BOSTON UNIVERSITY

COLLEGE OF ENGINEERING

Thesis

ROBUST ESTIMATION OF STOCHASTIC SEGMENT MODELS FOR WORD RECOGNITION

by

ASHVIN KANNAN

B.E.(Hons) Electronics & Communication,
University of Roorkee, Roorkee, India, 1990.

Submitted in partial fulfillment of the

requirements for the degree of

Master of Science

1992

Approved by

First Reader: ______________________________

Mari Ostendorf, Ph.D.
Assistant Professor of Electrical, Computer and Systems Engineering
Boston University

Second Reader: ______________________________

Jan Robin Rohlicek, Ph.D.
Senior Scientist
Bolt Beranek and Newman Inc.

Third Reader: ______________________________

David Castañon, Ph.D.
Associate Professor of Electrical, Computer and Systems Engineering
Boston University

Acknowledgments

About three years ago, I wrote to Mari Ostendorf expressing interest in working on speech coding with her. I was not really expecting a reply, as no other professors to whom I wrote similar letters bothered to reply. I was excited by just the reply and had my fingers crossed, till one fine day she offered me a research assistantship. I was indeed fortunate that she did not hold my *faux pas*, of addressing her all along as "Dear Sir", against my candidature. I am very grateful for the opportunity to work with Mari. By setting very high standards and being a perfectionist she has set an example for me to follow. Her patience with my many (often naive) questions and constant guidance have been largely responsible for the completion of this work.

Interacting with Robin Rohlicek has been rewarding and many interesting ideas in this thesis resulted from meetings with him and Mari. David Castañon, while being on my committee, also had a position of being an "outside-referee" and offered constructive criticism to improve the quality of this thesis. I thank Carol Espy-Wilson for her help in the design of questions related to tree design for triphone clustering and subsequent tree interpretation.

The environment in SPI lab has been very enjoyable and conducive to productive work (even though at times it felt like the Arctic !). I wish to thank Vassilios Digalakis, Owen Kimball, Nanette Veilleux, Burhan Necioglu, Ken Ross, Cynthia Fong and Becky Bates for many interesting, distracting and desultory discussions. Vassilios showed great patience in answering my questions even after he joined SRI and moved to the other coast. To Owen, I remain indebted for dazzling me with the powers of Emacs when I was very impressionable. Working with him has been an eye-opener in many other ways too. Special thanks go to Becky for taking care of all the administrative work.

Venkatesh Chari and Yogen Patel provided spirited opposition in many late-night enjoyable (and inconclusive) arguments about whether one should model “knowledge” with “ignorance” or “ignorance” with “knowledge”.

This thesis could not have been completed but for a very pleasant home away from home. It has been wonderful to stay with Abhijit Kulkarni and Dinesh Venkatesh and share our experiences, from sedate course-work to wild escapades. They were (and still are) very understanding and this, to some extent, is exemplified by their stoic endurance to so much of my *rajma*.

I could not have come this far if it wasn't for the constant support of my family. I look forward to the day I can be of more help to them. This thesis is dedicated to my parents, for the innumerable sacrifices they have made to keep us together and to Sridhar, for shouldering much more than his share of responsibilities.

This research was jointly supported by NSF and DARPA under NSF grant number IRI-8902124.

ROBUST ESTIMATION OF STOCHASTIC SEGMENT MODELS

FOR WORD RECOGNITION

(Order No.)

ASHVIN KANNAN

Boston University, College of Engineering, 1992

Major Professor: Mari Ostendorf,
Assistant Professor of Electrical, Computer
and Systems Engineering

Abstract

In this work, we develop robust estimation techniques for a continuous-word recognition system using the Stochastic Segment model (SSM). This work is done under the N-best rescoring formalism, where a less complex system than the SSM is used to generate candidate hypotheses which are then rescored and reranked by the SSM. Components of the system that are the focus of this work include estimation of weights for score combination and robust parameter estimation using clustering techniques to model context. In particular, we develop several agglomerative and divisive clustering techniques for multivariate Gaussian distributions, which we use to cluster triphone models. This leads to better estimates with fewer parameters resulting in reduction in word error and storage/computation costs over using unclustered triphones. We also implement an SSM system based on microsegments which combines mixture modeling with trajectory modeling and examine the tradeoffs involved between the allocation of distributions for time sequences versus mixture components. Word recognition performance is reported on the speaker-independent corpus of the Resource Management database. The recognition rates achieved with the SSM, 7.0% word error for context-independent and 4.7% for context-dependent systems, are comparable to the state-of-the-art in speech recognition. Combining the SSM scores with Hidden Markov model scores reduces word error rate from 3.8% to 3.1%.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

As speech is considered the "natural" mode of communication for humans, it would be desirable to have a human-machine interface with speech communication capabilities. Speech recognition, the transcription of speech into text by a computer, would be the front-end of such an interface. There are many advantages of such a device. The obvious application would be in the form of a voice typewriter. More ambitiously, it would be a part of a larger speech understanding system such as an airline information and reservation system which would enable remote access to the system via telephone and/or for unskilled computer users. Various modes of speech recognition can be investigated including, speaker-dependent or speaker-independent speech, isolated words or continuous speech, read or spontaneous speech, and clean or noisy speech. In this thesis we will consider the problem of speaker-independent, continuous speech recognition (CSR) with clean read speech.

An important advance in acoustic modeling for speech recognition has been the development of segment-based models like the stochastic segment model (SSM), proposed by Ostendorf and Roukos [32]. The SSM is a statistical framework for modeling a whole segment of speech, which allows for the incorporation of segmental features and for directly modeling correlation in time of frame-based features (a frame is a fixed-duration window of speech). The SSM represents an alternative to the commonly used hidden Markov model (HMM), which has yielded good results but suffers from some intrinsic shortcomings including the conditional independence assumption of the observations given the state sequence and the inability to incorporate segmental features. Since the SSM has a more costly recognition search, most work has been done in phone recognition [13, 22] and small vocabulary word recognition [32] where it has been shown to be a viable alternative to the HMM.

The goal of this study is to develop an SSM-based word recognition system and demonstrate the advantages of such a system over systems based on HMMs. We address two important problem areas; the reduction of search space to be manageable and capturing co-articulation effects.

The complexity of the SSM results in a very large search space for an optimal search. This search space is drastically reduced by using the N-best rescoring formalism. In this, a less complex HMM based system generates a set of candidate sentence hypotheses for an utterance that are then rescored and reranked by the SSM system. This formalism also has the potential of achieving better performance that any one system by the combination of the scores of the hypotheses from the different systems. We discuss the issues involved in combining scores in an effective manner. We have also investigated the effect of changing different system parameters.

Speech is strongly influenced by co-articulation effects which may be captured by modeling context, explicitly using information of the identity of neighboring phones in the acoustic model. An important unit of detail is the triphone, a phonetic model that is conditioned on the identities of its adjacent phones. Context-modeling and specifically triphones, have been responsible for some of the biggest gains in CSR [27, 42]. The SSM has many more parameters than the HMM so robust parameter estimation is an important issue. To address this problem, we investigated clustering of triphones, using both agglomerative and divisive clustering techniques. As an alternative approach, we also explored the use of microsegment (parts of a segment) mixtures in word recognition and examine the trade-offs between modeling context using mixtures instead of explicitly modeling trajectories.

The rest of the thesis is organized as follows. In Chapter 2, a background to speech recognition using statistical models (with emphasis on the SSM) is given together with a summary of past work done in modeling context. A description of the experimental paradigm used in this work along with system development related issues is in Chapter 3. Chapter 4 describes clustering advances in context modeling for the SSM. An SSM recognition system based on microsegments is developed in Chapter 5. Finally, Chapter 6 summarizes the conclusions of this thesis, with a discussion of the significance and implications of the results for future work.

Chapter 2

Background

Speech recognition has been an area of active research for a number of years. Initial approaches were "knowledge-based" and usually consisted of a series of "if-then" type of questions on acoustic-phonetic "features". However, this approach met with limited success, and most successful recognition systems today are based on a statistical approach. In this chapter we provide a background to research in speech recognition using the statistical approach. We shall compare the Hidden Markov Model (HMM) with the Stochastic Segment Model (SSM) and look at the SSM structure in greater detail. Since context modeling will be an important aspect of this thesis, previously developed techniques for modeling context with the HMM and SSM will also be examined. Finally, there is a brief description of the evaluation methods and the corpus that will be used in this study.

2.1 Statistical Models

Using a stochastic model of speech has many advantages. For example, we can draw upon the rich research advances in statistics. In addition, the algorithms used to train models and for recognition are well understood. Perhaps the biggest advantage of a statistical approach is that most, if not all, of the training is done automatically, thus saving costly human time.

2.1.1 Hidden Markov Models

Hidden Markov Models [5, 38] are currently the most popular statistical models used in speech recognition. The time-varying nature of speech is modeled as a Markov process with a discrete state space. Each state produces observations according to a probability distribution characteristic of that state. These observations may take on discrete or continuous values. In either case, the observations represent a fixed-duration window of the speech signal (a frame). The states are not directly observable, and thus the name Hidden Markov Model. The HMM is characterized by the number of states in the model, probability distributions for the observations conditioned on the states, state transition probability distributions and the initial state distribution.

To construct word models for speech recognition, HMMs may represent entire words, or HMMs for phones (a linguistic unit subword model, e.g. "eh", "t", "m") may be connected in a network according to a pronunciation dictionary. Each subword model may consist of a number of states. An efficient algorithm for training (i.e., estimating the parameters of) the models is the Baum-Welch algorithm (or equivalently the Expectation-Maximize algorithm [12]). Recognition, which typically involves finding the most likely state sequence, can be done efficiently using the Viterbi algorithm.

Despite their many attractive properties, HMMs have some disadvantages. Observations from adjacent frames are assumed to be conditionally independent given the state sequence, clearly an unrealistic assumption as it is a well known fact that adjacent frames are fairly correlated. Incorporation of segmental features like voicing, stress and duration, which can be useful in speech recognition, is also difficult in the HMM framework.

2.1.2 Stochastic Segment Model

Given the fact that adjacent frames of speech are highly correlated, researchers have considered modeling the segment (phone) as a whole. Bush and Kopec first proposed a segmental network-based digit recognition [10]. Since then, Ostendorf and Roukos proposed the Stochastic Segment Model (SSM) [32], MIT has developed the SUMMIT system which uses segment models and linguistically motivated features [35], and more recently segmental neural networks (SNN) are also being used [2, 29]. As the proposed research will be done in the SSM framework, we shall examine only this approach in greater detail.

In the SSM a phone is represented as a variable length observation sequence (segment) $\mathbf{Y}$, which is a matrix of dimension $L \times q$:

$$\mathbf{Y} = [y_1 y_2 \ldots y_L]^t \tag{2.1}$$

each column y_i being the q-dimensional feature vector for the i^{th} time frame. The models for a phone α consists of 1) a family of joint density functions (one for every observation length), and 2) a collection of mappings that specify the particular density function for a given observation length. Typically, the model assumes that segments are described by a fixed-length sequence of locally time-invariant regions (or regions of tied distribution parameters).

The specific version used in this work uses Gaussian density functions and assumes that frames within a segment are conditionally independent given the segment length, and the distributions used for each frame are determined by a linear time warping of the L-long observation to the m regions of the model. In this case, the probability of a segment given phone α, $p(\mathbf{Y}|\alpha)$, is the product of the probability of each observation y_i. This is similar to an HMM with a constrained state trajectory and does not take full advantage of the SSM formalism.

Recently the stochastic segment model has been reformulated as a dynamical system (DS) within the SSM framework [13]. In the DS approach, a phone is modeled as having a Gauss-Markov state process with the observed speech being a noisy transformation of the state. Recognition is performed using a Kalman predictor to compute the probability of a segment in terms of its innovations. This formulation permits modeling of time-correlation of the observation sequence. Although we shall not consider the DS model in this work, we anticipate algorithms developed in this thesis to be extended to this case.

The training algorithm for both the independent frame SSM and the DS model is iterative and consists of alternating between segmenting data with the current parameters and finding maximum-likelihood estimates for the parameters given the newly segmented data.

Phone recognition with the SSM is performed by jointly choosing the maximum probability phone sequence and segmentation. For each hypothesized phone sequence, $\underline{\alpha}$, the most likely segmentation is determined by maximizing the likelihood over all possible segmentations, $\underline{s}$, :

$$l(\underline{\alpha}) = \max_{\underline{s}} \log\,[p(\underline{\mathbf{Y}} \mid \underline{\alpha})p(\underline{\alpha})]. \tag{2.2}$$

This search can be implemented efficiently using a dynamic programming algorithm to get the recognized phone sequence as:

$$\hat{\underline{\alpha}} = \arg\max_{\underline{\alpha}}\; l(\underline{\alpha}). \tag{2.3}$$

The obvious extension to word recognition would be to expand a network of words into a network of phones with the help of a dictionary (taking into consideration multiple pronunciations) and then search over the allowable word sequences (which effectively constrains the phone sequences). The computation required to use this technique is considerably greater than that for HMMs (due to the larger search space resulting from hypothesizing all possible segment lengths), making it impractical to implement the standard optimal dynamic programming search algorithms. Therefore, a constrained search for SSM word recognition will be given in Section 3.1.

2.2 Context Modeling

Context-independent models are made for phones by ignoring the identity of the neighboring phones. However, the acoustic characteristics of a phone are significantly affected by the neighboring phones, particularly near the boundaries. To account for the effects of phone context, or co-articulation, many more *context-dependent* models are used by representing phones conditioned on the neighboring phone identities. Modeling of context was first proposed for HMMs in [4] and implemented in [42]. A popular unit for modeling context is the triphone, a phone that is conditioned on its left and right phonetic contexts. More detailed models may also be considered, such as word dependent phone models [27].

Robust estimation of parameters is a key problem in research on context-dependent models. While there is motivation for capturing more detail, the number of such different models increases exponentially leading to severe estimation problems due to limited training data. One solution to robust estimation of context-dependent models for HMMs has been to use distribution smoothing techniques such as deleted interpolation [19]. Alternatively, shared mixtures could be used in which a set of distribution functions is shared across contexts and the weights used for the mixture components are context-dependent. In addition, models in “similar” contexts may be identified and merged to reduce the number of models and also to have more training data for each model. For triphones, the concept of "generalized triphones" was proposed which involves agglomerative clustering of the triphones [27]. Divisive clustering using binary branching trees have also been considered for clustering larger context such as the five preceding and five following phones [6] and for clustering based on stress, utterance position, syllable boundary, etc., as in [28]. The tree-based clustering has superior generalization capability and thus outperforms agglomerative clustering for vocabulary-independent tasks [28].

Recognizing the need for modeling context to be able to capture co-articulation, context-dependent models have been developed for the SSM resulting in error reduction similar to that observed for HMMs. This has been achieved by tying covariance parameters across classes of similar context [22] rather than using distribution smoothing or shared mixtures. (By "tying parameters", we mean that different distributions are defined to share the same values for these parameters). The specific classes over which parameters are tied were first based on less detailed context (e.g., left-context is less detailed than triphone context). A system of empirically determined thresholds is used in association with a hierarchy of context classes. The model chosen for use during recognition is the most detailed context model that is adequately trained according to a threshold on the number of training samples. Linguistic classes (e.g., manner of articulation) were used to provide a definition of similar context for covariance tying. Automatic clustering using the K-means algorithm was also performed, initializing with means given by the manner of articulation (e.g., stop, nasal) cluster means. For phone classification in TIMIT, slightly better results were reported for linguistic clustering than automatic clustering while no improvements were observed in word recognition on the Resource Management corpus [22].

2.3 Corpus and Performance Evaluation

The corpus used in the experiments reported here is DARPA's speaker-independent Resource Management (RM) task [37], which has continuous read speech based on a 1000-word vocabulary in the naval domain. We will use the speaker-independent SI-109 training set, which consists of a total of 3977 sentences spoken by 109 speakers. There are several test sets associated with this corpus, each consisting of 300 sentences (approximately 2500 words) spoken by a different set of 10 speakers (30 sentences each) not in the training set. Test and training sets contain both male and female speakers.

Word recognition performance is reported in terms of percent word error. This is computed as the ratio of the sum of words substituted, inserted and deleted to the true (reference) number of words in the sentence. Homophones, words that sound alike but differ in spelling, such as "too" and "two", are counted as errors because they can be distinguished when a grammar is used.

There are 950 sentence patterns (unique surface semantic-syntactic patterns) in the database. The standard grammar used in this task to constrain the recognition search is a word-pair grammar that allows all two-word sequences occurring in these sentence patterns and has a perplexity[1] of about 62 [25]. The word-pair grammar is used in all word recognition experiments in this study.

The signal processing of the speech that was used in our experiments involves computing the first 14 Mel-warped cepstral coefficients (excluding C_0) every 10 mil-

[1]The perplexity of the grammar is defined as $L = 2^K$, where

$$K = -\frac{1}{n} \sum_{i=2}^{n} \log_2[\hat{P}(w_i | w_{i-1}, \ldots, w_1)]$$

is the average per word entropy of the language model, and $\hat{P}(w_1) = 1$ where $w_1, w_2, \ldots, w_n$ is the word sequence formed by concatenating all sentences in the training set [25].

liseconds using a 20 millisecond window. The first differences of the cepstra and of energy were also computed at each 10 millisecond frame, yielding 16 more coefficients (including the first difference of C_0). This feature vector is transformed to 29 dimensions using linear discriminants to reduce dimensionality, as in [14].

Chapter 3

System Development

This chapter presents work done in the development of a basic SSM-based word recognition system. This work includes addressing the issues of reduced computation searches using the N-best rescoring formalism and robust estimation of weights for combining scores in this formalism. We also investigate some variations in the structure of the SSM.

3.1 The N-Best Rescoring Formalism

The exhaustive dynamic programming recognition technique given in Section 2.2 (for word recognition) is computationally too expensive to be practical. A solution to this problem is the N-Best rescoring formalism [31], jointly developed by researchers at BU and BBN (see Figure 3.1). Under this paradigm, a recognition system produces the N-best hypotheses for a sentence which are subsequently rescored by other (more complex) knowledge sources. The sentence hypotheses are then reranked according to a weighted linear combination of the different scores (e.g. log probabilities). This paradigm has the potential of achieving better performance than that of any

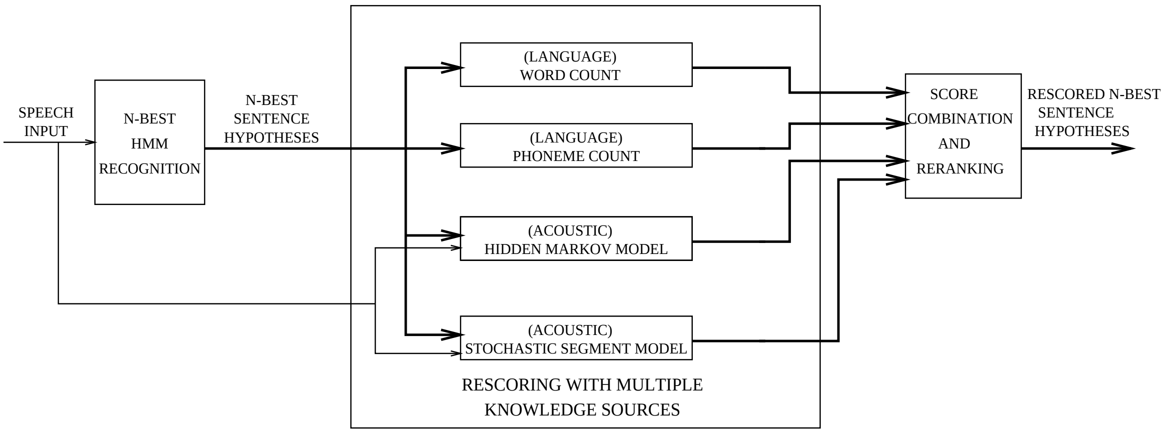

Figure 3.1: The N-best rescoring formalism, illustrated with the knowledge sources used in this work.

individual knowledge source.

The BBN BYBLOS system [11, 24] is used to generate the list of hypotheses. This is a high performance HMM system that uses context-dependent models including cross-word-boundary contexts. The HMM observation densities are modeled by tied Gaussian mixtures. For this problem, the cost of finding the exact top N-best hypotheses grows exponentially with N. As a result several approximate N-best algorithms have been proposed [40, 41]. The BYBLOS system generates an approximate N-best list.

Word recognition by the SSM is performed by rescoring candidate phone sequences that correspond to the words in the hypotheses. The phone sequences (actually networks) are made from the word sequences for each hypothesis using a dictionary of pronunciations, and then scored using the SSM. As this leads to a significant reduction in search space for the SSM, the computation is reduced to a reasonable amount.

Once the N-best list is rescored by the different knowledge sources (such as the SSM, and a grammar score, if used), they are reranked according to a linear combination of the scores from the different knowledge sources. We find that using SSMs with HMMs leads to an improvement over the HMM performance. If we perform a re-ranking of the list without considering the HMM scores, we have a method of evaluating performance for the SSM alone. In experiments, we have used N = 20 and it was seen that the correct sentence is included in this list about 98% of the time by the BYBLOS system.

In this work 61 phones were used, each with an underlying segment of 8 independent distributions. Out of these 61 phones, 9 do not occur in the training or test sets due to the particular word dictionary used, so effectively 52 phones are used. Gender-dependent models were used. Though the HMM scores can also be combined in the score combination as shown in Figure 3.1, we do not do so in most of this work. Unless otherwise noted, these conditions for the SSM and the N-best paradigm were maintained in this work.

3.2 Score Combination

The N-best paradigm has the potential of achieving better performance than that of any individual knowledge source, if these scores are combined in an effective manner. In this work, we chose to use a linear combination of log probabilities, specifically SSM log acoustic probability, number of words in the sentence (insertion penalty), number of phones in the sentence, and optionally, HMM log acoustic probability. Weights for the scores in this combination are estimated on a development test set (Feb 89). Performance using these weights on the same test set are optimistic and are used for testing different SSM systems only in system development. At the end, another test set (Oct 89) is rescored with this system using the weights that were estimated based on the earlier test set.

In the initial work [31], we estimated the set of weights that minimizes a generalized mean of the rank of the correct answer:

$$m(\mathcal{S}) = |\frac{1}{|\mathcal{S}|}\sum_{i\in\mathcal{S}} r(i)^p|^{1/p}, \tag{3.1}$$

where $r(i)$ is the rank of the correct answer in sentence i of a set $\mathcal{S}$ sentences and p determines the type of mean. For example, $p = 1$ specifies the average, $p = 2$ specifies the root-mean-square, $p = -1$ specifies the harmonic mean, and $p = -\infty$ only counts the percent correct. For speech recognition applications $p = -\infty$ would be appropriate, but for speech understanding applications, $p = 1$ might be more useful. In practice, we found that the different values of p did not have a significant impact on the results. Estimation of the weights is an unconstrained multi-dimensional minimization problem. The algorithm used initially was Powell's method [36], which iteratively minimizes the generalized mean (Equation 3.1) by optimizing the weights in successive conjugate directions.

Further experience using the generalized mean and Powell's method suggested that the result was very sensitive to the large number of local minima in the optimization criterion. Several steps were taken to address this issue [20]. First, the optimization criterion was changed to minimize the average word error in the top ranking hypothesis in the hope of obtaining a "smoother" weight-space having fewer local minima. Second, we evaluated the error function at a large number of points in the weight-space, specifically, on a multi-dimensional lattice spanning the range of probable weights. Powell's method may be used with points on the lattice as the initial estimate of weights to find the best performance, or the points on a fine lattice may be evaluated directly. Powell's method would be more suitable if there are not many local optima. As that was not so in our case, we found it more convenient to evaluate the points directly on a fine lattice.

The error function used as an optimization criterion is piece-wise constant over the weight space. The weights that result in a particular value of the error function vary in weight space defining a region (cell) by a set of inequalities that describe a polytope:

$$\mathbf{s}_{i1}.\mathbf{w} \geq \mathbf{s}_{ij}.\mathbf{w}, \qquad i = 1, \ldots, M. \tag{3.2}$$

where $\mathbf{w}$ is a vector of weights, $\mathbf{s}_{i1}$ is the vector of scores of the top ranking hypothesis for the ith sentence, $\mathbf{s}_{ij}$ is the vector of scores for the jth ranking hypothesis of the same sentence and M is the number of sentences in the test set. The top ranking hypothesis is constant for all sentences for all weights in this polytope.

We find the weights corresponding to the "center" of cell, in the hope that they will be more robust than other weights from the cell. To do this we measure the amount of slack for the different coefficients along the coordinate axes such that the weight remains within the cell:

$$\mathbf{s}_{i1}.(\mathbf{w} + \alpha\mathbf{e}) \geq \mathbf{s}_{ij}.(\mathbf{w} + \alpha\mathbf{e}) \tag{3.3}$$

where $\alpha\mathbf{e}$ represents the slack (perturbation) along the co-ordinate directions, $\mathbf{e}$, that is permissible while still maintaining the set of inequalities. These define the end-points of the polytope along the co-ordinate directions. We determine the "center" of this cell as the mean of these points. Since the cell is a convex region, we are assured that the "center" does indeed lie within the cell. This procedure of finding the "center" is iterated a few times to get a better "center". The product of the slacks in the co-ordinate directions at the "center" is an approximate indicator of the "volume" of the cell. Figure 3.2 illustrates this for a two-dimensional weight space. If more than one cell gives the same performance, we choose the one with the largest "volume". Weights which correspond to the "center" of this cell are used for combining scores in the test set. It should be noted that abutting cells may have the same performance. In such a case, the algorithm would still pick the largest cell as the best cell.

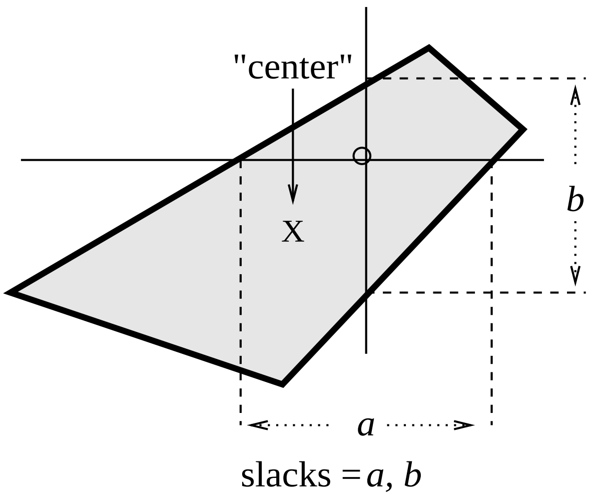

Figure 3.2: A polytope in two-dimensional weight space with an illustration of slacks.

Experiments were conducted to gain a better understanding of the weight space. The SSM system used context-dependent models with context tying based on automatic clustering (see Section 4 for a description of clustering). The weights were trained on the Feb 89 test set and then used to combine scores for the Oct 89 test set. The training of weights may be either gender-dependent or gender-independent and both were examined.

Figures 3.3 to 3.6 are contour plots for the word error distribution as a function of scores on the two test sets. The contours have been drawn for the ten lowest word errors, with intensity increasing with accuracy. When plotting the contours as a function of HMM vs. SSM scores, these scores were normalized by the average of the respective scores for the correct sentences to better illustrate their relative weight in the combined score.

Figures 3.3 and 3.4 show contour plots for the word error distribution as a function of normalized HMM and SSM scores on the two test sets, keeping the phone and word insertion penalties fixed at typical values. Figure 3.3 represents the case for gender-dependent optimization over male speakers. The weight space for the two test sets appears vastly different. The effects of gender-independent optimization is shown in Figure 3.4. Although the Oct 89 figure has fewer local optima, it must be noted that the best region for one test set still does not match that of the other. Normalizing the acoustic scores shows that the HMM is weighted higher than the SSM, but the weights are of the same order of magnitude. Figure 3.5 suggests that typical values of the word penalty are about 3-5 times that of the phone penalty.

We see that the major problem involved in the estimation of a robust set of weights is the significant mismatch between test sets. This may be because the weights are trained on a set containing only 10 speakers, which may not be representative enough. Gender-independent optimization was more robust than gender-dependent optimization possibly because there were more speakers included. This suggests that we should train on a larger test set. However we will not be doing so due to lack of time and resources.

An interesting question to ask is how much the error surface changes when we make minor changes to the recognition system. Optimizing weights every time we change a parameter in the recognizer is a bottleneck during experiments. If the regions are reasonably similar, we can find good weights once and fix them for many experiments. Figure 3.6 shows contour plots for two different SSM triphone systems. The top figure has no clustering while the lower one uses clustered triphones. We see that the regions of best weight are reasonably similar so we can indeed fix weights while running experiments involving minor modifications.

3.3 Investigation of the Structure of the SSM

The time-transformation component of the SSM maps a variable length phone to a fixed-length model, where the model length refers here to the number of distributions

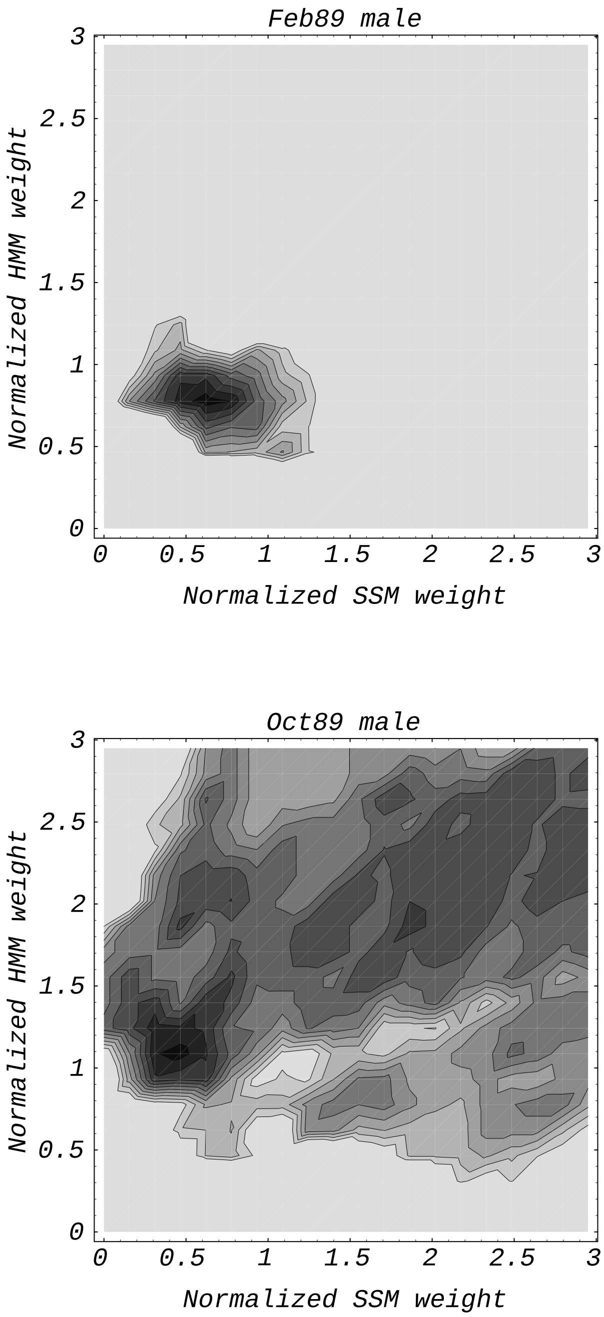

Figure 3.3: Contour plot of word error distribution as a function of normalized SSM weight and normalized HMM weight for optimization over male speakers. Feb 89 range: 2.9 – 3.6%, Oct 89 range: 2.8 – 3.3%. (Phone weight = -0.0005, Word weight = -0.0025)

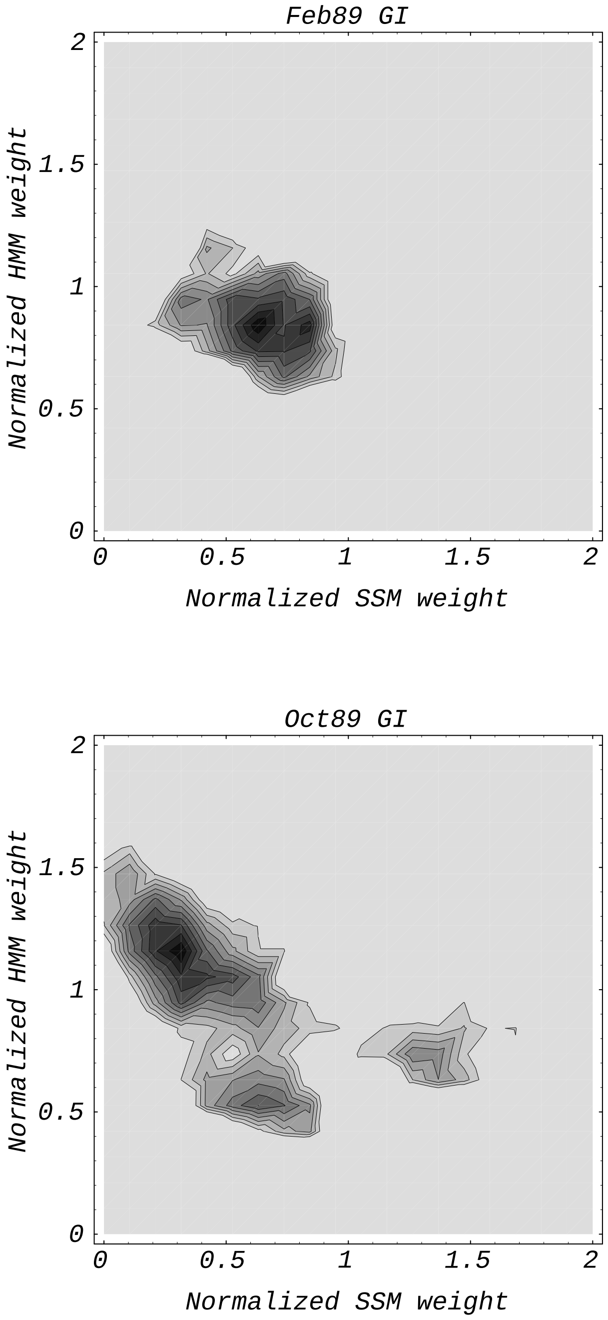

Figure 3.4: Contour plot of word error distribution as a function of normalized SSM weight and normalized HMM weight for gender-independent optimization. Feb 89 range: 2.8 – 3.2%, Oct 89 range: 3.2 – 3.6%. (Phone weight = -0.0005, Word weight = -0.0025)

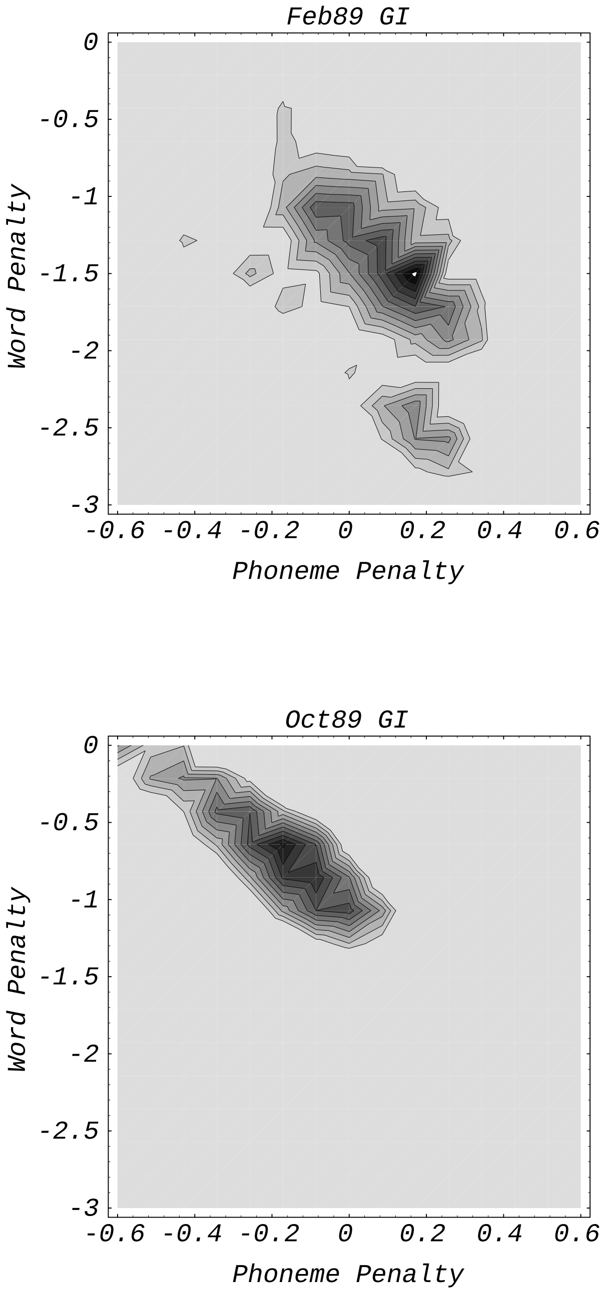

Figure 3.5: Contour plot of word error distribution as function of phone and word insertion penalties using gender-independent optimization (with SSM weight fixed at 0.03). Feb 89 range: 4.5 – 4.9%, Oct 89 range: 4.2 – 4.8%.

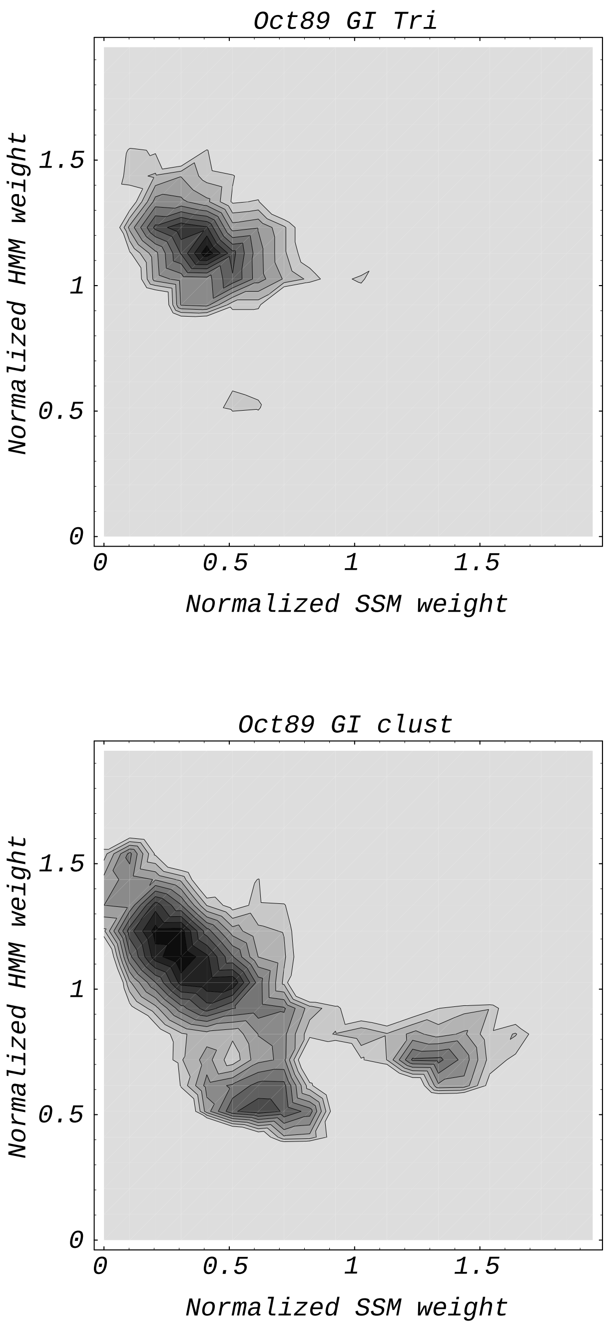

Figure 3.6: Contour plot of word error distribution as function of normalized SSM weight and normalized HMM weight using GI optimization for different SSM systems. The lower figure is for a system using clustered triphones while the top one does not. (Phone weight fixed at -0.0005 and word weight fixed at -0.0025).

in the model. However the theory does not tell us how long the fixed-length segment must be for best performance. This has been investigated earlier in the context of phone classification by Bechwati [7]. His conclusions were :

- If all models have the same fixed length, longer models perform better than shorter ones.
- Models with phone-dependent lengths as the average length of the phone are better than using fixed length models for the case of short and long phones, looking individually at those phones, although he did not get improved performance using variable length models overall.

We examined these suggestions for the SSM in word-recognition. An experiment was performed keeping the SSM length fixed for all models. Recognition was performed with "left+right (L+R)" models. These models have their left half made from left-context models and right half made from right-context models and model context better than either case alone. The feature vector was reduced from 30 to 29 features by using linear discriminants. Two fixed length models, of 5 and 8 distributions were examined, the results of which are in Table 3.1. There is a clear improvement in using 8 distributions over 5 distributions (reduction in word error rates for the SSM: 4.1% for females, 11.4% for males and 8.7% overall).

We also examined the possibility of having model lengths that are phone-dependent where the model length is characteristic of phone duration. For these experiments all the 30 features were used without transformation. The average duration of the phone was not used, as it possible that a few very long occurrences (possibly due to mislabeling) would skew this statistic. Therefore we chose the average of the middle 80% of the occurrences, i.e., we neglected the shortest and longest 10% of the occurrences, to be the length of the model. The model lengths used for the different phones is shown in Table 3.2. The results for this setup with L+R models is given in Table 3.3(a). Comparing this with results in Table 3.1, we see that the performance actually deteriorates (considering the SSM performance only). We thought that this might be due to very long models which may not be well trained since the model lengths vary from 3 distributions to 17 distributions. Therefore an experiment was carried out using the same middle 80% formula except that the maximum length is limited to 10 distributions. The results of this experiment are in Table 3.3(b). No improvement and in fact a slight degradation in performance was observed. The total number of distributions used for the variable length models was close to the number for 8-distribution fixed length model, so performance differences cannot be attributed to parameter dimensionality differences. It may be that longer phones have longer regions of approximate stationarity and therefore do not need longer models, while short phones may have more time variability.

Of all the systems considered, we see that the best model length is 8 distributions for word recognition. Even when triphones were considered the 8 distribution model was better than the 5 distribution model (results not shown here), so 8 distributions will be used in the remainder of the work.

The SSM recognizer uses linear discriminants for feature reduction. This has shown to have a better discriminative power than principal components for speech [9]. Reducing features leads to fewer parameters to estimate, suggesting that the resulting models may be more robust. In context-independent phone-recognition experiments with the SSM [14], it was found that feature reduction did not yield improvement in performance. However, context-dependent models involve estimation of many more parameters, and feature reduction might be more useful in this case. Therefore, experiments were performed to see the effect of linear transformations, including (i) no transformation, i.e., using 30 features, (ii) transforming to 29 features, and (iii) transforming to 26 features. In all these cases the 8-distribution L+R SSM was considered. The results for these experiments along with the case when variable length models are used (with no transformation) are in Table 3.4. We see that the SSM system that used 8 distributions and transformation to 29 features is the best. For this system the combined HMM+SSM system gave 2.8% word error (the HMM alone resulted in 3.4% word error). On examining the acoustic scores produced by the different systems, we observed that the (8, 29) system scores highest and this is used in all further experiments unless otherwise indicated. The 30 dimensional feature vector includes first difference energy and first differences of the Mel-warped cepstral coefficients (including C_0). As C_0 represents the energy in the signal it is quite likely that the first difference of Mel-warped C_0 has much the same information as that in the first difference energy and therefore by reducing the feature vector to 29 dimensions, we eliminate this redundancy. On the other hand, reducing the feature vector to 26 dimensions results in some loss of information which is reflected in the degradation of performance.

Gender (SSM length)	Percent word error		
	HMM	SSM	HMM+SSM
Female (5)	2.97	4.69	2.97
Female (8)	-	4.50	2.87
Male (5)	3.63	5.80	2.97
Male (8)	-	5.14	2.77
M+F (5)	3.36	5.35	2.97
M+F (8)	3.36	4.88	2.81

Table 3.1: Performance of two fixed length (5 and 8 distribution) left+right models on the Feb 89 test set.

Phone	iy	ih	ey	eh	ae	aa	ah	ao	uh	uw	ax	ix	er	ow
Male	10	6	11	7	11	12	8	11	5	7	3	3	12	13
Female	11	6	13	8	11	12	8	11	6	7	4	4	12	13
Phone	y	w	r	l	m	n	nx	p	t	k	b	d	g	hh
Male	9	9	9	9	7	6	9	9	6	9	6	4	7	5
Female	10	10	9	9	7	6	9	9	6	10	6	5	8	7
Phone	sh	v	dh	z	zh	ch	jh	el	en	dx	vl	vr	axr	urp
Male	13	5	3	9	8	14	9	10	11	3	10	6	9	5
Female	14	5	4	11	8	15	10	10	12	3	9	6	10	5
Phone	oy	tq	s	-	urk	urd	ay	th	aw	f	urt			
Male	16	6	12	11	5	2	16	10	14	11	4			
Female	3	3	13	11	3	3	17	10	14	10	4			

Table 3.2: Average length (in frames) of middle 80% phones

System	(a) Variable length models			(b) Var. len. (max = 10)		
	Female	Male	M+F	Female	Male	M+F
SSM	4.79	5.80	5.39	4.89	6.00	5.54
SSM+HMM	2.87	2.70	2.77	2.78	2.70	2.73

Table 3.3: Percentage word error for SSMs using (a) variable length and (b) variable length up to a maximum of 10 distributions. All cases used SSM left+right context-dependent models on the Feb 89 test set.

Gender	# of transformed features			
	26	29	30	(var, 30)
Female	5.29	4.50	4.50	4.79
Male	5.87	5.14	5.27	5.80
M+F	5.62	4.88	4.96	5.39

Table 3.4: SSM percent word error for different transformations using Left+Right models on the Feb 89 test set. Excluding the variable length case (var), all models were 8 distributions long.

Chapter 4

Context Modeling and Clustering

Modeling of context (allophonic variations of phones) has resulted in some of the biggest gains in continuous speech recognition. A popular unit for modeling context is the triphone. Triphones capture acoustic-phonetic detail by using different models for phones, conditioned on the identities of the left and the right neighboring phones. However there are N^3 possible models if there are N phones. This leads to severe training problems for the models (due to the paucity of training data) as well as a significant memory requirement. These two problems are both addressed by clustering, or grouping "similar" triphone models, which reduces the number of free parameters to estimate.

One approach would be to merge models for acoustically similar contexts using human knowledge. Alternatively, automatic clustering methods have been used successfully by several researchers. Initially, Lee proposed a framework for the agglomerative clustering of triphone models for HMMs with discrete distributions [27]. Others have proposed divisive clustering frameworks for discrete distributions [6, 28]. In this thesis, we investigate both agglomerative and divisive strategies for clustering multivariate Gaussian distributions, which we have applied to context modeling with the SSM.

In this work, clustering is performed on all triphones that occur for a given phone (having more than 5 observations), with each phone being clustered separately. The distributions (frames) are clustered independently as it is quite plausible that the distributions for the initial frames cluster differently than the ones for the ending frames. The distribution of the frames near the center can be expected to be close to the context-independent case, resulting in fewer clusters. The SSM recognizer was used to evaluate the effects of clustering on recognition performance. In this work, clustering has been performed only to obtain better covariance estimates for the distributions. The distribution means used in recognition are the same as have been estimated from previous work without clustering, since means can be reliably estimated with less data than covariances. The clusters define classes for covariance tying. If a triphone model belongs to a cluster with a "valid" clustered covariance associated with it, it is used in recognition. Otherwise the left half and the right half of the segment model are constrained to have similar regions of tying along the lines of [22]. The covariances of the distributions of the left half of the model are the same as the left half of the left-context model and similarly for the right half. A covariance is declared to be "valid" if the number of observations that went into estimating it is above an empirically determined threshold.

4.1 Agglomerative Clustering

A two-step clustering algorithm is used involving agglomerative clustering followed by K-means clustering. In agglomerative clustering, the nearest clusters are merged repeatedly till some stopping criterion is satisfied. A cluster $\mathcal{X}$ has as its constituents the means $\mu_1, \mu_2, \ldots, \mu_k$ and covariances $\Sigma_1, \Sigma_2, \ldots, \Sigma_k$, (of the respective distributions) with the corresponding number of occurrences during training being, $n_1, n_2, \ldots, n_k$. In computing distances between clusters, this cluster is characterized by its centroid, m, and the total number of observations in all distributions in this cluster, o. The algorithm is initialized by assuming that every distribution forms a singleton cluster located at the mean of that distribution. The key issues involved in clustering are the choice of distance measure and the stopping criterion.

Distance measure

A weighted Mahalanobis distance is used with the context-independent covariance, since in general this covariance has been well estimated. The standard Mahalanobis distance considers each cluster as a point (located at the mean of the cluster), and does not account for the number of constituents in the cluster. Experimentally this distance measure was seen to have an effect of merging large clusters. Therefore the distance measure used is

$$d(\mathcal{X}_i, \mathcal{X}_j) = \sqrt{\frac{o_i o_j}{o_i + o_j}} d_M(m_i, m_j) \tag{4.1}$$

where $\mathcal{X}_i$ represents the cluster i with centroid m_i, o_i is the total number of observations in all the distributions in this cluster and $d_M(m_i, m_j)$ represents the standard Mahalanobis distance between the two centroids using the CI covariance, Σ, i.e., $d_M(m_i, m_j) = \{(m_i - m_j)^t \Sigma^{-1} (m_i - m_j)\}^{1/2}$. This weighted distance measure favors growth by adding small clusters to large ones over merging medium-sized clusters. If we define J_e to be the total sum-of-squared errors of each mean from the centroid of the respective clusters, this weighting factor decreases J_e by the smallest possible amount at each stage of agglomeration and is thus stepwise-optimal [15].

If $\mathcal{X}_i$ and $\mathcal{X}_j$ are merged to form $\mathcal{X}_k$, the parameters of the new cluster, o_k and

m_k are computed as:

$$o_k = o_i + o_j \qquad\qquad m_k = \frac{o_i m_i + o_j m_j}{o_i + o_j} \tag{4.2}$$

and $\mathcal{X}_i$ and $\mathcal{X}_j$ would not be used anymore.

Stopping criteria

Since our primary reason for clustering is to improve the covariance estimates by pooling, we would ideally like to stop when the number of observations in all clusters are more than an empirical threshold, T_c, when we can assume that the covariances can be estimated reasonably well. However due to presence of "outlier" clusters it has been experimentally seen that satisfying this condition leads to over-clustering.

Therefore two other conditions when clustering may be stopped are:

- The ratio of minimum distance to maximum distance between clusters exceeds a threshold, T_d, and at least one cluster has $o_k > T_c$. This would be akin to a measure of similarity.

- The ratio of the number of clusters to the number of merges is less than a threshold, T_m, and at least one cluster has $o_k > T_c$. This results in specifying the maximum number of merges, or equivalently the minimum number of clusters.

The thresholds were determined empirically by running experiments. Once agglomerative clustering is done, we define all clusters whose observations fall below T_c as "outlier" clusters. In this work, we have also investigated the effect of just using agglomerative clustering, i.e., not doing the second step of K-means clustering. If that is the case, the covariances of the constituents of a cluster are tied to the cluster covariance. The outlier clusters may be assigned the covariance of the nearest non-outlier cluster (but not used in estimating the tied covariance) or else may have a less detailed covariance such as left+right or left/right-context. Both these schemes were evaluated.

The cluster covariance is determined by tying the covariances of the constituent distributions to get an unbiased estimate:

$$\Sigma_j = \frac{\sum_i (n_i \Sigma_{j,i})}{\sum_i n_i - \sum_i 1}. \tag{4.3}$$

4.1.1 K-means Clustering

K-means clustering is a general clustering algorithm for classification of points in space to K classes [15]. For clustering triphone models, we start with initial value of the centroids $\hat{m}_1, \ldots, \hat{m}_K$. The distributions are classified by assigning them to the class which minimizes the distance between the cluster centroid and distribution mean. The centroids are recomputed as the average of the means of the distributions in their class (weighted by the number of observations that went into estimating the means). This procedure is iterated till convergence. K-means clustering is performed after agglomerative clustering because some of the merges that take place during agglomeration may not be optimal. K-means clustering at least guarantees a local minimum for the total cluster distortion.

Clustering is performed with K equal to the number of non-outlier clusters and initializing with the centroids of these clusters. The distance measure used here is the standard Mahalanobis distance with the CI covariance. Each distribution is assumed as a point located at its mean and standard K-means clustering is performed till convergence. Convergence is declared when the percentage change in the within-cluster distortion is less than some threshold, T_k. For this clustering all distributions are considered (even those that lie in the outlier clusters). It was experimentally observed that convergence is very quick, typically in less than 3 iterations. Once convergence is reached, the covariance of each cluster is computed as the unbiased tied covariance estimate and all constituents of the cluster are assigned this covariance.

Pruned K-means

The K-means algorithm may be modified to prune atypical distributions at each iteration of the computation (where an "atypical" distribution is defined as having a mean that is far from its cluster centroid). This would be beneficial if after pruning outliers, the remaining distributions make the cluster more "homogeneous". A set of experiments were performed to test this hypothesis. Pruning of distributions was done as follows. For each cluster the average distance of the distributions from the centroid of the cluster is computed and all distributions that are more than a certain multiple (called the pruning threshold) of this distance away are declared as outliers and pruned. During recognition, the corresponding models may be assigned the covariance of the nearest non-outlier cluster or get the left+right covariance.

4.1.2 Experiments

The effects of the different schemes of clustering on the number of triphone covariances and the average number of observations in the clustered covariances is indicated in Table 4.1. Clustering results in a significant reduction in the size of model files; typically the size is halved, enabling the recognizer to have reduced virtual memory requirements. We investigated agglomerative clustering, with and without K-means clustering following it with two values for T_c, 150 and 250 observations. Results are reported in terms of average word error of the top ranking hypothesis after rescoring and reranking using gender-dependent optimization. In the experiments that were performed, the thresholds T_d and T_m were fixed at 0.4 and 0.2 respectively. These were seen to yield better results in preliminary experiments.

Two options were explored for outlier models while clustering only agglomeratively: assigning covariances for outliers as the covariance of the nearest non-outlier cluster and alternatively backing off to the left+right covariance. The results are shown in Table 4.2. This table also shows performance for the baseline system, which is the normal triphone SSM system as well as the triphone SSM system with linguistically motivated class-dependent tying of covariances for triphones [22] for comparison. It is seen that use of agglomerative followed by K-means clustering results in decreasing the error rate by 8.5 % over normal triphones and 7.7 % over class-dependent tying (an alternative to automatic clustering). The results also indicate that it is better to assign outlier clusters the covariance of the nearest non-outlier cluster than to back-off to the left+right covariance. There is a substantial improvement when K-means clustering is used as a second step. This is probably due to a more optimal partitioning of the data.

We also investigated the effect of varying the pruning threshold in K-means clustering on performance for the case of assigning outliers the covariance of the nearest cluster as well as assigning them left+right covariances in recognition. The results for this set of experiments are shown in Table 4.3. The word errors for all the experiments here have been computed fixing the weights to the values that resulted in the best performance without pruning (from Table 4.2) as we believe that the error contours would not change significantly. Decreasing the threshold for pruning for K, increases the number of model distributions pruned. This results in fewer models (and thus observations) to tie the covariance over. The outlier models pruned seem to have few observations associated with them as there is a lack of a significant drop in the average number of observations per covariance cluster when the K-pruning threshold is decreased. The performance is not sensitive to the value of T_c. The conclusion from this set of experiments is that pruning during K-means actually hurts performance,

gender	T_c	clustering	# of tri. covs	avg. # of obs. in cluster
female	N.A	none (baseline)	16324	N.A.
female	150	agglomerative	710	331
female	150	aggl. + K-means	710	443
female	250	agglomerative	521	397
female	250	aggl. + K-means	521	603
male	N.A	none (baseline)	27506	N.A.
male	150	agglomerative	1584	430
male	150	aggl. + K-means	1584	512
male	250	agglomerative	1073	545
male	250	aggl. + K-means	1073	756

Table 4.1: Characteristics of the clusters resulting from agglomerative clustering and subsequent K-means clustering.

T_c	cov. for outlier	clustering	word error (%)
N.A	N.A.	none (baseline)	4.61
N.A	N.A.	linguistic tying	4.57
150	nearest cluster	agglomerative	4.33
150	left+right	agglomerative	4.65
150	no outliers	aggl. + K-means	4.26
250	nearest cluster	agglomerative	4.69
250	left+right	agglomerative	4.72
250	no outliers	aggl. + K-means	4.22

Table 4.2: Results of agglomerative clustering on the Feb 89 test set using GD weight optimization.

instead of helping it.

In all the experiments listed in Table 4.2, the weights for linearly combining the scores were found by optimizing performance over the same test set. To evaluate performance on a new test set, the best clustering case was run on the October 89 test set. There is a slight improvement in performance on this test set, from 5.0% word error for the baseline triphone system to 4.8% word error with clustering.

4.2 Divisive Clustering

In the previous section, we developed agglomerative techniques for clustering. Alternatively, we can develop divisive clustering techniques in the spirit of the work done for HMMs in [6] and [28]. As was the case in the previous section, we shall cluster triphone models in a phone and distribution-dependent manner. A set of

T_c	Pruning threshold	# models pruned	word error (%)			
			clust. cov. for out.		L+R cov. for out.	
			GI	GD	GI	GD
150	4.0	153	4.53	4.23	4.49	4.29
	3.0	755	4.64	4.53	4.61	4.45
	2.5	1998	4.64	4.53	4.41	4.33
	2.0	6176	4.74	4.57	4.80	4.49
250	4.0	141	4.49	4.30	4.49	4.30
	3.0	755	4.45	4.45	4.57	4.53
	2.5	2023	4.76	4.65	4.76	4.69
	2.0	6566	4.84	4.65	5.00	4.61

Table 4.3: Results of pruned K-means clustering after agglomerative clustering on the Feb 89 test set. GI and GD represent gender-independent and gender-dependent weight optimization respectively.

binary linguistically-motivated questions related to contextual effects is asked at each node and the node yielding greatest reduction in distortion is split. Divisive clustering has some advantages over the agglomerative techniques we considered in the previous section. It allows us to explicitly incorporate acoustic-phonetic knowledge in a statistical framework. Also, when using agglomerative clustering, there are quite a few variables that required adjusting. In contrast, with divisive clustering the only variable that needs to be empirically determined is the covariance threshold, since it is reasonable to continue growing the tree as long as covariance of the nodes can be well estimated.

Clustering is performed by starting with a tree with 61 nodes, one for each phone. Let $\mathcal{N}$ be the set of leaf nodes in a tree and $N \in \mathcal{N}$. N has as its constituents, the means $\mu_1, \mu_2, \ldots, \mu_k$ and covariances $\Sigma_1, \Sigma_2, \ldots, \Sigma_k$, (of the respective distributions) with the corresponding number of occurrences during training being, $n_1, n_2, \ldots, n_k$, k being the number of distributions in the node. Let $\mathcal{Q}$ be the set of allowable questions and the reduction in distortion due to the split at node N resulting from question q be $D(q, N)$, where $q \in \mathcal{Q}$. Defining $d(N)$ as the distortion of node N, if node N is split into a left node, N_l and a right node, N_r:

$$D(q, N) \equiv d(N) - d(N_l) - d(N_r). \tag{4.4}$$

We have investigated the use of three different distortion metrics, $d(N)$, for tree-based clustering. Letting T_c be the minimum number of observations in a node for the node to have a well estimated covariance, we constrain the allowable splits of node N such that the number of observations in both N_l and N_r are greater than T_c. If such a split is not possible, node N is declared to be terminal. Termination of clustering occurs when all leaf nodes are declared to be terminal. In this work we shall limit the set of questions to be simple only, i.e., logical combination of questions cannot be asked at a node. If compound questions were used, the framework here would still be applicable but tree design would require more computation.

Questions related to the place and manner of articulation are asked at the nodes. They were designed with the help of an expert in acoustic-phonetics to differentiate contextual effects [16]. The list of classes used as questions are shown in Table 4.4. A glossary for these classes, essentially taken from [26], is in Appendix B. Allowable questions are of the form: "Does the left context phone belong to class A ?". When question q is asked at node N, all triphone models that answer "yes" go to N_l while the rest go to N_r and $D(q, N)$ is computed. Questions related to membership of all classes are allowed for both the left and the right contexts. For each node, N, we note the question resulting in largest reduction in distortion, $D(q, N)$. The node that results in largest reduction in distortion over the whole tree is split using the corresponding question.

4.2.1 Metric 1: Mahalanobis distance

We define node distortion to be

$$d(N) = \sum_i n_i d_M(\mu_i, m) \tag{4.5}$$

where $m = \frac{1}{\Sigma n_i} \sum n_i \mu_i$ and $d_M(\mu_i, m)$ is the Mahalanobis distance between the distribution mean and the centroid of the node. The covariance used to compute this distance is the context-independent covariance of that particular phone, again because it is well estimated. As this is the same metric as used in agglomerative clustering, this provides us with an opportunity to make a fair comparison between divisive and agglomerative clustering. It should be noted that this metric does not take into consideration triphone covariances during clustering, as was also the case in agglomerative clustering. At the end of clustering, the node covariances are determined by tying over the constituent triphone covariances.

Class	Phones
sonorant	iy ih ey eh ae aa ah ao uh uw ax ix er ow aw ay oy yu y w r l m n nx el em en vl vr axr uy
continuant	iy ih ey eh ae aa ah ao uh uw ax ix er ow aw ay oy yu y w r l hh f th s sh v dh z zh wh el em en vl vr axr uy uax
stop	p t k b d g dx urp urt urk urb urd urg tq q
fricative	f th s sh v dh z zh
nasal	m n nx
lateral	l vl el
vowel	iy ih ey eh ae aa ah ao uh uw ax ix er ow aw ay oy yu el em en axr uy uax
semivowel	y w r l vr vl wh
reduced vowel	ax ix el em en axr uax
front vowel	iy ih ey eh ae ix y
back vowel	aa ao uh uw ax ow aw w el vl
high vowel	iy ih uh uw y w
low vowel	ae aa ao ow
round	ow ao aw oy yu uy uw w r
silence	–
labial	w m p b f v wh urp urb
retroflex	r axr er vr
velar	nx k g
palatal	sh zh
affricate	ch jh
flap	dx
strident	s sh z zh ch jh
diphthong	ay oy aw yu uy
coronal	n t d th s dh z
lax	ih eh uh ax ix uax
syllabic consonant	el em en axr
aspiration	hh

Table 4.4: Classes used as questions for tree-based clustering. Labels are based on the Resource Management phone set.

To see the effectiveness of using linguistically-based questions for divisive clustering, we also performed an experiment using binary divisive clustering without the use of any questions. Instead, the partitions (to assign models to N_l and N_r) are determined using K-means clustering with K=2 with no constraints on how the node may be split. If m is the centroid of node N, the K-means is then initialized with the two centroids as $m_1 = m$ and $m_2 = m(1+\epsilon)$. The distributions are classified to one of two classes based on the Mahalanobis distance of the distribution mean from the two node centroids (with the context-independent covariance) and the two centroids are recomputed after classification. This procedure of classification and computation of the two centroids is repeated till convergence.

4.2.2 Metric 2: Entropy-based distance

We can alternatively take an information-theoretic approach, as in [28], to determining the distortion function. The d-variate Gaussian $p(x) \sim N(\mu, \Sigma)$ has entropy defined as

$$H(X) = \frac{d}{2}\left[\ln(2\pi) + 1\right] + \frac{1}{2}\ln|\Sigma|. \tag{4.6}$$

(See Appendix A for the derivation.) Defining $H(N)$ as the entropy of the distributions in node N with Σ defined as the (sample, not tied) covariance of that node, we can define the distortion of the node as

$$d(N) = H(N)\sum_{i=1}^{k} n_i \tag{4.7}$$

and use Equation 4.4 to compute the change in distortion. We weight the entropy by the number of observations used in estimating the Gaussian so that the change in entropy given by a split reflects the change for the entire tree. At the end of clustering the node covariances may be determined by either tying over the constituent triphone covariances or by computing the node sample covariance, and both methods are examined in our experiments.

4.2.3 Metric 3: Generalized likelihood-based distance

The generalized likelihood-based distance is a third splitting criterion for divisive clustering. As we wish to cluster Gaussian distributions which are similar, given a certain partition of distributions (in two parts using questions at the node, as before) we can use a hypothesis test along the lines of [18] to choose between one of two hypotheses:

- H_0: the observations were generated from two different distributions (that represent the distributions of the child nodes), and
- H_1: the observations were generated from one distribution (that represents the distribution of the parent node).

We test this hypothesis by using the generalized likelihood ratio test. Let O, O_l and O_r denote the observations in the parent node (N) and the two child nodes (N_l, N_r) with n, n_l and n_r observations respectively and $n = n_l + n_r$. If $L(O_l; \mu_l, \Sigma_l)$ denotes the likelihood of the training observations that went to the left node and $L(O_r; \mu_r, \Sigma_r)$ denotes the likelihood of the training observations that went to the right node then the likelihood of the two partitions being generated from two different distributions is given by

$$L(O_l; \mu_l, \Sigma_l) L(O_r; \mu_r, \Sigma_r). \tag{4.8}$$

The likelihood of the observations being generated from the same distribution is $L(O; \mu, \Sigma)$ where O is the union of O_l and O_r.

If we let λ denote the likelihood ratio, then

$$\lambda = \frac{L(O; \hat{\mu}, \hat{\Sigma})}{L(O_l; \hat{\mu}_l, \hat{\Sigma}_l) L(O_r; \hat{\mu}_r, \hat{\Sigma}_r)} \tag{4.9}$$

where the hat denotes maximum likelihood estimates.

For multivariate Gaussians [18], the likelihood ratio can be written as

$$\lambda = \lambda_{COV}\lambda_{MEAN}, \tag{4.10}$$

where λ_{COV} is the likelihood ratio that tests the hypothesis that the observations in both partitions were generated from the same Gaussian models with the same covariance matrix, with no assumption being made about the equality of the means, and λ_{MEAN} is the likelihood ratio that tests the hypothesis that the observations in both partitions were generated from the same Gaussian models with the same mean, with no assumption being made about the equality of the covariances. Further, λ_{COV} and λ_{MEAN} can be written in terms of the sufficient statistics of the Gaussians:

$$\lambda_{COV} = \left(\frac{|\hat{\Sigma}_l|^{\alpha} |\hat{\Sigma}_r|^{(1-\alpha)}}{|W|} \right)^{\frac{n}{2}} \tag{4.11}$$

where $\alpha = \frac{n_l}{n}$, and W is the frequency weighted tied covariance, viz., $W = \frac{n_l}{n}\Sigma_l + \frac{n_r}{n}\Sigma_r$, and

$$\lambda_{MEAN} = \left(1 + \frac{n_l n_r}{n^2} (\hat{\mu}_l - \hat{\mu}_r)^t W^{-1} (\hat{\mu}_l - \hat{\mu}_r) \right)^{\frac{-n}{2}}. \tag{4.12}$$

We define the reduction in distortion due to the partition formed by question q as

$$D(q, N) = -(\log \lambda_{COV} + \log \lambda_{MEAN}). \tag{4.13}$$

This is always a positive quantity as the generalized likelihood ratio is always greater than zero and less than unity. Under the null hypothesis, i.e., when the observations are from the same distribution, $2 * D(q, N)$, has a chi-square distribution when n is large [1]. For further discussion on the distribution of this distance, the reader is referred to [18].

Using the above definition of $D(q, N)$ the tree growing procedure is the same as has been previously described. At the end of clustering the node covariances may be determined by either tying over the constituent triphone covariances or by computing the node sample covariance. Again, we performed experiments with both the methods.

4.2.4 Experiments

Experiments were conducted to evaluate the performance due to the different metrics. Results are reported in terms of average word error of the top ranking hypothesis after rescoring and reranking using gender-dependent (GD) and gender-independent (GI) optimization on the Feb89 test set. Unless otherwise mentioned, the node covariance was determined by tying over the covariances of the constituent models. T_R is the recognizer's covariance threshold. If any estimated covariance has less than T_R observations, then the model is assigned the left+right covariance during recognition.

We first performed an experiment to compare the performance of trees grown using the linguistically motivated questions with that of trees grown using K-means splitting, both using the Mahalanobis distance as the metric. T_c and T_R were both set to 250 for this experiment. The results are shown in Table 4.5. It is seen that the tree grown with K-means has fewer terminal nodes (there is one cluster covariance for each terminal node). This happens because the K-means step may choose a partition such that one of the child nodes has less than T_c observations, leading to the parent node being declared terminal, because there is no simple method for considering alternative splits. On the other hand, while using questions, there might be some question which could still split the parent node into two valid child nodes. We also noticed that the total tree distortion is less for the tree grown using questions. This is partly due to the fact that there are more nodes in the tree which used questions, but may also be because the greedy growing algorithm is more "greedy" with the K-means approach.

Criterion to grow tree	# clust. covs.	word error (%)	
		GI	GD
Questions	3037	4.49	4.30
K-means	2458	4.81	4.69

Table 4.5: Comparison of trees grown using linguistically motivated questions with that grown using K-means, both using the Mahalanobis distance and evaluated on the Feb 89 test set.

The use of questions restricts the manner in which partitions of the triphones may be made, which appears to be better in the long run. In summary, this experiment justifies the use of linguistically motivated questions to guide divisive clustering.

Next we performed a set of experiments to compare the different metrics. The results are shown in Table 4.6. We observe from the results of Mahalanobis and entropy-based clustering that when T_c and T_R are set at 250 we achieve the best performance, which suggests that covariances with less than 250 training observations are not well estimated and performance deteriorates as more of these poorly estimated covariances are used. Therefore, the generalized-likelihood-based clustering was evaluated at only this setting. The results of $T_c = 250$ indicate that clustering using entropy and generalized likelihood metrics achieve better performance than clustering based on Mahalanobis distance. We attribute this to the fact that the Mahalanobis distance does not take into consideration triphone covariances while splitting nodes which both the other metrics take into account.

As both the entropy and generalized likelihood distance metrics use the sample covariance of the nodes for clustering, we performed an experiment to see the effect of assigning terminal nodes the sample covariance of models as the clustered covariance

Distance metric	Clustering T_c	# clust. covs.	word error (%)			
			$T_R = 250$		$T_R = 150$	
			GI	GD	GI	GD
Mahalanobis	250	3037	4.49	4.30	–	–
	150	4922	4.65	4.41	4.80	4.76
Entropy	250	3036	4.41	4.22	–	–
	150	4885	4.69	4.57	5.12	4.80
Generalized Likelihood	250	3035	4.37	4.18	–	–

Table 4.6: Comparison of clustering using different metrics evaluated on the Feb 89 test set.

instead of tying over the constituent models. T_c and T_R were both fixed at 250. The results for this experiment are shown in Table 4.7. Comparison of these results with the corresponding entries in Table 4.6 indicates that performance has deteriorated. The log acoustic scores were also observed to be significantly lower in this case, because the sample covariance is much broader than the tied covariance.

To see the relationship between total tree distortion and the number of terminal nodes, we analyzed the tree grown with the Mahalanobis distance asking questions. Figure 4.1 shows total tree distortion (over all phones) as the number of terminal nodes is increased for each distribution (frame) of the model. The fifth distribution has the maximum number of terminal nodes while the fourth distribution has the least. This is due to the linear warping with an eight distribution long model: observations of odd length warp to the fifth distribution at the expense of the fourth distribution due to the rounding-up involved. This difference in the number of observations does not

Distance metric	# clust. covs.	word error (%)	
		GI	GD
Entropy	3036	4.65	4.51
Generalized likelihood	3035	4.49	4.33

Table 4.7: Results of assigning the terminal nodes sample instead of tied covariances. Evaluated on the Feb 89 test set. T_c and T_R both equal to 250.

cause a problem as long as the same warping is used in training and recognition.

From the experiments, the best results are achieved by clustering using questions and the generalized likelihood metric. We analyzed the questions used to grow this tree on triphone models from the male speakers in the training corpus as they have more data than female speakers. Denoting $\mathcal{Q}_l$ as the set of questions asked about the left context of the triphone and $q_l \in \mathcal{Q}_l$, we define the relative importance of a question asked on the left context, $RI(q_l)$, as

$$RI(q_l) = \frac{\sum_{N \in \mathcal{N}} D(q_l, N)}{\sum_{N \in \mathcal{N}, q_l \in \mathcal{Q}_l} D(q_l, N)}. \tag{4.14}$$

$RI(q_r)$ can be defined in a similar manner for questions asked on the right context. To get the overall relative importance of a question, q, we sum over the reduction in distortions produced when asked over the left (q_l) and right (q_r) contexts:

$$RI(q) = \frac{\sum_{N \in \mathcal{N}} D(q_l, N) + \sum_{N \in \mathcal{N}} D(q_r, N)}{\sum_{N \in \mathcal{N}, q_l \in \mathcal{Q}_l} D(q_l, N) + \sum_{N \in \mathcal{N}, q_r \in \mathcal{Q}_r} D(q_r, N)}. \tag{4.15}$$

Table 4.8 shows the relative importance (in percent) of questions evaluated while growing this tree (including questions that did not result in the best split at the node) and Table 4.9 shows the relative importance of questions considering their contribution only when they result in splitting the node. Relative frequencies of occurrence of the phone classes in the database are also shown in Table 4.8. It should be noted that this column does not sum to 100% – a consequence of the class definitions not forming a partition. It is seen that the relative importance of questions is related to the relative frequencies when contributions are counted even when they do not result in a split. Relative importance associated with questions not chosen is included in Table 4.8 but not Table 4.9. We observe that the order of relative importance changes substantially between these two tables. This is because a question, e.g. "is a phone a continuant ?", that produces a large reduction in distortion, may not be the top ranking question and is therefore not used to split the node. In other words, Table 4.9 to some extent accounts for the fact that features contain redundant information. Tables 4.10 and 4.11 show the relative importance of questions (for the top eight classes) when the center phone is considered from a particular class. The relative importance of questions gives us an indication of contextual factors that have the greatest effect on the acoustic realization of a phone, e.g. labials and stops affect a vowel more when they occur before the vowel than after it, while nasals and retroflexed sounds affect a vowel more when they occur after the vowel than before it.

Figures 4.2 to 4.4 illustrate the order of questions asked to cluster triphones when the center phone is "iy" for three distributions (first, middle and last). The trees are unbalanced with more nodes on the right than on the left. This is because there are always fewer phones in a class than there are outside, for every phone class. As triphone models that answer "no" to the question are sent to the right, we have the observed effect. Looking at the order of questions we find that for the first distribution, questions are asked on the left context more often and higher in the tree than for right context. The opposite effect is seen with the right context on the last distribution. For the middle distribution it is seen that questions on left and right context are asked approximately equally. It can be expected that the effect of the right context should be most prominent on the last distribution while the left context's effect is most prominent on the first distribution. This is another indication that the tree growth is reasonable.

To evaluate performance on a new test set, the best clustering case was run on the October 89 test set. This resulted in 4.9% word error which was a slight improvement over the baseline system of 5.0% word error. In comparing this best case with the best case of agglomerative + K-means clustering (4.8%), we observe that agglomerative and divisive clustering yield similar performance. This observation was also noted by Lee *et al.* in [28] for vocabulary-dependent training, which is the case here.

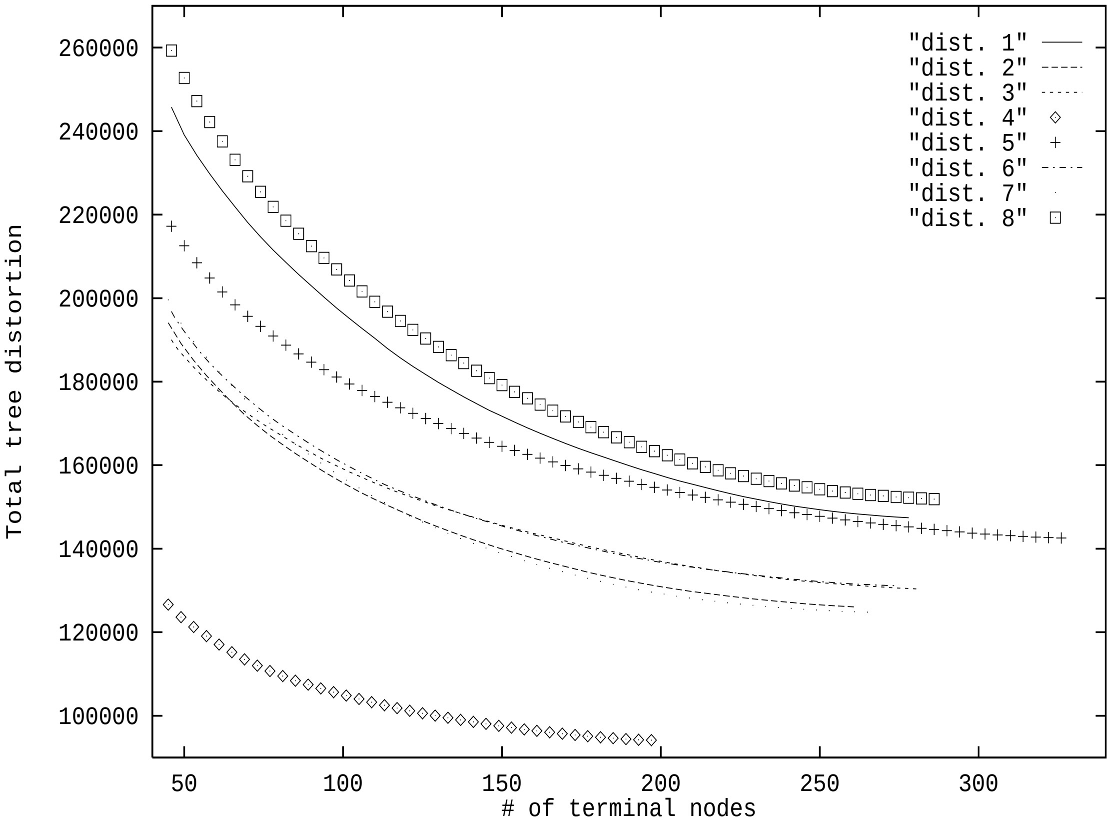

Figure 4.1: Total tree distortion as a function of number of terminal nodes for tree clustering of different distributions with the Mahalanobis distance criterion.

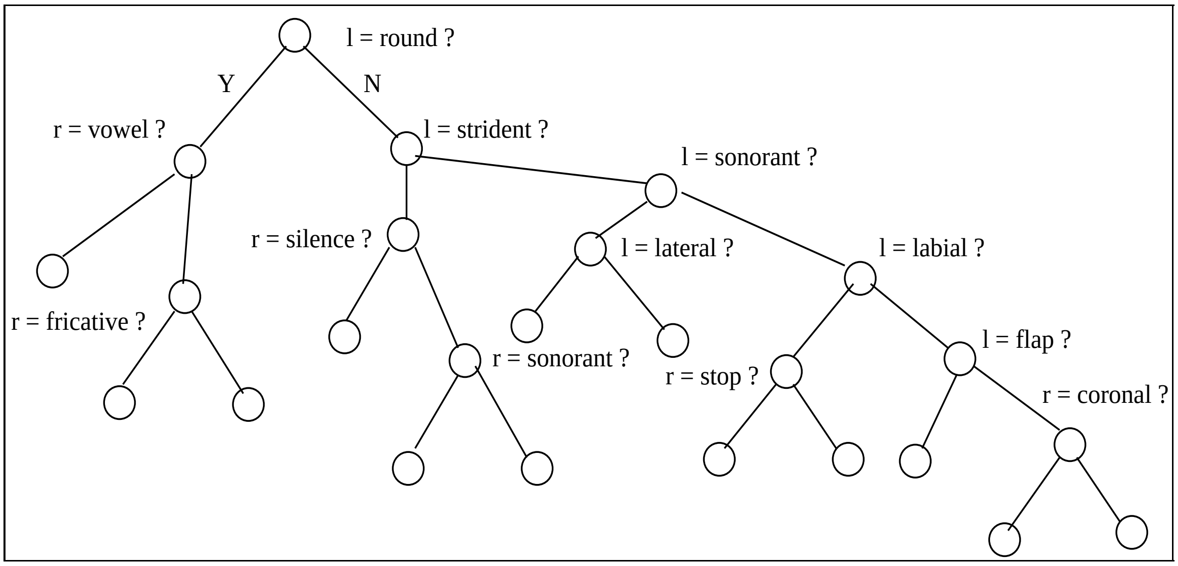

Figure 4.2: Questions asked for distribution 1 of phone "iy".

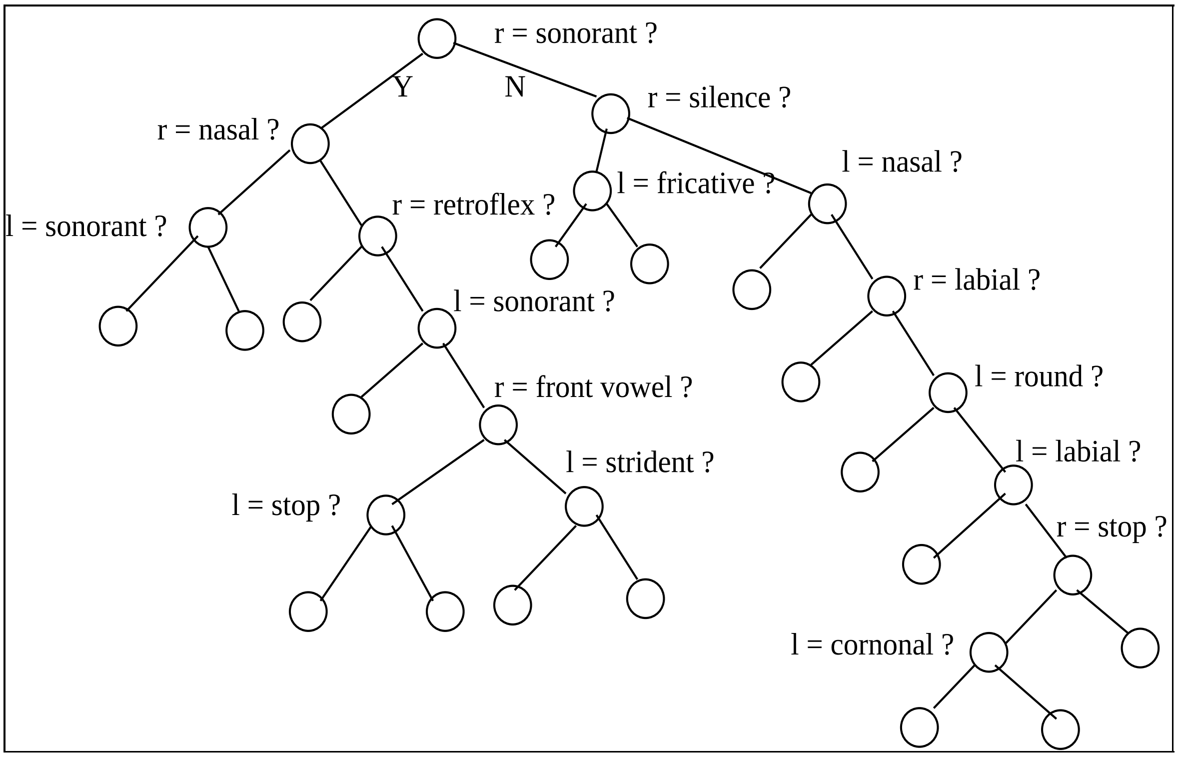

Figure 4.3: Questions asked for distribution 5 of phone "iy".

left context		right context		overall		
Class	rel. imp.	Class	rel. imp.	Class	rel. imp.	r.f.
sonorant	9.96	continuant	9.31	continuant	9.39	65.5
continuant	9.47	coronal	8.66	sonorant	9.30	57.2
coronal	8.71	sonorant	8.63	coronal	8.69	26.5
stop	7.08	fricative	7.09	stop	6.90	18.1
fricative	6.17	stop	6.73	fricative	6.63	16.7
vowel	5.56	front vowel	5.75	vowel	5.49	36.7
labial	5.23	vowel	5.41	front vowel	5.48	20.1
front vowel	5.21	labial	5.29	labial	5.26	12.3
semivowel	4.69	strident	5.28	strident	4.56	11.1
back vowel	4.33	lax	4.36	semivowel	4.36	11.3
lax	4.32	back vowel	4.04	lax	4.34	15.5
strident	3.83	semivowel	4.02	back vowel	4.18	15.8
round	3.81	high vowel	3.94	round	3.71	10.2
retroflex	3.45	round	3.62	high vowel	3.69	12.6
nasal	3.45	nasal	3.52	nasal	3.49	9.2
high vowel	3.45	reduced vowel	3.14	reduced vowel	3.04	11.1
reduced vowel	2.95	low vowel	2.93	retroflex	3.03	7.6
velar	1.80	retroflex	2.62	low vowel	2.32	7.5
low vowel	1.71	silence	2.47	silence	1.84	6.2
lateral	1.29	velar	1.13	velar	1.46	4.3
silence	1.20	lateral	0.63	lateral	0.96	3.8
palatal	0.78	palatal	0.45	palatal	0.62	1.4
aspiration	0.54	flap	0.40	flap	0.38	1.2
syllab. cons.	0.36	syllab. cons.	0.29	syllab. cons.	0.33	2.4
flap	0.35	diphthong	0.27	aspiration	0.27	0.9
affricate	0.25	affricate	0.00	diphthong	0.16	1.5
diphthong	0.05	aspiration	0.00	affricate	0.13	1.1

Table 4.8: Relative importance (in %) of questions while using the generalized likelihood ratio criterion to grow the tree for some center phone classes (including contributions even when the question is not used to split the node). Relative frequencies, r.f., (in %) of phone classes in the database are also shown.

left context		right context		overall	
Class	rel. imp.	Class	rel. imp.	Class	rel. imp.
sonorant	19.41	sonorant	13.79	sonorant	16.32
coronal	6.98	vowel	9.97	front vowel	8.14
labial	6.80	front vowel	9.56	vowel	7.65
front vowel	6.40	coronal	7.94	coronal	7.51
continuant	5.70	retroflex	7.03	labial	5.69
semivowel	5.58	continuant	5.48	continuant	5.58
vowel	4.83	fricative	5.27	retroflex	5.22
round	4.19	nasal	5.07	silence	4.43
silence	4.19	high vowel	5.07	strident	4.25
strident	4.14	labial	4.79	fricative	4.25
stop	4.00	silence	4.64	stop	3.98
lax	3.80	strident	4.33	high vowel	3.95
retroflex	3.01	stop	3.96	nasal	3.91
fricative	3.00	low vowel	2.56	semivowel	3.02
back vowel	2.91	reduced vowel	2.23	lax	2.86
high vowel	2.58	lax	2.08	round	2.81
nasal	2.49	velar	1.78	low vowel	2.15
velar	1.81	round	1.68	reduced vowel	1.84
low vowel	1.64	semivowel	0.92	back vowel	1.81
reduced vowel	1.35	back vowel	0.91	velar	1.79
palatal	1.22	flap	0.48	lateral	0.71
lateral	1.13	lateral	0.38	palatal	0.58
aspiration	1.06	palatal	0.06	flap	0.50
affricate	1.06	diphthong	0.04	aspiration	0.48
flap	0.51	syllab. cons.	0.00	affricate	0.48
syllab. cons.	0.21	affricate	0.00	syllab. cons.	0.09
diphthong	0.00	aspiration	0.00	diphthong	0.02

Table 4.9: Relative importance (in %) of questions while using the generalized likelihood ratio criterion to grow the tree (only contributions of questions that result in the splitting the node are considered).

left context		right context		overall	
Class	rel. imp.	Class	rel. imp.	Class	rel. imp.
Center phone is a vowel					
sonorant	15.88	sonorant	20.90	sonorant	18.72
labial	11.99	retroflex	10.49	labial	10.20
semivowel	11.97	nasal	10.30	coronal	9.08
coronal	8.25	coronal	9.72	retroflex	7.54
stop	6.04	labial	8.83	nasal	7.25
round	5.94	fricative	6.92	fricative	6.46
fricative	5.86	strident	6.84	semivowel	5.66
continuant	4.28	continuant	6.01	strident	5.66
Center phone is a semivowel					
sonorant	32.47	front vowel	25.43	sonorant	19.02
front vowel	10.93	high vowel	25.06	front vowel	17.38
silence	9.44	low vowel	10.68	high vowel	11.65
coronal	6.73	lax	7.33	low vowel	7.13
vowel	6.32	reduced vowel	6.14	lax	5.53
round	5.26	continuant	5.00	silence	5.24
labial	4.99	fricative	4.92	vowel	4.88
low vowel	4.28	back vowel	3.46	continuant	4.60
Center phone is a sonorant					
sonorant	20.47	sonorant	17.48	sonorant	18.81
labial	9.50	coronal	8.21	labial	7.90
semivowel	8.24	retroflex	7.56	coronal	7.83
coronal	7.36	nasal	7.42	front vowel	5.76
round	5.63	labial	6.61	retroflex	5.32
front vowel	5.02	front vowel	6.35	fricative	5.31
stop	4.97	continuant	6.18	nasal	5.26
fricative	4.34	fricative	6.09	continuant	5.22

Table 4.10: Relative importance (in %) of questions while using the generalized likelihood ratio criterion to grow the tree (only contributions of questions that result in the splitting the node are considered).

left context		right context		overall	
Class	rel. imp.	Class	rel. imp.	Class	rel. imp.
Center phone is a continuant					
sonorant	17.84	sonorant	14.97	sonorant	16.30
labial	8.41	coronal	9.24	coronal	8.66
coronal	7.98	retroflex	7.71	labial	7.13
semivowel	6.90	vowel	7.42	continuant	5.87
continuant	5.92	nasal	6.60	front vowel	5.80
front vowel	5.36	front vowel	6.17	retroflex	5.65
stop	4.86	labial	6.02	vowel	5.64
round	4.85	continuant	5.83	stop	4.84
Center phone is a coronal					
sonorant	17.79	vowel	25.59	vowel	18.67
lax	10.69	coronal	11.18	sonorant	11.57
vowel	9.24	silence	9.88	coronal	10.01
front vowel	8.60	high vowel	7.74	high vowel	7.71
coronal	8.43	sonorant	7.00	front vowel	6.58
high vowel	7.66	continuant	5.79	silence	5.96
continuant	6.03	front vowel	5.10	lax	5.93
retroflex	5.93	stop	4.74	continuant	5.89
Center phone is a stop					
sonorant	28.49	front vowel	25.27	front vowel	18.33
vowel	11.04	vowel	18.30	sonorant	16.19
front vowel	9.75	high vowel	7.91	vowel	15.05
strident	9.71	retroflex	7.73	strident	6.00
lax	6.56	round	7.02	retroflex	5.47
back vowel	6.35	fricative	6.28	high vowel	5.03
continuant	5.48	sonorant	6.23	lax	4.59
low vowel	4.30	low vowel	4.03	round	4.22

Table 4.11: Relative importance (in %) of questions while using the generalized likelihood ratio criterion to grow the tree (only contributions of questions that result in the splitting the node are considered).

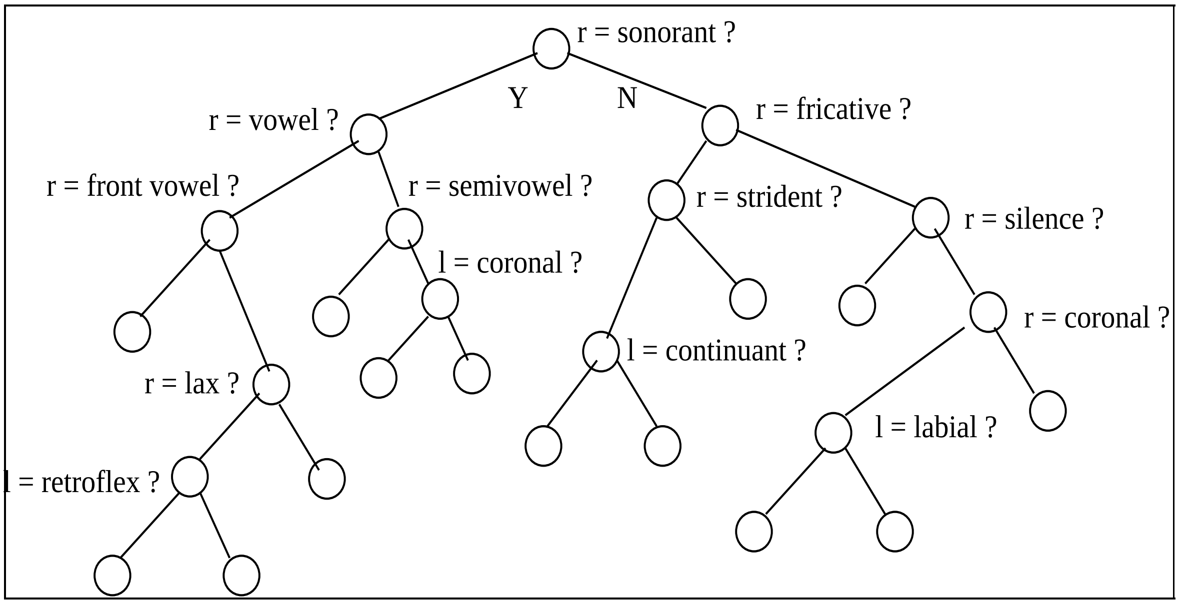

Figure 4.4: Questions asked for distribution 8 of phone “iy”.

Chapter 5

Microsegment Mixtures

Instead of using just one probability distribution to characterize the observations for a state in HMMs, researchers have looked at using mixtures of distributions [8, 24, 33]. Multimodal distributions can be made using a mixture of unimodal distributions such as Gaussians. Traditionally, mixtures of distributions are implemented for individual frames. In addition, Digalakis recently introduced a multi-level formalism for acoustic modeling and investigated the two-level framework with mixtures of microsegments (a unit smaller than a phone-segment) [13]. Initial experiments with context-independent (CI) phone classification suggested that microsegment (MS) mixtures provided a relatively simple means of capturing inter-frame correlation and improving performance. In this work, we modify the microsegment framework for word recognition, extend it to context-dependent modeling, and investigate the trade-offs of using more distributions (length of model) per microsegment versus more mixture components. In this chapter we describe the framework used, recognition and training (parameter estimation) procedures, extension to context-modeling and experiments conducted to examine this framework.

5.1 Microsegment Framework

The framework proposed by Digalakis consists of two levels: the upper level represented by phones and the lower level represented by microsegments (MS). Each phone-length segment is divided into a fixed number of regions. Each region can be characterized by more than one MS model, which is an independent-frame SSM with a fixed number of distributions (multivariate Gaussians) per MS model. The number of distributions may vary across regions but is constant for different MS models representing the same region. There is a phone-dependent stochastic grammar for the microsegment labels that functions as the mixture weights. Digalakis observed that using a dynamic warping for MS segmentation, while being more expensive, did not achieve better performance than a deterministic linear warping in phone classification [13]. Therefore, this work will use a deterministic linear warping to obtain the MS-level segmentation within a phone segment.

The evolution of MS labels is assumed to be a first-order Markov chain. Let α represent a phone, A represent the microsegment sequence, and Y the observation sequence associated with a segment. The segment acoustic probability in the MS framework is defined to be

$$p(Y|\alpha) = \sum_A p(Y|A)\, p(A,\alpha) = \sum_A \prod_i p(Y_i|a_i)\, p(a_i|a_{i-1},\alpha), \tag{5.1}$$

where Y_i and a_i represent observations and MS labels respectively for MS region i. The components of the MS mixture are MS models $p(Y_i|a_i)$, with the grammar probabilities $p(a_i|a_{i-1},\alpha)$ serving as mixture weights. The mixture components and grammar are phone-dependent. In earlier work [13], it was found that shared-mixtures (sharing the mixture components across all phones) produced poor results.

5.2 Recognition

To reduce computational requirements in recognition, we do not actually compute the mixture probability as in Equation 5.1, but rather use the dominant component of the mixture to find jointly the most probable phone and MS sequence. For a single phone segment, classification of observations Y involves

$$(\hat{\alpha}, \hat{A}) = \operatorname*{argmax}_{\alpha, A} \; p(Y|A)\, p(A, \alpha). \tag{5.2}$$

For most of our experiments, we used three MS regions to characterize a phone segment. In such cases, the segment observations are partitioned into these regions, resulting in Y_1, Y_2, Y_3, according to the number of distributions characterizing each region and Equation 5.2 can be expanded to be

$$(\hat{\alpha}, \hat{a}_1, \hat{a}_2, \hat{a}_3) = \operatorname*{argmax}_{\alpha, a_1, a_2, a_3} \; p^{[\alpha]}(a_1)p(Y_1|a_1)\; p^{[\alpha]}(a_2|a_1)p(Y_2|a_2)\; p^{[\alpha]}(a_3|a_2)p(Y_3|a_3) \tag{5.3}$$

where a_1, a_2 and a_3 represent MS labels from the three respective MS regions for the phone α. During recognition, if a short segment (of length 2 frames) is to be scored that warps only to the two corner regions, Equation 5.2 is expanded to be

$$(\hat{\alpha}, \hat{a}_1, \hat{a}_3) = \operatorname*{argmax}_{\alpha, a_1, a_3} \; p^{[\alpha]}(a_1)p(Y_1|a_1)\; p^{[\alpha]}(a_3|a_1)p(Y_3|a_3). \tag{5.4}$$

Equation 5.2 is performed using Viterbi decoding and is illustrated in Figure 5.1 for a case when the phone-segment is divided into three MS regions. In the figure, there are two MS models in the first region, four for the second and three in the third. The circles represent acoustic probabilities of the observations by the models while the dotted-lines between the circles represent bigram grammar probabilities. If log probabilities are considered, for a particular α, Equation 5.2 is a shortest-path problem solved using dynamic programming. In the figure, this path is indicated by the solid line with the solid circles showing the MS models constituting the sequence $\hat{A}$ for the particular α.

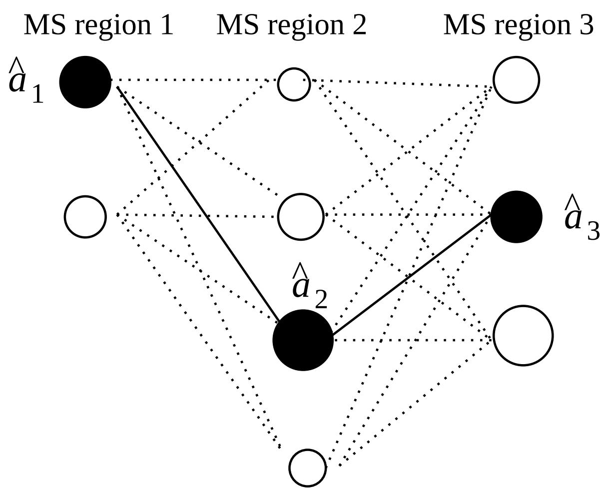

Figure 5.1: Viterbi decoding of MS labels for a phone-segment. Circles represent acoustic scores of MS models while connecting lines represent bigram probabilities.

As we allow for a variable number of microsegments components (not regions) for phones, choosing the dominant component of the mixture results in the grammar introducing differing penalties on phones with different numbers of mixture components. More specifically, phones with a small number of mixture components have higher probability MS transition probabilities than phones with a large number of mixture components, so there is a bias towards the smaller number of mixture components when recognizing according to the dominant mode. Therefore, the grammar is used in determining the best MS sequence but left out from the segment acoustic probability, i.e.,

$$p(Y|\hat{\alpha}) = p(Y|\hat{A}) \approx p(Y_1|\hat{a}_1)\ p(Y_2|\hat{a}_2)\ p(Y_3|\hat{a}_3) \tag{5.5}$$

The final probability for a segment is the acoustic probability multiplied by the phone duration probability.

5.3 Estimation of MS Parameters

As the microsegments do not correspond to any linguistic unit, we need to automatically determine and label them in the training database. Training of MS parameters involves the following steps:

1. With the phone segmentation fixed, find initial estimates of MS models –
   (a) Use binary divisive clustering on data to get initial means and partitions.
   (b) Use K-means to improve partitions and define microsegments labels.
   (c) Maximum-likelihood estimation of mixture components with the partitions found in 1 (b).
2. Use segmental K-means to estimate mixture components iteratively –
   (a) Segment speech with current MS parameters.
   (b) Maximum-likelihood estimation of parameters with current segments.

These steps are described in more detail below.

Initialization

Each MS region is initialized independently of other regions. For each MS region, an n-ary tree with one node for each phone is specified. Each node consists of all the observations for this particular phone and MS region from the training database using the deterministic linear warping. To split a node in step 1 (a) (binary divisive clustering), K-means clustering with K=2 is performed. Specifically, given initial means $(\mu_1^c, \ldots, \mu_m^c)$, $c = 1, 2$, iterate until convergence:

- For all MS in this node, associate l_i-frame observation, y_i, to cluster $\hat{c}$ where

$$\hat{c}(y_i) = \underset{c}{\operatorname{argmin}} \sum_{j=1}^{l_i} d_{M_k}(y_i, \mu_k^c) \tag{5.6}$$

- Find new means $(\mu_1^c, \ldots, \mu_m^c)$, $c = 1, 2$

where $k = f(j, l_i, m)$ is the distribution that frame j maps to resulting from linearly warping the l_i-long observation to the m-distribution MS region, μ_k^c is the centroid of the sub-cluster for distribution k and cluster c, and $d_{M_k}(.)$ is the Mahalanobis distance between the observation and the centroid with the context-independent covariance for distribution k of the phone of which N is a node.

A greedy-growing algorithm is used to split the node with the maximum reduction of node distortion resulting in an unbalanced tree. The reduction of node distortion is the difference between the node distortion of the parent node and the sum of the node distortions of the two child nodes. Node distortion for node N, $d(N)$, is defined to be

$$d(N) = \sum_{i=1}^{n} \frac{1}{l_i} \sum_{j=1}^{l_i} d_{M_k}(y_{i,j}, \mu_k) \tag{5.7}$$

where there are n microsegments of observations associated with this node and region, y_i is a microsegment of length l_i and $k = f(j, l_i, m)$ as defined earlier.

The number of terminal nodes is constrained so that the number of distribution parameters are comparable across experiments. Specifically, for the CI experiments the number terminal nodes is equal to three times the number of initial nodes, resulting in three times as many parameters as that used in the CI 8-distribution SSM. After the tree has been fully grown, K-means clustering is performed within each phone sub-tree, to obtain better estimates (Step 1(b)). The terminal nodes now define the phone-dependent MS alphabet with the means and covariances of the observations in the terminal nodes as the initial estimates for the MS models. This alphabet is referred to as the CI MS alphabet.

Iterative segmentation/re-estimation

Once initial estimates for the MS models are available, we should ideally use the Expectation-Maximize (EM) algorithm [12] to obtain better estimates. For the case of Gaussian mixtures this would involve iterating between finding ML estimates of the mixture parameters (the Gaussian means and covariance and mixture probabilities) and the probability of each observation having arisen from the respective mixture components, along the lines of pages 198–200 in [15]. This is a very expensive procedure as it involves repeated computation and inversion of sample covariances, and therefore we do not use the EM algorithm. Instead, with the initial estimates, a segmental K-means procedure is followed to obtain better estimates. This involves iterating between segmenting speech into microsegments using the current MS parameters and finding maximum-likelihood estimates for the MS parameters from the segmented speech.

As mentioned earlier, the evolution of MS labels is assumed to be a first-order Markov chain and CI bigram probabilities are estimated. These probabilities were padded maximum-likelihood estimates smoothed with the *a priori* probabilities of labels of the previous region. As the MS alphabet is much larger than the phone alphabet, methods specific to estimation of grammar from sparse data could have been used such as the Katz's method [21]. However, during recognition it was observed that the grammar score is two orders of magnitude smaller than the acoustic score of the microsegments and its exclusion does not affect recognition performance. We felt that the use of a more sophisticated technique to estimate these probabilities is not justified and was therefore not explored. Grammar probabilities for MS labels were used only for experiments using the 3-region CI MS alphabet experiments.

5.4 Context Modeling

The MS alphabet defined is substantially larger than the phone set used. Modeling context in the obvious way by having "diphone" and "triphone" equivalents, would lead to an explosion in the number of context models resulting in their poor training due to limitations of the size of the training data. Instead, we draw upon the work done in clustering triphone segment models, specifically that described in Section 4.2.3, to enable us to define context classes.

We define context classes by the collection of triphones at the terminal nodes of the context tree grown using divisive clustering with the generalized-likelihood ratio distance measure described in Chapter 4. As clustering was performed for a 8-distribution long SSM on each distribution independently, we use the cluster definitions from the 1st, 5th and 8th distributions for the three MS regions respectively. (Since there are no triphone occurrences for silence, all observations of silence are assumed to have arisen from the same context class.) By changing the stopping criterion for growing this tree we attain the desired number of context classes (number of terminal nodes), as will be described in Section 5.5.

Once we define context classes to use, we can model context using microsegments in different ways and two schemes were evaluated. First, we can retain the CI MS alphabet[1] and estimate models for these labels conditioned on the context classes. In this case, we estimate CD models for the labels from all the microsegments of data

[1] For context-modeling experiments, "CI MS alphabet" refers to using the MS labels that were produced from the CI MS tree. In the strict sense, this is not really CI as during re-estimation of the models we use context-dependent variants of these labels. However, we use this nomenclature to differentiate this from the "CD MS alphabet" that is introduced later.

given by a CI segmentation that occur only in this particular context class. Alternatively, we can incorporate information of the context classes in the MS initialization process and obtain a CD MS alphabet. In this case, the MS tree growing procedure is modified to start with a node for each context class for each phone, with observations arising from that specific context class and that MS region (see Figure 5.2). The tree is grown till we have the desired number of terminal nodes. The rest of the procedure is analogous to the estimation of CI MS acoustic models.

The recognition algorithm is the same as for the CI case. As the grammar probabilities represent the MS label evolution, for the case when we retain the CI MS alphabet (labels) and capture context by conditioning using context classes, we can use the same grammar as was used in the CI experiments. For the case when a CD MS alphabet is used, estimation of grammar probabilities is not straight-forward. In this case, context classes for a phone are defined at a MS region level, i.e., the context classes chosen for the MS regions in a given triphone differ depending upon the context defining tree which is region-dependent. As a result, the CD alphabet requires too many grammar probabilites to estimate reliably and does not allow backing off to the CI grammar. For example, if there are 3 MS regions and each region has n possible context classes there are $3n^2$ times as many bigram probabilities to estimate over the CI MS alphabet case. Therefore, grammar probabilities were not used with the the CD MS alphabet case. This should not affect performance, as CI experiments showed that the grammar probabilities have little effect on performance as noted earlier. Attempting to model context with CD grammar probabilities and CI models is an option, but was not explored for the same reason.

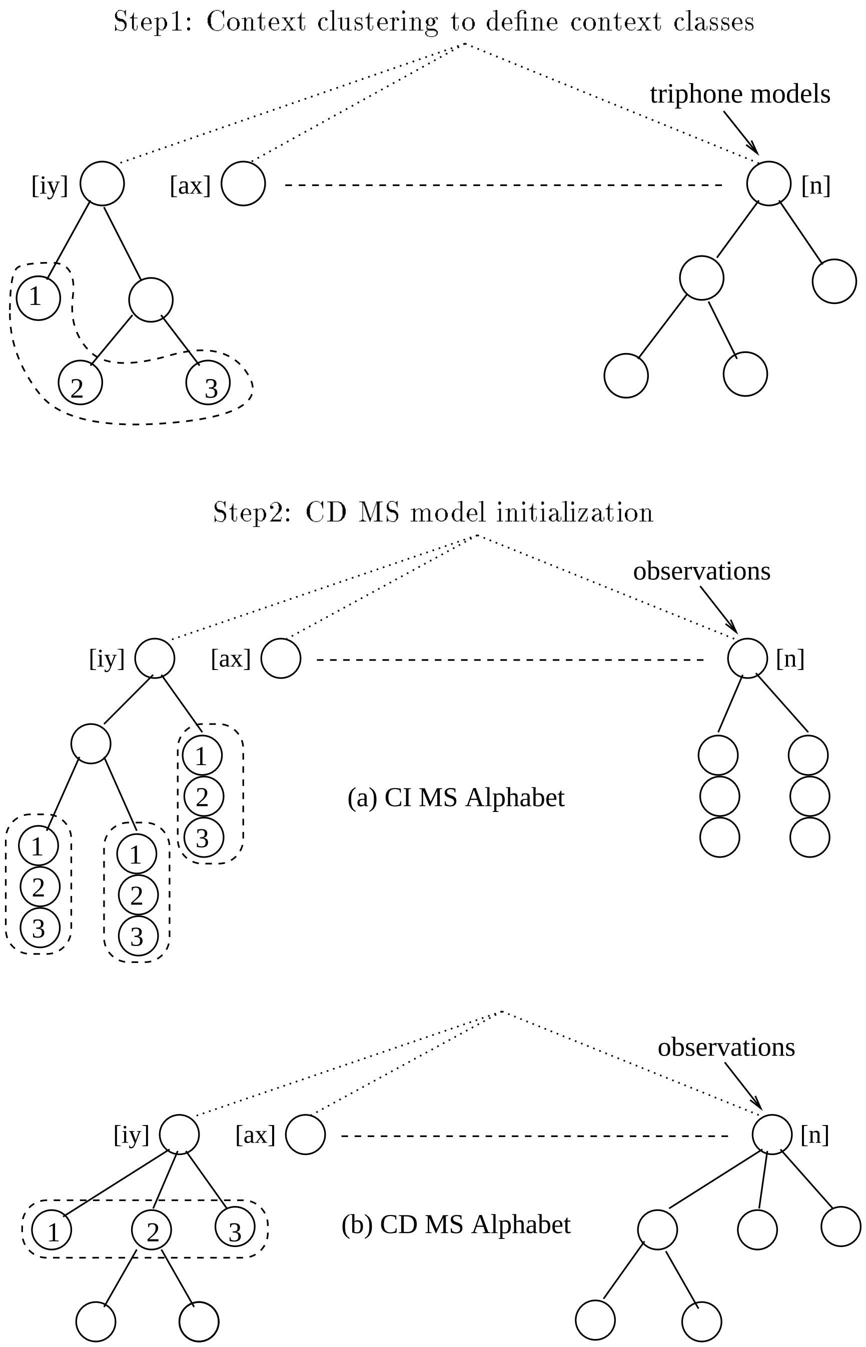

Figure 5.2: Initializing CD MS models: The top tree shows clustering of triphone models to define context classes while the two trees below show incorporation of context in MS model initialization. The CI MS alphabet (a) first finds CI MS clusters, then estimates models conditioned on context classes. The CD MS alphabet (b) finds MS clusters within the context classes.

5.5 Experiments

For all experiments 30 features were used per frame including energy, Mel-warped cepstra and first differences of energy and Mel-warped cepstra, i.e., feature reduction was not investigated. As before, gender-dependent models are used. Unlike the context experiments with the SSM, no feature reduction is performed. Word error reported here was found using gender-independent optimization and all development experiments were on the Feb89 test set. We implemented three MS systems and compared their performance with the 8-distribution long SSM. The (3,2,3) system used three MS regions in a segment with 3 distributions in the first and last MS region and 2 distributions in the middle MS region. The (1,1,1) system used three regions with one distribution length each, and the (8x1) system used 8 regions each one distribution long. These systems make different assumptions about the modeling of trajectories of features of speech. The (3,2,3) system assumes that trajectories move within a region, while the (1,1,1) system assumes trajectories are fixed within a region but has more mixture components. The (8x1) system assumes no restriction on the trajectories, and has the same form as 8-distribution SSM except that the distributions are mixtures. These trajectory assumptions are schematically illustrated for one feature in Figure 5.3.

For the first set of experiments, we used CI MS models to examine if it is more appropriate to have more components in the mixture than mixtures of sequences. All CI MS systems have three times as many parameters as the standard CI SSM. The (3,2,3) and (8x1) systems have an average of three MS models/region while the (1,1,1) system has an average of 9 MS models/region for the two corner regions and 6 MS models/region for the center region. The results of these systems as they vary with iterations of training are shown in Table 5.1. The first column indicates performance with initial estimates as obtained after the last step in the initializing

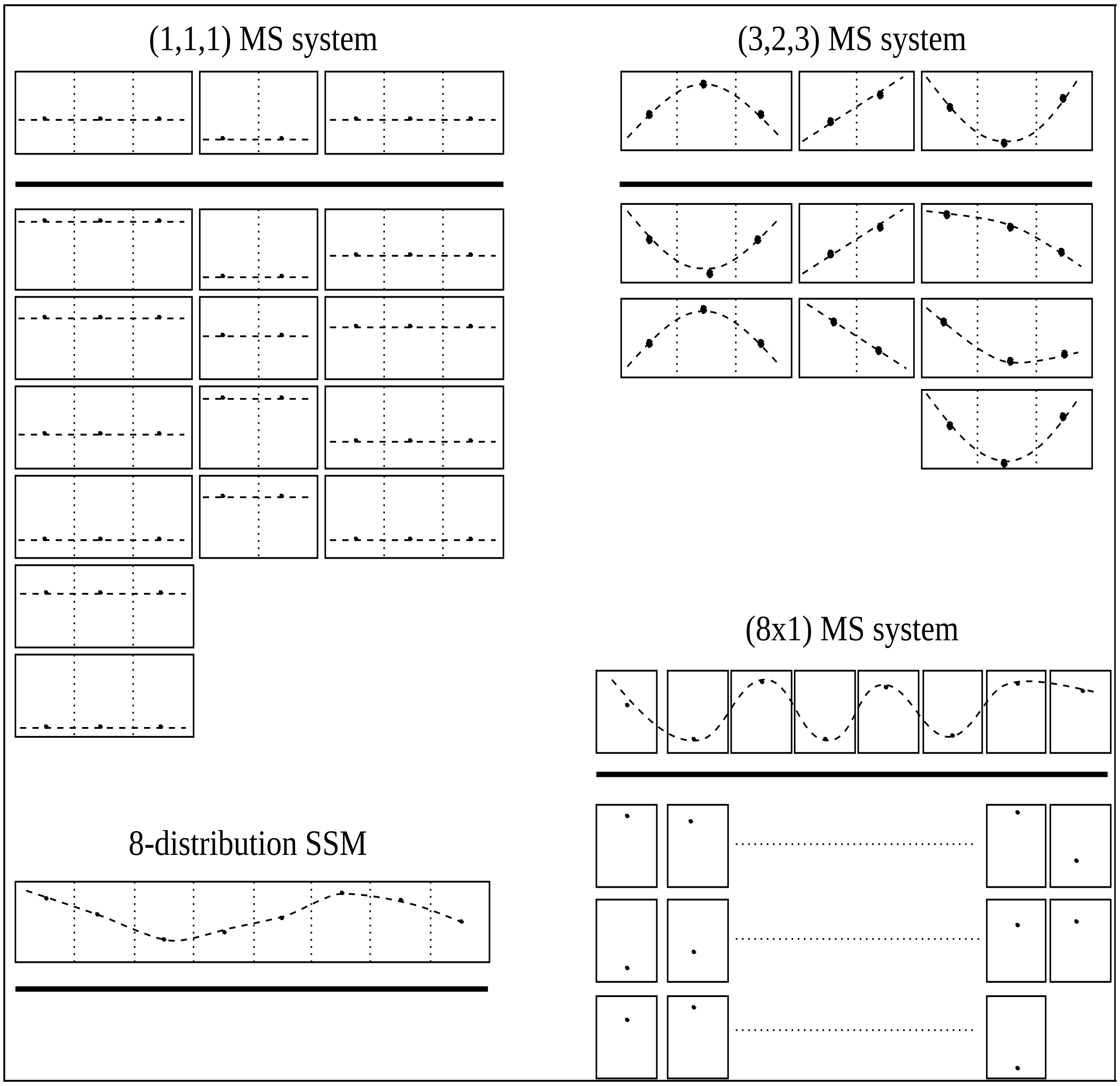

Figure 5.3: Trajectory assumptions (illustrated for one feature) for the SSM and MS systems. Clockwise from top-left, (1,1,1), (3,2,3), (8x1) MS systems and 8-distribution SSM. Mixture components (when present) are shown below the solid line.

procedure. Performance at the ith iteration is that achieved by models that were trained using microsegments from the training database that were determined using models from the $(i-1)$th iteration. These results (7.3% – 8.2%) compare favorably with the 8.9% word error of the standard 8-distribution long CI SSM system. For the CI MS systems, we see that imposing trajectory constraints is beneficial, in that the (8x1) MS system gives the highest error rate, but that it is better to have more mixture components than mixtures of sequences since the (1,1,1) system outperforms the (3,2,3) system.

For the (3,2,3) and (1,1,1) systems, an observation segment is first warped linearly to three regions. For the (3,2,3) system, each observation region is then linearly warped to the m-distribution MS model. In this scheme, the observations do not always warp to the same distributions in the (3,2,3) model as would be the case for the corresponding 8-distribution SSM. An example is illustrated in Figure 5.4. This effect is prominent for short observations for the (3,2,3) case. Therefore we performed an experiment in which we used the same warping as that used for the 8-distribution SSM. Though the original SSM warping results in higher log-likelihood scores during recognition, performance actually drops (8.0% after 3rd training iteration), and thus this scheme of warping is not used in further experiments. This surprising result suggests that warping should be further investigated, but this was beyond the scope of this thesis.

To see the effect of grammar probabilities on recognition performance, we performed recognition using the (1,1,1) system without grammar scores, i.e., the MS labels were assumed to be equiprobable. As the number of MS models/region was the largest for this case, the use of grammar probabilities would be most evident here. We found no change in recognition performance when the grammar scores were left out. As mentioned earlier, the MS acoustic scores are much greater than the MS

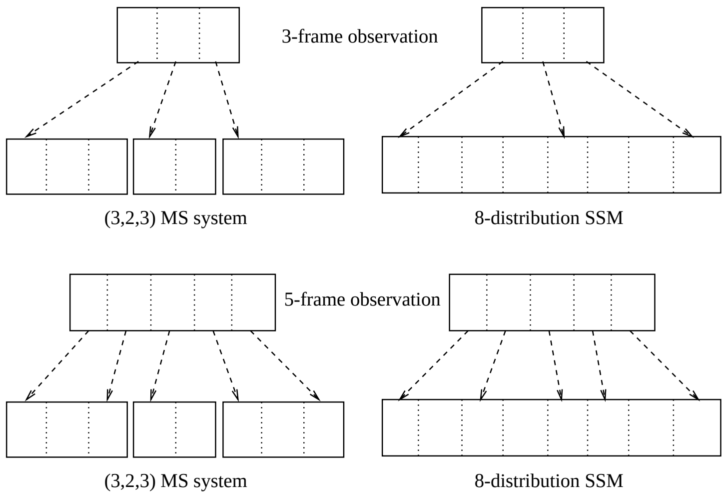

Figure 5.4: Differences in warping of two observation segments.

MS system	Word error (%)			
	Init. est.	1st iter	2nd iter	3rd iter
(8x1)	8.1	8.1	7.9	7.9
(3,2,3)	8.2	8.0	7.8	7.6
(1,1,1)	8.1	7.6	7.4	7.3

Table 5.1: Performance of context-independent MS systems on the Feb89 test set. All systems had approximately 1000 means and full covariances.

grammar scores and dominate the segment score. Therefore our efforts were directed toward improving the acoustic models.

For context-modeling experiments with MS mixtures we have to address the issue of robust parameter estimation in sparse data as the alphabet size is much larger. Since it would not be possible to estimate reliable full covariances for each of the context-dependent MS models, we performed an experiment to see what kind of covariance to use for the MS models. To simulate the effects of sparse data on the context models, we observed the effect of training CI models on a subset of the training database. The recognition experiments used three types of covariances: individual-full, individual-diagonal and shared-full. For the individual-full and individual diagonal cases all MS models have a full covariance or a diagonal covariance respectively. For the shared-full case, all MS models for a phone and region share the same full covariance which is obtained by tying over the covariances of MS models for that phone and region. In all cases this covariance (individual-full, individual-diagonal or shared-full) is used in recognition (with the individual means) if it exceeds a covariance threshold, T_R, otherwise the covariance formed by tying over all phones is used. Results of using these schemes on models trained on the entire training database and two fractions of it are shown in Table 5.2. It is seen that when the training set is reduced to one-fifth, individual diagonal covariances outperforms both the full and shared covariances. Diagonal covariances result in better performance than shared covariances when the training data is reduced, so its use was compared with individual full-covariances for context-modeling MS experiments. In addition, these results confirm that for CI models, the (1,1,1) case outperforms the (3,2,3) case.

For context-modeling with microsegments, we first attempted to model context by retaining the CI MS alphabet and estimating models of these labels conditioned on the context class defined by the tree grown using divisive clustering with the

MS System	covariance type	T_R	Word error (%)		
			full	1/3rd	1/5th
(3,2,3)	indiv-full	250	7.6	8.9	10.1
	indiv-diag	100	9.5	9.5	9.9
	shared-full	250	8.9	9.7	9.9
(1,1,1)	indiv-full	250	7.3	8.9	9.5
	indiv-diag	100	9.1	9.0	9.2
	shared-full	250	9.2	9.4	9.8

Table 5.2: Performance of context-independent MS systems on the Feb89 test set using models trained on the full, 1/3rd and 1/5th the training database.

generalized likelihood ratio distance measure. Experiments were performed with this tree grown with different cluster thresholds, T_c. The tree is grown as large as possible with the constraint that all terminal nodes have more than T_c observations. A large T_c results in fewer context classes. Since there are quite a few MS labels per phone-region, estimating models conditioned on these context classes leads to poorly estimated covariances. During recognition T_R was set to 250. If the number of observations in estimating the context covariance of a model falls below T_R, we can either back-off to the CI covariance for this model or else back-off first to the tied covariance of this model from the different context classes and if this also does not exceed T_R, we back-off to the CI model. Due to this back-off strategy, we did not investigate the use of diagonal covariances for this scheme of context modeling. The results for these experiments are shown in Table 5.3. This table also shows the number of means and covariances available to the recognizer. As quite a few covariances fall below T_R, they are mostly used for tying. Most of the means estimated by the trainer are used in recognition. The MS segmentations that were used in training models in Table 5.3 were those found after the third iteration using the CI MS systems from Table 5.1. These results indicate an improvement in performance over the CI MS systems, though not as dramatic as is usually observed when modeling context. Of course, the increase in the number of parameters from CI MS to CD MS is not as large as for the original SSM, as will be discussed later.

In the current implementation, covariances of the CD MS models are estimated as the sample covariance of the CI MS alphabet conditioned on the context class. From our divisive clustering experiments in Chapter 4.2 using entropy-based and generalized likelihood-ratio-based metrics, we found that using sample covariances of the terminal nodes in recognition was worse than using the tied covariance of models in the terminal node. Based on this observation, we anticipate an improvement if the trainer estimated MS models conditioned on triphones and then tied the covariances of these MS models across all triphones in the context class.

It is quite possible that the context that is inherently captured in the CI MS models is also captured in the context classes. Thus by further conditioning these models on these context definitions, our gain is not commensurate to the number of parameters used. Therefore, we ran experiments with a CD MS alphabet using context classes from the tree grown with T_c set to 500. To define the CD MS alphabet, the MS tree was grown as large as possible with the constraint that each terminal node have more than T_R observations. We found that keeping T_R equal to 250 results in few MS models/context class and region. To increase the number of MS models/context class we either had to reduce the number of context classes or reduce T_R. Reducing the number of context classes leads to a less detailed definition of context and reducing T_R means using less reliable estimates, both of which are undesirable. Therefore, we also implemented a system for which T_c was set at 500, but T_R was reduced to

MS System	T_c	# means	# covs	Word error (%)	
				back-off: CI	back-off: tied
(3,2,3)	250	7510	7097	6.2	6.0 (5.8)
	350	5857	5628	6.6	6.5
	500	4381	4277	6.6	6.6
(1,1,1)	250	7477	6877	6.6	6.3
	350	5911	5550	6.3	6.1 (6.1)
	500	4501	4324	6.6	6.2

Table 5.3: Performance of context-dependent MS systems with CI MS alphabet, conditioned on context classes, on the Feb89 test set. (Numbers in parentheses are obtained after one iteration of training with the best case CD systems.)

100 and only a diagonal covariance was used. Results for all these cases are shown in Table 5.4. We see that the use of diagonal covariances instead of full covariances usually increases word error. This is not inconsistent with Table 5.2 as we do not grow the tree beyond a point where we can reliably estimate the covariance. Therefore, reduction of training data in the CD MS alphabet case does not result in poorer estimates but rather results in fewer MS models. The (1,1,1) diagonal covariance system does slightly better than the (1,1,1) full covariance system since with more mixture components for the diagonal case it is less important to have full covariances.

In the CD case, the (3,2,3) system gives the best results for both approaches to CD MS suggesting that for modeling context it is more important to model trajectories than mixtures. The 8-distribution SSM, which does not use mixtures but models trajectories only at a segment-level instead of a microsegment level performs substantially better (4.8% word error) than any CD MS system. We also note that

System	covariance type	# means	# covs	Word error (%)			
				Init est	1st iter	2nd iter	3rd iter
(3,2,3)	full	1544	1532	6.8	6.6	6.3	6.4
	diagonal	3406	3394	8.6	7.8	7.4	7.0
(1,1,1)	full	1689	1555	6.6	6.7	6.5	6.5
	diagonal	3291	3288	7.8	6.6	6.4	6.1

Table 5.4: Performance of context-dependent MS systems with CD MS alphabet on the Feb89 test set. ($T_c = 500$, $T_R = 250$ for full covariance and 100 for diagonal covariance.)

the CD MS system based on the CI alphabet performs better than the one based on the CD alphabet further supporting the need to model trajectories.

The difference in performance between the CD MS system based on the CI and CD alphabet may also be due to the difference in the number of parameters used. The CD MS alphabet case with full covariance has roughly 1.5 times as many parameters as the CI MS system, while the CD models based on CI MS alphabet have between 4 to 8 times as many parameters as the CI MS system. Although the CD MS systems using the CD alphabet have far fewer covariances than the one based on the CI alphabet, almost all of them are well estimated and are used in recognition. However, the CI alphabet case has many more means and this may also account for improved performance. The CD 8-distribution SSM has about 35,000 triphone means and 37,000 covariances (22,000 triphone, 7,500 left context and 7,500 right context), a factor of about 35 times as many means and covariances as the CI MS system. Although almost all the triphone and many left/right context covariances are too poorly estimated to be used directly in recognition, they are used

System	# means	# covs
CI 8-dist. SSM	m	c
CI MS	$3m$	$3c$
8-dist. triphone SSM	$120m$	$110c$
8-dist. clustered triphone SSM	$65m$	$50c$
CD with CI MS alphabet	$(18-21)m$	$(18-21)c$
CD with CD MS alphabet	$4.5m$	$4.5c$
CD with CD MS alphabet (diag)	$9m$	$9c$ (diag)

Table 5.5: A comparison of number of free parameters of different recognition systems.

in tying over appropriate classes. However, most of the triphone means are used. The best case clustered triphone 8-distribution, discussed in Chapter 4 used about 800 clustered triphone covariances besides the 15,000 left/right context covariances and about 22,000 triphone means and outperforms the unclustered triphone SSM system. The free parameters of the different systems investigated are summarized in Table 5.5.

We evaluated the best case MS systems, the CI (1,1,1) system and the CD (3,2,3) system based on the CI alphabet, on the October 89 test set, combining scores with the weights determined using the GI optimization on the Feb89 test set. The recognition performances were 7.0% and 6.0% respectively. The 8-distribution SSM's word error rate on the Oct89 test set were 8.7% and 4.7% for CI and triphone systems respectively.

In summary, the microsegment framework for segment-based recognition enables us to examine assumptions about the trajectories of features across time in detail. Our results suggest that there is a trade-off in using mixture models and trajectory models, associated with the level of detail of the modeling unit (e.g., CI vs. CD), where it is preferable to use more mixture components for CI models although some level of trajectory constraint is useful even for CI models. The results support the use of whole segment models in the context-dependent case, and microsegment-level (and possibly segment-level) mixtures rather than frame-level mixtures.

Chapter 6

Conclusions

In this thesis, we have developed the basic building-blocks for a segment-based continuous word recognition system using the SSM, including the N-best rescoring formalism (in conjunction with BBN), and investigated methods for robust context-dependent modeling. The key results and contributions are summarized in the next section, followed by a discussion of possible directions for future work.

6.1 Summary

The research advances made in this work have resulted in the development of the N-best rescoring formalism and improvements to stochastic segment models for continuous speech recognition, with recognition results obtained on the Resource Management task that are comparable to the state-of-the-art. In addition, because of the independent frame assumptions made in this work, most of the algorithms can be extended to HMM-based systems.

The N-best rescoring formalism within which we perform recognition is useful in two ways: it reduces computational requirements to reasonable levels by constraining the search and by accommodating more knowledge sources, it has the potential of achieving better performance that any knowledge source alone. Advances in this formalism are useful to spoken language systems in general.

The N-best rescoring formalism's success hinges upon combining scores in an effective manner. We have investigated the issues related to estimation of weights for the linear combination of scores and found that there is a significant mismatch between the best set of weights for the linear combination on different test sets. This problem can be somewhat alleviated by training the weights on a larger test set and/or by gender-independent optimization. Test set mismatch is more apparent as the number of scores in the combination are increased. Benefits of research in this area extend beyond just the SSM system. Other systems that use this formalism, e.g., Segmental Neural Nets [2], will also be helped.

Clustering of triphone models has resulted in a more robust as well as parsimonious representation for triphones. We investigated both agglomerative and divisive clustering techniques. We found that it is important to use K-means clustering after agglomerative clustering to obtain better partitions. Divisive clustering using binary trees provides us with a framework to incorporate acoustic-phonetic knowledge in a statistical framework. The relative importance of questions used in clustering gives an indication of contextual factors that have the greatest effect on the acoustic realization of a phone. This information would be useful to researchers who wish to make more explicit use of this knowledge. Though gains were comparable (over 8% reduction in word error over normal triphone system on the Feb89 test set) for both techniques on the Resource Management task, we expect divisive clustering to perform better on a vocabulary-independent task as noted in [28]. As this research uses SSMs with the independent-frame assumption, these clustering techniques can be extended to HMMs with continuous observation distributions, specifically multivariate Gaussians.

The use of microsegment-based SSM models enables us to implement mixtures at a microsegment instead of frame or segment level and is a simple way of capturing some short-time correlation of features. For context-dependent modeling with microsegments, we use the context classes obtained from divisive clustering of triphone models. We examined, in detail, issues relating to the trade-offs between trajectory and mixture modeling in segment-based recognition. We find that modeling microsegments using mixtures is useful in the context-independent implementation as it gives the system more freedom in choosing a "best-fit" model resulting in a 19.5% reduction in word error on the Oct89 test set. In the context-dependent case, we find that it is more important to model the trajectory of features across time than model mixtures of sequences. The results support the use of whole segment models in the context-dependent case, and microsegment-level (and possibly segment-level) mixtures rather than frame-level mixtures.

Table 6.1 summarizes the best case results of the key experiments in this thesis. To examine the statistical significance of the results, the errors may be modeled to have a binomial distribution and as there are a large number of words in each test set, this may be approximated to have a Gaussian distribution. The resulting 2-sided confidence intervals are also shown in Table 6.1. The binomial test shows that all differences are significant except for the clustered triphone vs. unclustered triphone system and combining triphone SSM systems with HMM. The binomial model is too conservative for a confidence measure while comparing systems since the test sets are not independent (in fact all systems have been tested on the Oct89 test set). However, a better test, is the "matched-pairs" test proposed by Gillick and Cox [17], that might be used to demonstrate the significance of the difference between the different HMM-SSM systems. Even this test does not take into account other factors that affect system performance and due to time limitations, we did not perform these tests. One

	System	Word Error (%)	Conf. Int. (%)
CI	8-distribution CI SSM	8.7	±0.90
	CI (1,1,1) MS	7.0	±0.81
CD	8-dist. triphone SSM	4.7	±0.67
	8-dist. clust. tri. SSM	4.7	±0.67
	CD (3,2,3) MS based on CI MS alphabet	6.0	±0.75
With HMM	HMM (optimized)	3.8	±0.61
	8-dist. tri. SSM + HMM	3.4	±0.58
	8-dist. clust. tri. SSM + HMM	3.3	±0.57
	CD (3,2,3) MS based on CI MS alphabet + HMM	3.1	±0.55

Table 6.1: Summary of key recognition results of this thesis on the Oct89 test set.

such factor is variability across speakers; on the Oct89 test set, the word error range across speakers for the triphone HMM system was from 1.6% to 9.6%. Variability due to the choice of weights for N-best score combination is also not accounted for in these statistical tests. The word error range due to test set mismatch has been observed as 3.2% to 3.6% for the Oct89 test set for the SSM+HMM system.

The CI MS result for the Oct89 test set is among the best reported on this test set for speaker and context-independent systems. Renals *et al.* [39] report 7.9% on the Oct89 test set and Austin *et al.* [3] report 8.5% for the Feb89 test set. Ljolje *et al.* [30] report a "CI" result of 4.9% on the Feb89 test set but they used a trigram model of phonotactics and context-dependent acoustic models to pick the best scoring one as the "context-independent" score.

For context-dependent results on the Oct89 test set, BBN's Segmental Neural Nets + HMM (Byblos) system achieves a 3.0% word error [3] compared to our 3.1% error (on combining with the Byblos system). It should be noted that the neural networks were trained in a discriminative manner to "correct" mistakes made by the Byblos system, and we expect the combined SSM-HMM system could be further improved by discriminative training. Other sites have not reported recent results on the Oct89 test set. For a general idea of the range of word errors on this task by other sites, the Feb91 evaluation [34] reported the following results: SRI 4.8%, CMU 3.6%, AT&T 5.2% and MIT-Lincoln Labs 4.4%.

It has been shown earlier that segment models have performance as good or better than the HMM on phone classification and recognition. In this research, we show that the SSM, even with the independent-frame assumption, achieves word recognition performance similar to the best HMM systems. When we combine the best case context-dependent SSM systems with the Byblos (HMM) system, on the Oct 89 test set, we achieve 3.3% word error with the clustered triphone 8-distribution SSM system and 3.1% word error with the (3,2,3) CD MS system based on CI MS alphabet. The HMM system alone achieves 3.8% word error on this test set. This improvement clearly shows that the segment model and the N-best rescoring formalism is beneficial to HMMs.

6.2 Suggestions for Future Work

The research done in this thesis is by no means complete. There are many interesting extensions/modifications that can be done to the existing framework.

We used simple questions in divisive clustering of triphone models. However, the framework developed is valid for compound questions also. The design of compound questions would be much more elaborate but has the potential of capturing more detailed acoustic-phonetic knowledge in a statistical framework. Other possible extensions include questions on a bigger window of context and about other features (e.g., lexical stress).

While using the generalized likelihood ratio criterion in divisive clustering, we noted that for Gaussians, this ratio can be split into two terms, one of which depends on the individual covariances (of the two child nodes) while the other depends on the individual means. In the implementation in this thesis, we considered both terms to cluster distributions. However, since the aim of clustering distributions is to obtain better covariance estimates without affecting the mean estimates, it would be interesting to only consider the term which depends on covariances in determining clusters.

In our implementation of microsegment mixtures, we used only the dominant component of the mixture for reduced computation. The microsegment grammar score which is currently computed, but not used in recognition, can be thought of as a score for the hypothesis from a language model. We observed that this score is dominated by the acoustic score which is significantly larger. The grammar score can be included as one of the scores in the N-best score combination, which would provide a mechanism of weighting this score higher when using the dominant mode recognition algorithm and hopefully reduce the error rate. For future work, it would also be interesting to actually use all components in the form of a "true" mixture, summing rather than maximizing over the different microsegment models.

We experimented with using diagonal covariances for microsegment mixtures as they need less training data to estimate reliably than full covariances. Our implementation was simple and we did not perform any transformations to the observation features. It would be beneficial to transform the observation features using linear discriminant analysis to make the covariance "more diagonal" to introduce less of an modeling error by using diagonal covariances. Also, we can use linear discriminants to reduce the features from 30 to 29 (even for the full covariance case) as this has resulted in an improvement in performance in earlier experiments.

While analyzing the number of free parameters for the microsegment experiments, we hypothesized that a possible reason for the success of the triphone SSM system was the large number of means used. To include more means in the CD MS system based on the CD MS alphabet, the MS tree growing algorithm can be modified. The MS tree is currently grown with T_R set to 250 to estimate full covariances at the "terminal" nodes (called covariance nodes). In a possible modification for future work, the tree can be grown further with a threshold reduced, to say 50, to obtain "mean nodes". The mean nodes could constitute an "expanded" alphabet and share the covariance of their parent covariance node.

The system developed in this work is based on segment models that assume observation vectors are independent given the segment length. To take further advantage of a segment-based approach, we can explicitly model time correlation of features within the segment, as with the dynamical system formulation of the SSM [13]. In addition, the only segment-level information that we currently use is phone duration. Kimball in [23] has proposed a "classification-in-recognition" framework for the SSM to include a variety of acoustic features such as voicing and stress, including those from a broader time scale than a single segment. The N-best rescoring, context modeling and microsegment mixture methods developed here are extensible to both the dynamical system and the classification-in-recognition frameworks. When these advances are integrated together, it is expected that performance will be better than that achieved by HMM-based systems.

Appendix A

Entropy of a Multivariate Gaussian

Consider the m-variate Gaussian to be $p(x) \sim N(\mu, \Sigma)$ or

$$p(x) = \frac{1}{(2\pi)^{m/2}|\Sigma|^{1/2}} e^{-\frac{1}{2}[(x-\mu)^t \Sigma^{-1}(x-\mu)]}. \tag{A.1}$$

Its entropy is defined as

$$H(X) = -\int p(x)\ \ln p(x) dx. \tag{A.2}$$

Substituting Equation A.1 in the above equation we get

$$H(X) = \int_{R_x} \frac{1}{(2\pi)^{m/2}|\Sigma|^{1/2}} e^{-\frac{1}{2}[(x-\mu)^t \Sigma^{-1}(x-\mu)]}.\frac{1}{2}[m \ln(2\pi) + \ln|\Sigma| + (x-\mu)^t \Sigma^{-1}(x-\mu)] dx. \tag{A.3}$$

The first two terms of this equation result in $\frac{1}{2}[m\ \ln(2\pi) + \ln|\Sigma|]$. The third term can be seen to be

$$\begin{aligned} &\frac{1}{2}E\{(x-\mu)^t \Sigma^{-1}(x-\mu)\} \\ &= \frac{1}{2}E\{\ Tr[\Sigma^{-1}(x-\mu)(x-\mu)^t]\} \end{aligned}$$

$$= \frac{1}{2} Tr[\Sigma^{-1} E\{(x-\mu)(x-\mu)^t\}]$$
$$= \frac{1}{2} Tr[\Sigma^{-1}\Sigma] = \frac{1}{2} Tr[I] = \frac{m}{2},$$

where $\mathrm{Tr}(A)$ is the trace of the matrix A.

Finally, combining with the result of the first two terms we have

$$H(X) = \frac{m}{2} [\ln(2\pi) + 1] + \frac{1}{2} \ln |\Sigma|. \tag{A.4}$$

Appendix B

Glossary for Phonetic Classes used

Affricate A stop followed by a fricative made at the same place of articulation, as in /ch/ in "chat".

Aspiration A period of voicelessness after the release of an articulation. For this work this category has just the phone [hh].

Back vowel A vowel in which the tongue is close to the upper or back surface of the vocal tract, as in /u/ in "good".

Continuant In the production of continuant sounds, the primary construction in the vocal tract is not narrowed to the point where the air flow past the constriction is blocked; in stops the air flow through the mouth is effectively blocked. All phones except for nasals and stops are considered continuants.

Coronal Sounds produced with the blade of the tongue raised from its neutral position, as in /l/ in "bowl".

Diphthong A vowel in which there is a change in quality during a single syllable, as in /ay/ in "high".

Flap An articulation in which one articulator, usually the tongue tip, is drawn back and then allowed to strike against another articulator in returning to its rest position. The /t/ in "ditty" is often a flap in American English.

Fricative Narrowing of the distance between two articulators so the airstream is partially obstructed and a turbulent airflow is produced, as in /z/ in "zoo".

Front vowel A vowel in which the highest point of the tongue is in the front of the mouth, as in /e/ in "heed" and /a/ in "had".

High vowel A vowel in which the tongue body is close to the roof of the mouth, as in /e/ in "heed" and /o/ in "food".

Labial A sound where the articulation is produced with a constriction made between the upper lip and the lower lip, as in /b/ in "brown".

Lateral An articulation in which the airstream flows over the sides of the tongue, as in /l/ in "leaf".

Lax A term with no specific phonetic correlates, used when dividing vowels into classes on phonological grounds. In English the lax vowels are those that occur in monosyllables closed by /n/ such as "sing, length, long, hang".

Low vowel A vowel in which the tongue body is low down in the mouth, as in /ae/ in "head" and /a/ in "father".

Nasal A sound in which the soft palate is lowered so that there is no velic closure and air may go out through the nose, as in /m/ in "my".

Palatal An articulation involving the front of the tongue and the hard palate, as in /y/ in "you".

Reduced vowel A vowel that is pronounced with a noncontrasting centralized quality, although in the underlying form of a word it is part of a full set of contrasts.

Retroflex An articulation involving the tip of the tongue and the back part of the alveolar ridge, as in /r/ in "rye" and "row".

Round An articulation in which the lip is rounded, as in /o/ in "who".

Semivowel A sound articulated in the same way as a vowel, but not forming a syllable on its own, as in /w/ in "we".

Silence The absence of any speech sound.

Sonorant Sound produced when the vocal tract cavity configuration is such that spontaneous voicing is possible. These include vowels, semivowels, nasals and laterals.

Stop Complete closure of two articulators. This term usually implies an oral stop, i.e., complete closure of two articulators and a velic closure as in /b/ in "buy". Nasals, as in /m/ in "my" can also be considered stops, but are not classified as such for this work.

Strident Sounds that are marked acoustically by greater noisiness than their non-strident counterparts, as in /z/ in "zero".

Syllabic consonant Consonants that are at the end of a syllable, which otherwise has no vowel, as in /en/ in "button".

Velar An articulation involving the back of the tongue and the velum, or the soft palate, as in /g/ in "guy".

Vowel Vowels are produced with the vocal cords vibrating and when the vocal tract is essentially open or unconstrained.

Bibliography

[1] T. W. Anderson, *An Introduction to Multivariate Statistical Analysis*, J. Wiley & Sons, 1984.

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

[3] S. Austin, G. Zavaliagkos, J. Makhoul, and R. Schwartz, "Speech Recognition using Segmental Neural Nets", *Proceedings IEEE Int. Conf. Acoust., Speech and Signal Processing*, pp. 625–628, March 1992.

[4] L. R. Bahl, R. Bakis, P. S. Cohen, A. G. Cole, F. Jelinek, B. L. Lewis, and R. L. Mercer, "Further results on the recognition of a continously read natural corpus," *Proceedings IEEE Int. Conf. Acoust., Speech and Signal Processing*, pp. 872–875, 1980.

[5] L. R. Bahl, F. Jelinek and R. L. Mercer, "A Maximum Likelihood Approach to Continuous Speech Recognition," *IEEE Trans. on Pattern Analysis and Machine Intelligence,* March 1983, pp. 179–190.

[6] L. R. Bahl, P. V. de Souza, P. S. Gopalakrishnan, D. Nahamoo and M. A. Picheny, "Context Dependent Modeling of Phones in Continuous Speech Using Decision Trees," *Proceedings of the DARPA Workshop on Speech and Natural Language,* pp. 264–269, February 1991.

[7] I. M. Bechwati, *Structural Alternatives in Stochastic Segment Modeling for Speech Recognition*, Boston University M.S. Thesis, 1990.

[8] J. R. Bellegarda and D. Nahamoo, "Tied Mixture Continuous Parameter Models for Large Vocabulary Isolated Speech Recognition," *Proceedings IEEE Int. Conf. Acoust., Speech and Signal Processing*, pp. 13–16, 1989.

[9] P. Brown, *The Acoustic-Modeling Problem in Automatic Speech Recognition*, C.M.U. Ph.D. Dissertation, 1987.

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

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

[12] A. Dempster, N. Laird and D. Rubin "Maximum Likelihood from Incomplete Data via the EM Algorithm," *Journal of the Royal Statistical Society (B)*, Vol. 39 No. 1, pp. 1–22, 1977.

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

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

[15] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis,* Wiley-interscience, 1973.

[16] C. Espy-Wilson, Personal communications, 1992.

[17] L. Gillick and S. J. Cox, "Some Statistical Issues in the Comparison of Speech Recognition Algorithms," *Proceedings IEEE Int. Conf. Acoust., Speech and Signal Processing*, pp. 532–535, 1989.

[18] H. Gish, M. Siu, R. Rohlicek, "Segregation of Speakers for Speech Recognition and Speaker Identification", *Proceedings IEEE Int. Conf. Acoust., Speech and Signal Processing*, pp. 873–876, 1991.

[19] F. Jelinek and R. L. Mercer, "Interpolated Estimation of Markov Source Parameters from Sparse Data," *Pattern Recognition in Practice*, E. S. Gelesma and L. N. Kanal, Eds. Amsterdam, The Netherlands: North-Holland, pp. 381-397, 1980.

[20] A. Kannan, M. Ostendorf, J. R. Rohlicek, "Weight Estimation for N-Best Rescoring," *Proceedings of the DARPA Workshop on Speech and Natural Language,* February 1992.

[21] S. M. Katz, "Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recognizer," *IEEE Trans. Acoust., Speech and Signal Processing*, Vol. ASSP-35(3), pp. 400-401, March 1987.

[22] O. Kimball, M. Ostendorf and I. Bechwati, "Context Modeling with the Stochastic Segment Model,"*IEEE Trans. Signal Processing*, Vol. ASSP-40(6), pp. 1584-1587, June 1992.

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

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

[25] F. Kubala, *et al.*, "Continuous Speech Recognition Results of the BYBLOS System on the DARPA 1000-Word Resource Management Database," *Proceedings IEEE Int. Conf. Acoust., Speech and Signal Processing*, pp. 291–294, April 1988.

[26] P. Ladefoged, *A Course in Phonetics*, 2nd Edition, Harcourt Brace and Jovanovich, 1982.

[27] 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(4), pp. 509–609, April 1990.

[28] K.-F. Lee, S. Hayamizu, H.-W. Hon, C. Huang, J. Swartz, R. Weide, "Allophone Clustering for Continuous Speech Recognition," *Proceedings IEEE Int. Conf. Acoust., Speech and Signal Processing*, pp. 749–752, April 1990.

[29] H. C. Leung, I. L. Hetherington and V. Zue, "Speech Recognition Using Stochastic Explicit Segment Modeling," *Second European Conference on Speech Communication and Technology*, Genova, Italy, September 1991.

[30] A. Ljolje and M.D. Riley, "Optimal Speech Recognition Using Phone Recognition and Lexical Access", To appear in in the proceedings of *International Conference on Spoken Language Processing*, Banff, October 1992.

[31] 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," *Proceedings of the DARPA Workshop on Speech and Natural Language,* pp. 83–87, February 1991.

[32] M. Ostendorf and S. Roukos, "A Stochastic Segment Model for Phoneme-Based Continuous Speech Recognition," *IEEE Trans. on Acoust., Speech and Signal Processing,* pp. 1857–1869, December 1989.

[33] D. B. Paul, "The Lincoln Tied-Mixture HMM Continuous Speech Recognizer," *Proceedings of the DARPA Workshop on Speech and Natural Language*, pp. 332–336, June 1990.

[34] D. S. Pallett, "Session summary: Darpa Resource Management and ATIS Benchmark Test Poster Session," *Proceedings of the DARPA Workshop on Speech and Natural Language,* pp. 49–58, February 1991.

[35] M. Phillips, J. Glass and V. Zue, "Modelling Context Dependency in Acoustic-Phonetic and Lexical Representations," *Proceedings of the DARPA Workshop on Speech and Natural Language,* pp. 71–76, February 1991.

[36] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes*, Cambridge University Press, Cambridge, pp. 294–301, 1986.

[37] P. J. Price, W. M. Fischer, J. Bernstein, D. S. Pallet, "A Database for Continuous Speech Recognition in a 1000-word domain," *Proceedings IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, pp. 651–654, April 1988.

[38] L. R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," *Proceedings of the IEEE,* pp. 257–285, February 1989.

[39] S. Renals, N. Morgan, M. Cohen and H. Franco, "Connectionist Probability Estimation in the Decipher Speech Recognition System", *Proceedings IEEE Int. Conf. Acoust., Speech and Signal Processing*, pp. 601–604, March 1992.

[40] R. Schwartz, and S. Austin, "Efficient, High Performance Algorithms for N-Best Search", *Proceedings of the DARPA Workshop on Speech and Natural Language*, pp. 6–11, June 1990.

[41] R. Schwartz, and S. Austin, "A Comparison of Several Approximate Algorithms For Finding Multiple (N-BEST) Sentence Hypotheses," *Proceedings IEEE Int. Conf. Acoust., Speech and Signal Processing*, pp. 701–704, May 1991.

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