp. 599–609, April 1990.

[44] V. Lesser, R. Fennel, L. Erman, and D. Raj Reddy, "Organization of the Hearsay II Speech Understanding System," *IEEE Trans. on Acoust., Speech and Signal Processing,* Vol. ASSP-23, pp. 11–23, 1977.

[45] H. Leung, "The Use of Artificial Neural Networks for Phonetic Recognition," *Ph. D. Thesis,* Massachussets Institute of Technology, May 1989.

[46] H. Leung, I. Hetherington and V. Zue, "Speech Recognition Using Stochastic Explicit-Segment Modeling," in *Second European Conference on Speech Communication and Technology,* Genova, Italy, Sept. 1991.

[47] H. Leung, I. Hetherington and V. Zue, "Speech Recognition Using Stochastic Segment Neural Networks," in *Proc. of the IEEE Int. Conf. Acoust., Speech, Signal Processing,* pp. 613–616, March 1992.

[48] Y. Linde, A. Buzo and R. Gray, "An Algorithm for Vector Quantization Design," in *IEEE Trans. Commun.* vol. COM-28, no. 1, pp. 84-95, Jan. 1980.

[49] N. Morgan and H. Bourlard, "Continuous Speech Recognition using Multilayer Perceptrons with Hidden Markov Models," in *Proc. of the IEEE Int. Conf. Acoust., Speech, Signal Processing,* pp. 413–416, April 1990.

[50] H. Murveit, J. Butzberger and M. Weintraub, "Speech Recognition in SRI's Resource Management and ATIS Systems," *Proc. of the DARPA Workshop on Speech and Natural Language*, pp. 94–100, Feb. 1991.

[51] H. Murveit, P. Monaco, V. Digalakis and J. Butzberger, "Techniques to Achieve an Accurate Real-Time Large-Vocabulary Speech Recognition System," *Proc. of the ARPA Human Language Technology Workshop*, 1994.

[52] H. Murveit and M. Weintraub, "1000- Word Speaker-Independent Continuous-Speech Recognition Using Hidden Markov Models," in *Proc. of the IEEE Int. Conf. Acoust., Speech, Signal Processing,* pp. 115–118, April 1988.

[53] A. Nadas and D. Nahamoo, "Automatic Speech Recognition via Pseudoindependent Marginal Mixtures," in *Proc. of the IEEE Int. Conf. Acoust., Speech, Signal Processing,* pp. 1285–1287, April 1987.

[54] H. Ney, "Speech Recognition in a Neural Network Framework: Discriminative Training of Gaussian Models and Mixture Densities as Radial Basis Functions," in *Proc. of the IEEE Int. Conf. Acoust., Speech, Signal Processing,* pp. 573–576, May 1991.

[55] H. Ney, R. Haeb-Umbach, B.-H. Tran and M. Oerder, "Improvements in Beam Search for 10000-Word Continuous Speech Recognition," in *Proc. of the IEEE Int. Conf. Acoust., Speech, Signal Processing,* pp. I-9–12, April 1992.

[56] M. Ostendorf, A. Kannan, S. Austin, O. Kimball, R. Schwartz, J. R. Rohlicek, "Integration of Diverse Recognition Methodologies Through Reevaluation of N-Best Sentence Hypotheses," *Proc. of the DARPA Workshop on Speech and Natural Language,* pp. 83–87, Asilomar, CA, Feb. 1991.

[57] M. Ostendorf and S. Roucos, "A Stochastic Segment Model for Phoneme-Based Continuous Speech Recognition," *IEEE Trans. on Acoust., Speech and Signal Processing,* Vol. ASSP-37, No. 12,pp. 1857–1869, Dec. 1989.

[58] D. Pallett, Results for the Sept. 1992 Resource Management Benchmark, presented at the DARPA Workshop on Artificial Neural Networks and CSR, Sept. 1992.

[59] J. Patterson and B. Womack, "An Adaptive Pattern Classification Scheme," *IEEE Trans. on Systems, Science and Cybernetics*, Vol. SSC-2, pp. 62–67, Aug. 1966.

[60] D. Paul, "Speaker Stress-Resistant Continuous Speech Recognition," in *Proc. of the IEEE Int. Conf. Acoust., Speech, Signal Processing,* pp. 283–286, May 1988.

[61] D. Paul, "The Lincoln Robust Continuous Speech Recognizer," in *Proc. of the IEEE Int. Conf. Acoust., Speech, Signal Processing,* pp. 449–452, May 1989.

[62] D. Paul, "The Lincoln Tied-Mixture HMM Continuous Speech Recognizer," in *Proc. of the IEEE Int. Conf. Acoust., Speech, Signal Processing,* pp. 329–332, May 1991.

[63] D. Paul and J. Baker, "The Design for the Wall Street Journal-based CSR Corpus," in *Proc. of the ARPA Speech and Natural Language Workshop* , pp. 357–362, Feb. 1992.

[64] M. Phillips, J. Glass and V. Zue, "Modeling Context Dependency in Acoustic-Phonetic and Lexical Representations," *Proc. of the DARPA Workshop on Speech and Natural Language,* pp. 71–76, Asilomar, CA, Feb. 1991.

[65] A. Poritz and A. Richter, "On Hidden Markov Models in Isolated Word Recognition," in *Proc. of the IEEE Int. Conf. Acoust., Speech, Signal Processing,* pp. 705–708, April 1986.

[66] P. Price, W. Fisher, J. Bernstein, D. Pallett, "The DARPA 1000-Word Resource Management Database for Continuous Speech Recognition," in *Proc. of the IEEE Int. Conf. Acoust., Speech, Signal Processing,* pp. 651–654, April 1988.

[67] L. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," *Proceedings of the IEEE*, Vol. 77, no. 2, Feb. 1989.

[68] T. Robinson and F. Fallside, "A Recurrent Error Propagation Network Speech Recognition System," *Computer Speech and Language,* Vol. 5, no. 3, pp. 259–274, July 1991.

[69] S. Roucos, M. Ostendorf, H. Gish and A. Derr, "Stochastic Segment Modeling Using the Estimate-Maximize Algorithm," in *Proc. of the IEEE Int. Conf. Acoust., Speech, Signal Processing,* pp. 127–130, April 1988.

[70] S. Roucos and M. Ostendorf, "A Stochastic Segment Model for Phoneme-Based Continuous Speech Recognition," in *Proc. of the IEEE Int. Conf. Acoust., Speech, Signal Processing,* pp. 73–76, April 1987.

[71] D. Rumelhart, G. Hinton and R. Williams, "Learning Representations by Back-Propagating Errors," *Nature*, Vol. 323, pp. 533–536, Oct. 1986.

[72] R. Schwartz, T. Anastasakos, F. Kubala, J. Makhoul, L. Nguyen and G. Zavaliagkos, "Comparative Experiments on Large Vocabulary Speech Recognition," in *Proc. of the ARPA Human Language Technology Workshop* pp. 75–80, March 1993.

[73] R. Schwartz and Y.-L. Chow, "The N-Best Algorithm: An Efficient and Exact Procedure for Finding the N Most Likely Sentence Hypotheses," in *Proc. of the IEEE Int. Conf. Acoust., Speech, Signal Processing,* pp. 1857–1869, April 1990.

[74] R. Schwartz, Y. Chow, O. Kimball, S. Roucos, M. Krasner and J. Makhoul, "Context-Dependent Modeling for Acoustic-Phonetic Recognition of Continuous Speech," in *Proc. of the IEEE Int. Conf. Acoust., Speech, Signal Processing,* pp. 1205–1208, March 1985.

[75] R. Schwartz, O. Kimball, F. Kubala, M. Feng, Y. Chow, C. Barry, and J. Makhoul, "Robust Smoothing Methods for Discrete Hidden Markov Models," in *Proc. of the IEEE Int. Conf. Acoust., Speech, Signal Processing,* pp. 548–551, May 1989.

[76] R. Schwartz and V. Zue, "Acoustic-Phonetic Recognition in BBN SPEECHLIS," in *Proc. of the IEEE Int. Conf. Acoust., Speech, Signal Processing,* pp. 21–24, Philadelphia, April 1976.

[77] K. Sugawara, M. Nishimura, K. Toshioka, M. Okochi and T. Kaneko, "Isolated Word Recognition using Hidden Markov Models," in *Proc. of the IEEE Int. Conf. Acoust., Speech, Signal Processing,* pp. 1–4, March 1985.

[78] J. Tebelskis, A. Waibel, B. Petek and O. Schmidbauer, "Large Vocabulary Speech Recognition Using Neural Prediction Model," in *Proc. of the IEEE Int. Conf. Acoust., Speech, Signal Processing,* pp. 61–64, May 1991.

[79] A. Viterbi, "Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm," in *IEEE Trans. on Information Theory,* IT-13, pp. 260–269, 1967.

[80] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano and K. Lang, "Phoneme Recognition Using Time-Delay Neural Networks," in *Proc. of the IEEE Int. Conf. Acoust., Speech, Signal Processing,* pp. 107–110, April 1988.

[81] C. Wayne, "A Snapshot of Two DARPA Speech and Natural Language Programs," *Proc. of the DARPA Workshop on Speech and Natural Language,* pp. 403–404, Feb. 1991.

[82] P. Woodland and S. Young, "The HTK Tied-State Continuous Speech Recognizer," *Proc. of Eurospeech,* pp. 2207–2210, Sept. 1993.

[83] G. Zavaliagkos, R. Schwartz and J. Makhoul, "Elliptical Basis Functions for Segment Modeling," in *Proc. of the IEEE Int. Conf. Acoust., Speech, Signal Processing,* pp. 513–516, April 1993.

[84] G. Zavaliagkos, Y. Zhao, R. Schwartz and J. Makhoul, "A Hybrid Segmental Neural Net/Hidden Markov Model System for Continuous Speech Recognition", *IEEE Trans. on Speech and Audio Processing*, Vol. 2, No. 1, Part 2, pp. 151–160, Jan 1994.

[85] V. Zue, J. Glass, M. Phillips and S. Seneff, "Acoustic Segmentation and Phonetic Classification in the SUMMIT Speech Recognition System," in *Proc. of the IEEE Int. Conf. Acoust., Speech, Signal Processing,* pp. 389–392, May 1989.

OWEN KIMBALL

57 Marlboro St. Apt. 2

Belmont MA 02178

(617) 489-3958

EDUCATION:

Ph.D. (Electrical Engineering) Boston University, 1994.

Thesis Supervisor: M. Ostendorf.

M.S. (Computer Science) Northeastern University, 1988.

B.A. (Mathematics) University of Rochester, 1982, *magna cum laude.*

EXPERIENCE:

1989 – present **Boston University,** Boston MA

Research Assistant. Research in segmental acoustic models for continuous speech recognition. Investigation of a segmental recognition framework based on classification and segmentation scoring. Development of a generalized segmental mixture model that captures both segment-length dynamics and finer time-scale variability.

1982 – 1989 **BBN Laboratories, Inc.,** Cambridge MA

Scientist. Research and design of algorithms for continuous speech recognition. Acoustic modeling techniques using hidden-Markov models. Design and implementation of parallel algorithms for a large-scale multiprocessor speech recognition system. Research into recognition-based aids for the hearing-impaired.

1980 – 1982 **University of Rochester,** Rochester NY
Undergraduate Research Assistant. Computer vision research in the areas of shape from shading and rigid-body motion detection.

HONORS:

Presidential University Graduate Fellow, Boston University, 1989-1990.
Member *Phi Beta Kappa.*

PROFESSIONAL AFFILIATIONS:

Member IEEE Signal Processing Society

PUBLICATIONS:

"On the Use of Tied-Mixture Distributions," O. Kimball and M. Ostendorf, *Proceedings of the DARPA Workshop on Human Language Technology,* March 1993.

"Continuous Word Recognition Based on the Stochastic segment Model," M. Ostendorf, A. Kannan, O. Kimball, and J. R. Rohlicek, *Proceedings of the DARPA Continuous Speech Recognition Workshop,* September 1992.

"Context Modeling with the Stochastic Segment Model," O. Kimball, M. Ostendorf, and I. Bechwati, *IEEE Transactions on Signal Processing,* June 1992, pp. 1584-1587.

"Recognition Using Classification and Segmentation Scoring," O. Kimball, M. Ostendorf and J. R. Rohlicek, *Proceedings of the DARPA Workshop on Speech and Natural Language,* February 1992, pp. 197-201.

"Integration of Diverse Recognition Methodologies Through Reevaluation of N-Best Sentence Hypotheses," M. Ostendorf, A. Kannan, O. Kimball, S. Austin, M. Ostendorf, R. Schwartz, J. R. Rohlicek, *Proceedings of the DARPA Workshop on Speech and Natural Language,* February 1991, pp. 83-87.

"Robust Smoothing Methods for Discrete Hidden Markov Models," R. Schwartz, O. Kimball, F. Kubala, M-W. Feng, Y. L. Chow, C. Barry, and J. Makhoul, *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing,* May 1989, pp. 548–551.

"Continuous Speech Recognition Results of the BYBLOS System on the DARPA 1000-Word Resource Management Database," F. Kubala, Y. Chow, A. Derr, M. Feng, O. Kimball, J. Makhoul, P.Price, J. Rohlicek, S. Roucos, R. Schwartz, and J. Vandegrift, *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing,* April 1988, pp. 291–294.

"Efficient Implementation of Continuous Speech Recognition on a Large Scale Parallel Processor," O. Kimball, L. Cosell, R. Schwartz, and M. Krasner, *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing,* April 1987, pp. 852–855.

"BYBLOS: The BBN Continuous Speech Recognition System," Y.L. Chow, M.O. Dunham, O. Kimball, M. Krasner, F. Kubala, J. Makhoul, P. Price, S. Roucos, and R. Schwartz, *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing,* April 1987, pp. 89–93.

"Continuous Speech Recognition on a Butterfly Parallel Processor," L. Cosell, O. Kimball, R. Schwartz, and M. Krasner, *Proceedings of the International Conference on Parallel Processing* August 1986, pp 717-720.

"The Role of Word-Dependent Coarticulatory Effects in a Phoneme-Based Speech Recognition System," Y.L. Chow, R. Schwartz, S. Roucos, O. Kimball, P.

Price, F. Kubala, M.O. Dunham, M. Krasner, and J. Makhoul, *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing,* April 1986, pp. 1593–1596.

"Recognition Performance and Grammatical Constraints," O. Kimball, P. Price, S. Roucos, R. Schwartz, F. Kubala, Y.-L. Chow, A. Hass, M. Krasner, and J. Makhoul, *Proc. of the DARPA Speech Recognition Workshop,* February 1986, pp. 53–59.

"Context-Dependent Modeling for Acoustic-Phonetic Recognition of Continuous Speech," R. Schwartz, Y. L. Chow, O. Kimball, S. Roucos, M. Krasner, and J. Makhoul, *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing,* March 1985, pp. 1205–1208.

"Development of a Vidvox Speech Communication Aid," M. Krasner, A.W.F. Huggins, R. Schwartz, O. Kimball, Y. L. Chow, and J. Makhoul, presented at the AVIOS Conf. Sept, 1984, Arlington, VA.

"Rigid Body Motion from Depth and Optical Flow," D.H. Ballard and O. Kimball, *Computer Vision, Graphics, and Image Processing,* April 1983.

"Constraint Interaction in Shape from Shading Algorithm," D.H. Ballard, C.M. Brown, and O. Kimball, DARPA Workshop on Image Understanding, June 1982.

REFERENCES:

Prof. Mari Ostendorf

Boston University.

Electrical, Computing and Systems Engineering

44 Cummington St.

Boston, MA 02215

(617) 353-5430

Dr. John Makhoul

BBN Inc.

70 Fawcett St.

Cambridge, MA 02138

(617) 873-3332

Dr. J. Robin Rohlicek

BBN Inc.

70 Fawcett St.

Cambridge, MA 02138

(617) 873-3894