The Use of Speaker Correlation Information for Automatic Speech Recognition

by

Timothy J. Hazen

S.M., Massachusetts Institute of Technology, 1993

S.B., Massachusetts Institute of Technology, 1991

Submitted to
the Department of Electrical Engineering and Computer Science
in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

at the

Massachusetts Institute of Technology

January, 1998

Signature of Author ....
Department of Electrical Engineering and Computer Science
January 30, 1998

Certified by ....
James R. Glass
Principal Research Scientist
Department of Electrical Engineering and Computer Science

Accepted by ....
Arthur C. Smith
Chair, Department Committee on Graduate Students

The Use of Speaker Correlation Information for Automatic Speech Recognition

by

Timothy J. Hazen

Submitted to the Department of Electrical Engineering and Computer Science in January, 1998 in partial fulfillment of the requirements for the Degree of Doctor of Philosophy

Abstract

This dissertation addresses the independence of observations assumption which is typically made by today's automatic speech recognition systems. This assumption ignores within-speaker correlations which are known to exist. The assumption clearly damages the recognition ability of standard speaker independent systems, as can seen by the severe drop in performance exhibited by systems between their speaker dependent mode and their speaker independent mode. The typical solution to this problem is to apply speaker adaptation to the models of the speaker independent system. This approach is examined in this thesis with the explicit goal of improving the rapid adaptation capabilities of the system by incorporating within-speaker correlation information into the adaptation process. This is achieved through the creation of an adaptation technique called *reference speaker weighting* and in the development of a speaker clustering technique called *speaker cluster weighting*. However, speaker adaptation is just one way in which the independence assumption can be attacked. This dissertation also introduces a novel speech recognition technique called *consistency modeling*. This technique utilizes *a priori* knowledge about the within-speaker correlations which exist between different phonetic events for the purpose of incorporating speaker constraint into a speech recognition system without explicitly applying speaker adaptation. These new techniques are implemented within a segment-based speech recognition system and evaluation results are reported on the DARPA Resource Management recognition task.

Keywords: speech recognition, speaker adaptation, speaker constraint, speaker clustering, consistency modeling.

Thesis Supervisor: James R. Glass
Title: Principal Research Scientist

Acknowledgments

There are many people to acknowledge and thank for their support during the time I spent working on this dissertation.

First and foremost, my thesis supervisor Jim Glass and our group leader Victor Zue receive my deepest appreciation. Jim and Victor both provided invaluable and insightful suggestions during the course of my research. I am indebted to them for their friendly and supportive supervision of my work and their fostering of the wonderful atmosphere surrounding the Spoken Language Systems group.

This thesis could not have been possible were it not for the efforts of many members of our group. While I am grateful for all of the support I have received from everyone over the years there are several people in particular that I feel must be recognized and thanked:

- Mike McCandless and Lee Hetherington for developing the SAPPHIRE system and assisting me in learning how to use it.

- Ed Hurley, Christine Pao and Joe Polifroni for their tireless efforts in maintaining our network and systems.

I have called upon each of them for help more than my fair share of times in the last several years and I only hope I can return the favor in time.

Next, Prof. Louis Braida deserves thanks not only for the assistance he provided in his role on my thesis committee but also for the other multiple roles he's played in my graduate career including being my graduate counselor and a member of my area exam committee.

Finally, I must thank my wife Aneta and the rest of my family. Their patience, love and understanding eased the last few years of study allowing my to keep my sanity as pressure surrounding the completion of this dissertation increased. I'm sure they are all now relieved that I am finished and will be getting a "real job".

This dissertation is based upon work supported by the National Science Foundation under Grant No. IRI-9618731 and by DARPA under Contract N66001-96-C-8526, monitored through Naval Command, Control, and Ocean Surveillance Center.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Problem Definition

Automatic speech recognition systems are asked to perform a task which humans do quite easily. Because it is a task that even young children are able to perform proficiently, it is easy to overlook its complexity. In particular, speech recognition is difficult because of the large variability that is possible in any given utterance. Consider the illustration in Figure 1.1 which presents some of the sources of variability which are present in a digital recording of a speech utterance. Variability in the speech signal can result from changes in the individual speaker, the speaker's environment, the microphone and channel of the recording device, and/or the mechanism which converts the signal into its digital representation. It is important for a speech recognizer to be able to handle these types of variability.

While a speech recognizer must be able to handle a change in any of the items in Figure 1.1, it is very likely that each of the elements in the figure will remain fixed during any particular spoken utterance. In other words, typical speech utterances come from one speaker who stays in the same environment and is recorded using a fixed set of equipment. This knowledge can be used to provide constraint to the recognizer. By learning a little information about the current speaker, environment, microphone, and channel, a speech recognizer should be able improve its performance by adapting to the characteristics particular to the current utterance.

While developing a speech recognition system, the importance of the two goals stated above should be recognized. A recognizer should account for both the variability across different speech utterances and the constraints present within individual speech utterances. While being able to handle differing environments, microphones and channels is important, this dissertation will focus solely on the problems of speaker variability. Currently, typical speech recognition systems strive to achieve

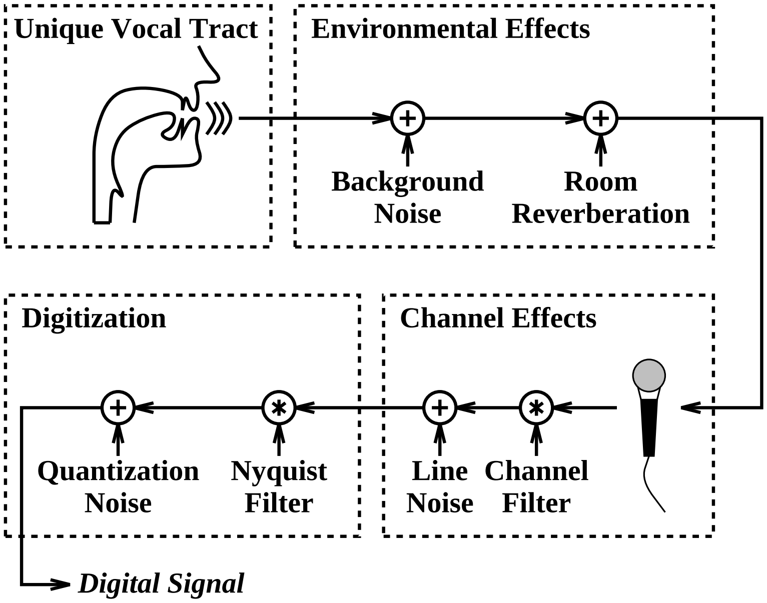


Figure 1.1: Illustration of the sources of variability in a typical recording of a spoken utterance.

strong *speaker independent* performance. However, many systems neglect the issue of speaker constraint. Thus, this dissertation will specifically examine the issues involved in utilizing speaker constraint for speech recognition.

Over the last ten to twenty years, dramatic improvements in the quality of speaker independent speech recognition technology have been made. With the development and refinement of the Hidden Markov Model (HMM) approach [7, 8, 57], today's speech recognition systems have been shown to work effectively on various large vocabulary, continuous speech, speaker independent tasks. However, despite the high quality of today's speaker independent (SI) systems [5, 25, 49], there can still be a significant gap in performance between these systems and their speaker dependent (SD) counterparts. As will be discussed in Chapter 3, The reduction in a system's error rate between its speaker independent mode and its speaker dependent mode can be 50% or more.

The reason for the gap in performance between SI and SD systems can be attributed to flaws in the probabilistic framework and training methods employed by typical speech recognizers. One primary problem lies in the fact that almost all speech recognition approaches, including the prevalent HMM approach, assume that all observations extracted from the same speech waveform are statistically independent. As will be demonstrated in Chapter 2, different observations extracted from speech from the same speaker can be highly correlated. Thus, assuming independence between observations extracted from the same utterance ignores *speaker correlation information* which may be useful for decoding the utterance. Speaker correlation information will be defined here as the statistical correlation between different speech events produced by the same speaker.

In SI systems, the independence assumption is particularly troublesome because of the manner in which SI acoustic models are trained. Because SI systems need to be robust in the face of changing speakers, their acoustic models are usually trained from a pool of data which includes all of the available observations from all available training speakers. As a result the acoustic models capture information about speech produced by a wide variety of speakers and can be quite robust to a change in speakers. Unfortunately this perceived strength is also a weakness. A large heterogeneous SI acoustic model has a much larger variance than a typical SD acoustic model trained on speech from only one speaker. The speech of one individual speaker falls into only a small constrained portion of the acoustic space occupied by speech produced by all speakers. Thus, SD models work well because they tightly match the acoustic characteristics of the one speaker on which they are used. SI models do not match any one speaker well despite the fact that they may perform adequately across all speakers.

To illustrate the differences between SI and SD models consider the contour plots shown in Figure 1.2. Each contour in this figure represents an equal likelihood contour extracted from a probability density function. The density functions were created from actual acoustic feature vectors extracted from a corpus of continuous speech. For each density function the contour which has a value of $\frac{7}{10}$ of the peak likelihood is drawn. In other words, each contour outlines its model's *high likelihood region.*

There are two main points to be learned from Figure 1.2. First, each SD model has a smaller variance and different mean location than the SI model. The SD models can also be quite different from each other. Second, the relative locations of the [i] and [e] models for each speaker are indicative of the types of within-speaker correlations which may be present between different phones. In Figure 1.2, the models for speaker HXS0 each have higher mean values than the models for speaker DAS0 for both dimensions of both phones. Knowledge of this type of relative positioning of the models across all speakers could be used to predict the location of one speaker's [i] model based on knowledge of that same speaker's [e] model, and *vice versa.*

Next consider the task of classifying tokens as either [i] and [e]. These two phones are often confused by phonetic classifiers. Figure 1.2 demonstrates that the difficulty in classifying these sounds stems from the fact that the acoustic features of [e] tokens from one speaker can be very similar to the acoustic features of [i] tokens from some other speaker. Because typical SI models are trained using speech from many different speakers, the probabilistic density functions of the acoustic models of these two phones

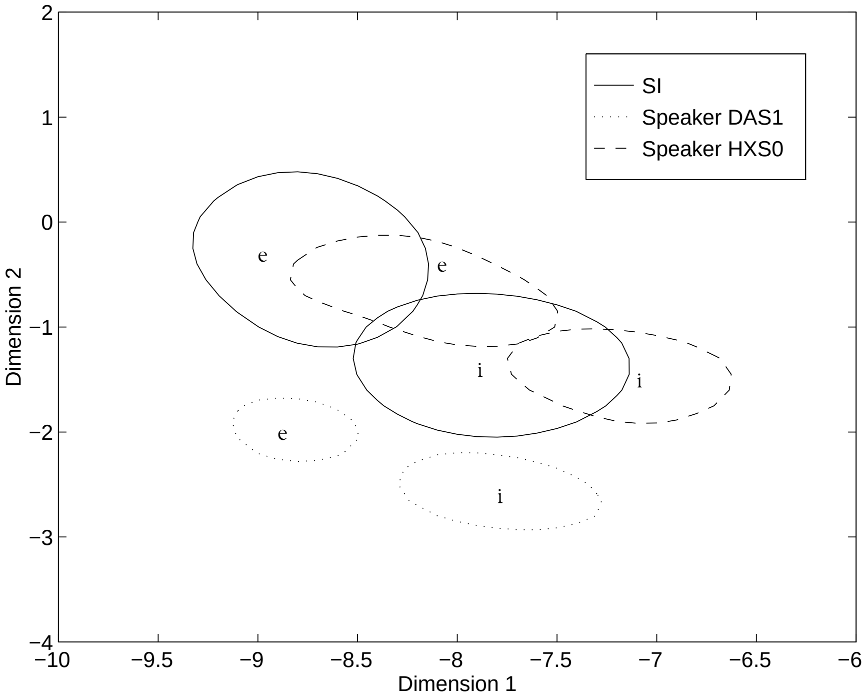


Figure 1.2: Contours plots of mixture Gaussian models for the phones [i] and [e] for a 2 dimensional feature vector. The contours are shown for the SI models and for the SD models from two different speakers.

can have significant overlap due to the similarity between [e] and [i] tokens spoken by different speakers. For example, Figure 1.2 indicates a sizeable amount of overlap exists between the high likelihood region of speaker HXS0's [e] model and the high likelihood region of the SI [i] model. On the other hand, individual speakers typically produce these two phones in acoustically distinct fashions. As such, the overlap in the SD density functions of [e] and [i] from any one typical speaker is considerably smaller then the overlap of the SI models for these phones.

To illustrate the potential use of speaker correlation information during recognition, consider the spectrogram of the utterance in Figure 1.3. This utterance is a male speaker uttering the words "*she sees the dee-jay*". This utterance contains examples of the [i] and [e] phones. During standard SI recognition, the phone [i] in the words *she*, *sees*, and *dee-jay* will all be classified independently of each other and all other vowels in the utterance. In this case it is possible for the recognizer to misclassify any one of these segments as an [e]. However, if each of the [i] tokens are considered simultaneously along with the [e] token at the end of *dee-jay*, misclassification should be less likely. Clearly the three [i] tokens are acoustically similar to each other and have distinctly different formant locations from the [e] token.

Speaker correlation can be similarly used to compare the strong fricatives in the utterance. Clearly, after observing the [š] at the beginning of the utterance, it is reasonable to assume that the [s] in *sees* is not a [š]. Likewise, a comparison of the [s] and [z] in *sees* should indicate that these two tokens, though similar in spectral content, must be different phones because of the obvious disparity in their durations. Though these types of observations are regularly used by expert human spectrogram readers, these across segment considerations are ignored in the statistical framework of most recognizers. Ideally, instead of classifying the tokens in the utterance independently of each other, these tokens should be classified *jointly* using the knowledge of the strong within-speaker correlations that exist between the phones.

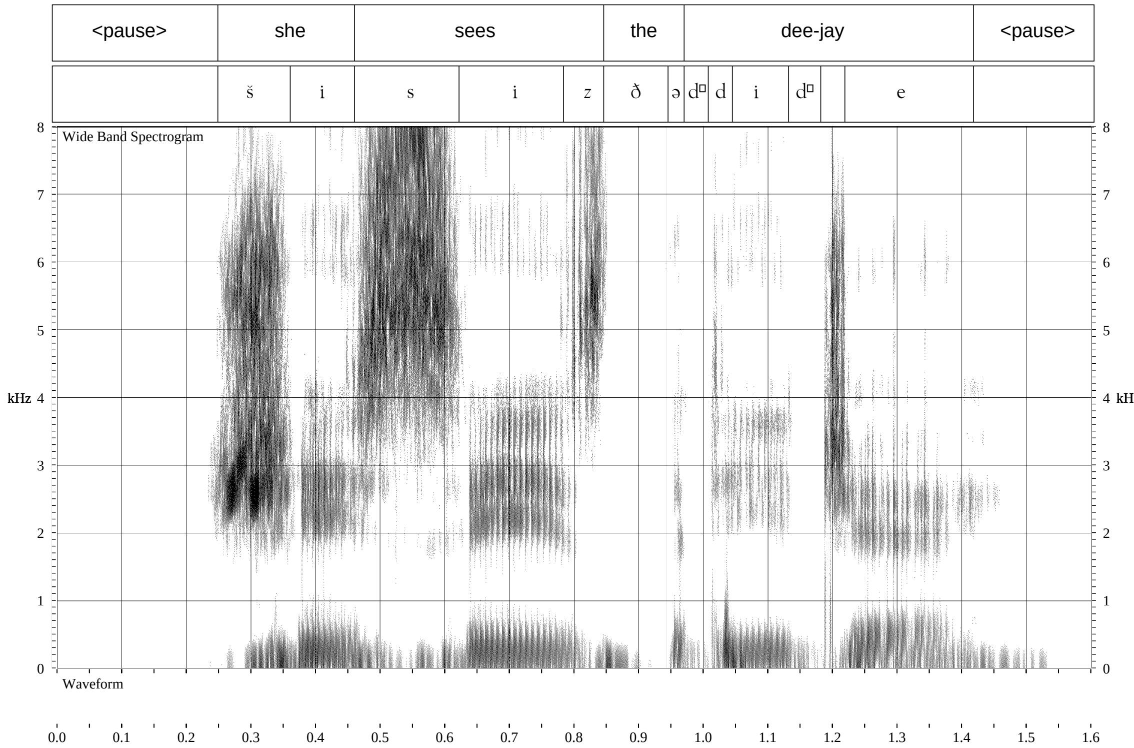


Figure 1.3: Spectrogram and aligned transcription of the utterance “*she sees the dee-jay*”.

1.2 The Independence Assumption

1.2.1 Probabilistic Description

To fully understand how the modeling decisions mentioned in the problem definition can affect typical speech recognition systems, a mathematical description of the problem must be presented. To simplify the description we will concern ourselves, for now, only with the acoustic modeling problem. The job of an acoustic model is to provide a likelihood estimate that a particular sequence of phonetic units spoken by a person could have been realized as a particular set of acoustic observations. The discussion that follows below is appropriate for practically all speech recognition systems based on a probabilistic framework. This includes the prevalent HMM approach, which is described in Appendix A, and the SUMMIT speech recognition system which is utilized in this thesis and is described in detail in Appendix B.

To describe the problem mathematically, begin by letting P represent a sequence of phonetic units. If P contains N different phones then let it be expressed as:

$$P = \{p_1, p_2, \ldots, p_N\} \tag{1.1}$$

Here each p_n represents the identity of one phone in the sequence. Next, let X be a sequence of feature vectors which represent the acoustic information of an utterance. If X contains one feature vector for each phone in P then X can be expressed as:

$$X = \{\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_N\} \tag{1.2}$$

Given the above definitions, the probabilistic expression for the acoustic model is given as $\mathrm{p}(X|P)$.[1]

In order to develop effective and efficient methods for estimating the acoustic model likelihood, typical recognition systems use a variety of simplifying assumptions. To begin, the general expression can be expanded as follows:

$$\mathrm{p}(X|P) = \mathrm{p}(\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_N|P) = \prod_{n=1}^{N} \mathrm{p}(\vec{x}_n|\vec{x}_{n-1}, \ldots, \vec{x}_1, P) \tag{1.3}$$

At this point, speech recognition systems almost universally assume that the acoustic feature vectors are independent. With this assumption the acoustic model is expressed as follows:

$$\mathrm{p}(X|P) = \prod_{n=1}^{N} \mathrm{p}(\vec{x}_n|P) \tag{1.4}$$

[1] In many texts, care is taken to distinguish between probability density functions, which are typically represented using the notation $\mathrm{p}(\cdot)$, and probability distributions, which are typically represented using either $\mathrm{P}(\cdot)$ or $\mathrm{Pr}(\cdot)$. To simplify the notation, no such distinction will be utilized in this thesis. All probability density functions and probability distributions will be notated using $\mathrm{p}(\cdot)$.

Because this is a standard assumption in most recognition systems, the term $\mathrm{p}(\vec{x}_n|P)$ will be referred to as the *standard acoustic model.*

Speech recognition systems often simplify the problem further by utilizing only a portion of the context available in P when scoring any given feature vector $\vec{x}_n$. The most extreme simplification is the assumption of context independence. In this case the output feature vector is dependent only on the identity of its corresponding phone. Thus, a context independent acoustic model is represented as:

$$\mathrm{p}(X|P) = \prod_{n=1}^{N} \mathrm{p}(\vec{x}_n|p_n) \tag{1.5}$$

Many systems provide a small amount of neighboring context as well. For example, a triphone acoustic model utilizes the identity of the phones before and after the current feature vector. The triphone model can be expressed as:

$$\mathrm{p}(X|P) = \prod_{n=1}^{N} \mathrm{p}(\vec{x}_n|p_{n-1}, p_n, p_{n+1}) \tag{1.6}$$

The utilization of context dependency in the acoustic model is an important problem which has been widely studied. It will not, however, be a primary concern of this thesis. As such, the derivations that follow will provide the full context P in the probabilistic expressions even though the actual implementation of the acoustic model may utilize only a small portion of the available context.

In Equation (1.3), the likelihood of a particular feature vector is deemed dependent on the observation of all of the feature vectors which have preceded it.[2] In Equation (1.4), each feature vector $\vec{x}_n$ is treated as an independently drawn observation which is not dependent on any other observations, thus implying that no statistical correlation exists between the observations. What these two equations do not show is the net effect of making the independence assumption. Consider applying Bayes rule to the probabilistic term in Equation (1.3). In this case the term in this expression can be rewritten as:

$$\mathrm{p}(\vec{x}_n|\vec{x}_{n-1}, \ldots, \vec{x}_1, P) = \mathrm{p}(\vec{x}_n|P)\frac{\mathrm{p}(\vec{x}_{n-1}, \ldots, \vec{x}_1|\vec{x}_n, P)}{\mathrm{p}(\vec{x}_{n-1}, \ldots, \vec{x}_1|P)} \tag{1.7}$$

After applying Bayes rule, the conditional probability expression contained in (1.3), is rewritten as a product of the standard acoustic model $\mathrm{p}(\vec{x}_n|P)$ and a probability ratio which will be referred as the *consistency ratio.* The *consistency ratio* is a multiplicative factor which is ignored when the feature vectors are considered independent. It represents the contribution of the correlations which exist between the feature vectors.

[2] The derivation contained in this section presumes time sequential processing. This presumption is made to simplify the discussion and is not a requirement of the theory presented here.

1.2.2 The Consistency Ratio

The consistency ratio is represented by the following expression:

$$\frac{\mathrm{p}(\vec{x}_{n-1}, \ldots, \vec{x}_1 | \vec{x}_n, P)}{\mathrm{p}(\vec{x}_{n-1}, \ldots, \vec{x}_1 | P)} \tag{1.8}$$

To understand what information is conveyed by this ratio, it is important to understand the difference between the numerator and denominator. Both the numerator and denominator provide a likelihood score for all of the feature vectors preceding the current feature vector $\vec{x}_n$. In the numerator, this likelihood score is conditioned on $\vec{x}_n$ while in the denominator it is not. In essence, this ratio is determining if all of the previous observed feature vectors are more likely or less likely given the currently observed feature vector $\vec{x}_n$ and the given phonetic sequence P.

Consider what this ratio represents during recognition when the phonetic string P is merely a hypothesis which may contain errors. When scoring a hypothesis, the standard acoustic model would be responsible for scoring each $\vec{x}_n$ as an independent element. The consistency ratio would then be responsible for determining if the current feature vector and its phone hypothesis is *consistent* with the previous feature vectors and their phone hypotheses under the assumption that the entire utterance was spoken under the same conditions, (i.e., by the same individual, in the same environment, etc). If the hypotheses for all of the previous feature vectors are *consistent* with the hypothesis for the current feature vector then it is expected that the value of numerator value will be greater than that of the denominator. However, if the current feature vector's hypothesis is *inconsistent* with the hypotheses of the previous feature vectors then it is expected that the numerator would be less than the denominator.

Given the above description, it is easy to see that the consistency ratio can be used to account for the within-speaker correlations which exist between phonetic events. As such the consistency ratio provides a measure of speaker constraint which is lacking in the standard SI acoustic model. Hypotheses whose aggregate consistency ratio is greater than one are deemed consistent with the assumption that all of the phones were spoken by the same person. These hypotheses thus have their standard acoustic model likelihoods boosted by the application of the consistency ratio. Likewise, hypotheses deemed to be inconsistent by the consistency ratio have their standard acoustic model likelihoods reduced.

If an accurate estimate of the consistency ratio can be obtained then all of the speaker correlation information which is ignored in the standard acoustic model will be accounted for in the estimate for $\mathrm{p}(X|P)$. However, this ratio requires an estimate for the likelihood of a large joint feature vector $(\vec{x}_{n-1}, \ldots, \vec{x}_1)$ under two different conditions. This is a very difficult modeling problem. Never the less, Chapter 6 investigates the estimation and use of the consistency ratio for speech recognition.

1.2.3 SD Recognition

The independence assumption is a major weakness of typical SI systems. By ignoring the correlations which exist between different observations, these systems are unable to provide any speaker constraint. On the other hand, SD systems provide full speaker constraint. Because SD systems have been trained with a large amount of speech from the one speaker of interest, there is relatively nothing new to be learned about the speaker's models from newly observed speech from that speaker. Because of this, the consistency ratio can be approximated as follows:

$$\frac{\mathrm{p}_{sd}(\vec{x}_{n-1}, \ldots, \vec{x}_1 | \vec{x}_n, P)}{\mathrm{p}_{sd}(\vec{x}_{n-1}, \ldots, \vec{x}_1 | P)} \approx 1 \tag{1.9}$$

Taking this into account, the acoustic model can utilize the following approximation when the recognition is performed in speaker dependent mode:

$$\mathrm{p}_{sd}(\vec{x}_n | \vec{x}_{n-1}, \ldots, \vec{x}_1, P) \approx \mathrm{p}_{sd}(\vec{x}_n | P) \tag{1.10}$$

In short, the independence assumption is relatively sound for SD systems.

Strictly speaking, the independence assumption is not completely validated when the system is a well trained SD system. Other factors could contribute to the existence of correlations between different observations. Some additional sources of constraint which may also affect the speech signal are the speaker's physiological state (healthy or sick), the speaker's emotional state (happy or sad), and the speaking style (read or spontaneous speech). While these are important constraints to be aware of, this thesis will not address them. Instead, it will be assumed that knowledge of the speaker is the only relevant information needed for providing constraint to the system.

1.2.4 Addressing the Independence Assumption

If it is assumed that the independence assumption is valid for SD systems, then it is reasonable to believe that the invalidity of the independence assumption in SI mode is a major factor in the severe drop in performance when a system is moved from SD mode to SI mode. This being said there are two ways of addressing the problem. The first way is to try to adjust the set of standard acoustic models used during recognition to match, as closely as possible, the characteristics of the current speaker (even if the current speaker is a stranger in the system's eyes). This is the approach taken by systems which utilize *speaker adaptation*. The second possible way to attack the problem is to utilize speaker correlation information directly within the probabilistic framework of the SI system. One way to accomplish this is to create models which can be used to estimate the contribution of the consistency ratio. This approach will be called *consistency modeling*. Both approaches will be examined in this thesis.

1.3 Thesis Overview

This thesis addresses the problem of incorporating speaker correlation information into a speech recognition system. In order to study the technical issues of the problem and assess the actual usefulness of the techniques proposed in this thesis, a series of analysis and recognition experiments will be presented. In order to conduct these experiments a source of data and a speech recognition system are required. For the sake of continuity, all of the experiments in this thesis were conducted using the same speech recognition system applied to the same speech corpus. The speech recognition system used in this thesis is the SUMMIT system. The SUMMIT system is a segment-based recognizer developed by members of the Spoken Language Systems Group at MIT. A full description of SUMMIT is provided in Appendix B. The corpus utilized in this thesis is the DARPA Resource Management corpus. The corpus contains continuous spoken utterances as read from prepared sentences by a variety of speakers. A full description of the corpus is provided in Appendix C.

The first step towards incorporating speaker correlation information into the framework of a speech recognition system is to study and understand the types of correlations that exist and what the magnitudes of these correlations are. This issue is investigated in Chapter 2 where two different paradigms for measuring the correlations which exist between acoustic events produced by the same speaker are explored. However, the acoustic model is not the only model for which speaker correlation information exists. Correlations are also present in models capturing segment duration information or phonological pronunciation information. Small experiments demonstrating the correlations which exist in these models are also presented.

To compensate for the inadequacies of standard SI recognition approaches, the most obvious route that can be pursued is the investigation of speaker adaptation techniques. The goal of speaker adaptation is to adjust the parameters of a generic SI system, using recently acquired speech (or *adaptation data*) from a new speaker, to match the new speaker as closely as possible. Using such techniques, the performance of the system should never be worse than the performance of the SI system, but would improve as more and more adaptation data became available. Given enough data a speaker adapted model's performance would eventually converge to the performance achieved by the fully trained speaker dependent system. Ideally, the system should converge towards its optimal speaker dependent performance using as little adaptation data as possible. Speaker adaptation has been studied widely by the speech recognition community with many significant advances being made. An overview of the basic tenets of speaker adaptation as well as a discussion of the most significant past approaches to speaker adaptation will be discussed in Chapter 3.

Although speaker adaptation attempts to bridge the gap between SI and SD systems, many adaptation techniques do not address the issue of correlation between different phonetic events. For example, standard maximum *a posteriori* probability (MAP) adaptation adapts each phone's acoustic model based on observations from the adaptation data of only that phone. Clearly this is a sub-optimal approach. The existence of correlation between sounds produced by the same speaker should allow the model of a phone to be adapted based on observations of other phones. This would be particularly useful in cases where the amount of adaptation data is limited, and only a subset of the phones have been observed in the adaptation data. To demonstrate how speaker correlation information can be incorporated into a traditional speaker adaptation framework, Chapter 4 presents a novel adaptation algorithm called *reference speaker weighting*.

One common method of incorporating speaker constraints into speech recognition systems is the use of speaker clustering. The goal of speaker clustering is to identify a cluster of training speakers which the current speaker most closely resembles, and to use models derived from these speakers for recognition. Chapter 5 presents an approach to hierarchical speaker clustering based on gender and speaking rate information. A novel variation of speaker clustering, called *soft speaker clustering*, is also presented.

Though speaker adaptation has received the lion's share of the effort devoted to this topic of research, a second route for incorporating speaker correlation information is available. The standard probabilistic framework used by typical recognizers can be retooled to account for the correlations existing between different observations within the same speech waveform. In this case, models incorporating the correlation information between different speech events would be integrated directly into the SI recognition framework. This differs from adaptation in that the SI models would never actually be altered. Chapter 6 presents a novel speech recognition technique called *consistency modeling* which is specifically designed to incorporate speaker correlation information directly into SI recognition framework by estimating the consistency ratio.

Chapter 7 presents the instantaneous adaptation problem. Instantaneous adaptation requires that a speech recognition system perform adaptation based on the same utterance that it is trying to recognize. The chapter begins with a discussion of the engineering issues which make instantaneous adaptation a difficult problem. The chapter also presents a series of experiments which combine reference speaker weighting, speaker clustering and consistency modeling into a unified framework for performing instantaneous adaptation.

Chapter 8 summarizes the research that will be presented, discusses the key issues of the thesis, and introduces potential future extensions to the ideas presented throughout the dissertation.

Chapter 2

Speaker Correlation Analysis

2.1 Overview

Before attempting to incorporate speaker constraints into a speech recognition system, it is important to understand the actual correlations which exist within the speech of individual speakers. It is obvious that correlations must exist because each speaker has a unique vocal tract, speaking rate, speaking style, regional dialect, *etc.* These speaker constraints can affect the acoustic realization, prosody, and pronunciation of a spoken utterance. What is not obvious is the relative strengths of these correlations and their potential usefulness within a speech recognition system. This chapter will address these issues. The chapter is divided into two primary sections. The first section covers acoustic model correlations. The second section examines the relationship between specific speaker properties and the models of the recognition system.

2.2 Acoustic Correlations

2.2.1 Basic Ideas

As already discussed, speech produced by any particular speaker is constrained by various factors which are unique to that speaker. For example, all of the speech from one individual is constrained by the physical configuration of that person's vocal tract. The speech production mechanism is flexible enough to allow a person to produce the many different sounds which compose the phonetic alphabet of our language. However, the size and shape of each individual vocal tract constrains the exact acoustic realization of the various phonetic elements. In fact, there is considerable literature dedicated to the relationship between the acoustic speech signal and the underlying vocal tract configuration and dimensions [21, 75]. The physical characteristics of the speaker, combined with other factors such as the speaker's regional dialect, speaking rate and emotional state all combine to constrain the acoustics produced by a speaker.

For the purposes of the analyses that follow, it is not required to understand the exact relationships between the specific characteristics of an individual speaker and the acoustics that are produced by that person. It is enough to understand that all of the phonetic sounds are all jointly constrained by these relationships. However, it is important to recognize that these relationships manifest themselves as constraints which can be examined statistically. Their effects can be measured by examining the statistical correlations which exist between the different phonetic events produced by individual speakers.

In order to be able to quantify the statistical correlations which exist within the speech of individual speakers, a model which captures the relevant information must be specified. There are two main paradigms that will be examined here. These paradigms can be summarized by the following questions:

Paradigm 1 How are two model parameters from the same speaker correlated?

Paradigm 2 How are two acoustic observations from the same speaker correlated?

Paradigms 1 would primarily be used in the context of speaker adaptation, i.e., when adjusting the parameters of a model based on the current observations and an a priori knowledge of the correlations between model parameters. Paradigm 2 would be used in the context of direct incorporation of speaker correlation information into the speech recognition modeling scheme, such as the consistency modeling approach. In this section, these two methods will be explored on data from the DARPA Resource Management (RM) corpus [22, 69]. A complete description of the corpus is provided in Appendix C.

2.2.2 Paradigm 1

One way in which speaker correlation information can be used is to constrain the space of possible speaker dependent (SD) models. Let Θ represent the set of parameters in a SD model set. Furthermore let Θ be subdivided into M different subsets, each one corresponding to a specific phonetic model as follows:

$$\Theta = \{\theta_1, \theta_2, \ldots, \theta_M\} \tag{2.1}$$

Here each θ_m represents a set of model parameters for the m^{th} phonetic class. Although many training and adaptation techniques assume independence between the different phonetic models in Θ, correlations do exist between the parameters of these models. To demonstrate this fact the statistical correlations between the model parameters of training speakers can be examined. Because typical speech recognition systems often use complicated models with many parameters, such as mixtures of Gaussian models, it is not practical to examine the correlations between all possible parameters. However, a reasonable understanding of the correlations which exist between different phonetic models can be obtained by examining a reduced set of parameters.

To begin, it will be assumed that each speaker can be represented by a single *speaker vector* which contains a set of model parameters describing that speaker. The speaker vector can be constructed by concatenating a set of vectors each of which contains parameters corresponding to a particular phonetic model. In the analysis that follows, 54 different phonetic units are utilized. The parameters used to represent each phonetic unit are the mean vectors of the 36 segment-based measurements used by SUMMIT (see Appendix B). These measurements are predominately spectrally-based and include some measurements which extend over the segment boundaries into the preceding or following segments. Thus, each speaker is represented by a vector with $36 \times 54 = 1944$ different dimensions. Let the mean vector for phone i, as averaged over all examples from a single speaker, be represented as $\vec{\mu}_i$. The vector representing the entire speaker space for a speaker can then be represented as:

$$\vec{m} = \begin{bmatrix} \vec{\mu}_1 \\ \vdots \\ \vec{\mu}_{54} \end{bmatrix} \tag{2.2}$$

In this analysis, the data from all 149 speakers in the training, development, and evaluation sets of the SI section of the RM corpus is used. A speaker vector $\vec{m}$ is created for each speaker. It should be noted that some phones were not spoken by all of the speakers. As a result some of the speakers have empty values for some of their $\vec{\mu}_i$ vectors.

In order to learn which sets of phones have high within-speaker correlations the cross correlation coefficients of the $\vec{m}$ vectors over all speakers need to be examined. This is done by calculating the standard correlation matrix over all 149 speaker vectors. Using these definitions the entire correlation matrix can be represented as:

$$\mathbf{C} = \begin{bmatrix} \mathbf{C}_{1,1} & \mathbf{C}_{1,2} & \cdots & \mathbf{C}_{1,54} \\ \mathbf{C}_{2,1} & \mathbf{C}_{2,2} & & \\ \vdots & & \ddots & \\ \mathbf{C}_{54,1} & & & \mathbf{C}_{54,54} \end{bmatrix} \tag{2.3}$$

where each sub-matrix $\mathbf{C}_{i,j}$ has dimension 36×36 and represents the cross-correlation coefficients between the mean vectors $\vec{\mu}_i$ and $\vec{\mu}_j$.

Mathematically, each sub-matrix $\mathbf{C}_{i,j}$ can be found using the expression:

$$\mathbf{C}_{i,j} = \frac{1}{R}\sum_{r=1}^{R}(\vec{\mu}_{i,r} - \mathrm{E}(\vec{\mu}_i))(\vec{\mu}_{j,r} - \mathrm{E}(\vec{\mu}_j))^T \tag{2.4}$$

Here each $\vec{\mu}_{i,r}$ represents the mean vector for the i^{th} phone for speaker r. The expected value of $\vec{\mu}_i$ is estimated as:

$$\mathrm{E}(\vec{\mu}_i) = \frac{1}{R}\sum_{r=1} R\vec{\mu}_{i,r} \tag{2.5}$$

Because some of the speakers have missing values for some $\vec{\mu}_i$ vectors, each sub-matrix $\mathbf{C}_{i,j}$ is computed using only the speakers who have non-empty vectors for $\vec{\mu}_i$ and $\vec{\mu}_j$.

The 1944×1944 correlation matrix, $\mathbf{C}$, has a sub-structure of 54×54 sub-matrices representing the cross-correlations of the parameters of each pair of phones. If we are interested in the *overall* correlation between two phones it is necessary to reduce each 36×36 sub-matrix to a single number. One simple way to do this is to add the absolute values of all 36×36 components within each sub-matrix to get a single *summed correlation* value. Each sub-matrix $\mathbf{C}_{i,j}$ can be represented as:

$$\mathbf{C}_{i,j} = \begin{bmatrix} c_{1,1}^{(i,j)} & c_{1,2}^{(i,j)} & \cdots & c_{1,36}^{(i,j)} \\ c_{2,1}^{(i,j)} & c_{2,2}^{(i,j)} & & \\ \vdots & & \ddots & \\ c_{36,1}^{(i,j)} & & & c_{36,36}^{(i,j)} \end{bmatrix} \tag{2.6}$$

For each $\mathbf{C}_{i,j}$ the summed-correlation value is given as:

$$s_{i,j} = \sum_{m=1}^{36}\sum_{n=1}^{36}|c_{m,n}^{(i,j)}| \tag{2.7}$$

By taking this approach, the 1944×1944 full correlation matrix can be reduced to a single 54×54 summed correlation matrix. This matrix can be represented as:

$$\mathbf{S} = \begin{bmatrix} s_{1,1} & s_{1,2} & \cdots & s_{1,54} \\ s_{2,1} & s_{2,2} & & \\ \vdots & & \ddots & \\ s_{54,1} & & & s_{54,54} \end{bmatrix} \tag{2.8}$$

The $\mathbf{S}$ matrix can be normalized to form a new *pseudo-correlation* matrix, $\mathbf{\Phi}$, where the elements of $\mathbf{\Phi}$ are found with the expression:

$$\phi_{i,j} = \frac{s_{i,j}^2}{s_{i,i}s_{j,j}} \tag{2.9}$$

Thus, $\mathbf{\Phi}$ is a matrix whose diagonal values are 1 and whose off-diagonal values represent the within-speaker *pseudo-correlation* between the model parameters of different pairs of phones. In essence, these cross-correlation values give an indication as to how easy or difficult it is to predict the mean vector parameters of one phone for a given speaker given the mean vector parameters of another phone.

Figure 2.1 shows a graphical representation of the pseudo-correlation matrix $\mathbf{\Phi}$. In this matrix dark colors represent regions of high correlation while lighter colors represent smaller correlations. As expected, there is a high amount of within-speaker correlation within different broad phonetic classes. For example, the vowels are highly correlated with each other. Similar correlations are also clearly evident between the strident fricatives as well as between the nasals. It is also not surprising that the phones [p] and [θ] exhibit relatively small amounts of correlation with other phones because these phones are produced at the extreme end of the vocal tract and their acoustic realizations are practically not affected by the dimensions or shape of the vocal tract behind their point of production.

There are also some unexpected observations that can be made from the figure. First, each of the stops is highly correlated with its respective closure. This may seem unusual because stop bursts are acoustically dissimilar to stop closures. However, as discussed in Appendix B, the set of segment-based measurements that were used for this experiment include measurements which extend backward into the previous segment and forward into the next segment. Thus, because the left context of a stop is almost always its own stop closure, the measurements for an average stop will include information about the closure preceding it, and *vice versa* for the stop closure. Similar arguments can be used to explain why [ɾ] and [v] are highly correlated with the vowels. Unexpectly the figure shows that the phone [ɔ] is most correlated with the phone [r]. This can be explained by the fact that the phone [ɔ] is very often followed by the phone [r] in the RM corpus. This occurs in words like "for" and "or".

The information in $\mathbf{\Phi}$ can be used to organize the phones into sets which contain high within-set correlations but lower across-set correlations. One way to do this is with bottom-up hierarchal clustering. The elements of $\mathbf{\Phi}$ can be turned into *pseudo-distances* using the expression

$$d_{i,j} = 1 - \phi_{i,j}. \tag{2.10}$$

Figure 2.2 shows a bottom-up clustering based on the pseudo-distance values. The clustering was initialized with each phone representing its own cluster. The hierarchical cluster tree was then created in an iterative fashion with two clusters being combined into one during each iteration. During each iteration the clustering algorithm combined the two clusters so as to minimize the increase in total cluster distortion. As would be expected the cluster tree separates the phones into distinct clusters such as vowels, consonants, strident fricatives, nasals, labial stops, etc.

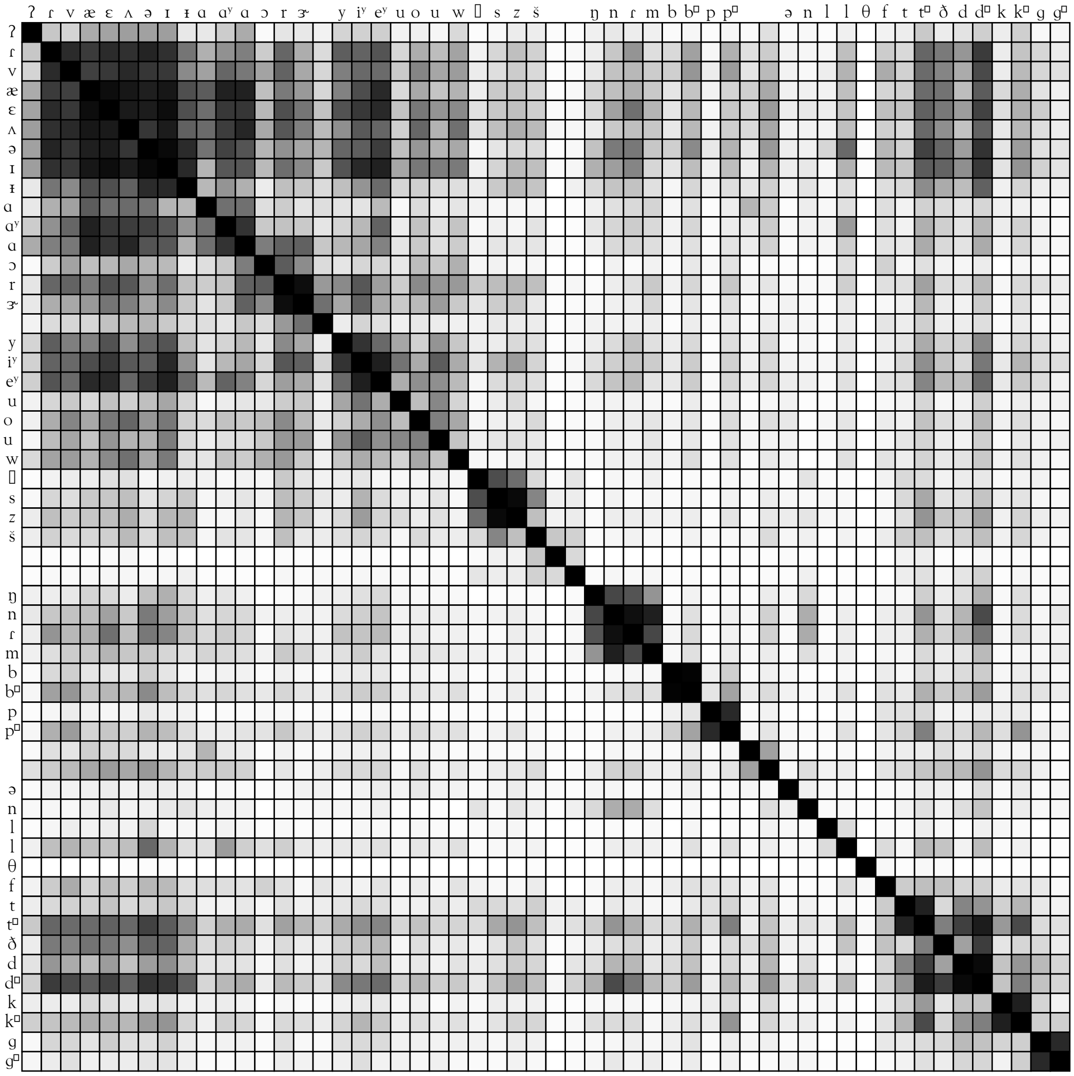


Figure 2.1: Plot of within-speaker pseudo-correlations across the mean vectors of 54 phones.

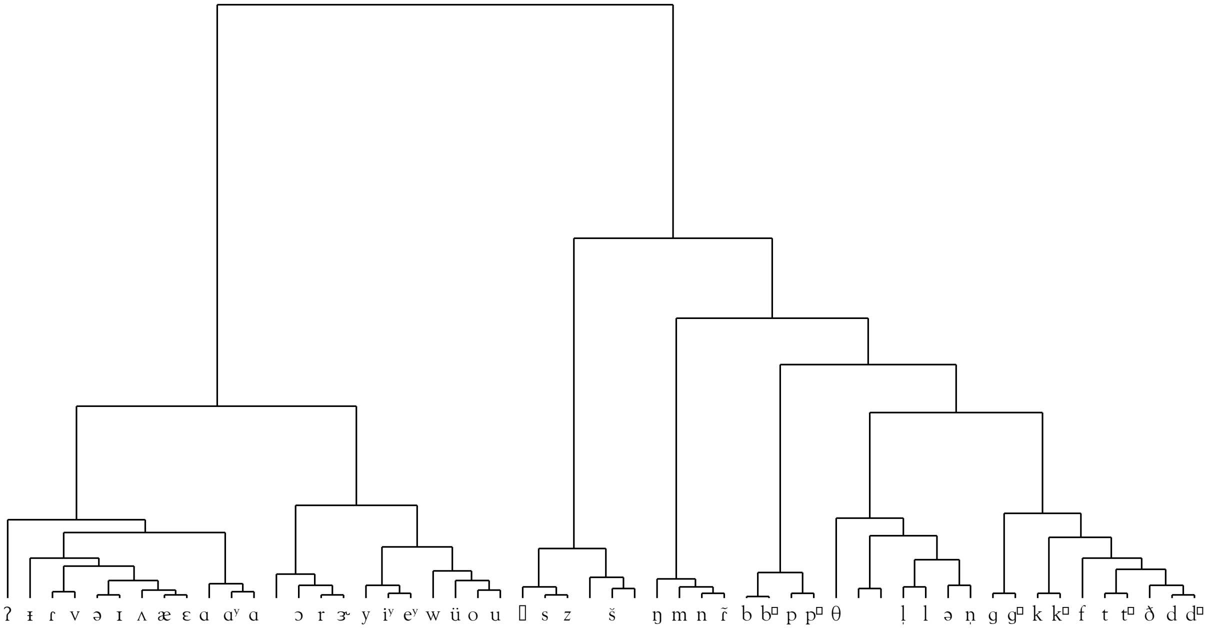


Figure 2.2: Phone clustering of 54 phones based on within-speaker correlations.

2.2.3 Paradigm 2

In Paradigm 1 the correlations between different model parameters from the same speaker are examined. While this information may be helpful for constraining the space of possible speaker dependent models during speaker adaptation, this analysis is not suitable for techniques which examine the speech from one speaker at the level of the phonetic observation, such as consistency modeling. Consider the problem of modeling the consistency ratio introduced in Chapter 1. Once again, the consistency ratio is expressed as:

$$\frac{\mathrm{p}(\vec{x}_{n-1},\ldots,\vec{x}_1|\vec{x}_n,P)}{\mathrm{p}(\vec{x}_{n-1},\ldots,\vec{x}_1|P)} \tag{2.11}$$

This ratio accounts for the correlations which exist between different acoustic observations. In particular, the ratio stresses the importance of the correlations which exist between the current observation $\vec{x}_n$ and the observations that precede it.

To simplify the problem, the correlations between phones will be investigated by examining the different phones in a pairwise fashion. By examining pairwise phone correlations, it is possible to learn which phones have large within-speaker correlation and which phones have very little within-speaker correlation.

One method for determining the within-speaker correlations of a pair of phones is to estimate correlations for the joint density function $\mathrm{p}(\vec{x}_j,\vec{x}_k|p_j,p_k)$. The joint density is intended to capture the likelihood of the two phones under the assumption that they were produced by the same speaker. There are various different ways in which the correlations contained in the joint density function can be estimated. The method presented in the analysis that follows utilizes a two step process.

The first step of the process is to create joint vectors of a particular phone pair by concatenating individual vectors from each of the two phones from one speaker. For example, suppose the data set for a particular speaker contains two instances of the phone [s] and three instances of the phone [t]. The observation vectors for the [s] exemplars can be represented as $\vec{x}_{s,1}$ and $\vec{x}_{s,2}$. Likewise observation vectors for the [t] exemplars can be represented as $\vec{x}_{t,1}$, $\vec{x}_{t,2}$, and $\vec{x}_{t,3}$. From the observations of these two phones a set of joint vectors, $X_{s,t}$, can be created for this one speaker. If all combinations of the two phones are considered then six total joint vectors would be created. The joint vectors contained in $X_{s,t}$ would be represented as:

$$X_{s,t} = \left\{ \begin{bmatrix} \vec{x}_{s,1} \\ \vec{x}_{t,1} \end{bmatrix}, \begin{bmatrix} \vec{x}_{s,1} \\ \vec{x}_{t,2} \end{bmatrix}, \begin{bmatrix} \vec{x}_{s,1} \\ \vec{x}_{t,3} \end{bmatrix}, \begin{bmatrix} \vec{x}_{s,2} \\ \vec{x}_{t,1} \end{bmatrix}, \begin{bmatrix} \vec{x}_{s,2} \\ \vec{x}_{t,2} \end{bmatrix}, \begin{bmatrix} \vec{x}_{s,2} \\ \vec{x}_{t,3} \end{bmatrix} \right\} \tag{2.12}$$

The second step of the process is to pool all of the joint vectors from all speakers together, and estimate the correlation from the entire pooled set of joint vectors. Figure 2.3 provides a fictitious illustration of how the joint vectors from three different speakers can be created and combined. In this figure each phone observation is

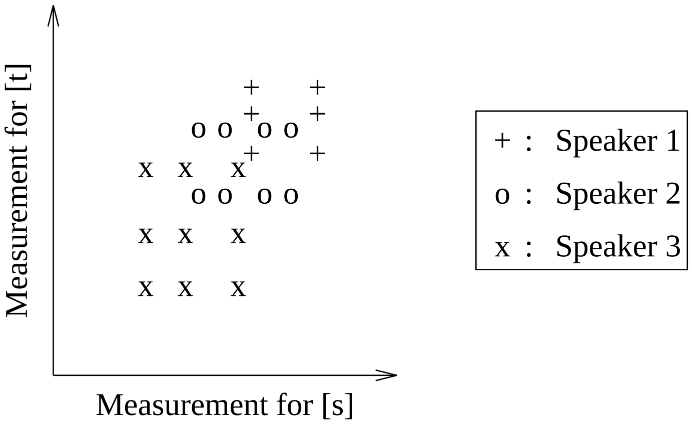


Figure 2.3: Fictitious illustration of joint vectors created for the pair of phones [s] and [t] as collected from three different training speakers.

represented by a one dimensional measurement, giving the joint phone vectors two dimensions. For this example, speaker 1 has two examples of [s] and three examples of [t]. Similarly, speaker 2 has four examples of [s] and 2 examples of [t], while speaker 3 has three examples each of [s] and [t].

An examination of Figure 2.3 shows that no correlation between phones [s] and [t] exists if only the observations of one speaker are considered. This is consistent with the supposition made in Chapter 1 that acoustic feature vectors can be treated as independent when the models are speaker dependent. However, correlation between the observations of two phones does exist when the joint vectors of many speakers are considered simultaneously.

To determine which phone pairs would be most useful to use, the within-speaker correlations between phones can be examined. For any phone pair, a collection of all of the joint vectors collected from all of the training speakers is created as demonstrated in Figure 2.3. Each exemplar phone is represented using a 36 dimensional feature vector. Thus, each phone pair joint vector contains 72 dimensions. From the entire collection of joint vectors from one phone pair, a 72×72 correlation matrix is computed. The correlation matrix, $\mathbf{C}$ can be subdivided into four 36×36 submatrices as follows:

$$\mathbf{C} = \begin{bmatrix} \mathbf{C}_{1,1} & \mathbf{C}_{1,2} \\ \mathbf{C}_{2,1} & \mathbf{C}_{2,2} \end{bmatrix} \tag{2.13}$$

Using this notation, $\mathbf{C}_{2,1}$ represented the submatrix corresponding to the cross correlation information between the two phones. It is this submatrix which determines how much correlation exists between observations of the two phones. There are several ways to reduce this submatrix to a single value representing the correlation between the two phones. The simplest way is to sum all 1296 correlation values in the 36×36 cross correlation submatrix. Because of the limited amount of data available for some phones, this method may create a summed correlation value which is very noisy due to the accumulated estimation error summed over all 1296 correlation values. To reduce the noise in the summed correlation value only the 36 diagonal elements of $\mathbf{C}_{2,1}$ were summed to compute the summed correlation value used to determine the relative correlations between the same measurements of different phone pairs. This method gives a summed correlation with a maximum value of 36.

Table 2.1 shows a selection of phones and the 5 phones which are most correlated with each selected phone based on the summed correlation measure described above. The first column of the table contains a selected phone. The columns to the right of that phone display the 5 phones which contain the most within-speaker correlation with that selected phone. The number beside each of the top 5 phones is the value of summed cross correlation estimate described above. This table excludes phones which had only a limited amount of available training data, such as [ɔ^{y}] and [ž].

As can be seen in the table, some expected correlations exist. For example, observations of the phone [ɑ] are highly correlated with other observations of [ɑ] as well as with the other low vowels such as [ɔ], [æ], [ʌ] and the diphthong [ɑ^{y}]. Similarly, all of the nasals are highly correlated. In some cases, it is observed that the place or articulation of a phone contributes more correlation information than the phone's manner class. For example, the phones exhibiting the most within-speaker correlation with the closure [ɡ$^{□}$] are the phones [ɡ$^{□}$], [ŋ], [ɡ], [k$^{□}$] and [y]. All five of these phones share a similar place of articulation to [ɡ$^{□}$], but only two of them are stop closures themselves.

Another interesting observation is that phones which exhibit a large amount of variance in their possible acoustic realizations, such as [d] and [t] which have multiple allophones which may be produced, are shown to exhibit very little correlation with observation of any other phones (including themselves). Similarly, of the phones most correlated with [ə], the top five do not even include [ə], indicating that an observation of [ə] gives less information about how other examples of [ə] might be realized by the same speaker than an observation of [ə] gives about less variant vowels such as [ɛ] and [ʌ].

Overall, the table provides rich information about the types of correlations which exist between different speech observations. This information will be used extensively in Chapter 6 when consistency modeling is presented.

Phone	Most Correlated Phones									
	1		2		3		4		5	
ɑ	ɑ	5.631	ɑʸ	5.231	ɔ	4.332	æ	4.227	ʌ	4.104
æ	æ	6.083	ɑʸ	5.399	ɛ	5.150	e	4.490	ʌ	4.266
ʌ	ʌ	5.157	o	4.674	ɑʸ	4.474	ɛ	4.379	æ	4.265
ɔ	ɔ	5.263	ɑ	4.332	o	3.866	ɑʸ	3.770	ʌ	3.472
ə	ɛ	3.499	ʌ	3.460	æ	3.425	o	3.360	ɑʸ	3.320
ɑʸ	ɑʸ	7.537	æ	5.399	ɑ	5.230	ɛ	4.649	ʌ	4.475
b	b	3.137	ð	2.321	b□	2.308	ɑʸ	2.046	u	2.040
b□	b□	3.367	v	2.764	m	2.757	ɾ̃	2.585	ŋ	2.533
d	d	2.717	ð	2.053	g	2.039	š	1.921	ɾ̃	1.914
d□	n	2.901	ɾ̃	2.889	ɾ	2.844	d□	2.813	ŋ	2.705
ð	ð	4.467	ɾ	2.948	ɾ̃	2.752	æ	2.555	ɛ	2.485
ɾ	ɾ	6.401	ɾ̃	4.368	æ	3.682	ɛ	3.665	n	3.532
ɛ	ɛ	5.508	æ	5.149	ɑʸ	4.648	e	4.394	ʌ	4.380
ɝ	ɝ	5.121	ɑ	3.633	ɛ	3.547	r	3.537	u	3.386
e	e	6.367	i	4.843	æ	4.489	ɛ	4.393	o	4.320
f	f	3.056	š	2.268	ð	1.893	v	1.760	ɾ	1.741
g	g	4.218	g□	2.701	y	2.261	ð	2.107	ŋ	2.088
g□	g□	4.082	ŋ	3.039	g	2.701	k□	2.433	y	2.388
ɪ	ɛ	4.088	ɪ	4.063	e	4.032	æ	3.849	ü	3.682
i	i	5.317	e	4.843	y	4.015	ü	3.983	u	3.952
k	k	2.078	g	1.800	š	1.388	s	1.370	t	1.268
k□	k□	2.704	g□	2.433	ŋ	2.366	n	1.910	ɾ̃	1.843
l	l	4.432	o	2.981	ɑʸ	2.875	ɑ	2.674	ɾ̃	2.669
m	m	6.468	ŋ	5.372	ɾ̃	5.364	n	5.110	o	3.442
n	n	7.144	ɾ̃	6.396	ŋ	6.260	m	5.110	ɾ	3.532
ŋ	ŋ	8.931	n	6.260	ɾ̃	5.771	m	5.372	y	3.649
o	o	6.631	ʌ	4.674	e	4.320	ɑʸ	4.190	u	4.130
p	p	2.287	b	1.653	f	1.577	š	1.439	ð	1.422
p□	p□	2.848	b□	2.139	v	2.070	ŋ	2.039	ɾ̃	1.992
r	r	3.978	ɝ	3.537	ɑ	2.833	ɾ	2.727	ɔ	2.711
s	s	5.054	z	4.263	š	2.692	□	2.070	ð	1.781
š	š	8.478	s	2.692	ɾ	2.493	z	2.369	ŋ	2.276
t	t	1.740	š	1.739	d	1.438	s	1.270	k	1.268
t□	n	1.976	d□	1.812	ɾ̃	1.812	ɾ	1.809	t□	1.805
u	u	6.200	ü	5.088	o	4.130	e	4.003	i	3.952
ü	ü	6.076	u	5.088	e	4.136	i	3.983	y	3.776
v	v	3.899	ɾ	3.166	o	2.841	ʌ	2.799	ɾ̃	2.772
w	w	4.276	o	3.286	u	2.981	ɔ	2.930	m	2.889
y	y	5.154	i	4.015	e	3.844	ü	3.776	u	3.759
z	z	4.592	s	4.265	š	2.369	□	2.055	ɨ	1.970

Table 2.1: Estimates of the top 5 phones which display the most within-speaker correlation with the set of selected phones listed.

2.3 Speaker Property Correlations

2.3.1 Basic Ideas

Section 2.2 demonstrated how the acoustic features of different phones can contain significant within-speaker correlation. This correlation can be attributed to a number of factors such as the physical characteristics of the speaker's vocal tract, the speaker's speaking rate, and the speaker's regional dialect. Knowledge of some of these *speaker properties* can be very useful in reducing the variance of the models used during recognition such that the models more closely match the characteristics of the current speaker. Two of the most significant properties are the speaker's gender and speaking rate. These two properties each have a significant effect on the acoustic realization of the speech that is being produced. Both properties are also relatively easy to estimate from a typical speech utterance. This section will examine the effects of gender and speaking rate on the acoustic realization of speech.

2.3.2 Gender Correlations

The largest factor in determining the physical dimension of a person's vocal tract is the person's gender. On average a female vocal tract is 15% shorter than a male vocal tract, with a majority of the difference occurring in the length of the pharynx [21, 46]. This results in vowel formant locations which are measurably different between male and female speakers. As a result acoustic models trained on only male speakers are distinctly different than acoustic models trained on only female speakers. Figure 2.4 shows the male and female models for three different vowels ([i],[e], and [o]). These plots show the first two dimensions of 36 dimensional models trained using a mixture of diagonal Gaussian density function. The contours represent the equal likelihood contour corresponding to $\frac{7}{10}$ of the peak likelihood for each model. Distinct differences between the male and female models are obvious.

2.3.3 Speaking Rate Correlations

Speaking rate is a speech property whose effect is observable in multiple different models including the duration model, the acoustic model, and the pronunciation model. It is obvious that increasing the rate of speech will have an effect on the duration of the phones in an utterance, however the speaking rate also directly influences the manner in which people articulate and the phonological variations of the words they produce. Each of these phenomena will be discussed below.

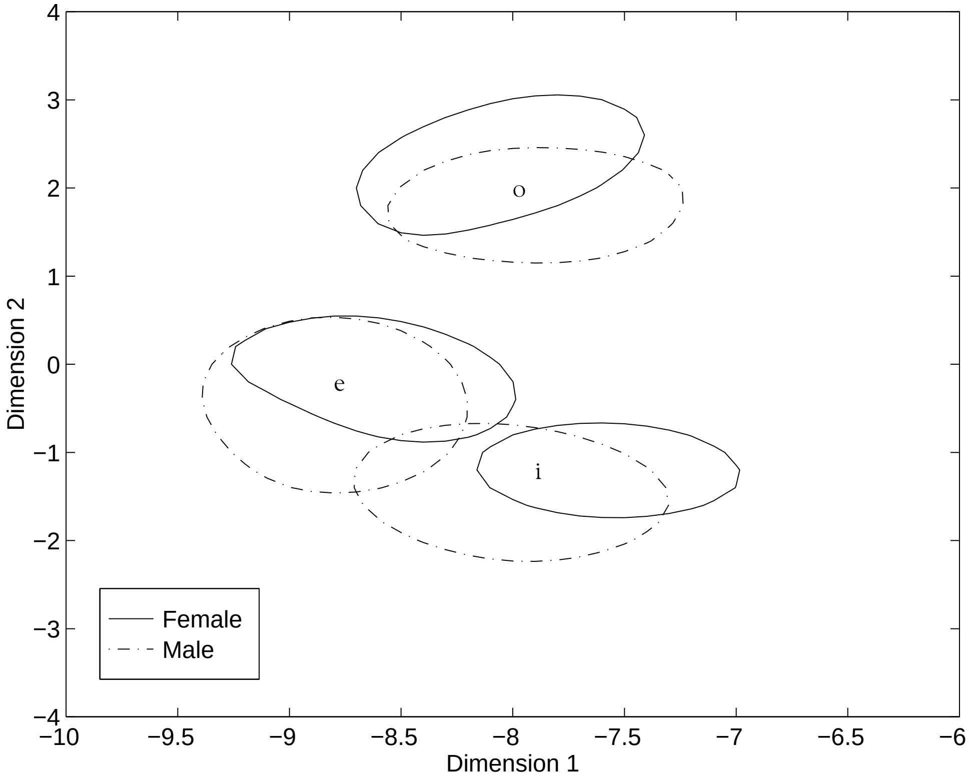


Figure 2.4: Plot of high likelihood regions for two dimensions of female and male models for three different vowels.

Definition of Speaking Rate

In order to investigate the effects of speaking rate, a mathematical definition of the speaking rate of an utterance must be defined. The speaking rate measure used in the thesis was calculated on a segment by segment basis and then averaged over all segments in an utterance. If an example phone (having the phonetic label p) has a duration of d, then the segment-based speaking rate r for this phone is calculated using the following equation:

$$r = \frac{\mu_p - d}{\sigma_p} \tag{2.14}$$

In this equation, μ_p is the speaker independent average of the duration of phone p, and σ_p is the speaker independent standard deviation of the duration of phone p. This measure transforms the duration of a segment into a variance-normalized, zero-centered value. By zero-centering the speaking rate measure with the average duration of the phone, all of the segments are easily classified as *fast* ($r > 0$) or *slow* ($r < 0$) based solely on whether their segment speaking rate is positive or negative. By normalizing with the speaker independent standard deviation each segment's speaking rate can be compared on an equal scale with all other segments.

Suppose an utterance has N phones. Let the phone labels be represented as:

$$P = \{p_1, p_2, \ldots, p_N\} \tag{2.15}$$

The the sequence of durations corresponding to these phones be represented as:

$$D = \{d_1, d_2, \ldots, d_N\} \tag{2.16}$$

Using these definitions, the average speaking rate $\bar{r}$ across all segments in the utterance is defined as:

$$\bar{r} = \frac{1}{N} \sum_{n=1}^{N} r_n = \frac{1}{N} \sum_{n=1}^{N} \frac{\mu_{p_n} - d_n}{\sigma_{p_n}} \tag{2.17}$$

Using the speaking rate definition provided above, the average speaking rate of the 109 speakers in the RM train and development sets was calculated using all available training utterances from each speaker. To avoid problems caused by long silences or pauses, only segments containing linguistically relevant phonetic units were used to calculate the speaking rate. The average speaking rates for individual training speakers varied from .36 at the fast end to -.49 at the slow end. A speaking rate between .15 and -.15 was utilized by more than half of the speakers. A speaking rate between .10 and -.10 was utilized by 36% of the training speakers.

Effect on Duration Model

One item of interest that can be examined is the relationship between speaking rate and the duration of particular phones. One means of investigating this is to examine the correlation between the average speaking rate of individuals and the average duration of particular phones uttered by these individuals. To do this for any particular phone, a joint vector is created for each training speaker. This joint vector contains the speaking rate for that speaker and the average duration of all examples of the particular phone of interest that were uttered by that speaker. The correlation between the average speaking rate and the average duration of the phone was calculated from the joint vectors collected over all 109 speakers in the RM corpus. These correlations are shown for a selected set of phones in Table 2.2.

Observation of Table 2.2 reveals many expected results. First, the set of 12 phones whose average SD duration is most correlated with the speaker's speaking rate contains 10 vowels. The two non-vowels in this set are the strong fricatives [s] and [š]. This is expected because these are the phones whose perception is predominantly determined through their spectral content. Most of these sounds can experience large variations in their duration without affecting their perception by human listeners. On the other hand, the bottom of the table's list contains phones that require a specific set of dynamic motions whose absolute timing is important in order for the phone to be perceived correctly. Thus, the phones whose duration is least correlated with the speaking rate include stop releases and semivowels. In particular, three of the four phones at the very bottom of the list are voiced stops releases.

Effect on Acoustic Model

The speaking rate not only has an effect on the duration of phones but also on their full acoustic realization. For example, as speaking rate increases a person is less likely to achieve the extreme articulatory positions when producing vowels. As a result formants move towards more central or *laxer* position when speaking rate is increased [50]. Overall, an increase in speaking rate generally results in speech which is more relaxed and less carefully articulated than speech which is spoken slower [15]. As a result, distinct acoustic differences in the acoustic models of fast and slow speakers can be observed. Figure 2.5 shows the high likelihood region contours for models from fast and slow speakers. In this case fast speakers are defined as speakers with an average speaking rate greater than zero while slow speakers have a speaking rate less than zero. For each of the four phones shown, distinct differences between the fast and slow speakers are observable. In particular, the first dimension's feature appears to be positively correlated with speaking rate.

Phone	Correlation with Speaking Rate
ɑ	0.867036
i	0.861611
ɪ	0.843518
ʌ	0.836272
ə	0.828570
e	0.810271
ɛ	0.802337
æ	0.793887
s	0.784364
o	0.782406
ɝ	0.768465
ɑʸ	0.767269
š	0.735199
k□	0.717915
m	0.683802
ᵻ	0.663470
l	0.660407
z	0.649163
ɔ	0.639630
n	0.624609
u	0.596110
t□	0.536246
d□	0.528784
ɾ̃	0.504753
t	0.464951
ŋ	0.459353
p□	0.435856
v	0.415123
w	0.395703
k	0.395109
y	0.389849
ü	0.321752
f	0.304044
b□	0.277442
r	0.274498
g□	0.267334
p	0.249298
□	0.241992
b	0.090592
d	0.085704
ʔ	0.017234
g	0.003755

Table 2.2: Estimates of the correlation between the SD average duration of a phone and the speaker's average speaking rate.

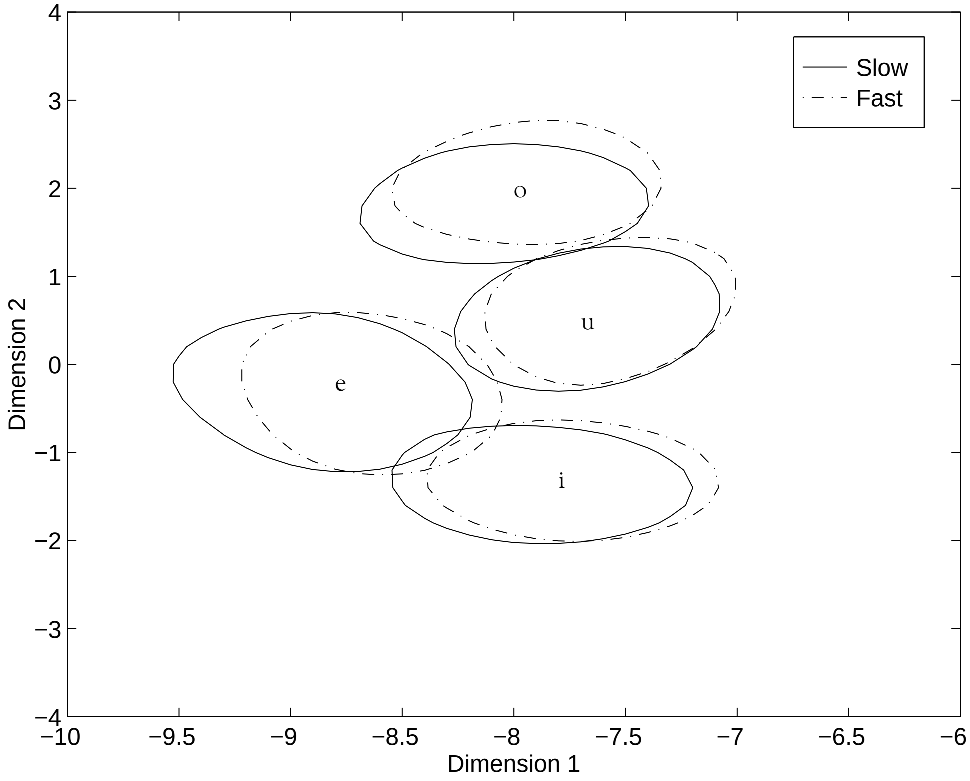


Figure 2.5: Plot of high likelihood regions for acoustic models of fast and slow speakers for four different phones.

Effect on Pronunciation Model

In addition to the acoustic differences that result between speech spoken at different speaking rates, the rate in which different phonological rules are applied is also affected by the speaking rate. As the speaking rate is increased speakers tend to produce speech which is more *casual* in nature, tending towards pronunciations which are reduced or under-specified versions of the underlying form [15]. Consider the following phonological rules as expressed in rewrite form:

1. stə → sə

2. šən → šn̩

3. i → ɪ

4. Vdə → Vɾə -or- Vtə → Vɾə where V represents any vowel.

The first rule occurs when the [stə] cluster is reduced to [sə] because the user does not produce the closure that typically precedes precedes the [t]. For example, the word *distance* is typically pronounced [dɪst$^{\square}$təns]. However, if the speaker does not articulate the [t$^{\square}$] clearly, the word might be pronounced as [dɪsəns] instead.

The second rule often occurs in words containing the syllable "tion". The rule simply states that the sequence [ən] can be realized as the syllabic [n̩] when preceded by a [š]. For example, word "position" would be realized as [pəzɪšn̩] instead of as [pəzɪšən].

The third rule occurs when the phone [i] occurs in an unstressed environment. In this case the speaker could substitute the lax phone [ɪ] in its place. This occurs often in words beginning with the unstressed prefixes "re-" and "de-". For example, the word *define* can be pronounced as either [difɑ^{y}n] or as [dɪfɑ^{y}n].

The last rule occurs when an intervocalic [t] or [d] can be realized as a flap. For example, the word "edit" can be pronounced as either [ɛd$^{\square}$dət$^{\square}$t] or as [ɛɾət$^{\square}$t].

To examine the relationship between speaking rate and phonological variation, the probability of the phonological rules stated above can be examined as the speaking rate is varied. Table 2.3 shows the probability of each of the four specified rules firing under different speaking rate considerations. The speakers in the training set were classified as either fast speakers ($\bar{r} > 0.1$), medium speakers ($0.1 > \bar{r} > -0.1$), or slow speakers ($\bar{r} < -0.1$). The probability of the different rules firing was based on the occurrence of the rules firing when the forced phonetic path was automatically generated for each utterance in the training set.

In examining Table 2.3 some expected results are encountered. In the case of Rules 1 through 3, the probability of each rule firing increases as the speaking rate increases. This is expected because each of these rules simplifies the articulatory

	SI	Speaking Rate Models		
Rule	Model	Slow	Medium	Fast
1	.164	.120	.134	.273
2	.371	.269	.362	.497
3	.632	.599	.616	.702
4	.709	.750	.737	.642

Table 2.3: Probability of four different phonological rules firing under different speaking conditions.

motions necessary to complete the phonetic sequence. This simplification or reduction of articulatory motion is necessary in order for the speaker to effectively increase the rate at which the words are spoken. On the other hand, the probability of the fourth rule firing (i.e., flapping of the phones [d] and [t]) decreases as the speaking rate increases. This seem counterintuitive since flapping is also a means of reducing the articulation time of a phone. In order to explain this phenomenon, an examination of the specific words which were affected by this rule was conducted. This examination revealed that many of the examples of the phones [d] and [t] in the fast speech which were not labeled as flaps in the forced path transcription (which is provided by SUMMIT as discussed in Appendix C) were actually produced in a rapid flap-like fashion. However, these same exemplars also contained short burst-like artifacts. Thus, as the speaking rate was increased, and the articulators such as the tongue were forced to move quicker, the flap motion of the tongue was more likely to produce short click-like bursts which the recognizer classified as standard stop releases.

2.4 Summary

The chapter has presented several analyses of the within-speaker correlations which exist in the speech signal. Each of these analyses produced information which can be used in the development of speech recognition algorithms which intend to capture speaker correlation information. Though only a few analysis techniques were presented, and some of the printed results are largely anecdotal, this chapter hopefully provides a succinct description of the types of correlations which exist and which are not always taken advantage of by speech recognition systems.

Chapter 3

Principles of Speaker Adaptation

3.1 Overview

To date, speaker correlation information has primarily been applied to the task of acoustic model speaker adaptation. The goal of speaker adaptation is to adjust the density functions used by the acoustic model to match the current speaker as closely as possible using whatever adaptation data is available. In this sense, speaker adaptation is simply an estimation problem. During adaptation a set of *adaptation parameters* must be learned. The purpose of the adaptation parameters is to capture relevant information about the acoustic properties of the current speaker. These adaptation parameters are then used during the construction of the speaker adapted density functions used during recognition. By utilizing some *a priori* knowledge about the statistical properties and correlations of these adaptation parameters, the underlying speaker dependent model can be learned more rapidly, i.e. with less adaptation data.

Before being able to incorporate speaker correlation into speaker adaptation routines, it is important to understand the mathematical frameworks used for adaptation and the past adaptation techniques which have been proposed. Thus, the goals of this chapter are as follows:

- Present the mathematical framework and underlying tenets of the speaker adaptation problem.

- Empirically demonstrate some of the basic ideas of speaker adaptation on a word recognition task.

- Discuss the important past and present approaches to the adaptation problem.

3.2 Mathematical Framework for Adaptation

3.2.1 Adaptation Styles

The first step in defining the adaptation problem is determining the context in which adaptation will be performed. Because different applications have different requirements, there are several different styles of adaptation. The different manners in which adaptation is applied can be described using the following three descriptors:

- supervised or unsupervised
- enrolled or instantaneous
- batch or on-line

In supervised adaptation, the words spoken in the adaptation utterances are known. In unsupervised adaptation the words that were spoken in the adaptation data are not available. In enrolled adaptation, the set of adaptation data is recorded ahead of time and used to adapt the models prior to their use on unseen data. In instantaneous adaptation, the system must adapt its models on the same data that it is trying to recognize. In batch adaptation the system has all of its adaptation data available to it when adaptation is being performed. In on-line adaptation, the system is allowed to use each new utterance that is presented to it for adaptation, but the system is not able to reexamine old utterances, i.e., after each utterance is processed it is then forgotten.

Of the different styles, the easiest to perform is supervised, enrolled, batch adaptation. This is the approach taken by many speaker dependent dictation applications [18, 42]. For these tasks, the time spent recording the user's speech on predetermined sentences is small compared to the many hours that the system will be used by the specific individual. Thus, the time needed to provide the initial set of adaptation data is viewed as a worthwhile investment.

On the other hand, there are many applications, such as the Jupiter weather information server [89], where a user interacts with the system for only a few utterances. For applications such as these, it is not practical for the user to provide enrollment data for the purpose of adaptation. In this case the most difficult style of adaptation must be utilized: unsupervised, instantaneous, on-line adaptation.

For the rest of this chapter, it will be assumed that supervised, enrolled, batch adaptation is being performed. It should be noted that the main principles of adaptation remain the same no matter what style of adaptation is used. Chapter 7 will discuss the engineering issues involved in performing unsupervised, instantaneous adaptation.

3.2.2 Stochastic Representation

Acoustic Model Density Functions

The first step in creating a probabilistic framework for speaker adaptation is developing an understanding of the stochastic modeling that is performed. In standard speech recognition systems, probabilistic acoustic models are created to predict the likelihood of an acoustic observation given its underlying phonetic class. Let the likelihood of an acoustic feature vector $\vec{x}$ for some arbitrary phonetic class be represented as $\mathrm{p}(\vec{x})$. Typically, this likelihood function is modeled using a parametric or semi-parametric probability density function.

For speaker independent recognition, the parameters of a model are usually trained using all available data from all available speakers. Once the model is trained, these parameters remain fixed. However, if the model's parameters are trained using only the data from one speaker (i.e., in speaker dependent mode) the parameters could be quite varied from speaker to speaker. Let θ represent the SD parameters of the observation density function for one particular speaker. The parameters in θ can be different for every speaker. If a particular speaker's parameters, θ, are known, the SD density function can be represented as $\mathrm{p}(\vec{x}|\theta)$.

Now, suppose that the speaker is unknown. In this case, θ is unknown and can be treated as a set of random variables which are generated by a separate random process, $\mathrm{p}(\theta)$. Viewed in this light, the density function for the acoustic observation $\vec{x}$ is doubly stochastic since $\vec{x}$ is a random variable which is dependent on the model parameters θ of the current speaker, which are themselves generated by a random process. If the distribution $\mathrm{p}(\theta)$ is known then the speaker independent density function for $\vec{x}$ can be represented by the following integral:

$$\mathrm{p}_{si}(\vec{x}) = \int_\theta \mathrm{p}(\vec{x}|\theta)\ \mathrm{p}(\theta)\ \mathrm{d}\theta \tag{3.1}$$

In practice, the distribution of $\mathrm{p}(\theta)$ is not known and $\mathrm{p}_{si}(\vec{x})$ is estimated directly from training data pooled from many speakers. Generating maximum likelihood estimates of the parameters of a speaker independent density function directly from the pooled set of training observations is the most common approach. However, $\mathrm{p}_{si}(\vec{x})$ could also be estimated by creating a mixture model from SD models. For example, suppose L different training speakers are available and each training speaker l has a unique set of parameters, θ_l. By assigning equal weight to each training speaker, the SI density function can be estimated using the following mixture model expression:

$$\mathrm{p}_{si}(\vec{x}) \approx \sum_{l=1}^{L} \frac{1}{L} \mathrm{p}(\vec{x}|\theta_l) \tag{3.2}$$

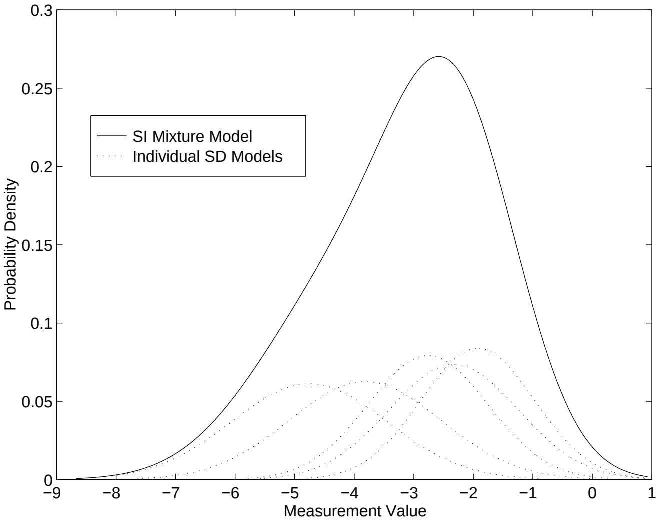


Figure 3.1: Plot showing an SI mixture model for the phone [i] created from five different SD models, along with each of the five SD models all scaled by $\frac{1}{5}$.

Figure 3.1 shows an example of a SI mixture density function created from 5 SD Gaussian density functions. The models use actual observations of a particular measurement extracted from example segments of the phone [i]. As can be seen the SD models show a large variation in their mean values and a relatively small amount of overlap between some pairs of the speakers. This results in an SI model with a noticeably higher variance then the individual SD models from which it was created.

SI vs. SD Classification

Figure 3.1 illustrates how specificity is lost when the SI model is used instead of an SD model. However, the loss in specificity is only a problem if it harms multi-class classification performance. In order to examine this problem, let us first expand upon our notation. For an M-class classification problem, let the full set of M different acoustic models be represented as:

$$\Theta = \{\theta_1, \theta_2, \ldots, \theta_M\} \tag{3.3}$$

The goal in classification is to identify the underlying phonetic class of an observation vector. The *a posteriori* probability that a particular observation vector $\vec{x}$ belongs to a particular phonetic class p given a particular set of models Θ is represented as $\text{p}(p|\vec{x}, \Theta)$. The Bayes minimum error decision rule for determining the most likely

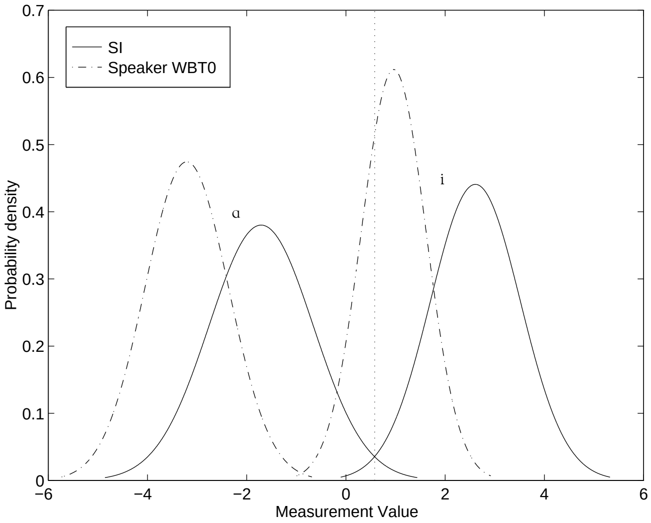


Figure 3.2: Comparison of density functions for one measurement from the phones [ɑ] and [i] for the SI model set and for speaker WBT0's model set.

phone hypothesis, p' is represented by the expression:

$$p' = \arg\max_p \mathrm{p}(p|\vec{x},\Theta) = \arg\max_p \mathrm{p}(\vec{x}|p,\Theta)\mathrm{p}(p|\Theta) \tag{3.4}$$

To demonstrate the potential deficiencies of the SI model set, consider the two-class classification problem presented in the plot in Figure 3.2. In this figure the distribution of a generic measurement is shown for the phones [ɑ] and [i] for both the SI model set (created from 149 training speakers in the RM corpus) and the SD model set for speaker WBT0. Using the expression in (3.4) a test token can be classified as either an [ɑ] or an [i] given the token's acoustic measurement. When using the SI model set, the Bayes minimum error decision boundary (assuming equal *a priori* probabilities for [ɑ] and [i]) is shown with the vertical dotted line. The probability of making an error given a random observation drawn from a randomly selected speaker is calculated by integrating the area of the regions of the SI [i] and [ɑ] density functions which fall on the wrong side of the decision boundary.

Now consider using the SI model set on an observation known to have been generated by WBT0. The probability of making an error in this case increases dramatically because an inordinately large number of WBT0's [i] tokens could fall on the wrong side of the SI decision boundary (even though practically all of WBT0's [ɑ] tokens should be classified correctly). However, if WBT0's SD model set is used for Θ in (3.4), instead of the SI model set, then the SD classification error for WBT0 will be considerably less than the SI error rate across all speakers.

Adaptation Notation

Figure 3.2 is one example demonstrating the potential improvements that can be made from utilizing the SD model parameters for each class. Unfortunately, if the speaker is unknown the parameters of Θ are also unknown. Similarly, if the speaker is known, but only a limited amount of training (or adaptation) data from that speaker has been observed then a standard maximum likelihood estimates for the parameters in Θ are likely to be insufficiently trained. Thus, the goal of a speaker adaptation algorithm is to create a speaker adapted set of parameters which, during recognition, performs as accurately as possible for whatever amount of adaptation data is available.

The first step in describing a framework for adaptation is to define the standard notation that will be used. To begin let $\mathcal{A}$ represent the set of acoustic observations contained in the adaptation data. To avoid confusion $\mathcal{A}$ will be used to represent previously seen adaptation data while X will be used to represent unseen test data. Let the adaptation set $\mathcal{A}$ be subdivided as follows:

$$\mathcal{A} = \{A_1, A_2, \ldots, A_M\} \tag{3.5}$$

Here, M is the total number of phonetic classes for which an acoustic model exists. Each A_m represents the set of observed acoustic feature vectors belonging to the specific phonetic class m. Furthermore let each A_m be represented as:

$$A_m = \{\vec{a}_{m,1}, \vec{a}_{m,2}, \ldots, \vec{a}_{m,N_m}\} \tag{3.6}$$

Here, each $\vec{a}_{m,n}$ is an independent observation from the m^{th} phonetic class. Note that N_m could be zero if no observations from the m^{th} class have been observed in the adaptation data. Next, let Θ represent the set of density functions covering the entire inventory of phonetic units. Given M different phonetic units, Θ is represented as:

$$\Theta = \{\theta_1, \theta_2, \ldots, \theta_M\} \tag{3.7}$$

Now consider the task of adapting the full set of model parameters in Θ given the full set of adaptation data $\mathcal{A}$. Two common estimation techniques for finding Θ are maximum likelihood (ML) estimation and maximum *a posteriori* probability (MAP) estimation. The general expression for finding the ML estimate Θ^{ml} is given by:

$$\Theta^{ml} = \arg\max_{\Theta} \mathrm{p}(\mathcal{A}|\Theta) \tag{3.8}$$

The general expression for finding the MAP estimate Θ^{map} is given by:

$$\Theta^{map} = \arg\max_{\Theta} \mathrm{p}(\Theta|\mathcal{A}) = \arg\max_{\Theta} \mathrm{p}(\mathcal{A}|\Theta)\mathrm{p}(\Theta) \tag{3.9}$$

Practical considerations of both estimation techniques will be examined next and empirical results of adaptation algorithms using these techniques will be presented later in this chapter.

3.2.3 ML Estimation

Maximum likelihood estimation, as expressed in Equation 3.8 is the primary method used to estimate the parameters of SI models. The goal is to find the set of model parameters which best describes the training data. ML estimation makes no assumptions about the likelihood of any particular set of parameters being the true underlying set of parameters for the system. ML estimation is typically used because it will yield the theoretically optimal set of models for Bayesian classification when given a sufficient amount of data. However, ML estimation could result in poorly estimated models if an insufficient amount of data is available to provide accurate estimation of the model parameters or the model is overly restrictive.

During ML estimation, the adaptation observations are treated as independent. This allows Equation 3.8 to be rewritten as:

$$\Theta^{ml} = \arg\max_{\Theta} \prod_{m=1}^{M} \mathrm{p}(A_m|\Theta) = \arg\max_{\Theta} \prod_{m=1}^{M} \prod_{n=1}^{N_m} \mathrm{p}(\vec{a}_{m,n}|\Theta) \tag{3.10}$$

Because the likelihood of an observation is dependent only on the model parameters of its own class and not on the model parameters of other classes, the expression can be simplified as follows:

$$\Theta^{ml} = \arg\max_{\Theta} \prod_{m=1}^{M} \prod_{n=1}^{N_m} \mathrm{p}(\vec{a}_{m,n}|\theta_m) \tag{3.11}$$

Because each observation is dependent only on the model parameters of its own class, the model parameters of each class can be estimated independently of the other classes. The estimate for the parameters of any particular model θ_m is given by the following expression:

$$\theta_m^{ml} = \arg\max_{\theta_m} \prod_{n=1}^{N_m} \mathrm{p}(\vec{a}_{m,n}|\theta_m) \tag{3.12}$$

One particularly troublesome aspect of ML estimation is that it is unable to provide estimates for a model's parameters if no observations from that model's phonetic class have been observed. This is because ML estimation makes no presumptions about the *a priori* likelihood of the model parameters' possible values. In effect, ML estimation assumes that all possible model parameter estimates are equally likely to occur.

3.2.4 MAP Estimation

During SI training, ML estimation is typically used because no sufficient model describing the likelihood of the underlying model parameters is available. However, for speaker dependent training, a model for the likelihood of the underlying parameters for a new speaker's SD model might be approximated from examination of the SD models of speakers in the training set. This makes MAP estimation a viable approach for adaptation.

Like ML estimation, MAP estimation assumes that all observations are independent. This allows Equation 3.9 to be rewritten as:

$$\Theta^{map} = \arg\max_{\Theta} \prod_{m=1}^{M} \mathrm{p}(A_m|\Theta)\mathrm{p}(\Theta) = \arg\max_{\Theta} \prod_{m=1}^{M} \prod_{n=1}^{N_m} \mathrm{p}(\vec{a}_{m,n}|\Theta)\mathrm{p}(\Theta) \tag{3.13}$$

As with ML estimation, the likelihood of an observation is dependent only on the model parameters of its own class. This lets the expression be simplified to:

$$\Theta^{map} = \arg\max_{\Theta} \mathrm{p}(\Theta) \prod_{m=1}^{M} \prod_{n=1}^{N_m} \mathrm{p}(\vec{a}_{m,n}|\theta_m) \tag{3.14}$$

At this point, the standard MAP estimation routine used for speaker adaptation assumes that the model parameters for each class are independent from the model parameters of all other classes. This assumption thereby ignores speaker correlation information which exists between the model parameters of different classes. However, this assumption also simplifies the problem so that the estimation of a set of model parameters is performed using only the adaptation observations from its own class. This allows the standard MAP expression for any particular model θ_m to be written as:

$$\theta_m^{map} = \arg\max_{\theta_m} \mathrm{p}(\theta_m) \prod_{n=1}^{N_m} \mathrm{p}(\vec{a}_{m,n}|\theta_m) \tag{3.15}$$

If the model parameters from different classes are not considered independent, as in Equation (3.15), and the speaker correlations between the model parameters are accounted for in the model p(Θ), the estimation method is referred to as extended maximum *a posteriori* probability (EMAP) estimation. This type of estimation will be discussed in more detail in Section 3.4.2.

3.2.5 Adaptation Parameters

Because adaptation is essentially just an estimation problem, many of the tenets of training standard SI acoustic models apply to the adaptation problem as well. One key principle is the notion that models with more parameters require more training data to achieve reliable estimates than models with fewer parameters. As such, it can not be expected that a small amount of adaptation data can be used to provide accurate estimates for a large number of model parameters. For this reason, adaptation algorithms seldom try to estimate all of the parameters of an acoustic model. Instead, some smaller set of *adaptation parameters* is often estimated. The full set of parameters is then generated from some function which utilizes these adaptation parameters.

To express this idea mathematically, let Λ be a set of adaptation parameters. The full set of speaker adapted model parameters can be expressed as a function of Λ as follows:

$$\Theta = f(\Lambda) \tag{3.16}$$

Using this idea, the general form of the ML estimate of Λ, given a set of adaptation data $\mathcal{A}$, is expressed as:

$$\Lambda^{ml} = \arg\max_{\Lambda} \mathrm{p}(\mathcal{A}|\Theta) = \arg\max_{\Lambda} \mathrm{p}(\mathcal{A}|f(\Lambda)) \tag{3.17}$$

Similarly, the general form of the MAP estimate of Λ is expressed as:

$$\Lambda^{map} = \arg\max_{\Lambda} \mathrm{p}(\mathcal{A}|\Theta)\mathrm{p}(\Theta) = \arg\max_{\Lambda} \mathrm{p}(\mathcal{A}|f(\Lambda))\mathrm{p}(\Lambda) \tag{3.18}$$

As an example of this idea, the simplest form of speaker adaptation is gender dependent modeling. In this case, Λ consists of only one binary variable, which will be represented as λ. If the value of λ is set to 1 for male speakers and to 0 for female speakers, then the function for determining the speaker adapted model set, Θ^{sa}, is written as:

$$\Theta^{sa} = f(\lambda) = \lambda * \Theta^{male} + (1 - \lambda) * \Theta^{female} \tag{3.19}$$

In the case of ML estimation, λ would simply be found by choosing the model set, Θ^{male} or Θ^{female}, which gives the adaptation data the highest likelihood.

One of the arts of adaptation is the ability to chose an appropriate set of adaptation parameters. There are two main issues to consider when choosing these parameters. First, the number of parameters in the set should be suitable for the amount of available adaptation data. As the number of adaptation utterances increases, the size of Λ can increase accordingly. Optimally, Λ will be chosen to provide as much detail as possible while still be being small enough in size to be reliably estimated. Second, it is important for Λ to efficiently account for the possible speaker variability.

In other words, Λ should be chosen in such a way that it can encode a large portion of the speaker dependent characteristics of the current speaker with as small a set of parameters as possible. If this is done properly, fewer utterances will be needed in order for the system to approach the performance achieved by well-trained speaker dependent models.

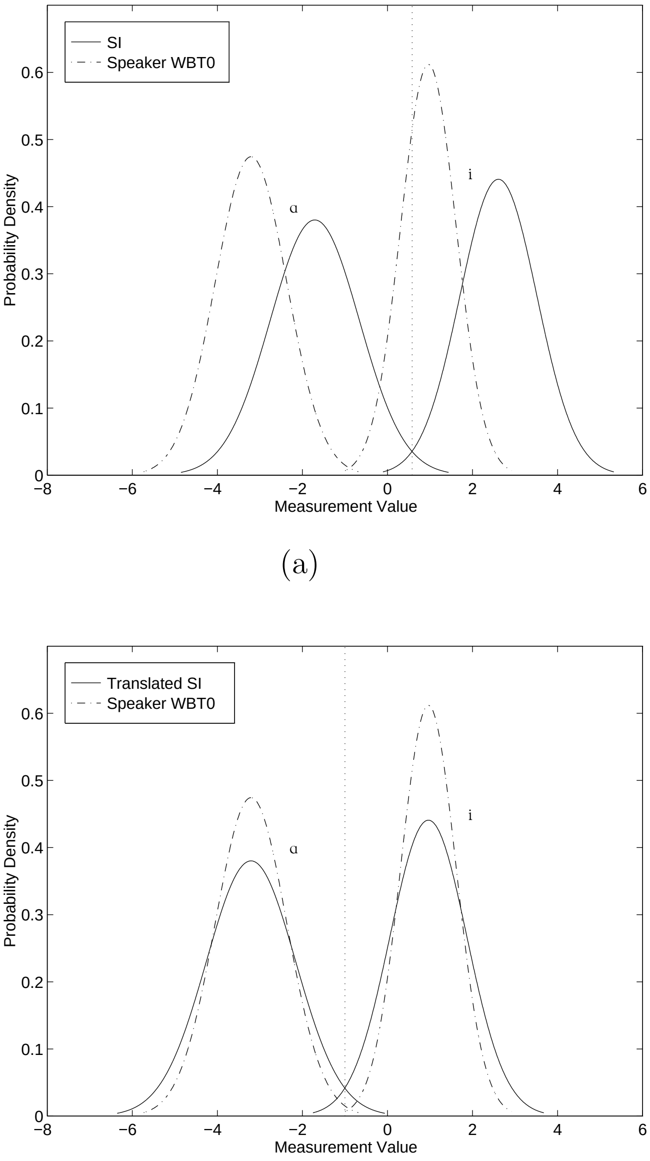
In addition to the two issues presented above, a third issue arises if the adaptation is being performed in an unsupervised fashion. In this case the adaptation parameters must also have the property that they can be estimated robustly in the face of the uncertainty of the underlying word string. Thus, parameters which require that the exact transcription be known may be useful for supervised adaptation but may not be suitable for unsupervised adaptation when the transcription of the utterance is not known *a priori*.

To provide some examples, the following list contains some of the possible adaptation parameters that could be learned or estimated:

- Gender
- Regional dialect
- Speaking rate
- Similarity to reference speakers or speaker clusters
- ML estimated mean vectors of acoustic observations
- ML estimated mixture Gaussian model parameters

Each one of these type of parameters would be utilized in a different fashion but each could contribute information which would help adapt the system's models to match the current speaker.

To provide an example of how a set of adaptation parameters which is smaller than the full set of acoustic model density function parameters can be effective for speaker adaptation consider the classification task illustrated in Figure 3.2 (and repeated in Figure 3.3). This figure shows how much disparity can exist between the set of SI density functions and any random set of SD density functions. However, in this particular problem it not vitally important that the exact SD density functions be learned. Instead, the classification performance is most dependent on learning the optimal classification decision boundary. Consider what happens if only the SI density functions' mean values are adapted to the current speaker but not the variances. This is demonstrated in Figure 3.3. In this figure, the SI models from Figure 3.2 are shifted such that their mean values are the same as the SD mean values for speaker WBT0 but the original SI variances are maintained. By adjusting only the mean vectors the optimal SD error rate is almost fully achieved.



(a)

(b)

Figure 3.3: Gaussian models for a one dimensional measurement of the phones [i] and [ɑ] for the training speaker WBT0 as compared to (a) the original SI models and (b) the translated SI models where the SI mean values have been shifted to match speaker WBT0's mean values.

3.3 Experiments Using Basic Adaptation Methods

3.3.1 Experimental Setup

This section demonstrates some of the principles of speaker adaptation with a series of experiments utilizing basic adaptation techniques. For all of the experiments used in this chapter, the recognition was performed by the SUMMIT system. The corpus used for the experiments was the RM corpus. This corpus contains a data set specifically designed for speaker dependent and speaker adaptive experiments. Only a word pair grammar is used for these experiments thus forcing the acoustic model to provide the bulk of the responsibility for the recognition task. The baseline SI recognizer for these experiments was trained on the 149 speakers in the SI set. The 12 speakers in the SD set were used for adaptation and testing. For each of the 12 test speakers, the adaptation data was extracted from the speaker's 600 utterance training set and the speaker's 100 utterance development set was used for testing.

The experiments presented here are all conducted on the task of supervised, enrolled, batch speaker adaptation. In other words, speaker adapted models were created from a set of adaptation utterances for which a transcription of the underlying word string was provided. In this case, the forced paths generated by the baseline SI models are used to provide the aligned transcriptions.

The focus of these experiments is to examine the performance of several basic adaptation routines as the amount of adaptation data that is used is varied. In these experiments, speaker adaptation performance is examined when the number of available adaptation utterances is varied from 1 to 600. The specific quantities of adaptation utterances that were tested were 1, 3, 5, 10, 20, 50, 100, 300, and 600.

Because the quality of the adaptation can vary depending on the vocabulary and phonetic content of the adaptation utterances, it is important to run multiple adaptation trials for each specific quantity of adaptation data that is being investigated. The more adaptation trials that are available the better the estimate of the adaptation routine's expected performance will be. To create the adaptation sets, utterances were randomly chosen from the 600 available utterances from each speaker. Table 3.1 shows the number of randomly chosen adaptation sets created for each speaker for

Number of adaptation utterances	1	3	5	10	20	50	100	300	600
Number of randomly chosen sets	5	5	5	3	3	2	2	1	1

Table 3.1: The number of randomly chosen sets of adaptation utterances used to estimate the word error rate for each speaker given a set number of adaptation utterances.

each number of adaptation utterances. No utterance was used for more than one trial for any particular number of adaptation utterances (i.e., there was no intersection in the sets used for different trials). As the size of the adaptation sets increases, the variance in the distribution of the phonetic content across different trials decreases. This allows fewer trials to be necessary when the number of adaptation utterances increases.

To evaluate the performance of a speaker adapted model, the model is tested on the 100 utterances in the development set of that individual speaker. The experiments use average word error rate as the evaluation metric. The average word error rate for each speaker for a particular number of adaptation utterances is calculated by averaging the word error rates of each individual adaptation trial. The average word error across all twelve speakers is computed by simply averaging the average word error rates of the twelve speakers. Thus, each speaker contributes equally to the average word error rate even though some some speakers recited more words in their development set than others.

The word error rate evaluation metric was chosen so as to maintain consistency with past research efforts conducted using this corpus. It can be argued that the phone error rate measure would provide a more revealing picture of the capabilities of the acoustic model component of a system. A discussion of the choice of the corpus and evaluation metric used for the experiments in this thesis is presented in Chapter 8.

3.3.2 SI Performance

As a baseline against which all other experiments are compared, an SI model set was created. These models were trained on the utterances from the 149 speakers in the SI portion of the corpus. These 149 speakers form a set which is disjoint from the 12 speakers from the SD portion of the corpus used for these evaluations. The SI corpus was used to train the acoustic model set and the pronunciation network weights. In all of the experiments in this chapter, the fixed SI pronunciation network is utilized. Thus, adaptation is only performed on the acoustic model set.

The speaker independent acoustic models were created using standard mixtures of diagonal Gaussians. The number of Gaussian components used per mixture was dependent on the amount of training data available for each phonetic model. A maximum of 120 Gaussians per mixture was allowed for all 60 phonetic models as well as the anti-phone model. The complete set of SI models used a total of 5,439 mixture components each containing 73 different parameters for a grand total of 397,047 model parameters. The average word error rate of the SI models when tested across all 12 speakers in the SD development set was 7.4%.

3.3.3 ML Estimation

The first step in examining speaker adaptation performance is to examine the system when it is trained using standard ML estimation techniques. In this experiment the number of adaptation utterances was varied from 1 to 600 as described in Section 3.3.1. When the size of the adaptation data set was small the system often did not encounter the minimum of two examples needed to created a single Gaussian component for some of the phones. In these cases the system borrowed the equivalent SI model as a substitute model. As such, the performance for small amounts of data is not a good indication of the true ML performance since many SI models may be used to fill in the holes left by the lack of available SD data.

Figure 3.4 shows the performance of ML training as the number of adaptation utterances was varied from 1 to 600. With 600 adaptation utterances, the speaker dependent system achieved an average error rate of 3.7%. Thus, the error rate was cut in half when the system was trained using the full adaptation set for each speaker. Additionally, the speaker dependent systems trained using the full 600 adaptation utterances only utilized 1135 mixture components on average. This resulted in an average parameter set size of 82,855 which is only one fifth of the size of the speaker independent model set. Unfortunately, the SD trained models did not outperform the SI models when less than 300 adaptation utterances were used.

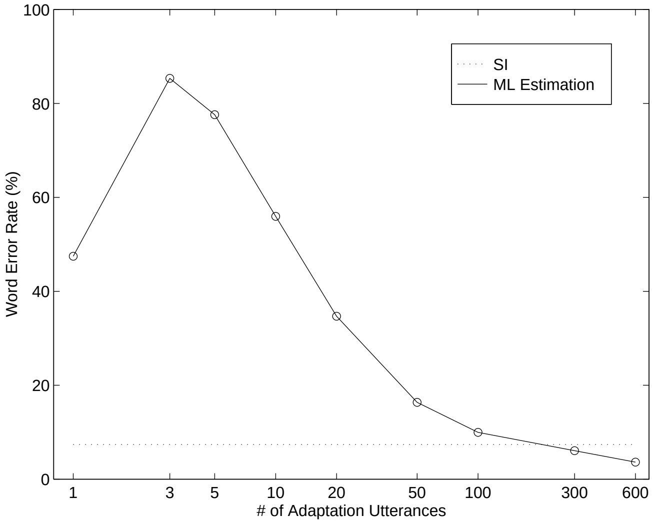


Figure 3.4: Performance of ML trained SD models as the number of adaptation utterances is varied from 1 to 600.

3.3.4 Density Function Interpolation

ML estimation is a sub-optimal approach to adaptation because the technique learns too slowly. The learning rate is slow because no prior knowledge about what the speaker dependent model is likely to be is utilized. Many adaptation techniques, such as MAP estimation, utilize *a priori* models or information to guide the adaptation process.

One of the easiest methods for incorporating prior information is to simply interpolate an SI density function with the ML estimated density function. The interpolation can be done independently for each phonetic class using an interpolation method which is sensitive to the amount of adaptation data that is available. Using this method, the speaker adapted probability density function for a particular phonetic unit p can be expressed as:

$$\mathrm{p}_{sa}(\vec{x}|p) = \frac{N_p}{N_p + K}\,\mathrm{p}_{ml}(\vec{x}|p) + \frac{K}{N_p + K}\,\mathrm{p}_{si}(\vec{x}|p) \tag{3.20}$$

In this expression, N_p is the number of examples of phone p that have been observed in the adaptation data and K is an interpolation factor. As can be seen in the expression, when N_p is small relative to K then the speaker adapted model relies heavily on the original SI model, $\mathrm{p}_{si}(\vec{x}|p)$. However, as N_p is increased, the ML estimated model, $\mathrm{p}_{ml}(\vec{x}|p)$, gains more and more weight. The speaker adapted model eventually asymptotes to the ML estimate when $N_p \gg K$. It should be noted that the density function interpolation method shares some similarities with the interpolation performed by standard MAP adaptation. The similarities between the two methods will be discussed in Section 3.4.1.

Figure 3.5 shows the density interpolation adaptation method with a K value of 300 in comparison to standard ML training. As can be seen, the density function interpolation method slowly improves in performance from the 7.4% error rate of the SI model set as more adaptation data becomes available. With 600 adaptation utterances the error rate for the density function interpolated models is 3.5%, which is slightly better than the performance of the standard ML trained models.

An appropriate value for K can be determined empirically. Experiments showed that values of K between 200 and 500 all perform comparably. Figure 3.6 shows the performance for three different values of K: 100, 200 and 300. The figure shows that $K = 200$ and $K = 300$ yield roughly the same performance curve. When K is set to 100, the adaptation performance improves at a slightly lower rate. If K is set too low then performance could suffer because the system relies too heavily on the ML estimated density functions. If K is set too high then performance will suffer because the system backs off to the SI density functions too readily. The figure also demonstrates that the system is not sensitive to the exact value of K.

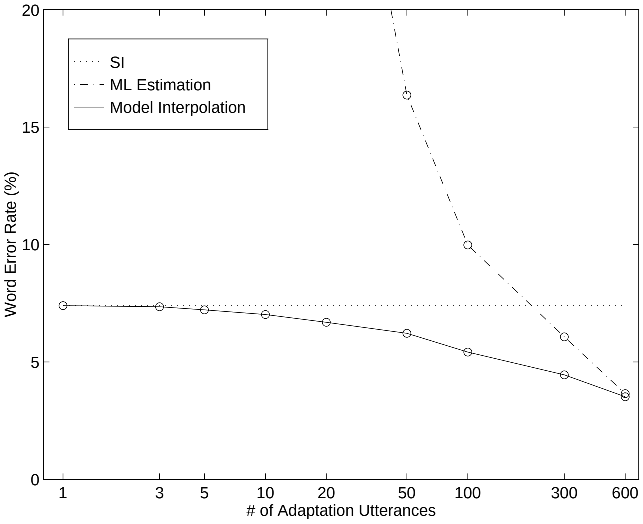


Figure 3.5: Performance of maximum likelihood trained models and density function interpolated models as the number of adaptation utterances is varied from 1 to 600.

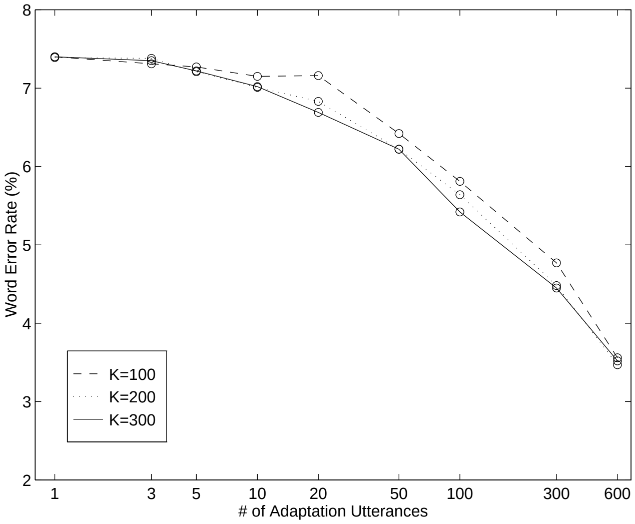


Figure 3.6: Performance of density function interpolated models using three different values of K as the number of adaptation utterances is varied from 1 to 600.

3.3.5 MAP Model Translation

Figure 3.3 hinted that an accurate estimate of a model's mean value is more important than an accurate estimate of the model's variance. This observation provides the impetus for implementing MAP adaptation of the mean vectors of a set of acoustic models. However, mixture Gaussian models do not utilize a single mean vector, but rather contain a separate mean vector for each mixture component. Thus, care must be taken in defining the terms that willed be used. The term *center of mass* will be used, instead of the term *mean*, when referring to the central location of a mixture Gaussian model. The center of mass, $\vec{c}$, for a mixture model containing G Gaussians can be expressed as:

$$\vec{c} = \sum_{g=1}^{G} w_g \vec{\mu}_g \tag{3.21}$$

In this expression, each $\vec{\mu}_g$ is a mean vector for a particular Gaussian mixture component and w_g is the component's weight. It can be noted that the vector $\vec{c}$ is also simply the mean vector of all of the data used to train the mixture Gaussian model. Using $\vec{c}$, we can re-express each mixture mean vector as follows:

$$\vec{\mu}_g = \vec{c} + \vec{\nu}_g \tag{3.22}$$

In this expression $\vec{\nu}_g$ is simply an offset which, when added to $\vec{c}$, yields the mixture component mean, $\vec{\mu}_g$. Using, these new definitions it can be seen that the location of a model can be altered without changing the model's shape simply by adjusting the vector $\vec{c}$. This type of adjustment will be referred to as *model translation*.

When model translation is the chosen type of speaker adaptation, the full set of adaptation parameters consists of one center of mass vector for each of the M different phonetic classes. Thus, the set of adaptation parameters can be expressed as:

$$\Lambda = \{\vec{c}_1, \vec{c}_2, \ldots, \vec{c}_M\} \tag{3.23}$$

The center of mass vectors can be estimated using standard MAP estimation techniques. The standard MAP estimation expression for any given center of mass vector $\vec{c}_m$, as derived from Equation 3.15, is given as:

$$\vec{c}_m^{\,map} = \arg\max_{\vec{c}_m} \prod_{n=1}^{N_m} \mathrm{p}(\vec{a}_{m,n}|\vec{c}_m)\mathrm{p}(\vec{c}_m) \tag{3.24}$$

To simplify the estimation process, the density function $\mathrm{p}(\vec{a}_{m,n}|\vec{c}_m)$ can be modeled using a single Gaussian density function instead of the mixture Gaussian density function which is actually used by the recognizer. With this assumption the acoustic model density function is expressed as:

$$\mathrm{p}(\vec{a}_{m,n}|\vec{c}_m) \equiv \mathcal{N}(\vec{c}_m, \mathbf{S}_m) \tag{3.25}$$

In other words, $\mathrm{p}(\vec{a}_{m,n}|\vec{c}_m)$ is modeled with a Gaussian density function with mean vector $\vec{c}_m$ and covariance matrix $\mathbf{S}_m$. Assume that during adaptation the value of $\vec{c}_m$ may be altered but the covariance matrix $\mathbf{S}_m$ will remain fixed. As such $\mathbf{S}_m$ is simply the SI covariance matrix found from the pooled data from all 149 SI speakers.

Note that the SI $\mathbf{S}_m$ covariance matrix has a much larger variance than the true underlying SD covariance matrix. Some past adaptation efforts have investigated methods for reducing the variance of this matrix to be more in line with the variance of a typical speaker dependent model. The most successful of these approaches is *speaker adaptative training* [4, 3]. This issue will not be investigated in this thesis.

Next, each *a priori* density function $\mathrm{p}(\vec{c}_m)$ will also be modeled with a single Gaussian density function. With this modeling decision the following definition can be made:

$$\mathrm{p}(\vec{c}_m) \equiv \mathcal{N}(\vec{\mu}_m^{ap}, \mathbf{S}_m^{ap}) \tag{3.26}$$

In other words, $\vec{\mu}_m^{ap}$ and $\mathbf{S}_m^{ap}$ represent the mean and covariance values of a Gaussian density function which models the likelihood of a particular speaker's density function for phone m possessing a center of mass value of $\vec{c}_m$.

Using the definitions provided above, it can be shown that the MAP estimated value of $\vec{c}_m$ given the adaptation data A_m can be found using the following equation [19]:

$$\vec{c}_m^{map} = \mathbf{S}_m \left(N_m \mathbf{S}_m^{ap} + \mathbf{S}_m\right)^{-1} \vec{\mu}_m^{ap} + N_m \mathbf{S}_m^{ap} \left(N_m \mathbf{S}_m^{ap} + \mathbf{S}_m\right)^{-1} \vec{c}_m^{ml} \tag{3.27}$$

It should be noted that $\vec{c}_m^{ml}$ is the ML estimated value of $\vec{c}_m$ and is defined as:

$$\vec{c}_m^{ml} = \frac{1}{N_m} \sum_{n=1}^{N_m} \vec{a}_{m,n} \tag{3.28}$$

When examining Equation (3.27) one should note that $\vec{\mu}_m^{ap}$ should be approximately the same as the SI center of mass vector for class m. Thus, the value of $\vec{c}_m^{map}$ obtained by MAP adaptation is simply an interpolation between the SI center of mass vector and the maximum likelihood estimated center of mass vector. When the number of observations of class m is small, the MAP estimate relies more heavily on the SI center of mass. As the number of observations increases, the MAP estimate asymptotes to the ML estimate. In Equation (3.27), $\mathbf{S}_m^{ap}$ and $\vec{c}_m^{ap}$ are trained from the estimated values of $\vec{c}_m$ for each of the 149 speakers in the SI set.

Figure 3.7 shows the performance of the MAP model translation method of speaker adaptation in comparison with the density function interpolation method of speaker adaptation. As can be seen, by adapting only the center of mass of the SI models the MAP model translation method achieves a much faster rate of adaptation than the density function interpolation method when the number of adaptation utterances

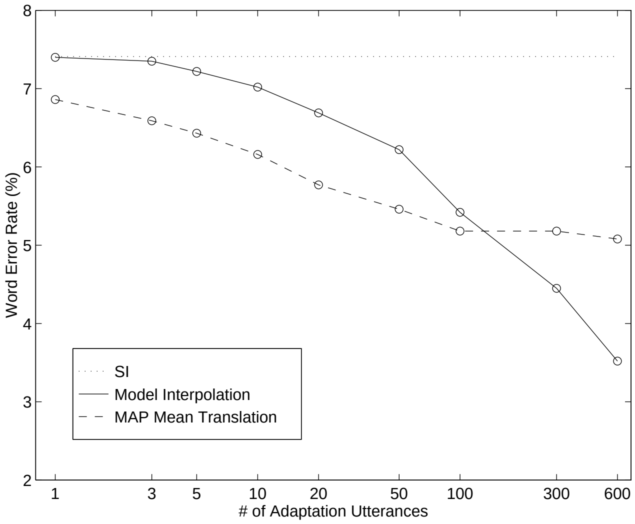


Figure 3.7: Performance of MAP translated SI models vs. SI and ML density function interpolated models as the number of adaptation utterances is varied from 1 to 600.

is small. After only 1 adaptation utterance has been presented to the system the error rate using MAP model translation is reduced from 7.4%to 6.9%. Despite the fact that MAP model translation is only adapting the location of the acoustic models and not their shape, it takes 100 adaptation utterances before the density function interpolation method achieves the same error rate as MAP model translation. With 600 adaptation utterances, MAP model translation achieves an error rate of 5.1% compared to 3.5% for model interpolation adaptation. In other words, by adapting only 2160 center of mass parameters (which is less than 1% of the total number of parameters in the SI model), the system is able to achieve 60% of the error reduction that adapting the entire set of model parameters allows.

This experiment shows the importance of allowing the number of adaptation parameters to vary depending on the amount of adaptation data that is available. As will be discussed later, different methods of adaptation (each with with different numbers of adaptation parameters) will be optimal for different amounts of adaptation data. Figure 3.7 shows that MAP model translation outperforms model interpolation when 100 or fewer adaptation utterances are available, but that model interpolation is better when more than 100 adaptation utterances are available. This suggests that, for the experiments just shown, an adaptation algorithm which relies upon MAP model translation for small to medium sized adaptation sets and but uses model interpolation for large adaptation sets is optimal. Methods for combining different adaptation algorithms within a unified framework will be discussed in Chapter 4 and Chapter 7.

3.4 Past and Present Approaches

3.4.1 MAP Adaptation

The standard expressions for MAP adaptation[1] of Gaussian model parameters have long been derived and can be found in many standard texts [19]. However, the standard text book implementation may encounter problems when it is implemented. Consider the MAP estimation expression for a Gaussian mean vector, as explained in Section 3.3.5. The expression is as follows:

$$\vec{\mu}_m^{map} = \frac{1}{N_m}\mathbf{S}_m \left(\mathbf{S}_m^{ap} + \frac{1}{N_m}\mathbf{S}_m\right)^{-1} \vec{\mu}_m^{ap} + \mathbf{S}_m^{ap} \left(\mathbf{S}_m^{ap} + \frac{1}{N_m}\mathbf{S}_m\right)^{-1} \vec{\mu}_m^{ml} \tag{3.29}$$

Here, the MAP estimate of a mean vector, $\vec{\mu}_m^{map}$, is found to be an interpolation of the ML estimate of the mean vector, $\vec{\mu}_m^{ml}$, and the *a priori* mean vector, $\vec{\mu}_m^{ap}$. The interpolation weights are a function of the acoustic model variance, $\mathbf{S}_m$, the *a priori* model variance, $\mathbf{S}_m^{ap}$, and the number of observations used to find the ML estimate, N_m. In this expression, it is assumed that only the mean vector is being adapted and that the variance used during recognition will remain fixed. It is also assumed that $\vec{\mu}_m^{ap}$ and $\mathbf{S}_m^{ap}$ have been accurately estimated.

In practice, the values of $\vec{\mu}_m^{ap}$ and $\mathbf{S}_m^{ap}$ must be computed from estimates of $\vec{\mu}_m$ from many training speakers. If the number of training speakers is limited or the estimates of $\vec{\mu}_m$ from each training speaker are poor then $\vec{\mu}_m^{ap}$ and $\mathbf{S}_m^{ap}$ may be poorly estimated. Because of this practical concern, it may be useful to limit the number of parameters in the *a priori* model by only estimating and utilizing the diagonal components of $\mathbf{S}_m^{ap}$. In this case $\mathbf{S}_m^{ap}$ can be defined with the following equation:

$$\mathbf{S}_m^{ap} = \begin{bmatrix} (\sigma_{m,1}^{ap})^2 & 0 & \cdots & 0 \\ 0 & (\sigma_{m,2}^{ap})^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & (\sigma_{m,D}^{ap})^2 \end{bmatrix} \tag{3.30}$$

In this equation, D is the number of dimensions of the observation space. Similarly the SI model covariance matrix, $\mathbf{S}_m$ can be forced to be diagonal as in the following equation:

$$\mathbf{S}_m = \begin{bmatrix} \sigma_{m,1}^2 & 0 & \cdots & 0 \\ 0 & \sigma_{m,2}^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_{m,D}^2 \end{bmatrix} \tag{3.31}$$

[1] MAP adaptation is also frequently referred to as Bayesian learning.

By assuming diagonal covariances any correlation between the dimensions can be eliminated thus allowing each element of $\vec{\mu}_m$ to be adapted independently of all other elements. If $\mu_{m,d}$ is defined to be the mean estimate of the d^{th} dimension of the m^{th} class then Equation (3.27) can be simplified to:

$$\mu_{m,d}^{map} = \frac{N_m}{N_m + K_{m,d}} \mu_{m,d}^{ml} + \frac{K_{m,d}}{N_m + K_{m,d}} \mu_{m,d}^{ap} \tag{3.32}$$

In this equation, the variable $K_{m,d}$ is defined as follows:

$$K_{m,d} = \left(\frac{\sigma_{m,d}}{\sigma_{m,d}^{ap}} \right)^2 \tag{3.33}$$

In this light, Equation (3.32) is only a simple interpolation between the *a priori* model parameters and the maximum likelihood estimated parameters with $K_{m,d}$ being an interpolation factor. In this case, the problem is essentially reduced to finding appropriate values for each $K_{m,d}$. There are a variety of ways in which these *a priori* parameters can be determined. They can be estimated from the training data, selected to optimize recognition performance on some development set of data, or even set to fixed arbitrary values.

The similarity between the interpolation in Equation (3.33) and the density function interpolation scheme presented in Section 3.3.4 should be noted. Although the density function interpolation method is not based on a formal mathematical framework, it is clear that its interpolation scheme will perform in a similar manner to the standard MAP estimation method derived above. As such, it may be possible to substitute simpler interpolation methods, such as the one utilized in Section 3.3.4, in the place of more complicated MAP algorithms without sacrificing accuracy.

The difficult issues surrounding MAP adaptation have been thoroughly investigated and expounded upon by Gauvain and Lee [26, 27, 28, 29, 30, 56]. In particular, in [30] they extend the mathematical framework of MAP adaptation to mixture Gaussian density functions and HMM recognizers. More recent investigations of the problem have introduced methods for performing on-line MAP adaptation [38, 39] and methods for smoothing MAP estimates of mixture Gaussian parameters (i.e., vector field smoothing) [78, 79]. Despite the extensive research that has been devoted to improving upon the standard MAP adaptation techniques, MAP adaptation remains a sub-optimal approach because it ignores the within-speaker correlations which exist between different acoustic models.

3.4.2 Extended MAP Adaptation

Standard MAP adaptation techniques have the property that the parameters they are estimating converge asymptotically to their ML estimates as more adaptation data becomes available. However, these techniques assume independence between the classes, thus ignoring the within-speaker correlations between the classes that are known to exist. By ignoring these correlations, adaptation towards the correct underlying parameter set for a new speaker may be slower than possible. To account for these correlations within a MAP framework Lasry and Stern developed the Extended MAP adaptation (EMAP) approach [52, 77]. The derivation of the EMAP adaptation scheme begins with Equation (3.18):

$$\Lambda^{map} = \arg\max_{\Lambda} \mathrm{p}(\mathcal{A}|f(\Lambda))\mathrm{p}(\Lambda) \tag{3.34}$$

Assuming the observations are independent, the expression becomes:

$$\Lambda^{map} = \arg\max_{\Lambda} \mathrm{p}(\Lambda) \prod_{m=1}^{M} \prod_{n=1}^{N_M} \mathrm{p}(\vec{a}_{m,n}|f(\Lambda)) \tag{3.35}$$

When describing the standard MAP approach to adaptation, the adaptation parameters set consisted of the mean vectors of the density functions of each class. The same set of parameters will be used to describe EMAP adaptation. Instead of using separate, independent *a priori* models for each mean vector $\vec{\mu}_m$, a generalized mean vector $\vec{m}$ is used to represent to entire set of mean vectors. Thus, $\vec{m}$ contains the full set of adaptation parameters and is defined as follows:

$$\Lambda = \vec{m} = \begin{bmatrix} \vec{\mu}_1 \\ \vec{\mu}_2 \\ \vdots \\ \vec{\mu}_M \end{bmatrix} \tag{3.36}$$

It will also be assumed, as it is with MAP adaptation, that an observation is only dependent on the parameters of the acoustic model of its own class. Under this condition, the EMAP estimation problem is expressed as:

$$\vec{m}^{map} = \arg\max_{\vec{m}} \mathrm{p}(\vec{m}) \prod_{m=1}^{M} \prod_{n=1}^{N_M} \mathrm{p}(\vec{a}_{m,n}|\vec{\mu}_m) \tag{3.37}$$

As with MAP adaptation, the observation density functions will be modeled with single Gaussian densities and only the mean vectors of these density functions will be adapted. Thus, the following definition will be utilized:

$$\mathrm{p}(\vec{a}_{m,n}|\vec{\mu}_m) \equiv \mathcal{N}(\vec{\mu}_m, \mathbf{S}_m) \tag{3.38}$$

The generalized mean vector $\vec{m}$ is also modeled with a single Gaussian density as follows:

$$\mathrm{p}(\vec{m}) \equiv \mathcal{N}(\vec{m}^{ap}, \mathbf{S}^{ap}) \tag{3.39}$$

Within this definition, the covariance matrix $\mathbf{S}^{ap}$ will be represented as follows:

$$\mathbf{S}^{ap} = \begin{bmatrix} \mathbf{S}^{ap}_{1,1} & \mathbf{S}^{ap}_{1,2} & \cdots & \mathbf{S}^{ap}_{1,M} \\ \mathbf{S}^{ap}_{2,1} & \mathbf{S}^{ap}_{2,2} & \cdots & \mathbf{S}^{ap}_{2,M} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{S}^{ap}_{M,1} & \mathbf{S}^{ap}_{M,2} & \cdots & \mathbf{S}^{ap}_{P,M} \end{bmatrix} \tag{3.40}$$

In examining $\mathbf{S}^{ap}$, it should be noted that the submatrices $\mathbf{S}^{ap}_{j,k}$ where $j \neq k$ represent the within-speaker correlations which exist between different phonetic classes. From the three definitions provided above it can be easily shown that EMAP and MAP yield this same results if, within $\mathbf{S}^{ap}$, $\mathbf{S}^{ap}_{j,k} = 0$ when $j \neq k$.

Before presenting the final EMAP adaptation equation, three more definitions must be made. First, let the matrix $\mathbf{N}$ be defined as follows:

$$\mathbf{N} = \begin{bmatrix} N_1\mathbf{I} & 0 & \cdots & 0 \\ 0 & N_2\mathbf{I} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & N_M\mathbf{I} \end{bmatrix} \tag{3.41}$$

Within $\mathbf{N}$, each subdiagonal matrix, $N_m\mathbf{I}$, has dimension $D \times D$. Next, let the matrix $\mathbf{S}$ be defined as:

$$\mathbf{S} = \begin{bmatrix} \mathbf{S}_1 & 0 & \cdots & 0 \\ 0 & \mathbf{S}_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{S}_M \end{bmatrix} \tag{3.42}$$

Finally, the maximum likelihood estimate of $\vec{m}$ will be defined as follows:

$$\vec{m}^{ml} = \begin{bmatrix} \vec{\mu}^{ml}_1 \\ \vdots \\ \vec{\mu}^{ml}_M \end{bmatrix} \tag{3.43}$$

Using the above the definitions it can be shown that the EMAP adaptation equation for mean vectors is:

$$\vec{m}^{emap} = \mathbf{S}^{ap}(\mathbf{N}\mathbf{S}^{ap} + \mathbf{S})^{-1}\mathbf{N}\vec{m}^{ml} + \mathbf{S}(\mathbf{N}\mathbf{S}^{ap} + \mathbf{S})^{-1}\vec{m}^{ap} \tag{3.44}$$

As with MAP adaptation, EMAP adaptation also converges to the ML estimate as the number of adaptation examples of each class increases. Unfortunately, EMAP adaptation has far more parameters than standard MAP adaptation and is thus more susceptible to problems due to sparse training data. In particular, the matrix $\mathbf{S}^{ap}$ is $MD \times MD$ in dimension. Thus, in order for $\mathbf{S}^{ap}$ to be non-singular and invertible, the matrix must be trained from at least $MD+1$ different training vectors. In order to generate $MD+1$ training vectors, $MD+1$ different training speakers are needed, each of whom has a reliably estimated mean vector for all M different phones. If the dimension of each mean vector is $D = 36$ and the number of phonetic classes used by the system is $M = 60$ then $MD = 1 = 2161$ different training speakers are required. Because this is almost an order of magnitude larger than the number of speakers in the standard speech databases used today, full covariance EMAP adaptation is not yet feasible.

To make EMAP adaptation practical, simplifying assumptions must be made. The most typical assumption that is made is to assume that each dimension is independent of all other dimensions. In this case, the correlations between different sounds are retained only for the same dimensions within each mean vector. By ignoring the across dimension correlations the number of covariance parameters that need to be trained within the *a priori* models drops from $(MD)^2$ to M^2D. Additionally, the covariance matrix $\mathbf{S}^{ap}$ will only need $M+1$ different training speakers, instead of $MD+1$, in order to be invertible. For comparison, full covariance MAP adaptation requires MD^2 covariance parameters be trained. Even with simplifying assumptions, EMAP adaptation is extremely difficult to implement. Despite the difficulties, research efforts by Huo and Lee [40, 41] and by Zavaliagkos, *et al* [81, 82], have utilized EMAP adaptation with modest success.

3.4.3 Model Prediction Approaches

In addition to the EMAP approach there have been a number of approaches which follow the same motivation as EMAP, but utilize different slightly frameworks. Typically, these techniques utilize a standard MAP estimates when observations for a particular model are available. Predictive methods which utilize within-speaker correlation information are then used to adapt the models for which no adaptation data has been seen. Techniques that fall into this general class of adaptation approaches have been developed by Ahadi and Woodland [2], Cox [14], Hazen [34], Chen and DeSouza [11], and Afify, *et al.* [1].

3.4.4 Transformational Approaches

Overview

As an alternative to predictive methods such as EMAP, which require the training of *a priori* models, some researchers have investigated transformational approaches to the problem. Like EMAP, these approaches allow a model's parameters to be adapted even if no adaptation data from that particular phone has been observed. However, they do not require the use of an *a priori* statistical model. The fundamental idea driving these approaches is that similar phones will be adapted in similar fashions and, as such, can be *tied* during the adaptation process. These methods are especially useful in context dependent systems where many different context dependent allophones of the same phonetic element are allowed to share the same adaptation *transformation*.

To begin, let the set of all phones be presented as $\mathcal{P}$. Next, suppose the full set $\mathcal{P}$ can be subdivided into S subsets. Let the set of subsets be defined as:

$$\mathcal{C} = \{C_1, C_2, \ldots, C_S\} \tag{3.45}$$

Furthermore, let each phone belong to one and only one sub-class. Mathematically, this is represented by:

$$C_1 \cup C_2 \cup \cdots \cup C_S = \mathcal{P} \quad \text{and} \quad C_i \cap C_j = \emptyset \ \ \forall i \neq j \tag{3.46}$$

Here each subset C_i represents a particular class of phones which are believed to possess some similarity with respect to the adaptation algorithm that is being used. If the phone for the m^{th} acoustic model is represented as p_m and this phone belongs to the i^{th} class, then we can say $p_m \,\epsilon\, C_i$. The exact number of classes in $\mathcal{C}$ and the breakdown of the phones given to each C_i is something that varies from algorithm to algorithm depending on the nature of the algorithm's adaptation scheme and the amount of available adaptation data. During adaption each class utilizes a set of adaptation parameters which are *shared* amongst all phones contained within that class.

Tied Model Translation

The simplest transformational approach can be referred to as *tied model translation* [44, 73]. In this approach it is assumed that the speaker adapted center of mass of a phone is simply the addition of the SI center of mass and a translation vector. The translation vector, in this case, will be *shared* or *tied* with all other phones contained in the same class. This can be expressed as:

$$\vec{c}_m^{\,sa} = \vec{c}_m^{\,si} + \vec{v}_i \ \text{ where } \ p_m \,\epsilon\, C_i \tag{3.47}$$

In other words, all center of mass vectors (and consequently all mean vectors of the individual Gaussian components) belonging to the models of phones contained in class C_i will be translated using $\vec{v}_i$.

The number of classes, S, can be varied to allow for varying degrees of parameter tying. When only a small amount of adaptation data is available then it is desirable to use only a small number of free parameters. In this case the number of classes may be very small. As more and more adaptation data becomes available the number of classes can be increased.

A variety of different methods can be used to estimate the translation vector, $\vec{v}_i$, for each class. The simplest method is via ML estimation. This can be expressed as:

$$\vec{v}_i^{ml} = \arg\max_{\vec{v}_i} \prod_{p_m \epsilon C_i} \prod_{n=1}^{N_m} \mathrm{p}(\vec{a}_{m,n}|\vec{v}_i) \tag{3.48}$$

When using this expression, the following definition can be utilized:

$$\mathrm{p}(\vec{a}_{m,n}|\vec{v}_i) \equiv \mathcal{N}(\vec{c}_m^{\,si} + \vec{v}_i, \mathbf{S}_m) \tag{3.49}$$

Under the above conditions it can be shown that:

$$\vec{v}_i^{ml} = \mathbf{U}^{-1}\vec{w} \tag{3.50}$$

Here $\mathbf{U}$ is defined as:

$$\mathbf{U} = \sum_{p_m \epsilon C_i} N_m \mathbf{S}_m^{-1} \tag{3.51}$$

Also, $\vec{w}$ is defined as:

$$\vec{w} = \left(\sum_{p_m \epsilon C_i} \sum_{n=1}^{N_M} (\vec{a}_{m,n} - \vec{c}_m^{\,si})^T \mathbf{S}_m^{-1} \right)^T \tag{3.52}$$

Tied Model Transformation

Tied model transformation is simply an extension of the model translation approach discussed in the last section. In model translation the mean vectors of the system are translated with the following equation:

$$\vec{\mu}_{m,g}^{\,sa} = \vec{\mu}_{m,g}^{\,si} + \vec{v}_i \quad \text{where} \quad p_m \,\epsilon\, C_i \tag{3.53}$$

Here $\vec{\mu}_{m,g}$ is the g^{th} Gaussian component of the mixture model for the m^{th} phone. In model transformation, the models can be rotated and scaled as well as translated. This is performed with the following equation:

$$\vec{\mu}_{m,g}^{\,sa} = \mathbf{R}_i \vec{\mu}_{m,g}^{\,si} + \vec{v}_i \quad \text{where} \quad p_m \,\epsilon\, C_i \tag{3.54}$$

In this equation, the matrix $\mathbf{R}_i$ performs rotation and scaling of the mean vectors while $\vec{v}_i$ performs the translation. This approach was independently proposed by Leggetter and Woodland [59, 60] and by Digalakis, *et al.* [17, 16, 71]. The expression can be condensed by using the following definitions:

$$\mathbf{Q}_i = [\mathbf{R}_i \; ; \; \vec{v}_i] \tag{3.55}$$

$$\vec{\xi}_{m,g} = \left[\begin{array}{c} \vec{\mu}^{si}_{m,g} \\ 1 \end{array} \right] \tag{3.56}$$

With these new definitions, the adapted mean vector for component g of phone m can be written as:

$$\vec{\mu}^{sa}_{m,g} = \mathbf{Q}_i \vec{\xi}_{m,g} \quad \text{where} \quad p_m \,\epsilon\, C_i \tag{3.57}$$

A similar maximum likelihood approach as the one described above for tied model translation adaptation can be taken to find the values for each $\mathbf{Q}_i$ matrix. A full description of this process can be found in [60]. Leggetter and Woodland call their implementation *maximum likelihood linear regression* (MLLR). Since its introduction, MLLR has become one of the most widely used adaptation methods. Its usefulness spawns from the fact that it is capable of blindly adapting to both the speaker and environment simultaneously without requiring any explicit *a priori* models. Despite its popularity MLLR has difficulties when the number of adaptation utterances is small. These difficulties arise because of the large number of parameters in each $\mathbf{Q}_i$ matrix which must be estimated. In fact, MLLR has been shown to harm recognition performance when the number of adaptation utterances is small (3 or less) [60, 83]. As such, it has not proven useful for rapid or instantaneous adaptation.

3.4.5 Adaptation to Speaker Properties

Overview

This section describes a series of adaptation techniques which capitalize on prior knowledge about the effects of various speaker properties on the speech signal. It is known that different speaker properties can contribute to systematic variations in the speech waveform. For example, the length of a persons vocal tract is a primary factor in determining the location of the person's formants during the production of vowels. Thus, if the models of a speech recognition system can be normalized to match the current speaker's vocal tract length, one major source of speaker variability can be removed. Other sources of systematic variation in the speech signal arise from the speaker's gender, speaking rate, regional dialect, etc. A number of research efforts have attempted to account for different speaker properties within the modeling schemes of their systems.

Vocal Tract Length Normalization

Recently there have been numerous research efforts directed at the problem of *vocal tract length normalization* (VTLN) [20, 43, 58, 83]. The basic idea of VTLN is to warp the spectrum of the speech through *stretching* or *compressing* so that the relative formant locations of the current speaker match the formant locations of some generic speaker model as closely as possible. The warping function is intended to simulate the effects of lengthening or shortening the vocal tract of the speaker. The ultimate goal is to remove the variability in the acoustics of different speakers that is caused by the differences in vocal tract length.

Speaking Rate Adaptation

Although there is evidence that the speaking rate can have a significant effect on the acoustic realization, duration, and pronunciation of speech, there has been relatively little research directed towards accounting for speaking rate within the models of a speech recognition system. Both Siegler and Stern [74], and Morgan, *et al.* [64] were able to improve their HMM recognition systems by utilizing speaking rate dependent HMM transition probabilities. The transition probabilities provide the durational modeling capability for an HMM. Siegler and Stern also created speaking rate specific acoustic and pronunciation models as well, but were not able to achieve any improvement from these new models.

Speaker Clustering

One of the most common approaches for providing speaker constraint to a recognition system is speaker clustering [13, 23, 24, 48, 47, 61, 65, 62, 66]. The basic of idea of speaker clustering is to create a selection of different models by clustering the training speakers based on some similarity measure. It is hoped that each cluster represents some specific type of speaker, i.e. all of the speakers possess a common set of speaker properties. Each cluster model would then be trained using a set of speakers deemed similar by the clustering criterion. During recognition the test speaker would be compared to each speaker cluster. The model of the cluster to which the test speaker is most similar would be used to recognize the utterance. For example, the simplest form of speaker clustering is gender dependent modeling. The many different ways in which the clustering, training, and recognition can be performed are too varied to be discussed in detail here. One specific method of speaker clustering is presented and analyzed in Chapter 5.

3.5 Summary

This chapter has discussed the underlying tenets of speaker adaptation, presented a series of experiments which demonstrate the basic ideas of adaptation, and summarized some of the major past and present approaches. The ultimate goal of speaker adaptation is to learn the underlying SD models for the current speaker using as little adaptation data as possible. In order to achieve this goal a speaker adaptation algorithm should possess the following attributes:

1. It should converge asymptotically with the ML estimate of the SD model as the amount of adaptation data grows.

2. It should account for the within-speaker correlations which exist between different speech events.

3. It should utilize a set of adaptation parameters whose size can be varied to permit reliable estimation using whatever amount of adaptation data is available.

4. It should utilize adaptation parameters which can account for and encode sources of systematic speaker variability.

In examining the different past approaches presented in Section 3.4, it is clear that none of them possess all four of the attributes discussed above. MAP adaptation only has the first attribute. EMAP adaptation possesses both the first and second attributes. The transformational approaches can be implemented so as to possess both the first and third attribute. It may also be argued that the parameter sharing within the different phonetic classes utilized by the transformational approaches accounts for some of the existing within-speaker correlation information discussed in the second attribute. Methods such as speaker clustering and vocal tract normalization possess the second and fourth attributes but not the first and third.

In order to achieve optimal speaker adaptation, a method which incorporates the strengths of all of the past approaches into one framework is needed. One approach that can be taken is to combine some of the past approaches into a single framework. For example, speaker clustering and MAP adaptation can be combined within a two-step process. The first step is to select the speaker cluster model which is the closest fit with the current speaker. The second step is to apply MAP adaptation to the selected speaker cluster model. A number of research efforts have investigated the combination of two or more of the techniques presented above [16, 71, 81, 83]. The next three chapters will present three new adaptation approaches which attempt to incorporate some or all of the attributes listed above. A method for combining the new adaptation techniques for the purpose of combining their strengths is discussed in Chapter 7.

Chapter 4

Reference Speaker Weighting

4.1 Overview

In Chapter 3 a series of experiments demonstrated the relative capabilities of model interpolation adaptation and MAP model translation on a word recognition task. What these methods lack is the use of within-speaker correlation information or of any form a speaker constraint. Within-speaker correlation information could be included into an adaptation algorithm using an approach, such as EMAP, which encodes the correlation information in an *a priori* statistical model. However, such approaches have only been moderately successful so far while also being difficult to implement. On the other hand, transformational approaches which tie the adaptation of different phone models to the same set of transformation parameters are easy to implement and have provided respectable adaptation results. The transformation approaches have performed well despite not using explicit speaker correlation information. In this chapter, a novel adaptation approach called *reference speaker weighting* (RSW) is introduced. RSW adaptation attempts to take advantage of the strong points of the EMAP and transformation adaptation approaches while avoiding their weak points.

The basic idea of RSW adaptation is to incorporate speaker constraints into a tied adaptation parameter approach similar to the transformation approaches described in Section 3.4.4. As its name suggests, reference speaker weighting adaptation is performed by finding an optimal weighting of parameters provided by a set a reference speakers. In the RSW framework that will be presented, the task of finding an optimal weighting of a set of reference speakers will be performed in a manner similar to finding the optimal rotation matrix in the MLLR transformation approach.

4.2 Formulation of RSW Approach

4.2.1 Basic Framework

The basic premise of reference speaker weighting is that the model parameters of a speaker adapted model can be constructed from a weighted combination of model parameters from a set of individual reference speakers. Because model translation is an effective style of adaptation, the RSW algorithm presented in this chapter will be developed and evaluated within a model translation framework. In other words, the goal is to adjust the center of mass parameters of the mixture Gaussian density functions used for acoustic modeling (as discussed in Section 3.3.5).

To begin, assume a set of R different reference speakers has been extracted from the training data. Also assume that for each reference speaker a reasonably accurate estimate of the center of mass vector for each of M different phonetic classes has been obtained. Let the center of mass vector for phone model m of reference speaker r be represented as $\vec{c}_{m,r}$. Let the dimension of the mean vectors be defined as D. Thus, each speaker has a collection of center of mass vectors which define a single $M \times D$ length *speaker vector*. Let the speaker vector for reference speaker r be defined as $\vec{\gamma}_r$. The mathematical definition of the speaker vector $\vec{\gamma}_r$ is given as:

$$\vec{\gamma}_r = \begin{bmatrix} \vec{c}_{1,r} \\ \vdots \\ \vec{c}_{M,r} \end{bmatrix} \tag{4.1}$$

Furthermore, the entire set of speaker vectors can be represented with the matrix $\mathbf{\Gamma}$ which will be defined as:

$$\mathbf{\Gamma} = [\, \vec{\gamma}_1 \,;\, \vec{\gamma}_2 \,;\, \cdots \,;\, \vec{\gamma}_R \,] = \begin{bmatrix} \vec{c}_{1,1} & \vec{c}_{1,2} & \cdots & \vec{c}_{1,R} \\ \vec{c}_{2,1} & \vec{c}_{2,2} & \cdots & \vec{c}_{2,R} \\ \vdots & \vdots & \ddots & \vdots \\ \vec{c}_{M,1} & \vec{c}_{M,2} & \cdots & \vec{c}_{M,R} \end{bmatrix} \tag{4.2}$$

The portion of $\mathbf{\Gamma}$ which contains only the center of mass vectors for the m^{th} model can be represented as $\mathbf{\Gamma}_m$ and is expressed as:

$$\mathbf{\Gamma}_m = [\, \vec{c}_{m,1} \,;\, \vec{c}_{m,2} \,;\, \cdots \,;\, \vec{c}_{m,R} \,] \tag{4.3}$$

This allows $\mathbf{\Gamma}$ to be expressed as:

$$\mathbf{\Gamma} = \begin{bmatrix} \mathbf{\Gamma}_1 \\ \vdots \\ \mathbf{\Gamma}_M \end{bmatrix} \tag{4.4}$$

During adaptation, the goal is to determine the most likely speaker vector, $\vec{\gamma}$, for a test speaker given the available adaptation data. It is desirable to utilize the *a priori* knowledge provided by the reference speaker vectors without having to explicitly build *a priori* statistical models. One possible solution is to use the speaker vectors in $\mathbf{\Gamma}$ to constrain the speaker space in which $\vec{\gamma}$ may fall. Specifically, the value of $\vec{\gamma}$ is constrained to be a weighted average of the speaker vectors contained in $\mathbf{\Gamma}$. This can be expressed as:

$$\vec{\gamma} = \mathbf{\Gamma}\vec{w} \tag{4.5}$$

Here $\vec{w}$ is a weighting vector which allows a new speaker vector to be created via a weighting summation of the reference speaker vectors in $\mathbf{\Gamma}$. The portions of $\vec{\gamma}$ and $\mathbf{\Gamma}$ which represent model m can be expressed as $\vec{c}_m$ and $\mathbf{\Gamma}_m$, thus allowing the following expression:

$$\vec{c}_m = \mathbf{\Gamma}_m\vec{w} \tag{4.6}$$

4.2.2 ML-RSW

One method for determining the weights in $\vec{w}$ is to use a maximum likelihood approach. The goal is to find the value of $\vec{w}$ which maximizes the likelihood of the adaptation data $\mathcal{A}$. As shown in Chapter 3, $\mathcal{A}$ can be represented as:

$$\mathcal{A} = \{\, A_1,\, A_2,\; \ldots\;,\, A_M \,\} \tag{4.7}$$

Here each A_m is a set of example observations from the m^{th} phonetic class. Furthermore, the sets of observations from each class will be represented as:

$$A_m = \{\, \vec{a}_{m,1},\, \vec{a}_{m,2},\; \ldots\;,\, \vec{a}_{m,N_m} \,\} \tag{4.8}$$

Here each $\vec{a}_{m,n}$ is a specific observation vector of class m and N_m is the total number of adaptation observations available for class m. Using the above definitions the goal is to find the optimal value of $\vec{w}$ using the maximum likelihood expression:

$$\arg\max_{\vec{w}}\ \mathrm{p}(\mathcal{A}|\vec{w}). \tag{4.9}$$

Because the logarithm is a monotonic function, the same result can be obtained by using the expression:

$$\arg\max_{\vec{w}}\ \log \mathrm{p}(\mathcal{A}|\vec{w}). \tag{4.10}$$

In solving for the optimal $\vec{w}$ the assumption that all observations are independent is made. With this assumption the expression reduces to:

$$\arg\max_{\vec{w}}\ \log \prod_{m=1}^{M} \prod_{n=1}^{N_m} \mathrm{p}(\vec{a}_{m,n}|\vec{w}) \tag{4.11}$$

This can be rewritten as:

$$\arg\max_{\vec{w}} \sum_{m=1}^{M} \sum_{n=1}^{N_m} \log \mathrm{p}(\vec{a}_{m,n}|\vec{w}). \tag{4.12}$$

Next the density function must be defined. As in the MAP model translation method, a single Gaussian density function is used to approximate each phonetic class model. This density function can be expressed as:

$$\mathrm{p}(\vec{a}_{m,n}|\vec{w}) \equiv \mathcal{N}(\vec{c}_m, \mathbf{S}_m) \tag{4.13}$$

Here $\vec{c}_m$ is the subsection of $\vec{\gamma}$ representing the center of mass value for class m. Also, $\mathbf{S}_m$ represents the covariance matrix for class m, which will be assumed to remain constant. By applying the logarithm to each Gaussian density function the expression expands to:

$$\arg\max_{\vec{w}} \sum_{m=1}^{M} \sum_{n=1}^{N_m} -\frac{D}{2}\log(2\pi) - \frac{1}{2}\log(|\mathbf{S}_m|) - \frac{1}{2}(\vec{a}_{m,n} - \vec{c}_m)^T \mathbf{S}_m^{-1} (\vec{a}_{m,n} - \vec{c}_m). \tag{4.14}$$

By noting that only $\vec{c}_m$ is dependent on $\vec{w}$ the expression is equivalently written as:

$$\arg\max_{\vec{w}} \sum_{m=1}^{M} \sum_{n=1}^{N_m} -(\vec{a}_{m,n} - \vec{c}_m)^T \mathbf{S}_m^{-1} (\vec{a}_{m,n} - \vec{c}_m). \tag{4.15}$$

Next $\mathbf{\Gamma}_m \vec{w}$ can be substituted for $\vec{c}_m$ to yield the expression:

$$\arg\max_{\vec{w}} \sum_{m=1}^{M} \sum_{n=1}^{N_m} -(\vec{a}_{m,n} - \mathbf{\Gamma}_m \vec{w})^T \mathbf{S}_m^{-1} (\vec{a}_{m,n} - \mathbf{\Gamma}_m \vec{w}). \tag{4.16}$$

Next the expression can be expanded and terms that do not contain $\vec{w}$ can be removed to yield the expression:

$$\arg\max_{\vec{w}} \sum_{m=1}^{M} \sum_{n=1}^{N_m} 2(\vec{a}_{m,n}^T \mathbf{S}_m^{-1} \mathbf{\Gamma}_m \vec{w}) - (\mathbf{\Gamma}_m \vec{w})^T \mathbf{S}_m^{-1} (\mathbf{\Gamma}_m \vec{w}). \tag{4.17}$$

This is equivalently written as:

$$\arg\max_{\vec{w}} \sum_{m=1}^{M} \sum_{n=1}^{N_m} 2(\vec{a}_{m,n}^T \mathbf{S}_m^{-1} \mathbf{\Gamma}_m \vec{w}) - \vec{w}^T \mathbf{\Gamma}_m^T \mathbf{S}_m^{-1} \mathbf{\Gamma}_m \vec{w}. \tag{4.18}$$

The expression can then be rewritten as:

$$\arg\max_{\vec{w}} \left[2 \left(\sum_{m=1}^{M} \sum_{n=1}^{N_m} \vec{a}_{m,n}^T \mathbf{S}_m^{-1} \mathbf{\Gamma}_m \right) \vec{w} - \vec{w}^T \left(\sum_{n=1}^{M} \sum_{n=1}^{N_m} \mathbf{\Gamma}_m^T \mathbf{S}_m^{-1} \mathbf{\Gamma}_m \right) \vec{w} \right]. \tag{4.19}$$

Let $\mathbf{U}$ be defined as follows:

$$\mathbf{U} = \sum_{m=1}^{M} \sum_{n=1}^{N_m} \mathbf{\Gamma}_m^T \mathbf{S}_m^{-1} \mathbf{\Gamma}_m = \sum_{m=1}^{M} N_m \mathbf{\Gamma}_m^T \mathbf{S}_m^{-1} \mathbf{\Gamma}_m \tag{4.20}$$

Let $\vec{v}$ be defined as follows:

$$\vec{v}^T = \sum_{m=1}^{M} \sum_{n=1}^{N_m} \vec{a}_{m,n}^T \mathbf{S}_m^{-1} \mathbf{\Gamma}_m \tag{4.21}$$

Using the definitions for $\mathbf{U}$ and $\vec{v}$ the expression becomes:

$$\arg\max_{\vec{w}} \; 2\vec{v}^T\vec{w} - \vec{w}^T\mathbf{U}\vec{w}. \tag{4.22}$$

Finally, expression can be solved for $\vec{w}$ using the equation:

$$\frac{\mathrm{d}}{\mathrm{d}\vec{w}}(2\vec{v}^T\vec{w} - \vec{w}^T\mathbf{U}\vec{w}) = 0. \tag{4.23}$$

From here it can be shown that:

$$\vec{w} = \mathbf{U}^{-1}\vec{v}. \tag{4.24}$$

Thus, the optimal weighting vector for the ML-RSW formulation can be found using a closed form solution.

4.2.3 Incorporation of Constraints

The formulation in (4.24) is elegant in that a closed form solution for the speaker weighting can be found. However, this formulation also allows for solutions which do not have an obvious physical meaning. There are two main flaws in this formulation. First, the weights attached to each individual training speaker mean vector can be assigned a negative value. Second, the total sum of the weights can add up to a value other than one. If it is desired that the adapted speaker vector be a simple weighted interpolation of the reference speaker vectors then the following constraints must be imposed on the problem:

$$\forall r \; w_r \geq 0 \;\text{ and }\; \sum_{r=1}^{R} w_r = 1 \tag{4.25}$$

Without these two constraints the speaker vector $\vec{\gamma}$ which is found during adaptation may not be a simple interpolation of the reference speakers. Instead it is possible for the algorithm to create a speaker vector which extrapolates outside of the constrained space defined by $\mathbf{\Gamma}$.

It is possible to incorporate the constraint that the weights must sum to one and still have a closed form solution. This can be done through the use of a Lagrange multiplier. To begin let $\vec{o}$ be defined as a vector of dimension R which is simply a column of ones. This allows the constraint to be expressed as:

$$\vec{o}^T \vec{w} = \sum_{r=1}^{R} w_r = 1. \tag{4.26}$$

Using a Lagrange multiplier the expression in (4.22) can be rewritten to include the new constraint as:

$$\arg\max_{\vec{w}} \; 2\vec{v}^T \vec{w} - \vec{w}^T \mathbf{U} \vec{w} - \kappa(\vec{o}^T \vec{w} - 1) \tag{4.27}$$

Here κ is the Lagrange multiplier. This leads to two equations which now must be satisfied. These equations are as follows:

$$\frac{\mathrm{d}}{\mathrm{d}\vec{w}} \Big(2\vec{v}^T \vec{w} - \vec{w}^T \mathbf{U} \vec{w} - \kappa(\vec{o}^T \vec{w} - 1) \Big) = 0. \tag{4.28}$$

$$\frac{\mathrm{d}}{\mathrm{d}\kappa} \Big(2\vec{v}^T \vec{w} - \vec{w}^T \mathbf{U} \vec{w} - \kappa(\vec{o}^T \vec{w} - 1) \Big) = 0. \tag{4.29}$$

Solving these two equations simultaneously leads to the following solution for $\vec{w}$:

$$\vec{w} = \mathbf{U}^{-1} \vec{v} - \left(\frac{\vec{o}^T \mathbf{U}^{-1} \vec{v} - 1}{\vec{o}^T \mathbf{U}^{-1} \vec{o}} \right) \mathbf{U}^{-1} \vec{o}. \tag{4.30}$$

While a closed form solution exists which incorporates the constraint that the weights must sum to one, a closed form solution does not exist which incorporates the constraint that all weights must have a non-negative value. Thus, in order to find the optimal $\vec{w}$ under these constraints, an iterative approach to maximizing the expression in (4.22) must be taken. A simple hill-climbing algorithm exists which can perform this constrained maximization. The hill climbing routine iteratively maximizes the likelihood by altering only two weights at a time.

To begin, suppose the weights are initialized in some suboptimal fashion. The total likelihood of the adaptation data can be increased by changing the values of only two weights, w_a and w_b. Accounting for the fact that the expression in (4.27) will be maximized by altering only two of the weights, this expression can be rewritten as:

$$f(w_a, w_b, \kappa) = 2\vec{v}^T \vec{w} - \vec{w}^T \mathbf{U} \vec{w} - \kappa(\vec{o}^T \vec{w} - 1). \tag{4.31}$$

In this expression it is assumed that all w_i except w_a and w_b will remain constant. The expression can be rewritten in summation form as:

$$f(w_a, w_b, \kappa) = \left(\sum_{i=1}^{S} \sum_{j=1}^{S} -w_i w_p U_{i,j} \right) + \left(\sum_{i=1}^{S} 2v_i w_i \right) + \left(\kappa - \kappa \sum_{i=1}^{S} w_i \right). \tag{4.32}$$

In this expression, $U_{i,j}$ is the element in the i^{th} row and j^{th} column of $\mathbf{U}$. To maximize this expression the following set of equations must be solved:

$$\frac{d}{dw_a} f(w_a, w_b, \kappa) = 0 \tag{4.33}$$

$$\frac{d}{dw_b} f(w_a, w_b, \kappa) = 0 \tag{4.34}$$

$$\frac{d}{d\kappa} f(w_a, w_b, \kappa) = 0 \tag{4.35}$$

The following constants can be defined:

$$c_a = v_a - \sum_{i \neq a,b} w_i U_{i,a} \tag{4.36}$$

$$c_b = v_b - \sum_{i \neq a,b} w_i U_{i,b} \tag{4.37}$$

$$c_w = 1 - \sum_{1 \neq a,b} w_i \tag{4.38}$$

Using these constants the three equations above can be solved, resulting in the following update equations for w_a and w_b:

$$w_a = \frac{c_a - c_b + c_w(U_{b,b} - U_{b,a})}{U_{a,a} - U_{a,b} + U_{b,b} - U_{b,a}} \tag{4.39}$$

$$w_b = \frac{c_b - c_a + c_w(U_{a,a} - U_{a,b})}{U_{a,a} - U_{a,b} + U_{b,b} - U_{b,a}} \tag{4.40}$$

After the update equations for w_a and w_b are solved, the values must be checked to see if they are both greater than zero. It is possible that one of the two values may be less than zero thus violating the constraint that all weights must be non-negative. If this is indeed the case then the weights must be adjusted. Supposing $w_a < 0$, a new set of weights w'_a and w'_b can be found which result in the highest possible likelihood when all other weights are held constant and the constraints are obeyed simply by using the following updates:

$$w'_a = 0 \tag{4.41}$$

$$w'_b = w_b + w_a \tag{4.42}$$

By applying the above set of equations iteratively across all pairs of weights, the likelihood can be maximized while still adhering to the constraints of the system. This iterative process is essentially just a hill-climbing algorithm which is guaranteed to increase the likelihood of the data with every iteration until convergence is achieved. In practice, this hill-climbing search is found to operate quickly and efficiently, converging to the maximum likelihood in a small number of iterations.

4.2.4 MAP-RSW

The goal of ML-RSW center of mass estimation was to find a set of weights, $\vec{w}$, which maximized the likelihood of the adaptation data $\mathcal{A}$. This was expressed as:

$$\arg\max_{\vec{w}} \mathrm{p}(\mathcal{A}|\vec{w}) \tag{4.43}$$

This problem can be reformulated as a MAP estimation problem using the expression:

$$\arg\max_{\vec{w}} \mathrm{p}(\mathcal{A}|\vec{w})\mathrm{p}(\vec{w}) \tag{4.44}$$

In this expression the term $\mathrm{p}(\mathcal{A}|\vec{w})$ is calculated in the same fashion as it is for ML-RSW. The difficultly in the MAP-RSW approach is devising an adequate model to represent the *a priori* model term, $\mathrm{p}(\vec{w})$. One possible solution is to model the density function of the center of mass vectors that are generated by $\mathbf{\Gamma}\vec{w}$ rather than modeling the density function of $\vec{w}$ directly. In other words, the expression in (4.44) can be rewritten as:

$$\arg\max_{\vec{w}} \mathrm{p}(\mathcal{A}|\mathbf{\Gamma}\vec{w})\mathrm{p}(\mathbf{\Gamma}\vec{w}) \tag{4.45}$$

The expression can equivalently be represented in the log domain as:

$$\arg\max_{\vec{w}} \; \log\mathrm{p}(\mathcal{A}|\mathbf{\Gamma}\vec{w}) + \log\mathrm{p}(\mathbf{\Gamma}\vec{w}) \tag{4.46}$$

Next, three assumptions are made. First, assume that all observations are independent. Second, assume each observation is dependent only on the mean vector parameters of its own class. Third, assume that the parameters of each phonetic class represented in $\mathbf{M}\vec{w}$ are independent. Although this third assumption is clearly flawed, it is necessary to keep the problem tractable and the *a priori* models trainable. With these assumptions the expression becomes:

$$\arg\max_{\vec{w}} \sum_{m=1}^{M}\left(\log\mathrm{p}(\mathbf{\Gamma}_m\vec{w}) + \sum_{n=1}^{N_m}\log\mathrm{p}(\vec{a}_{m,n}|\mathbf{\Gamma}_m\vec{w})\right) \tag{4.47}$$

Next the density functions must be defined. Both $\mathrm{p}(\mathbf{\Gamma}_p\vec{w})$ and $\mathrm{p}(\vec{a}_{m,n}|\mathbf{\Gamma}_m\vec{w})$ will be modeled with single Gaussian density functions. Thus, the observation density function is modeled as:

$$\mathrm{p}(\mathbf{\Gamma}_m\vec{w}) \equiv \mathcal{N}(\mathbf{\Gamma}_m\vec{w}, \mathbf{S}_m) \equiv \mathcal{N}(\vec{c}_m, \mathbf{S}_m) \tag{4.48}$$

Here, $\mathbf{S}_m$ represents the covariance matrix for class m, which is assumed to remain constant. These covariances can be trained from a pooled set of training speakers. Similarly, the *a priori* density function is modeled as:

$$\mathrm{p}(\vec{a}_{m,n}|\mathbf{\Gamma}_m\vec{w}) \equiv \mathcal{N}(\vec{c}^{\,ap}_m, \mathbf{S}^{ap}_m) \tag{4.49}$$

Here $\vec{c}^{\,ap}_m$ is simply the mean of all of the column vectors in $\mathbf{\Gamma}_m$ and $\mathbf{S}^{ap}_m$ is the covariance of the column vectors in $\mathbf{\Gamma}_m$. With the modeling assumptions shown above the maximization process can be reduced down to the expression:

$$\arg\max_{\vec{w}} 2\vec{v}^T\vec{w} - \vec{w}^T\mathbf{U}\vec{w} \tag{4.50}$$

Within this expression it can be shown that $\mathbf{U}$ and $\vec{v}$ can be defined using the following expression:

$$\mathbf{U} = \sum_{m=1}^{M} \mathbf{\Gamma}_m^T \left((\mathbf{S}^{ap}_m)^{-1} + N_m\mathbf{S}_m^{-1}\right)\mathbf{\Gamma}_m \tag{4.51}$$

$$\vec{v}^T = \sum_{m=1}^{M}\left((\vec{\mu}^{\,ap}_m)^T(\mathbf{S}^{ap}_m)^{-1}\mathbf{\Gamma}_m + \sum_{n=1}^{N_m}\vec{a}^{\,T}_{m,n}\mathbf{S}_m^{-1}\mathbf{\Gamma}_m\right) \tag{4.52}$$

With these new definitions, the optimal $\vec{w}$ can be found in exactly the same fashion as in ML-RSW. The MAP estimation approach to the problem provides smoothing with an *a priori* model when there is a limited amount of data from which to estimate $\vec{w}$. Unfortunately, the *a priori* model is flawed by its simplifying assumptions. One result of this is that the system provides excessive smoothing towards the *a priori* model. To correct for this, a smoothing parameter which reduces the effects of the *a priori* model can be introduced. Using a smoothing parameter ζ, which would presumably be set to some value considerably less than one, the MAP expressions for $\mathbf{U}$ and $\vec{v}^T$ can be rewritten as follows:

$$\mathbf{U} = \sum_{m=1}^{M} \mathbf{\Gamma}_m^T \left(\zeta(\mathbf{S}^{ap}_m)^{-1} + N_m\mathbf{S}_m^{-1}\right)\mathbf{\Gamma}_m \tag{4.53}$$

$$\vec{v}^T = \sum_{m=1}^{M}\left(\zeta(\vec{\mu}^{\,ap}_m)^T(\mathbf{S}^{ap}_m)^{-1}\mathbf{\Gamma}_m + \sum_{n=1}^{N_m}\vec{a}^{\,T}_{m,n}\mathbf{S}_m^{-1}\mathbf{\Gamma}_m\right) \tag{4.54}$$

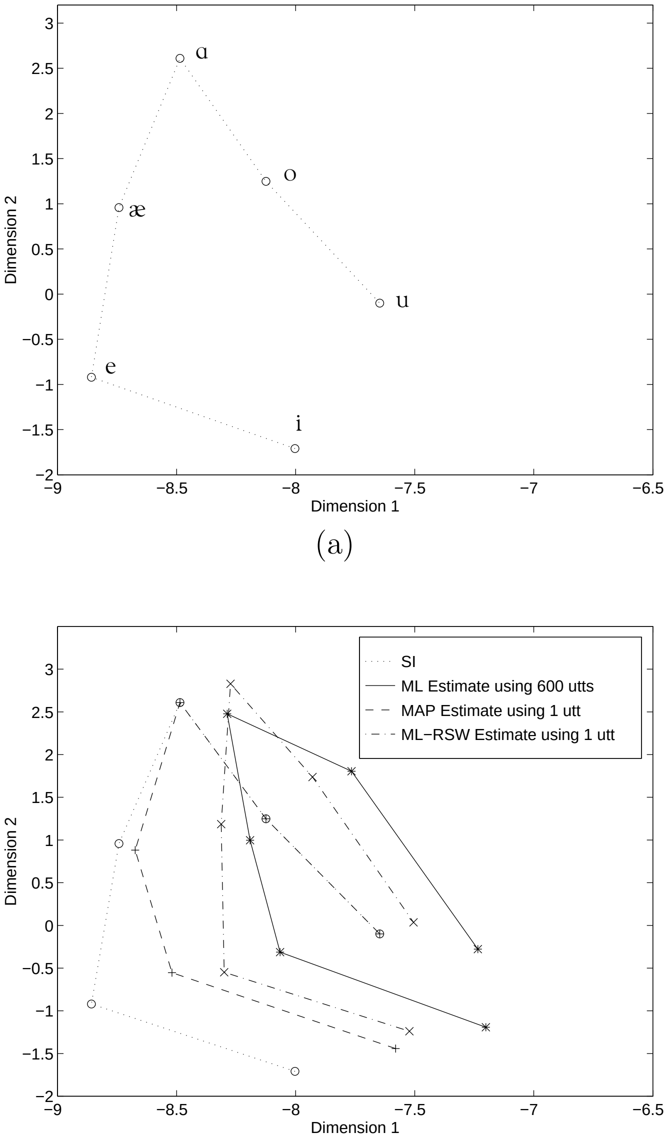
4.3 RSW Adaptation Experiments

4.3.1 Experimental Setup

To demonstrate the effectiveness of RSW adaptation, the same experimental setup as presented in Section 3.3 will be used, i.e., the algorithm will be tested with varying amounts of adaptation data on the 12 speakers in the SD portion of the RM corpus. The only additional training that must be performed is the creation of the reference speaker vectors in the matrix $\mathbf{\Gamma}$ and the MAP *a priori* parameters $\vec{c}_m^{ap}$ and $\mathbf{S}_m^{ap}$. All 149 speakers in the SI portion of the RM corpus were used as reference speakers in the creation of $\mathbf{\Gamma}$. For each speaker, a center of mass vector was created for all 60 phonetic units used by SUMMIT. Some of the training speakers did not have observations in their training utterances for one or more of the phonetic units. In these cases, the appropriate gender dependent center of mass was plugged in as a substitute vector. Once $\mathbf{\Gamma}$ is created, the *a priori* parameters $\vec{c}_m^{ap}$ and $\mathbf{S}_m^{ap}$ are trained directly from the vectors in $\mathbf{\Gamma}$.

4.3.2 RSW Center of Mass Estimation

The main advantage of RSW adaptation is that it adapts the models of all phones regardless of whether or not these phones have been observed in the adaptation data for the speaker. This is illustrated in Figure 4.1. In the figure, plot (a) a shows the locations of the SI center of mass estimates for two measurements as extracted from six different vowels ([i], [e], [æ], [ɑ], [o], and [u]). In plot (b) of the figure, the center of mass estimates are shown for four different conditions. The first condition is the same six SI estimates shown in plot (a). The second condition shows the center of mass locations from the SD model for speaker CMR0 as estimated from 600 utterances. Ideally, the center of mass locations should move from the SI estimates towards the ML estimates using as few adaptation utterances as possible. The third condition shows the result of MAP estimation as obtained from one adaptation utterance from speaker CMR0. As can be seen in the plot, the MAP estimates for the phones [i], [e], and [æ] have moved away from the SI locations towards the ML estimates. On the other hand, the estimates for [ɑ], [o], and [u] have not moved, indicating that the adaptation algorithm has not observed any exemplars of these phones in the adaptation utterances. The fourth condition shown is the ML-RSW estimates obtained from the same utterance used for MAP adaptation. Using ML-RSW adaptation all of the six phones, including the three for which no adaptation data has been observed, have moved away from the SI estimates towards the ML estimates. This demonstrates how the constraining power of the RSW algorithm can improve rapid adaptation performance over that of standard MAP estimation.



(a)

(b)

Figure 4.1: Location of center of mass for six different vowels ([i], [e], [æ], [ɑ], [o], and [u]) for the following estimates: (1) the SI model, (2) the ML estimate from 600 adaptation utterances, (3) the MAP estimate from one adaptation utterance, and (4) the RSW estimate from one adaptation utterance.

4.3.3 ML-RSW With Variable Classes

In Section 3.4.4, transformation approaches to adaptation were discussed. One key element of these approaches was that the number of transformations could be increased or decreased depending on the amount of available adaptation data. This same idea can be applied to RSW adaptation. As the amount of adaptation is increased the number of different weighting vectors can be increased. Each weighting vector could be optimized on a different class of phonetic units. For example, if two weighting vectors are used, the first could be used to adapt the model parameters for the set of vowels while the second could be used for consonants.

When the phonetic units are split into multiple classes, it is desirable for the phones in the same class to exhibit high within-speaker correlation while phones in different classes should exhibit low within-speaker correlation. To achieve this the phones can be clustered using the hierarchical clustering method presented in Section 2.2.2. When testing RSW, three different sets of classes will be presented. The first set simply places all phones in to one global class. The second set breaks the phones into four separate classes using the clustering in Figure 2.2 as a guide. The four classes roughly correspond to (1) vowels, (2) strong fricatives and affricates, (3) nasals, and (4) stops, weak fricatives and other consonants. The third set uses seven classes which roughly correspond to: (1) low and retroflexed vowels, (2) mid and high vowels, (3) strong fricatives and affricates, (4) nasals, (5) labial stops, (6) alveolar and velar stops, and (7) other consonants.

To test the capabilities of RSW adaptation, the algorithm was evaluated as the number of adaptation algorithms was varied from 1 to 600. Figure 4.2 shows the performance of ML-RSW adaptation using one, four and seven phonetic classes. When only one adaptation utterances is available, ML-RSW performs best using only one global weighting vector as opposed to four or seven. This is because one adaptation utterance does not provide enough data to accurately estimate the increased number of parameters present when more than one class is used. However, as the amount of adaptation data increases, the number of classes can be increased. As more classes are introduced, the amount of constraint is reduced giving the algorithm the ability to better match the characteristics of the current speaker. This is evidenced by the fact that ML-RSW using seven classes outperforms ML-RSW using either four classes or one class when 5 or more adaptation utterances are available.

In the extreme, ML-RSW could use a different weighting vector for each phone. In this case, ML-RSW estimation would be nearly identical to ML estimation with one exception. The estimates learned by ML-RSW would still be constrained to be an interpolation of the parameters of the reference speaker. If the test speaker's parameters fell within the constrained parameter space belonging to the reference speakers then ML-RSW estimation and ML estimation would produce the same estimate.

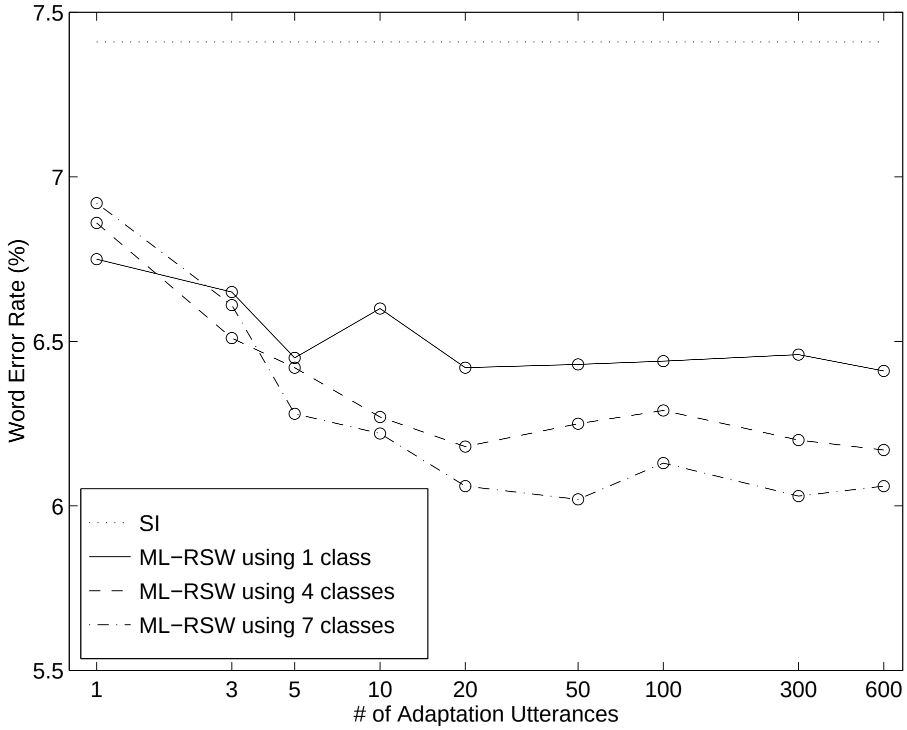


Figure 4.2: Performance of RSW model translation using different numbers of classes.

4.3.4 MAP-RSW vs. ML-RSW

In Section 4.2.4, an implementation of MAP-RSW was introduced. Even when only one global weighting vector is utilized, it is possible that the adaptation data may not contain enough observations to provide suitable center of mass estimates. In this case is is wise to introduce the *a priori* knowledge provided by the MAP formulation of the problem. For this experiment MAP-RSW was performed using a smoothing factor of $\zeta = 0.03$. With sixty phones in the weighting vector this means that the *a priori* information carries the equivalent weight of two adaptation observations (i.e., the *a priori* information and the adaptation data each contribute about equally when only two adaptation observations are present). As the number of adaptation observations becomes much larger than two, the ML-RSW estimates begins to dominate. This is demonstrated in Figure 4.3. In this figure, MAP-RSW slightly outperforms ML-RSW when there is only one adaptation utterance, but the performance of MAP-RSW converges with the performances of ML-RSW when the number of utterances is three or more. As in other interpolation schemes, if ζ is set too high then the system backs off to readily to the *a priori* estimate. If ζ is set too low then the ML estimate dominates too soon (i.e., before its estimate can be deemed reliable).

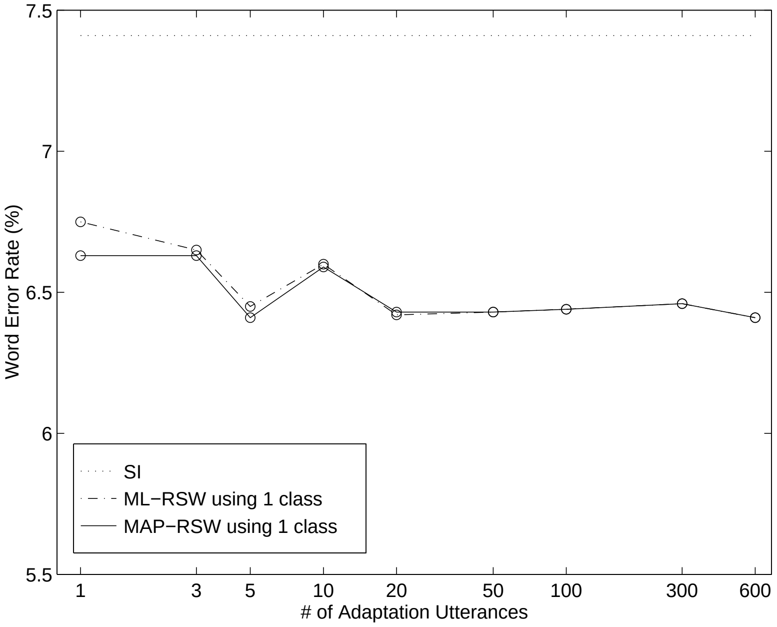


Figure 4.3: Performance of MAP-RSW and ML-RSW as the number of adaptation utterances is varied.

4.3.5 Combining RSW with Other Adaptation Methods

RSW adaptation provides a method for incorporating speaker constraint into a model translation adaptation routine. This makes RSW adaptation especially useful for rapid adaptation. However, the constraints imposed by RSW adaptation prohibit the RSW estimates from ever converging with their corresponding ML estimates. As such a method for combining RSW with other adaptation methods is needed.

Figure 4.4 shows the performances of MAP-RSW model translation using one global weighting vector versus MAP model translation adaptation and model interpolation adaptation. As can be seen in the figure, MAP-RSW model translation achieves an error rate of 6.6%, as compared to 6.9% for MAP model translation and 7.4% for the SI model, when only one adaptation utterance is used. However, the performance of MAP-RSW adaptation converges quickly to 6.4% as the amount of adaptation data is increased. As the number of adaptation utterances increases, MAP model translation improves steadily and surpasses MAP-RSW when 10 or more adaptation utterances are available. Likewise, model interpolation improves at a slower rate but eventually surpasses MAP model translation when more than 100 adaptation utterances are available.

As seen in Figure 4.4 each of the three methods has advantages and disadvantages. The model interpolation method provides adaptation which converges to the standard ML estimated models as the number of adaptation utterances increases to a sufficiently large number. However, this method is slow to adapt when the amount of adaptation data is small. MAP model translation adapts faster than model interpolation because it adapts only the locations of the SI models and not their shape. As a result, MAP model translation adapts more efficiently when the amount of adaptation data is small, but it fails to asymptote to the ML estimate when there is a large amount of adaptation data available. MAP-RSW model translation also provides a means for rapid adaptation because it accounts for the correlations between the different phone models in its formulation. However, RSW also has poor asymptotic properties and is incapable of learning the exact values of a model's center of mass parameters. By combining the three models in an appropriate fashion the strengths of each model can be incorporated together to create a model which (1) adapts rapidly when limited data is available and (2) has the proper asymptotic behavior when the amount of adaptation data increases.

In examining the number of parameters that must be trained from the adaptation data, it is clear that a small number of well chosen general parameters are best when the amount of adaptation data is small, while a large number of specific parameters are best when the amount of adaptation data is large. This belief is confirmed by our preliminary results. The MAP-RSW model translation approach requires that only 149 parameters (one weight for each reference speaker) be trained from the

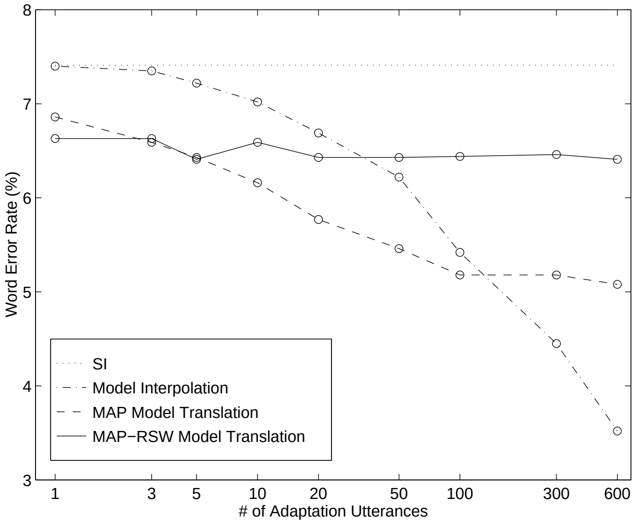


Figure 4.4: Performance of SI models which are translated using RSW vs. previous methods discussed in this paper.

adaptation data. The MAP model translation approach require that 2160 center of mass parameters (36 parameters for each of 60 different phones) be trained from the adaptation data. The standard ML training method requires that up to 90,000 mixture Gaussian parameters be trained for any given speaker.

In combining the different methods of adaptation two primary steps are performed. The first primary step is to translate the SI models using center of mass estimates obtained from a combination of the MAP-RSW center of mass and the ML center of mass estimates. These estimates are combined using a modified MAP estimation approach. Specifically, an RSW center of mass estimate, $\vec{c}_m^{rsw}$, is interpolated with the ML center of mass estimate, $\vec{c}_m^{ml}$, to yield the final model translation center of mass estimate, $\vec{c}_m^{mt}$, using the following equation:

$$\vec{c}_m^{mt} = \mathbf{S}_m \left(\rho N_m \mathbf{S}_m^{ap} + \mathbf{S}_m\right)^{-1} \vec{c}_m^{rsw} + \rho N_m \mathbf{S}_m^{ap} \left(\rho N_m \mathbf{S}_m^{ap} + \mathbf{S}_m\right)^{-1} \vec{c}_m^{ml} \tag{4.55}$$

There are only two differences between this equation and the equation used in MAP model translation, Equation (3.27). First, the SI center of mass estimate is replaced with the MAP-RSW center of mass estimate. Second, an interpolation rate factor, ρ, is introduced to prevent the final estimate from moving towards the ML estimate too quickly. This is necessary because the RSW estimate is also moving away from SI model towards the ML estimate. For these experiments, a value of 2.5 was used for ρ.

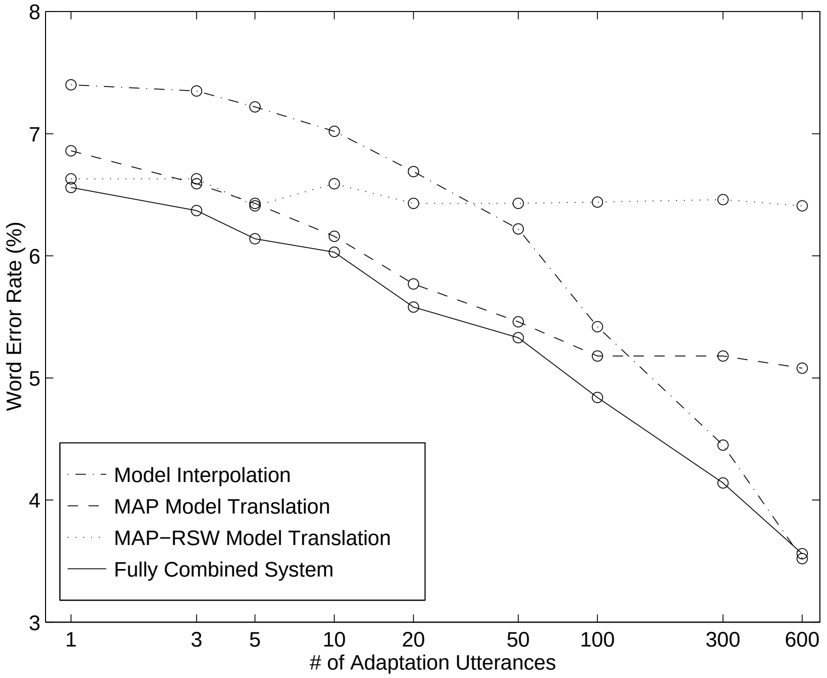


Figure 4.5: Performance of fully combined system

The second primary step in creating the final adapted model is the combination of the translated SI models with the ML estimated models. These models are combined using the same simple interpolation scheme presented in Equation (3.20). The interpolation in this case uses the following equation:

$$p_{sa}(\vec{a}) = \frac{N}{N+K}\, p_{ml}(\vec{a}) + \frac{K}{N+K}\, p_{mt}(\vec{a}) \tag{4.56}$$

In the above equation $p_{mt}(\vec{a})$ represents the probabilistic model obtained from the model translation method described above. For our experiments a value of 1000 was used for K.

Figure 4.5 shows the results using the fully combined system. As can be seen in the figure, the fully combined system takes advantage of the strengths of each of the 3 different adaptation methods that it utilizes. When the number of utterances is small the RSW model translation contributes the most to the system. As the number of utterances increases to 10 and above, the system relies primarily on the MAP model translation method. Finally, for large numbers of utterances, the system performance converges with the performance of the ML estimated model.

4.4 Summary

This chapter has presented a novel adaptation algorithm called reference speaker weighting. This algorithm incorporates speaker constraints into a model translation adaptation scheme. As seen in Figure 4.5, the use of the reference speaker weighting algorithm speeds up the adaptation process of standard algorithms such as MAP model translation. Unfortunately, the improvements obtained by RSW are relatively small. The improvements provided by RSW are also greatest when the number of adaptation utterances is small. By the time 10 or more adaptation utterances have been observed, RSW provides very little improvement over standard techniques which do not utilize any speaker correlation information. This can be explained by the fact that exemplars of almost all of the different phones have been observed after 10 adaptation utterances have been presented to the system. As such, the remainder of this thesis will concentrate on methods to effectively incorporate speaker correlation information into *rapid* or *instantaneous* adaptation routines.

Chapter 5

Speaker Clustering

5.1 Overview

In Chapter 4 the reference speaker weighting (RSW) algorithm was introduced. This algorithm incorporates speaker constraint into a speaker adaptation routine by forcing a set of speaker adapted parameters to be a weighted interpolation of the parameters of individual reference speakers. The RSW algorithm presented in Chapter 4 is limited because it only adapts the center of mass parameters of the SI model. Ideally an adaptation algorithm should adapt the *shape* of an acoustic model's density function as well as its *location*.

One potential means of addressing this problem is to reformulate the RSW approach so that the weighting vector $\vec{w}$ is applied to density functions instead of center of mass parameters. For example, if there are R different reference speakers, and each reference speaker has M different phone-based acoustic models, then the density function for the m^{th} phone for the r^{th} speaker can be expressed as $\mathrm{p}_r(\vec{x}|\, p{=}m)$. Using this definition, the speaker adapted version of the density function for the m^{th} phone can be expressed as a weighted combination of the density functions from all R reference speakers as follows:

$$\mathrm{p}_{sa}(\vec{x}|\, p{=}m) = \sum_{r=1}^{R} w_r \mathrm{p}_r(\vec{x}|\, p{=}m) \tag{5.1}$$

There exist simple algorithms, such as the expectation maximization (EM) algorithm, which can be used to find an optimal weighting of the reference speakers in an approach such as this.

Despite the simplicity of RSW density function interpolation, this type of approach has one main problem. In typical corpora of today there is seldom enough data to create reliable speaker dependent models for all of the reference speakers available in the training data. RSW center of mass adaptation is effective because only the center of mass variables for each speaker needs to be estimated reliably. RSW density function interpolation is currently not feasible because of the sparseness of the available data per speaker in today's corpora.

One potential solution to this problem is the use of speaker clustering. In this approach, similar reference speakers are grouped together into a speaker cluster for which one model is trained. When using speaker clustering, there is a trade-off between robustness and specificity. Large clusters are more general but can be trained more robustly. Smaller clusters can represent more specific speaker types but may lack a sufficient amount of training data required for accurate density function estimation.

In this chapter, two speaker clustering algorithms will be presented. Both algorithms utilize the same set of speakers clusters. The first algorithm is based on traditional hierarchical speaker clustering approaches. The second algorithm follows the spirit of reference speaker weighting and will be called *speaker cluster weighting* (SCW). All of the experiments conducted in this chapter were performed on the full SI evaluation set (40 speakers, 1200 utterances) of the Resource Management corpus. The SI training and development sets (109 speakers, 3990 utterances) were used for all training. The context independent SUMMIT recognizer was used in all of the experiments.

5.2 Hierarchical Speaker Clustering

5.2.1 Creating the Tree Structure

The basic idea behind hierarchical speaker clustering is that speakers can be classified into specific speaker types which can be organized within a hierarchical tree structure. The root node of the tree contains all of the training speakers. These speakers are then subdivided into different branches. Any branch node in the tree can also be recursively subdivided. In the extreme, the leaves of the tree represent individual speakers.

There are a variety of ways in which the speaker clustered tree can be constructed. The construction can be performed using unsupervised bottom-up clustering based on an acoustic similarity measure [47, 48], unsupervised top-down clustering based on an acoustic similarity measure [23, 62], or some supervised method. In the experiments presented here, the hierarchical tree is manually created. The data is first divided by gender and then subdivided by speaking rate. Three different speaking rate classifications are utilized: fast, medium and slow. Figure 5.1 illustrates the hierarchical tree that is constructed for these experiments. By creating the tree in this fashion, two of the main sources of acoustic variability, the speaker's gender and speaking rate, can be accounted for in the acoustic models of the tree's leaves.

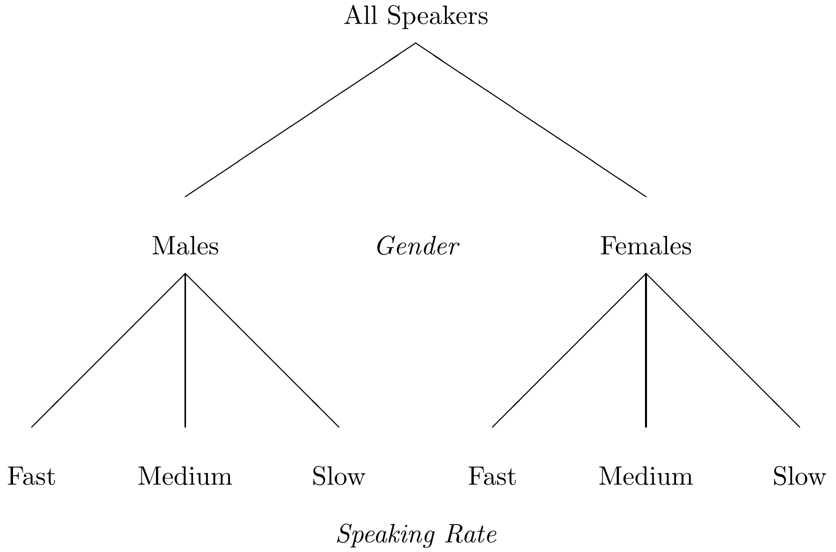


Figure 5.1: Manually created hierarchical speaker cluster tree.

In creating the tree, it is important that each leaf node contains enough speakers to be able to create a robust model. In order to do this, the tree that is utilized in these experiments is not a strict hierarchical tree. Some overlap of speakers between different leaf nodes was permitted. When the fast, medium and slow speaking rate nodes where created, some of the training speakers are shared by two different nodes. Specifically, the fast speaker node contained speakers with a speaking rate measure of $\bar{r} > 0.0$. The medium speaking rate nodes contained speakers with a speaking rate measure of $-0.15 > \bar{r} > 0.15$. The slow speaking rate nodes contained speakers with a speaking rate measure of $\bar{r} < 0.0$. The average speaking rate measure that is used here is the same measure introduced in Section 2.3.3 of Chapter 2. With this breakdown of the speakers, all of the medium speakers are also shared by the either the fast speaker node or the slow speaker node. In all, roughly 50% of the speakers are shared between two different leaf nodes.

5.2.2 Creating the Node Models

The first step in creating a set of models for each node in the tree is to utilize standard maximum likelihood training on the data from the speakers of each node. The nine different models in the tree will be referred to as the SI model, the gender dependent (GD) models (one for males and one for females), and the gender and speaker rate dependent (GRD) models (one each for fast males, medium males, slow males, fast females, medium females and slow females). Unfortunately, subdividing the speakers into smaller clusters reduces the amount of available training data for the models in the more specific clusters. This causes the tradeoff between robust general models and insufficiently trained specific models. One possible means of capitalizing on the robustness of general models and the specificity of cluster models is to create models which interpolate between these models. For example, an interpolated gender dependent (IGD) model for a specific phonetic model m can be created by combining a gender dependent model (GD) with an SI model as follows:

$$\mathrm{p}_{igd}(\vec{x}|\,p{=}m) = \lambda\mathrm{p}_{gd}(\vec{x}|\,p{=}m) + (1-\lambda)\mathrm{p}_{si}(\vec{x}|\,p{=}m) \tag{5.2}$$

Similarly, an interpolated gender and speaking rate (IGRD) dependent model can be created using the expression:

$$\mathrm{p}_{igrd}(\vec{x}|\,p{=}m) = \lambda_1\mathrm{p}_{grd}(\vec{x}|\,p{=}m) + \lambda_2\mathrm{p}_{gd}(\vec{x}|\,p{=}m) + (1-\lambda_1-\lambda_2)\mathrm{p}_{si}(\vec{x}|\,p{=}m) \tag{5.3}$$

There are several potential methods for determining the λ values. The method used in this thesis is called *deleted interpolation* [6, 37]. Deleted interpolation optimizes the λ values by maximizing the likelihood of data jack-knifed from the training set using the EM algorithm. A full description of the deleted interpolation algorithm is provided in Appendix D. Using the deleted interpolation algorithm, each phone model receives a different set of interpolation weights. If a particular speaker cluster has plenty of data to reliably estimate the density function for a particular phone, then the interpolation weights would favor the more specific cluster model. On the other hand, if a cluster contained only a small amount of data for a particular phone, then the interpolation weights could place emphasis on the more general model.

To provide an idea of the magnitude of the λ weights that are found by the deleted interpolation algorithm, the interpolated gender dependent models can be examined. As discussed above a different λ weight was trained for every phone. High values of λ indicate that the interpolated model relies more heavily on the GD model than on the SI model. For the male interpolated model the value of λ varied from a low of .429 for the phone [ɔʸ] to a high of .626 for the phone [ʊ]. For the female interpolated model the value of λ varied from a low of .321 for the phone [θ] to a high of .738 for the phone [ž]. The smaller range in the values of λ for the male interpolated models can be attributed to the fact that the SI model is already dominated by male speakers (78 males to 31 females). In general within each gender, vowels and strong fricatives received higher values for λ while stops, closures, and weak fricatives received lower values for λ. This indicates that gender differences are more prominent in the production of vowels and strong fricatives than they are in the production of stops and weak fricatives.

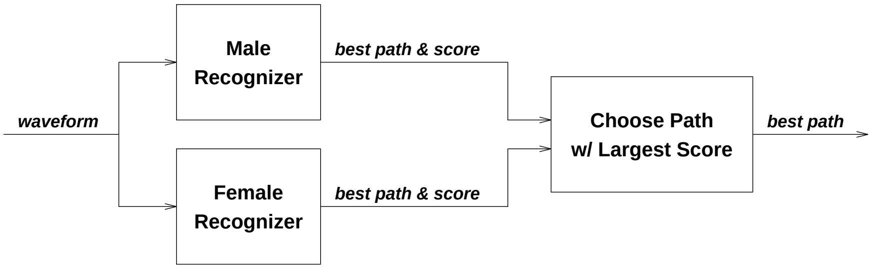


Figure 5.2: Diagram of a standard parallelized gender dependent recognition system.

5.2.3 Recognition Experiments

Gender Dependent Recognition

The first step required during recognition utilizing the tree-based cluster models shown in Figure 5.1 is identifying the gender of the speaker. While there are many acoustic properties that can be examined to distinguish between male and female speech, there may be no need to utilize a specialized algorithm for gender identification if the pre-existing gender dependent models can provide accurate gender identification. Systems which employ gender dependent modeling typically run male and female recognizers in parallel and determine the gender of the speaker by selecting the recognizer which produces the highest scoring best path. This procedure is demonstrated in Figure 5.2.

Using the standard process for parallel gender dependent recognition as detailed in Figure 5.2 presents potential problems when using the SUMMIT system. Specifically, the models which produce the best gender dependent recognition results do not necessarily possess good gender identification capabilities. Table 5.1 shows the performance of several different gender dependent model sets on the tasks of gender identification and word recognition. In creating the model sets used during testing, the 60 phonetic segment models and the anti-phone model could each be extracted from either the speaker independent (SI) models, the gender dependent (GD) models, or the interpolated gender dependent (IGD) models. Gender identification is performed by examining the scores of the highest scoring word string produced by each gender dependent recognizer. The word accuracy is calculated on recognition trials which always utilize the correct gender model.

There are two key points to observe in the results shown in Table 5.1. First, the word recognition performance on female speakers is clearly worse than the performance on male speakers. This is due to the disparity in the number of male and female speakers in the training set. Thus, the SI model is dominated by male speakers. Not unexpectedly, the female speakers had a larger improvement in accuracy

Model Set	Phone Models	Anti-Phone Models	Gender ID Error Rate	Word Error Rate		
				Males	Females	Overall
1	SI	SI	—	7.7%	9.7%	8.6%
2	GD	GD	1.08%	7.7%	8.8%	8.2%
3	GD	SI	0.67%	8.1%	9.3%	8.6%
4	IGD	IGD	16.9%	7.3%	8.4%	7.8%
5	IGD	GD	58.9%	7.2%	8.3%	7.7%
6	IGD	SI	0.33%	7.5%	8.6%	8.0%

Table 5.1: Performance of different model sets

when the GD models were used instead of the SI models.

The second key point is that gender dependent anti-phone models are not useful for performing gender identification even though they help improve word recognition accuracy. If both the male and female recognizers use different anti-phone models for normalization, their scores are not comparable and can not be used for accurate gender identification. As can be seen in the table, model sets 3 and 6 are the best for gender identification because the SI anti-phone model is used by both the male and female recognizers. On the other hand, model set 5 is the best model set to use for gender dependent recognition. The fact that one model set performs best at gender identification while a second model set performs best at word recognition means that the recognition strategy used in Figure 5.2 is not acceptable when using the SUMMIT system. The optimal routine using the given models is to first run the male and female recognizers using the IGD phone models and SI anti-phone model in parallel and choose the model set with the highest score to identify the gender. Next, the appropriate gender dependent model using IGD phone models and GD anti-phone models can be used to perform the actual recognition.

An alternative gender identification routine is presented in Figure 5.3. In this routine a single SI recognizer is run instead of two parallel gender dependent models. The best path is then rescored by the IGD phone models for each gender. The path does not need to utilize the anti-phone model during rescoring since the same path is being rescored by both IGD models. The system then selects the gender of the IGD phone models that produce the highest score. The error rate of this method is only 0.67% and this method is more efficient than any method which must run parallel gender dependent recognizers in order to identify the gender.

Achieving perfect gender identification is actually not necessary. On the handful of utterances for which the system has difficulty identifying the gender, the recognition accuracy is not affected by the choice of the wrong gender dependent recognizer. On these utterances, the incorrect IGD recognizer generally produces the same word

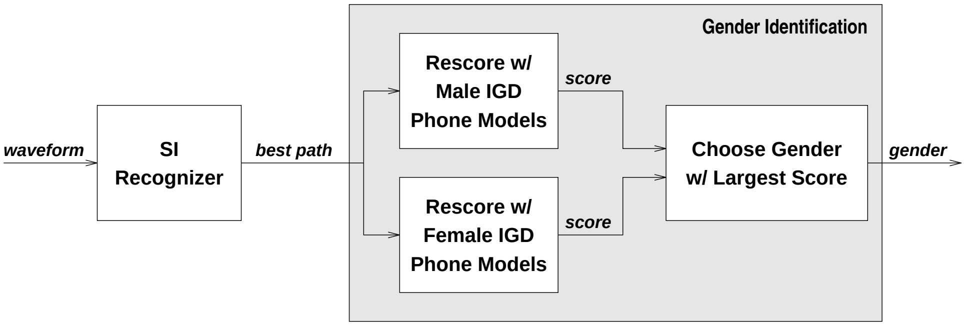

Gender Identification
waveform
SI Recognizer
best path
Rescore w/ Male IGD Phone Models
Rescore w/ Female IGD Phone Models
score
score
Choose Gender w/ Largest Score
gender

Figure 5.3: Diagram of a gender identification routine.

string as the correct IGD recognizer, and the overall recognition accuracy is the same over these utterances regardless of the IGD recognizer used. It is reasonable to assume that the gender dependent models do not generalize well to these particular speakers and the system relies heavily on the contribution of the SI models used in the IGD models when producing the best path, regardless of which gender model is selected.

The word error rate of the best gender dependent model set is 7.7%. This is a significant improvement over the word error rate of 8.6% achieved by the speaker independent system. In total the error rate is reduced by 10.5% relative to the SI system when using the best GD system.

Speaking Rate Dependent Recognition

Although the final system will be designed around the hierarchical tree shown in Figure 5.1, speaking rate dependent (RD) models can be examined without utilizing gender information. In this case the training data is used to create three different speaking rate dependent sets of models. The three models correspond to fast, medium, and slow speech and are created using the same deleted interpolation procedure described earlier with the only exception being that the training speakers are not subdivided by gender first.

In order to utilize RD models, an estimate of the speaking rate for the current test utterance must be determined in an unsupervised fashion. The same speaking rate measure described in Section 2.3.3 of Chapter 2 can be utilized. This speaking rate measure requires knowledge of the phonetic string of the utterance. Because the underlying phonetic string is unknown, the exact average speaking rate of the utterance can not be determined. However, it can be estimated from the phonetic string of the best path produced by the SI recognizer. Though the best path may contain errors, it is hoped that these errors will not drastically effect the estimate of the speaking rate.

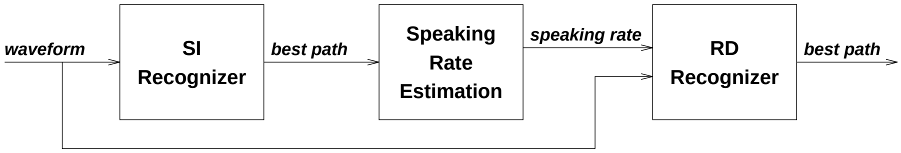


Figure 5.4: Diagram of the speaking rate dependent (RD) recognition system.

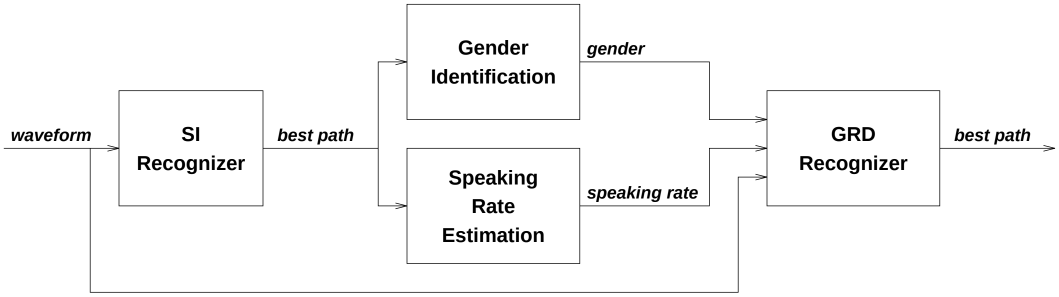


Figure 5.5: Diagram of the full gender and speaking rate dependent (GRD) recognition system.

Figure 5.4 diagrams the RD system used in these experiments. The RD system classifies the speaking rate as fast if $\bar{r} > 0.1$, medium if $-0.1 < \bar{r} < 0.1$, and slow if $\bar{r} < -0.1$. Using this approach the speaking rate dependent (RD) system achieves an error rate of 8.0%. This is relative error rate reduction of 6.5% from the SI system.

Full Tree Recognition

Both the gender and the speaking rate of the speaker can be accounted for using the hierarchical tree in Figure 5.1. To use this tree, the system must first determine the gender and speaking rate of the speaker and then choose the appropriate model from the leaves of the cluster tree. As in GD and RD recognition this can be accomplished with a two-pass recognition scheme. The first recognition pass is performed by the standard SI recognizer. The best path output of the SI recognizer can then be utilized for gender identification and speaking rate estimation. The second recognition pass would then be performed utilizing the appropriate gender and speaking rate dependent (GRD) model. Figure 5.5 diagrams the steps of this recognition process. Using this approach, the GRD system is able to achieve a recognition accuracy of 7.2%. This is a relative error rate reduction of 16.4% from the SI system.

5.3 Speaker Cluster Weighting

When using hierarchical speaker clustering, recognition is performed using a model selected from a finite set of predetermined models. The models in the set are themselves interpolations of various general and specific models. The weightings used to perform the interpolation are also predetermined using the deleted interpolation algorithm. An alternative approach is an interpolation scheme which determines the weighting factors on the fly to match the current speaker. This is the basic idea behind speaker cluster weighting (SCW).

In speaker cluster weighting, a set of predetermined models exist. The final SCW model is a weighted combinations of these models. Let $p_l(\vec{x}|\, p\!=\!m)$ represented the acoustic model from model set l for phonetic model m. If there are L different model sets then the final SCW model for phone m is a weighted combination of the L different models as represented by:

$$\mathrm{p}_{scw}(\vec{x}|\, p\!=\!m) = \sum_{l=1}^{L} w_l \mathrm{p}_l(\vec{x}|\, p\!=\!m) \tag{5.4}$$

The difficult part of the problem is to determine the values for each w_l weight. This process is very similar in nature to the reference speaker weighting problem in Chapter 4. As with RSW, one global set of weights can be used, or a different set of weights can be utilized for different classes of phones. For each class of phones the goal is to find the set of weights which maximizes the likelihood of the adaptation data for that class for the current speaker. To illustrate the SCW process, consider the problem of finding the single optimal set of global weights. The problem is cast in a maximum likelihood framework as follows:

$$\vec{w}' = \arg\max_{\vec{w}} \mathrm{p}_{csw}(X|P, \vec{w}) \tag{5.5}$$

Here, the weights are represented in the weighting vector $\vec{w}$ as follows:

$$\vec{w} = \begin{bmatrix} w_1 \\ \vdots \\ w_L \end{bmatrix} \tag{5.6}$$

By assuming each observation is independent of other observations and its surrounding context, this maximization process becomes:

$$\vec{w}' = \arg\max_{\vec{w}} \prod_{n=1}^{N} \mathrm{p}_{scw}(\vec{x}_n|p_n, \vec{w}) \tag{5.7}$$

This maximization process is easily performed by the EM algorithm.

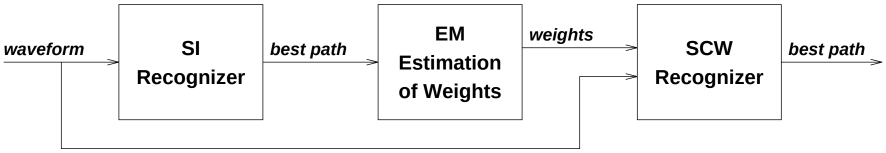


Figure 5.6: Diagram of the speaker cluster weighting (SCW) recognition system.

Class Set	Word Error Rate
1	7.3%
2	7.0%
3	6.9%
4	7.5%

Table 5.2: Performance of SCW recognition using different phonetic class sets.

To perform the maximization process for finding the optimal weights, a phonetic transcription must be provided. The phonetic transcription from the best path provided by the SI recognizer can be used to approximate the true phonetic transcription. Using this approach the full recognition process is demonstrated in Figure 5.6.

There are two finals steps in constructing an SCW system. The first step is determining the set of cluster models used by the SCW algorithm. The set of models used in these experiments contain the same nine ML trained models appearing at the nine nodes of the hierarchical tree shown in Figure 5.1. In other words, the model set contains one SI model, two GD models, and six GRD models.

The second step is determining the different phonetic classes, each of which will receive a different weighting vector. The four different sets of classes used in these experiments are defined as follows: (1) all classes utilize one global weighting vector, (2) all segment models utilize one weighting vector while the anti-phone model uses a different weighting vector, (3) different weighting vectors are used for phonetic models, silence models, and anti-phone models, and (4) vowel, consonant, silence and anti-phone models each receive a different weighting vector.

SCW recognition was performed using each of the four different sets of classes listed above. A different optimal set of weights was determined for each class in a set. The results are shown in Table 5.2. The optimal set of classes (one phonetic segment class, one silence class, and one anti-phone class) produced an error rate of 6.9% which is better than the 7.2% error rate produced by the hierarchical clustering. It is also a relative error rate reduction of 18.9% from the SI system.

5.4 Summary

This chapter has presented two different methods for recognition using speaker clustering: hierarchical speaker clustering and speaker cluster weighting. Table 5.3 shows the performances of the various different clustering approaches examined in this chapter. This table contains results for the speaker independent (SI) system, the speaking rate dependent (RD) system, the gender dependent (GD) system, the gender and speaking rate dependent (GRD) system, and the speaker cluster weighting (SCW) system. The SCW system performed the best, probably because of it's ability to adapt its interpolation weighting factors to the current speaker on the fly instead of having them predetermined as in the other methods. In conclusion, this chapter has shown that speaker clustering methods are a highly effective way of providing speaker constraint to a speech recognition system as evidenced by the 18.9% reduction in error rate provided by the best of these approaches.

	Total	Word	Error Rate
Method	Errors	Error Rate	Reduction
SI	882	8.6%	—
RD	825	8.0%	6.5%
GD	789	7.7%	10.5%
GRD	737	7.2%	16.4%
SCW	715	6.9%	18.9%

Table 5.3: Summary of recognition results using various speaker clustering approaches.

Chapter 6

Consistency Modeling

6.1 Overview

In Chapter 1, the acoustic modeling problem was presented and the *consistency ratio* was introduced. In standard speech recognition systems which assume that all observations are independent, the *consistency ratio* represents all of the correlation information which is disregarded by the independence assumption. It is reasonable to suspect that standard SI recognition can be improved if the speaker constraining information that is contained in the consistency ratio can be utilized. The difficulty lies in determining a feasible method for estimating the *consistency ratio*. This chapter will focus on this issue. In particular the goals of this chapter are:

- Present the derivation of the consistency model approach.
- Discuss the engineering issues involved in estimating the consistency ratio.
- Present potential techniques for creating consistency models.
- Analyze the capabilities of consistency modeling under different conditions.
- Report results on a speaker independent recognition task.

To remain consistent with the experiments presented in Chapter 5, the experiments conducted in this chapter are trained and evaluated using the same set of data. In particular, the system is evaluated on the 40 speaker SI evaluation set of the Resource Management corpus. The data from the 109 speakers in the SI training and development sets is used for all training including the training of the consistency models. Recognition is once again performed using the context independent version of SUMMIT.

6.2 Probabilistic Framework

Before describing the consistency modeling approach, the probabilistic framework for acoustic modeling that was introduced in Chapter 1 should be re-examined. Consider the task of scoring a sequence of N acoustic measurements. In a segment-based approach a measurement vector is created for each of N segments from the underlying acoustic information. The sequence of measurement vectors for a particular set of N segments can be represented as:

$$X = \{\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_N\} \tag{6.1}$$

For each particular set of N segments, a string of N phonemes can be hypothesized. This string can be represented as:

$$P = \{p_1, p_2, \ldots, p_N\} \tag{6.2}$$

Given a particular set of segments, the goal of an acoustic model is to estimate the likelihood that a particular string of phones could have produced the observed acoustic information. Thus, the acoustic model can be represented in probabilistic terms and expanded as follows:

$$\mathrm{p}(X|P) = \mathrm{p}(\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_N|P) = \prod_{n=1}^{N} \mathrm{p}(\vec{x}_n|\vec{x}_{n-1}, \ldots, \vec{x}_1, P) \tag{6.3}$$

At this point typical speech recognition systems assume that the acoustic observations are independent of each other. This assumption allows the acoustic model to be simplified as follows:

$$\prod_{n=1}^{N} \mathrm{p}(\vec{x}_n|\vec{x}_{n-1}, \ldots, \vec{x}_1, P) = \prod_{n=1}^{N} \mathrm{p}(\vec{x}_n|P) \tag{6.4}$$

As discussed in Chapter 1 a second major assumption that is often utilized is the context independence assumption. With this assumption, the acoustic information for the current segment is considered independent of the preceding and succeeding phonetic contexts. This allows the acoustic model to be simplified further as:

$$\prod_{n=1}^{N} \mathrm{p}(\vec{x}_n|P) = \prod_{n=1}^{N} \mathrm{p}(\vec{x}_n|p_n) \tag{6.5}$$

These independence assumptions are made for two primary reasons. First, sound modeling methods have not been developed to capture the inherent correlations between acoustic segments produced by one speaker. Second, when all segment observations are considered independent of each other and their surrounding contexts, efficient search mechanisms, such as the Viterbi search, can be used to decode the underlying phoneme sequence.

Effective methods for capturing contextual information have led to the development of systems which perform context dependent recognition. Systems which utilize a phone's left and right context are represented by the following probabilistic framework:

$$\prod_{n=1}^{N} \mathrm{p}(\vec{x}_n|P) = \prod_{n=1}^{N} \mathrm{p}(\vec{x}_n|p_{n-1}, p_n, p_{n+1}) \tag{6.6}$$

The creation of context dependent systems has relied on the development of methods addressing the computational issues of the search. In order to be run efficiently within the Viterbi search the contextual information is typically encoded directly into a phone model's identity. Methods for pruning the search space must also be used to avoid computing the scores of context dependent models that are unlikely. The use of context dependent models has also been spurred by the development of deleted interpolation smoothing algorithms which allow sparsely trained context dependent models to be smoothed with their context independent counterparts to improve their robustness.

As with context dependent modeling, eliminating the independence of segments assumption requires that two primary issues be addressed. First, how can the information about the correlation between segments be captured effectively and robustly. Second, how can this information be incorporated efficiently into the search mechanism of a speech recognition system.

To begin the derivation of the consistency modeling approach, consider the expression:

$$\mathrm{p}(X|P) = \prod_{n=1}^{N} \mathrm{p}(\vec{x}_n|\vec{x}_{n-1}, \ldots, \vec{x}_1, P) \tag{6.7}$$

Bayes' rule can be used to rewrite the probability terms in this expression as follows:

$$\mathrm{p}(\vec{x}_n|\vec{x}_{n-1}, \ldots, \vec{x}_1, P) = \mathrm{p}(\vec{x}_n|P)\frac{\mathrm{p}(\vec{x}_{n-1}, \ldots, \vec{x}_1|\vec{x}_n, P)}{\mathrm{p}(\vec{x}_{n-1}, \ldots, \vec{x}_1 \mid P)} \tag{6.8}$$

By rewriting the expression in this fashion, the original probability term can be viewed as the product of two separate terms. The first term is the *standard acoustic model*, i.e., the model that is used when the acoustic observations are considered independent. The second term is a ratio which will be referred to as the *consistency ratio*. This ratio compares the likelihood of the previously observed phones when considering and not considering the latest observation.

As discussed in Chapter 1, the consistency ratio determines whether or not the current observation is consistent or inconsistent with other observations in the same utterance under the assumption that the entire utterance was spoken by the same speaker. Observations which are deemed consistent with the past observations should result in a consistency ratio which is greater than one, while observations which are inconsistent with the past observations should result in a consistency ratio of less than one. Note that a ratio of one corresponds to a log ratio of zero. Thus, the log-based score for the ratio of consistent hypotheses should be positive while inconsistent hypotheses should receive negative scores.

Now that the consistency ratio is defined, the difficultly lies in devising a means of modeling this ratio. Modeling a large joint expression such as $\mathrm{p}(\vec{x}_{n-1}, \ldots, \vec{x}_1 \,|\, P)$ would be extremely difficult with anything but the simplest probabilistic models. Even the use of a single full covariance Gaussian model, though easy to construct, would be computationally expensive to use. For the purpose of practicality, one simplifying assumption will be made. It will be assumed that only the correlations between the current observation and the past observations are necessary to estimate the value of the consistency ratio. With this assumption the consistency ratio can be approximated as follows:

$$\frac{\mathrm{p}(\vec{x}_{n-1}, \ldots, \vec{x}_1 | \vec{x}_n, P)}{\mathrm{p}(\vec{x}_{n-1}, \ldots, \vec{x}_1 | P)} \approx \prod_{k=1}^{n-1} \frac{\mathrm{p}(\vec{x}_k | \vec{x}_n, P)}{\mathrm{p}(\vec{x}_k | P)} \tag{6.9}$$

This expression can be equivalently expressed as:

$$\prod_{k=1}^{n-1} \frac{\mathrm{p}(\vec{x}_k | \vec{x}_n, P)}{\mathrm{p}(\vec{x}_k | P)} = \prod_{k=1}^{n-1} \frac{\mathrm{p}(\vec{x}_n, \vec{x}_k | P)}{\mathrm{p}(\vec{x}_n | P)\mathrm{p}(\vec{x}_k | P)} \tag{6.10}$$

The full score for a hypothesized path can thus be written as:

$$\mathrm{p}(X|P) = \prod_{n=1}^{N} \mathrm{p}(\vec{x}_n | P) \prod_{k=1}^{n-1} \frac{\mathrm{p}(\vec{x}_n, \vec{x}_k | P)}{\mathrm{p}(\vec{x}_n | P)\mathrm{p}(\vec{x}_k | P)} \tag{6.11}$$

This can be rewritten as:

$$\mathrm{p}(X|P) = \left(\prod_{n=1}^{N} \mathrm{p}(\vec{x}_n | P) \right) \left(\prod_{n=1}^{N} \prod_{k=1}^{n-1} \frac{\mathrm{p}(\vec{x}_n, \vec{x}_k | P)}{\mathrm{p}(\vec{x}_n | P)\mathrm{p}(\vec{x}_k | P)} \right) \tag{6.12}$$

Typically the score of a hypothesized path is expressed in the log domain. In this case the expression becomes:

$$\log \mathrm{p}(X|P) = \left(\sum_{n=1}^{N} \log \mathrm{p}(\vec{x}_n | P) \right) + \left(\sum_{n=1}^{N} \sum_{k=1}^{n-1} \log \frac{\mathrm{p}(\vec{x}_n, \vec{x}_k | P)}{\mathrm{p}(\vec{x}_n | P)\mathrm{p}(\vec{x}_k | P)} \right) \tag{6.13}$$

In examining the final score of a hypothesized path using consistency modeling it can be seen that the consistency model contributes a sum of log ratios modeling individual pairs of acoustic observations. In information theory, the log ratio computed for each pair of observations is known as the pair's *mutual information.*

6.3 Consistency Example

To illustrate how the joint phone pair modeling employed by the consistency model works, consider the illustrated example in Figure 6.1. In this figure, a contour representing the joint density function of two phones is shown, [s] and [t]. Two different joint observations are also shown. One joint observation is labeled with an x and the other with an o. Let x_s represent the portion of x that corresponds to [s] while x_t represents the portion of x corresponding to [t]. Similarly let o_s and o_t represent the different components of o. Consider what happens when these two joint observations are scored by the consistency model. When examining the marginal probabilities for the different components of x and o it can be deduced from the figure that:

$$\mathrm{p}(o_s|s) \approx \mathrm{p}(x_s|s) \quad \text{and} \quad \mathrm{p}(o_t|t) \approx \mathrm{p}(x_t|t) \tag{6.14}$$

In other words, the standard acoustic model will produce approximately the same scores for the observations x_s and x_t as it does for o_s and o_t. However, when the correlations between the two phones in the pair are considered, it becomes clear that the joint pair for o is far more likely to be produced than the joint pair for x. Thus, the following expression will hold in this case:

$$\mathrm{p}(o_s, o_t|s, t) > \mathrm{p}(o_s|s)\mathrm{p}(o_t|t) \approx \mathrm{p}(x_s|s)\mathrm{p}(x_t|t) > \mathrm{p}(x_s, x_t|s, t) \tag{6.15}$$

In other words, the observations in o can be considered to be consistent with each other for the phone hypotheses of [s] and [t] (i.e., its consistency ratio is greater than one) as determined from the within-speaker correlation information provided in the consistency model, while the observations in x can be considered inconsistent.

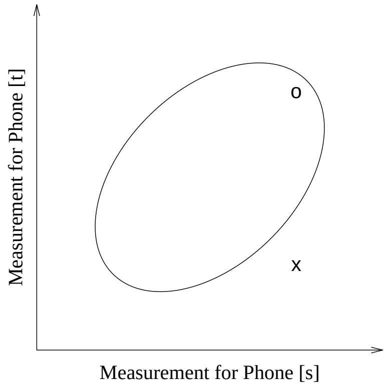


Figure 6.1: Fictitious illustration of joint consistency model created for the phones [s] and [t]. The "x" and "o" labels represent two potential joint observations.

6.4 Engineering Issues

In order to utilize the consistency model framework in an actual speech recognition system several engineering issues most be addressed. These issues are summarized by the following 5 questions:

1. How can a recognizer's search mechanism incorporate consistency modeling?
2. How should the consistency model be scaled relative to the standard acoustic model?
3. How are the consistency model's joint probability density functions created?
4. What acoustic measurements should the consistency model utilize?
5. What phone pairs should be scored by the consistency model?

6.4.1 Search Issues

As discussed earlier, when the utterance is processed in a time synchronous fashion, the acoustic model score for a particular segment is represented as:

$$\mathrm{p}(\vec{x}_n|\vec{x}_{n-1},\ldots,\vec{x}_1,P) = \mathrm{p}(\vec{x}_n|P)\frac{\mathrm{p}(\vec{x}_{n-1},\ldots,\vec{x}_1|\vec{x}_n,P)}{\mathrm{p}(\vec{x}_{n-1},\ldots,\vec{x}_1|P)} \tag{6.16}$$

From this equation it is clear that the score for a particular segment is dependent on all segments preceding it. Because of this dependence on the full past context, the consistency model can not be incorporated into a standard Viterbi search [70]. Furthermore, because the number of phones pairs that could be scored by the consistency model could be $O(n^2)$, it may be very inefficient to incorporate the consistency model into a best-first search such as the A^* search [45, 76].

An alternative to incorporating the consistency model directly into an A^* search is to use an A^* search to generate an N-best list and then rescore the N-best hypotheses using the consistency model. This approach greatly reduces the amount of computation that would potentially be performed by an A^* search directly incorporating the consistency model. If the N-best list has a high probability of containing the correct answer then this approach is not likely to suffer any severe degradation in performance as compared to implementing an A^* search which utilizes the consistency model. In the case of the Resource Management test set, the correct answer is one of the top two hypotheses 75% of the time and is one of the top ten hypotheses 90% percent of the time when the standard SI recognizer is used. For the experiments presented here, the consistency model is used to rescore the 10-best hypotheses proposed by the recognizer.

6.4.2 Consistency Model Scaling

As will be discussed in the next section, the training of the consistency model is a difficult estimation problem. This makes it necessary for the consistency model score to be scaled relative to the score of the standard acoustic model. The scaling factor will be represented as κ (where κ is typically set to 0.2) allowing the full acoustic model score to be expressed as:

$$\log \mathrm{p}(X|P) = \left(\sum_{n=1}^{N} \log \mathrm{p}(\vec{x}_n|P) \right) + \kappa \left(\sum_{n=1}^{N} \sum_{k=1}^{n-1} \log \frac{\mathrm{p}(\vec{x}_n, \vec{x}_k|P)}{\mathrm{p}(\vec{x}_n|P)\mathrm{p}(\vec{x}_k|P)} \right) \tag{6.17}$$

6.4.3 Constructing Joint Density Functions

Constructing Joint Observations

When utilized in a context independent mode, the consistency ratio is modeled utilizing the following expression:

$$\frac{\mathrm{p}(\vec{x}_j, \vec{x}_k|p_j, p_k)}{\mathrm{p}(\vec{x}_j|p_j)\mathrm{p}(\vec{x}_k|p_k)} \tag{6.18}$$

This expression requires the creation a joint density function $\mathrm{p}(\vec{x}_j, \vec{x}_k|p_j, p_k)$. The independent density functions $\mathrm{p}(\vec{x}_j|p_j)$ and $\mathrm{p}(\vec{x}_k|p_k)$ are simply the marginal densities for $\vec{x}_j$ and $\vec{x}_k$ and can be extracted directly from $\mathrm{p}(\vec{x}_j, \vec{x}_k|p_j, p_k)$.

The consistency ratio is intended to test the validity of the hypothesis that a pair of phones, p_j and p_k, could have been realized acoustically as $\vec{x}_j$ and $\vec{x}_k$ under the assumption that they were spoken by the same speaker. As such the training method employed to estimate the joint density function $\mathrm{p}(\vec{x}_j, \vec{x}_k|p_j, p_k)$ must account for the fact that each pair of observations $\vec{x}_j$ and $\vec{x}_k$ must be spoken be the same person.

In order to train $\mathrm{p}(\vec{x}_j, \vec{x}_k|p_j, p_k)$ using standard methods, a set of joint vectors representing joint observations of $\vec{x}_j$ and $\vec{x}_k$, as spoken by the same speaker, must be constructed. One potential method for constructing joint vectors was presented in Section 2.2.3 of Chapter 2. The basic idea is to create joint vectors for a particular phone pair by concatenating individual observation vectors from each of the two phones collected from one speaker. For example, suppose a training speaker has spoken 2 examples of the phone [s] and 3 examples of the phone [t]. The observation vectors for the [s] examples can be represented as $\vec{x}_{s,1}$ and $\vec{x}_{s,2}$. Likewise observation vectors for the [t] examples can be represented as $\vec{x}_{t,1}$, $\vec{x}_{t,2}$, and $\vec{x}_{t,3}$. From the examples of these two phones a set of joint vectors, $X_{s,t}$, for this one speaker can be created. If all combinations of the two phones are considered then six total joint

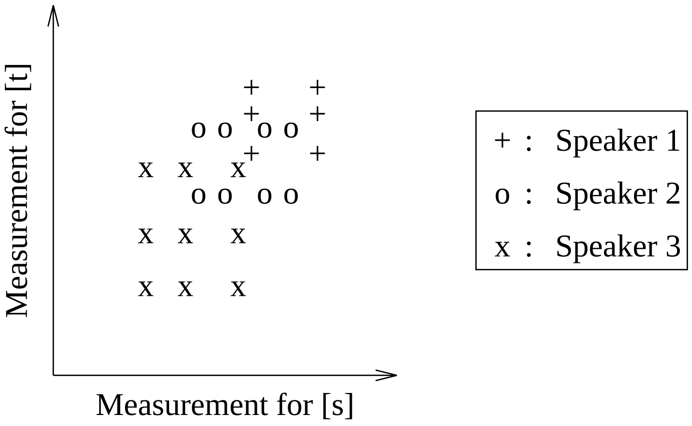


Figure 6.2: Fictitious illustration of joint vectors created for the pair of phones [s] and [t] as collected from three different training speakers.

vectors would be created. The joint vector of the $X_{s,t}$ set would be represented as:

$$X_{s,t} = \left\{ \begin{bmatrix} \vec{x}_{s,1} \\ \vec{x}_{t,1} \end{bmatrix}, \begin{bmatrix} \vec{x}_{s,1} \\ \vec{x}_{t,2} \end{bmatrix}, \begin{bmatrix} \vec{x}_{s,1} \\ \vec{x}_{t,3} \end{bmatrix}, \begin{bmatrix} \vec{x}_{s,2} \\ \vec{x}_{t,1} \end{bmatrix}, \begin{bmatrix} \vec{x}_{s,2} \\ \vec{x}_{t,2} \end{bmatrix}, \begin{bmatrix} \vec{x}_{s,2} \\ \vec{x}_{t,3} \end{bmatrix} \right\} \tag{6.19}$$

This process of constructing joint vectors must then be repeated for the remaining training speakers in the training set. Figure 2.3 illustrates how the joint vectors from three different speakers can be created. In this figure each phone observation is represented by a single measurement, giving the joint phone vectors 2 dimensions. For this example, speaker 1 has two examples of [s] and three examples of [t]. Similarly, speaker 2 has four examples of [s] and two examples of [t], while speaker 3 has three examples each of [s] and [t]. From the whole collection of joint vectors a joint density function can be trained using standard techniques. In this thesis two different methods for training the joint density model will be investigated. Both methods train models containing a mixture of diagonal Gaussian density functions.

The construction of joint vectors discussed above is designed to capture within-speaker correlations. The consistency model framework can also be utilized to account for other sources of correlation such as the channel, speaking style, etc. Accounting for these additional sources of correlation would only require a slight alteration in the construction of the joint vectors, and not a complete overhaul of the framework. For example, the joint vectors could be constructed from vectors which must be extracted from the same utterance instead of only restricting the two vectors to being from the same speaker. This would allow the consistency model to capture *within-utterance* correlation information instead of just within-speaker correlation information.

Direct Training

The most obvious method for creating a mixture Gaussian density function from a collection of joint vectors is to use the standard K-means/EM training algorithm. The data is first partitioned into clusters using the K-means algorithm with random initializations. These clusters are used to train the initial set of parameters for the components of the mixture Gaussian model. In this thesis the mixture components are diagonal Gaussian density functions. The mixture Gaussian parameters are then optimized on the training data using the EM algorithm.

There are two difficulties with this approach. First, the proper number of mixture components must be determined. Too few components does not provide the model adequate freedom to model the details of the underlying density function which generated the data. Too many parameters allows the system to over-fit the training data preventing the model the ability to generalized to unseen data. The second training difficulty lies in the fact that the training procedure is not guaranteed to find a globally optimal set of model parameters. A different random initialization of the K-means algorithm is likely to result in a different final set of mixture Gaussian parameters. Thus, a means for determining how to utilize different randomized training runs needs to be determined.

One technique which helps alleviate the two problems discussed above is aggregation. Aggregation simply combines the models created by different training trials into one model. For example, suppose T different training trials were run, each using a different K-means initialization, to estimate the expression for the joint likelihood $\mathrm{p}(\vec{x}_i, \vec{x}_j | p_i, p_j)$. If the expression $\mathrm{p}_t(\vec{x}_i, \vec{x}_j | p_i, p_j)$ is used to represent the model generated by the t^{th} training trial then the model created by aggregating all T trials can be expressed as:

$$\mathrm{p}_{agg}(\vec{x}_i, \vec{x}_j | p_i, p_j) = \frac{1}{T} \sum_{t=1}^{T} \mathrm{p}_t(\vec{x}_i, \vec{x}_j | p_i, p_j) \qquad (6.20)$$

Aggregating T different mixture Gaussian density functions simply results in one large mixture Gaussian density function containing T times as many Gaussian components as a model obtained from one training trial.

By combining together multiple trials, a more robust mixture model can be created. Aggregation has also been shown to alleviate problems caused when individual training trials tend to over-fit the training data [35]. A full discussion of aggregation is presented in Appendix E.

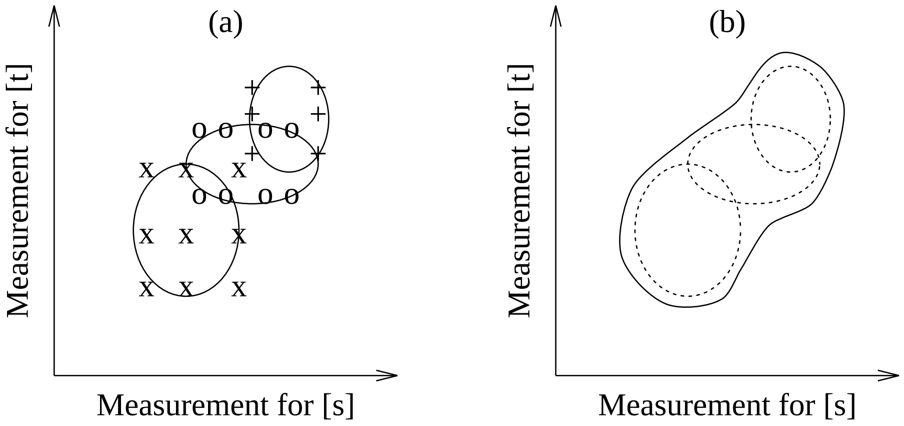


Figure 6.3: Illustration of joint models created for the pair of phones [s] and [t] as collected from three different training speakers. In (a) diagonal Gaussian models are created for each speaker. In (b) the individual diagonal Gaussians for each speaker are combined to make one large mixture of Gaussians.

Speaker Mixture Training

An alternative training approach is called speaker mixture training. In this approach, a joint model is first created for each individual training speaker. Next, the final model is created by combining all of the individual joint models from each speaker into one large mixture model. In this mixture model, the models from each individual speaker receive an equal weighting. In these experiments each individual speaker is modeled using only a single diagonal Gaussian density function.

Formally, the training procedure used when creating the joint model for any particular phone pair is as follows:

1. Train a single diagonal Gaussian model for the acoustic measurements for each individual phone of the phone pair for each speaker in the training set.

2. For each training speaker concatenate the diagonal Gaussians from each of the two phones into one joint diagonal Gaussian.

3. Giving all training speakers equal weight, combine the joint diagonal Gaussians from each training speaker into one large mixture Gaussian model.

This approach is illustrated with a fictitious example in Figure 6.3. This approach will not suffer the over-fitting problem of direct training since the EM algorithm is not used. It is also more efficient than direct training because no aggregation is needed.

6.4.4 Measurement Selection

Because the consistency model score can be computed independently of the standard acoustic model score, the measurement sets used by the two different models need not be the same. Because the consistency model is more difficult to train, a small set of measurements which exhibit a large amount of cross-phoneme correlation may be more appropriate than the full set of measurements used by the standard acoustic model.

The recognizer used in these experiments utilizes 36 different acoustic measurements in the standard acoustic model. These measurements are rotated using eigen analysis. The rotated measurements are ranked by the amount of variance they contribute across all acoustic observations. The ranked set of rotated measurements is also known as the set of *principal components* of the system [80].

In order to reduce the set of measurements used by the consistency model, it is useful to learn which of the principal components account for the most within-speaker correlation information. In Section 2.2.3 of Chapter 2 a method for measuring the within-speaker correlation exhibited between two phones was presented. This method can be adjusted to measure which principal components contributes the most within-speaker correlation across all phones. Recall that the within-speaker correlation between the observations of two phones can be represented as a 72 × 72 correlation matrix as follows:

$$\mathbf{C} = \begin{bmatrix} \mathbf{C}_{1,1} & \mathbf{C}_{1,2} \\ \mathbf{C}_{2,1} & \mathbf{C}_{2,2} \end{bmatrix} \tag{6.21}$$

Here $\mathbf{C}_{2,1}$ is a 36 × 36 matrix representing the cross correlation coefficients between the feature vector for phone 1 and the feature vector for phone 2. The contribution of any particular measurement in the feature vector can be determined from the column vector in $\mathbf{C}_{2,1}$ corresponding to that measurement. The total contribution of that measurement across all phones can be determined by summing the values in that measurement's cross correlation column vectors across all phone pairs. After finding the summed cross correlation values for each measurement, the final values for each measurement can be compared to determine which measurements contribute the most correlation across all phone pairs.

Figure 6.4 shows the estimated percentage of the contribution to the total across-phone cross correlation exhibited by each of the 36 principal components. As seen in the figure, there is a negative correlation between the order of the principal components and the relative contribution of the principal components towards the total across-phone correlation. Thus, a majority of the across-phone correlation is contributed by the top principal components. The relative contribution to the total correlation, in general, decreases as the rank of the principal components increases.

One simple way to reduce the dimensionality of the acoustic measurement vectors

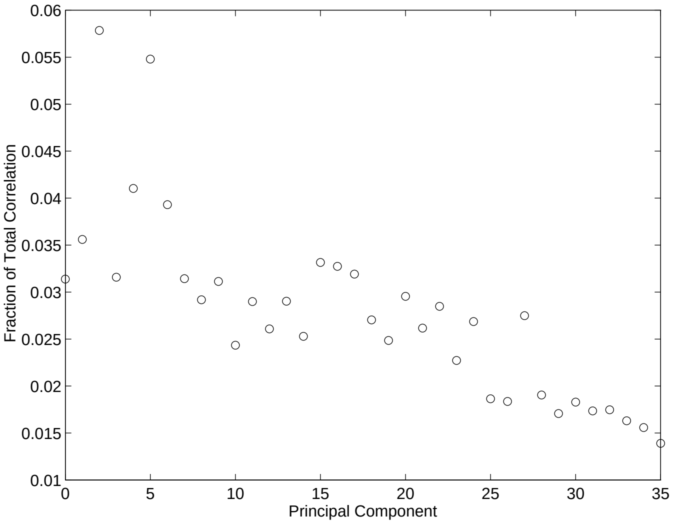


Figure 6.4: Estimated percentage of the across phone correlation contributed by each principal component of the acoustic measurement set as averaged across all phone pairs.

is to simply use only the top n principal components. Thus, the joint vector used by the consistency model would be of length $2n$. This approach is partially justified by the observed negative correlation between the principal component ranking and the percentage of the total correlation it contributes. An even better solution might be to take the top n principal components as ranked by their correlation contribution (as as opposed to their variance contribution). In these experiments the first approach is utilized, i.e. simply choosing the top n as based on the standard variance contribution. Our final experiments, as will be discussed later utilize only the top 10 principal components.

6.4.5 Phone Pair Selection

The consistency model need not score all of the phone pairs that it encounters. Because creating robust consistency models is a difficult estimation problem, it is wise to score only the phonetic pairs which exhibit a high amount of within-speaker correlation. If two phones do not exhibit a high amount of correlation then the estimation noise inherent in the phone pair's model could be more significant than the actual information to be gained from the correlation between the two phones. In these cases is is wise to simply assume that these phone pairs are uncorrelated and not score them. Phone pairs that are not used simply contributed a score of zero to the final log score, the same score that truly uncorrelated pairs should contribute.

To decide which pairs the consistency model will score, two criteria will be examined. First, only pairs with high within-speaker correlation values will be scored. Second, only pairs with enough training data to sufficiently train a joint model will be used. To determine which pairs have high within-speaker correlation, refer to Table 2.1 in Chapter 2. The phone pairs in this table can be ordered by their estimated within-speaker correlation value. Phone pairs with an insufficient amount of training data can be deleted from the list. For these experiments, phone pairs were eliminated from the list if the training corpus contained less than 3000 joint vector exemplars of the pair in the training data.

The phone-pairs that have a suitable amount of training data can be ranked by their within-speaker correlation values. In examining the ranked list, several obvious patterns are obvious. The top of the list is dominated self pairs, vowel-vowel pairs and nasal-nasal pairs. Of the top 60 phone pairs, 36 are self pairs, 31 are vowel-vowel pairs, 10 are fricative-fricative pairs, 8 are nasal-nasal pairs, and only 1 is a stop-stop pair. Table 6.1 shows the top ten phone pairs as ranked by their within-speaker correlation.

Rank	Phone Pair	Rank	Phone Pair
1	[ŋ],[ŋ]	6	[o],[o]
2	[š],[š]	7	[m],[m]
3	[ɾ̃],[ɾ̃]	8	[ɾ],[ɾ]
4	[ɑʸ],[ɑʸ]	9	[n],[ɾ̃]
5	[n],[n]	10	[e],[e]

Table 6.1: Top ten phone pairs as ranked by the amount of their within-speaker correlation.

6.5 Experimental Results

6.5.1 Overview

As discussed in the previous section, there are several system parameter that must be determined when using the consistency modeling approach. These include the training method for the mixture Gaussian models, the number of mixture components per model, the number of principal components in the joint measurement vectors, the set of phone pairs to be used, and the scale factor of the consistency model. This section will demonstrate the performance of the system as each of these parameters is varied. Because it would be difficult to show the effect of all of the parameters simultaneously, the experiments will show the performance of the system as two parameters are varied at a time. The best setting of these parameters was found to be as follows:

- Method of Training: Direct training with 24-Fold Aggregation
- Number of Gaussian Components per Mixture: 200
- Number of Measurement Principal Components: 10
- Number of Phone Pairs: 60
- Consistency Model Scale Factor: 0.2

All experiments will demonstrate the performance of different parameter settings relative to the best parameter settings shown above. All of the experiments were conducted using N-best rescoring of the 10-best hypotheses provided by the SI recognizer.

6.5.2 Effect of Scaling

The consistency model scale factor is an important parameter in consistency modeling. Because the consistency model is difficult to estimate, the model must be scaled relative to the standard acoustic model in order to improve the performance of the system. In each of the experiments discussed in the following sections, the results will be shown with a varying scale factor. In nearly every experiment a value of approximately 0.2 is optimal for the scaling factor. If the scaling factor is too large then the consistency model score could become too large relative to the standard acoustic model and the performance of the system could be harmed. In the experiments that follow note that the performance becomes more sensitive to the scaling factor when the consistency model uses more parameters, and hence suffers from more estimation noise.

6.5.3 Effect of Aggregation

As discussed earlier, training mixture Gaussian models using the standard K-means and EM training procedures can result different models given different initializations of the procedure. This method could cause the model to over-fit the training data in some regions of the acoustic space and underfit the data in other regions. Aggregation is one method for countering the problems of the standard training method. Aggregation is simply the process of combining together the models from a number of different training trials using the same measurement set. If N different training trials are combined into one mixture model this is called N-fold aggregation.

In these experiments the effects of aggregation are demonstrated. All system parameters are held constant with the exception of the number of training trials in the N-fold aggregation and the scale factor. To demonstrate the effectiveness of aggregation 24 different independent training trials were run. These 24 trials were used to run 24 independent 1-fold recognition experiments, 6 independent 4-fold recognition experiments, and 1 24-fold experiment. The average performance of the 1-fold, 4-fold, and 24-fold systems are shown in Figure 6.5. As can be seen, aggregation can significantly improve the performance of the consistency models. As the estimation of the consistency models improves with increased folds in the aggregate models, the optimal scale factor also appears to increase slightly. This is expected since the decrease in estimation noise will allow the correlation information captured in the consistency model to contribute more weight to the consistency score.

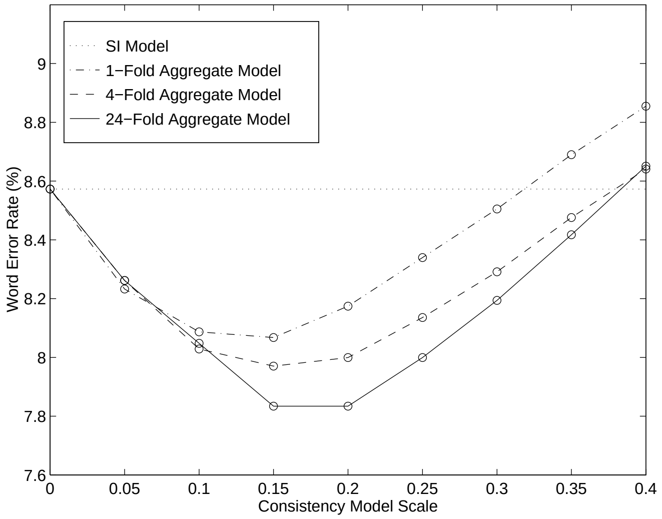


Figure 6.5: System performance using consistency modeling as the number of aggregated training trials and the scale factor are varied.

6.5.4 Effect of the Number of Mixture Components

Using the standard training techniques, the number of Gaussian components for each individual mixture model must be determined. If too few components are used the model may be overly restrictive and fail to capture the details of the underlying density function. If too many components are used then the model is susceptible to over-fitting of the training data and will fail to generalize well to unseen data. To account for this effect the number of Gaussian components can be varied depending on the amount of available training data. In these experiments, a new Gaussian component is added for every 300 joint vectors present in the data when training a mixture model for a particular phone pair. Also, a requirement of 3,000 joint vectors in the training data was enforced in order for the model of a phone pair to be used.

Figure 6.6 shows the performance of the system as the maximum number of allowed Gaussian components per training trial is varied from 100 to 300. Because the system aggregates 24 individual trials this actually means that the final models will have a maximum number of Gaussian components which vary from 2400 to 7200 over the three tests. As can be seen in the figure, a maximum of 100 mixture components per model (for one training trial) does not achieve as large a reduction in the error rate as 200 and 300 components per mixture when the scale factor is around 0.2. However, as the scale factor is increased beyond 0.2, the performance of the system drops off more rapidly in the models with more mixture components.

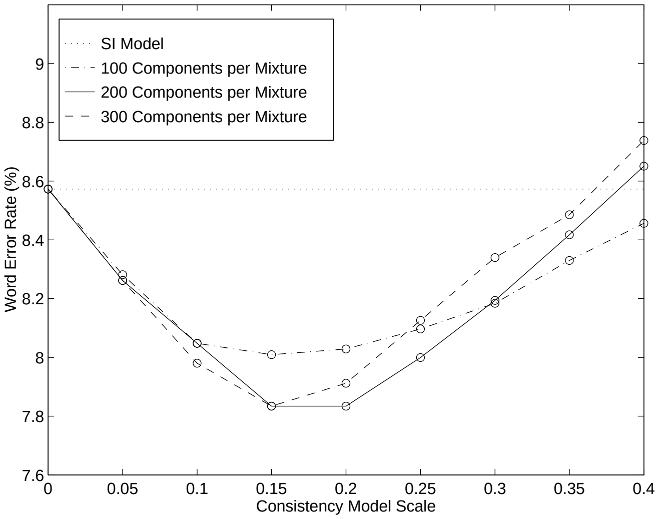


Figure 6.6: System performance using consistency modeling as the number of Gaussian components per mixture model and the scale factor are varied.

6.5.5 Effect of Training Method

In Section 6.4.3 two different methods are presented for training the joint mixture models used by the consistency model. One method is the standard method of direct K-means and EM training on the full collection of joint vectors. The second method is training individual joint models for each training speaker and then combining the individual speaker models to form the final mixture model.

The standard training method has difficulties in creating reliable models because of the uncertainty inherent in its training procedure as well as the potential for over-fitting. Aggregation is one method for overcoming these difficulties, but the benefits of improved model estimation from aggregation are offset by the potentially large increase in computation. The best set of consistency models utilized 24 aggregated models each with up to 200 Gaussian components. This results in models with up to 4800 Gaussian components.

The second training method avoids some of the problems of the standard training method by creating the mixture components at the individual speaker level first, and then combining the individual speakers' models to form the final mixture model. In these experiments the joint vectors for each individual speaker are modeled using only one diagonal Gaussian density function. Because of this, this method will not exhibit any uncertainty or over-fitting problems during training. Figure 6.7 compares the performance of the two methods and shows that the speaker mixture training method performs nearly as well as the standard method while using far fewer parameters.

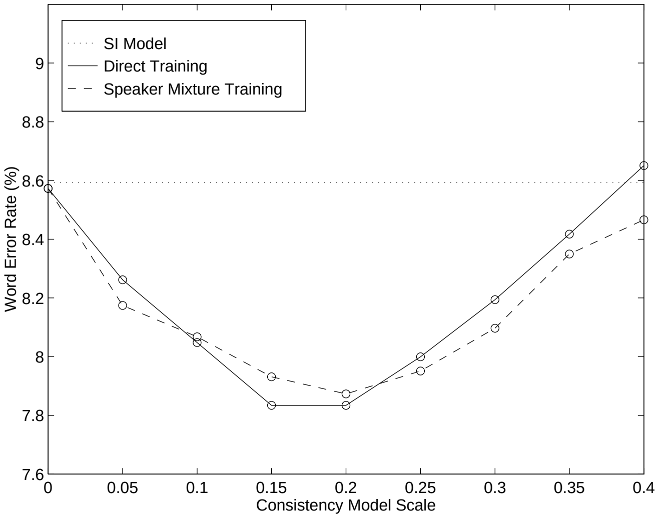


Figure 6.7: System performance using consistency modeling as the training method and the scale factor are varied.

6.5.6 Effect of Number of Principal Components

When altering the number of principal components used in the consistency model, there is a tradeoff between over-generalization of the training data and over-fitting of the training data. Too few components and the feature vector is unable to capture sufficient enough detail for the consistency model to contribute significant performance improvements. Too many components and the consistency model will suffer from over-fitting during the estimation phase and will not generalize well to the test data.

Figure 6.8 demonstrates the performance improvements provided by the consistency model as the number of principal components used in the construction of the consistency model feature vector is varied. When 5 principal components per phone are used, the performance gain is small and does not vary much as the scale factor is increased. This indicates the consistency model is not providing much new information but is also not contributing excessive estimation noise. With 20 principal components per phone the consistency model quickly degrades the system as the scale factor is increased indicating that the model is poorly estimated. The best performance is achieved with around 10 principal components.

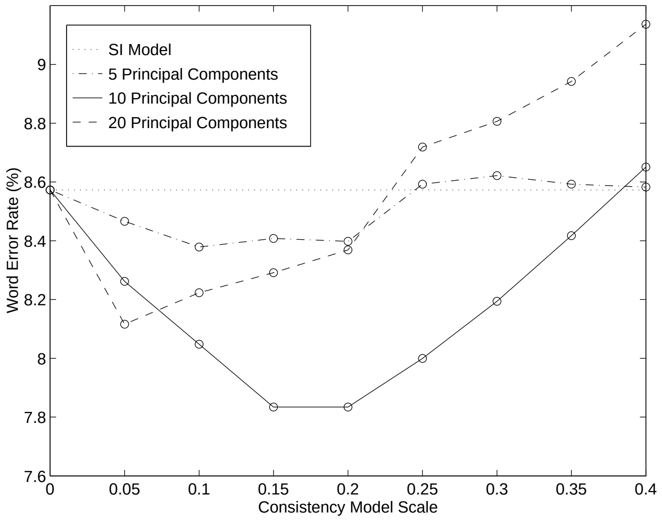


Figure 6.8: System performance using consistency modeling as the number of principal components and the scale factor are varied.

6.5.7 Effect of Phone Pair Set

The final modeling decision that must be made is the choice of phone pairs to use during consistency modeling. The goal is to utilize as many of the most correlated phone pairs as possible without introducing phone pairs whose estimation noise exceeds the contribution of their underlying within-speaker correlation. This is accomplished by varying the number of phone pairs extracted from the top of the list of most correlated phone pairs. For these experiments three different sets of phone pairs are utilized. The first set has the top 35 phone pairs, the second set has the top 60 phone pairs and the third set has the top 85 phone pairs.

Figure 6.9 shows the performance of the three sets as the scale factor is varied. As can be seen, the set of 60 phone pairs easily outperforms the set with only 35 phones indicating that the 25 additional phones in the set 60 are able to contribute additional useful correlation information beyond that provided by the initial 35 phone pairs. Increasing the number of phone pair models to 85 causes a slight drop-off in performance, presumably because the new phone pair models that are added in this set do not contribute enough correlation information to overcome their own estimation noise.

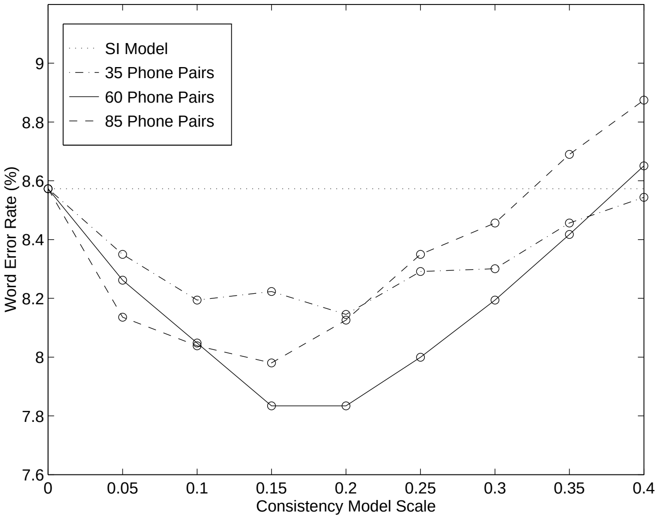


Figure 6.9: System performance using consistency modeling as the set of phone pair models and the scale factor are varied.

6.5.8 Contribution of Individual Phone Pairs

While the estimated within-speaker correlation between different phone pairs can be used to predict which phone pairs may contribute useful information during consistency modeling, it cannot predict the actual performance gains that are likely from utilizing the individual phone pairs. Table 6.2 shows the performance of some phone pairs when tested on an individual basis within the consistency model. The table shows the twelve phone pairs which corrected the most number of errors as well as the thirteen phone pairs which introduced the most number of errors (as represented by negative numbers in the table). This table gives some indication as to which phone pairs are actually the most useful to the consistency model. However, these numbers do not necessarily reflect the usefulness of the pairs when they are used in conjunction with other pairs during the rescoring process.

As the table shows, phone pairs with a high amount of correlation are not necessarily the most useful pairs for correcting errors. There are two other primary factors which determine the usefulness of a phone pair. First, pairs which contain common phones will obviously be used more often than pairs containing uncommon phones, thus offering the possibility that they correct more errors. This explains the presence of the four pairs containing [n] in the top twelve. Second, pairs that contain phones which are commonly confused with other phones will be more useful than pairs containing phones which are typically recognized correctly. This is highly dependent on the vocabulary of the task. The phones contained in words that are commonly confused for a given task are more useful to the consistency model than phones that appear in words that are seldom misrecognized. This explains why pairs containing [i] introduce more errors than they correct. Few of the word errors caused by the recognizer on the RM ask involve words containing the phone [i], thus giving the consistency model more opportunity to introduce errors than to correct errors in words containing [i].

This table suggests that the choice of phone pairs to be utilized by the consistency model should consider not only the correlation between the phones in each pair but also the potential for the pair to correct errors. Pairs which contain phones which are often confused with other phones are suited for inclusion in the consistency model, while pairs with phones which are typically classified correctly are best not used. The final set of pairs might best be determined by testing their performance on an independent development set.

Phone Pair	Net Errors Corrected	Rank from Correlation
[ɛ],[e]	13	39
[n],[n]	10	5
[n],[ŋ]	9	11
[ɛ],[n]	9	59
[ɑ],[ʌ]	8	54
[ɪ],[ɪ]	8	57
[z],[z]	7	34
[n],[ɾ̃]	6	9
[e],[e]	6	10
[ɛ],[ɛ]	6	17
[s],[s]	6	30
[ɑ],[æ]	6	49
⋮	⋮	⋮
[æ],[ɛ]	-1	26
[e],[o]	-1	43
[e],[ü]	-1	52
[ŋ],[ɾ̃]	-2	15
[l],[l]	-2	38
[e],[ɪ]	-3	58
[æ],[e]	-3	35
[æ],[ʌ]	-4	47
[ɾ],[ɾ]	-5	8
[ʌ],[ɛ]	-5	40
[i],[i]	-6	21
[e],[i]	-6	31
[i],[y]	-7	60

Table 6.2: Top twelve and bottom thirteen phone pairs from the set of 60 as ranked by the total number of errors corrected when only one phone pair is used by the consistency model.

6.6 Summary

This chapter has presented the theoretical derivation and engineering issues of a new speech recognition modeling technique called consistency modeling. This new method of modeling is designed to incorporate the correlation information which exists within an utterance which is ignored when the individual acoustic observations of the utterance are assumed to be statistically independent. In this chapter, consistency modeling is presented as a means for incorporating within-speaker correlation information into the probabilistic framework. Using the consistency modeling technique presented in this chapter, the word error rate (WER) is reduced by a relative 8.8%, from 8.6% WER to 7.8% WER, when the system's parameters are set appropriately and the direct training method using aggregation is utilized. Using the speaker mixture training approach, the system is able to reduce the error rate by a relative 8.2%, from 8.6% WER to 7.9% WER, while also being far more computationally efficient.

Finally, it is important to note that consistency modeling does not perform any explicit adaptation. This technique relies entirely on pre-trained models and does not alter any of the model parameters used by the standard recognizer. As such the novelty of consistency modeling is that it is capable of incorporating speaker constraint directly into the recognizer framework instead of relying of adaptation techniques to supply the speaker constraint.

Chapter 7

Instantaneous Adaptation

7.1 Overview

In Chapters 3 and 4, several techniques for speaker adaptation were discussed and results for supervised, enrolled, batch adaptation were presented. However, there are many applications where supervised, enrolled, batch adaptation is not feasible. For example, the GALAXY and JUPITER systems typically operate with speakers who are unknown to the system and will only utilize a small number of conversational exchanges to achieve their goal [33, 54, 89, 84]. In these cases adaptation must by performed in an unsupervised fashion. Because the number of utterances is small, it is also desirable to perform adaptation in an instantaneous fashion, i.e., adaptation is applied using the same utterance that the system is trying to recognize.

The speaker clustering weighting (SCW) technique discussed in Chapter 5 is an example of unsupervised, instantaneous adaptation. With this technique, the weights associated with each speaker cluster are adapted to optimally match the acoustic observations of the current utterance. Because the true word string is unknown, the adaptation is performed using a hypothesized word string generated by the SI recognizer. The adapted models are then used for a second recognition pass on the utterance.

In this chapter instantaneous adaptation experiments using reference speaker weighting, speaker clustering and consistency modeling will be presented. Each of these three techniques operates in an independent fashion. Speaker clustering can be used to determine an initial model to be used by the recognizer, reference speaker weighting can be used to adapt an initial model, and consistency modeling can be used to rescore a recognizer's N-best output. Because each technique can be applied independently from the other techniques, it is possible to utilize any two or all three of the techniques simultaneously.

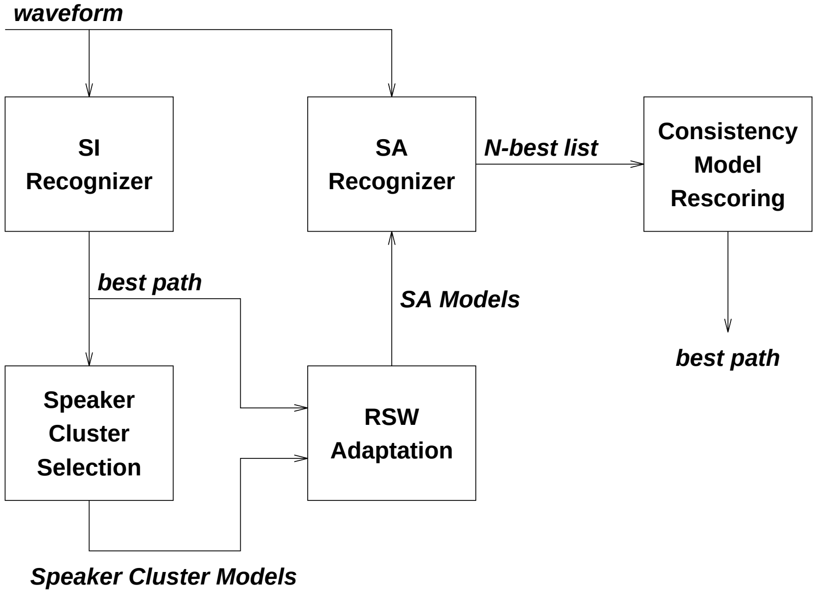


Figure 7.1: Architecture of recognizer using instantaneous adaptation techniques.

7.2 System Design

All of the techniques presented in this thesis require a transcription of the adaptation data when performing adaptation. Unfortunately, the underlying transcription of an utterance is not known during unsupervised adaptation. The simplest solution to this problem is to run the standard SI recognizer on the adaptation data and then use the best path proposed by the recognizer as a substitute for the true transcription when performing adaptation. This approach can cause troubles if the adaptation routine is sensitive to errors in the transcription. This is especially problematic for techniques which try to adapt a large number of specific parameters (such as the standard MAP technique) instead of a small number of general parameters (such as the RSW technique). When adapting a small number of general parameters it is possible for the correct segments in the best path to overwhelm the errors during the adaptation routine's estimation phase. This is the case with the reference speaker weighting and speaker cluster weighting techniques.

Figure 7.1 diagrams the system architecture used for the instantaneous adaptation experiments presented in this chapter. The system uses a two-pass recognition approach. First, the SI recognizer is run to generate a best path. This best path is then utilized by the speaker cluster selection module. If hierarchical speaker clustering is being used then this module determines the gender and speaking rate of the utterance and outputs the appropriate gender and speaking rate dependent set of models. If speaker cluster weighting is being used then this module determines the optimal weighting of the different cluster models and outputs the final speaker cluster weighted set of models. The best path from the SI recognizer is also used by the RSW adaptation module. This module takes the set of models provided from the speaker clustering module and adapts them using RSW adaptation based on the best path provided by the SI recognizer. The RSW module outputs a speaker adapted (SA) set of models which can then be utilized for the second recognition pass. The SA recognizer is then used to generate an N-best list which can be rescored by the consistency model module.

7.3 Context Independent Experiments

7.3.1 Experimental Conditions

The system is first evaluated using the context independent (CI) version of SUMMIT on the SI evaluation set of the Resource Management corpus, as is done in Chapters 5 and 6. The SI training and development sets are used for all training. The instantaneous adaptation is performed using various combinations of the three different techniques presented in this thesis: speaker clustering, reference speaker weighting, and consistency modeling. The experiments all use the system architecture presented in Figure 7.1.

When speaker clustering is used, the speaker cluster models are the same as those utilized in Chapter 5. For these experiments three different cluster selection techniques are used in addition to standard speaker independent (SI) recognition. These clustering techniques are gender dependent (GD) modeling, gender and speaking rate dependent (GRD) modeling, and speaker cluster weighting (SCW).

When reference speaker weighting is utilized, the system uses the same algorithm presented in Chapter 4. There are several minor differences between the experiments here and the experiments in Chapter 4. First, the experiments here are conducted in unsupervised mode while in Chapter 4 they were conducted in supervised mode. Second the reference speaker vectors and covariance matrices used during RSW adaptation are trained using the 109 speakers in the SI training and development sets instead of the full 149 speakers from the entire SI set as is done in Chapter 4. Finally, when RSW is used to adapt the gender dependent (GD) cluster models, the RSW parameter set is constrained to only the reference speakers of the appropriate gender. In other words, the RSW adaptation is performed using gender dependent reference speaker vectors and covariance matrices when GD cluster models are used for recognition.

In Chapter 5, RSW model translation adaptation is compared with standard MAP model translation. In that chapter, RSW model translation is shown to outperform MAP model translation when only one adaptation utterance is available, but not by a large amount. However, those experiments are performed using supervised adaptation. In unsupervised instantaneous adaptation, both RSW and MAP adaptation use the best path output of the SI recognizer to guide the adaptation. Under these conditions, MAP adaptation is not nearly as effective because it does not adapt to general speaker properties but rather adapts the specific parameters of individual models. This causes the algorithm to reinforce the recognizer's mistakes during adaptation instead of correcting them. This will be shown by substituting the MAP model translation algorithm for the RSW algorithm in one experiment that follows.

When performing consistency model rescoring, the system uses the same set of 60 speaker mixture trained models utilized in Chapter 6 for all experiments. These models each use 10 principal components per phone. The scaling factor is initially fixed at 0.2 as in Chapter 6. However, as the standard acoustic model set changes it seems appropriate that the optimal consistency model scale factor should change as well. As the model set begins to estimate the true underlying density functions for a particular speaker, it is reasonable to assume that the consistency model will become less useful and will require a smaller scale factor (although this turns out not be the case in these experiments). Thus, the performance of the system will be examined with a fixed scale factor of 0.2 in the first set of experiments and with a scale factor that is varied in the second set of experiments.

7.3.2 Experimental Results

Table 7.1 shows the instantaneous adaptation results using various combinations of the different adaptation algorithms presented in this thesis. The table is broken down into four subsections corresponding to the four different speaker clustering routines (SI, GD, GRD, and SCW). For each type of speaker clustering both RSW adaptation and/or consistency modeling (CM) can be applied in addition to the speaker clustering. The type of adaptation that is performed is listed in the first column. The next three columns show the total number of errors, the word error rate, and the reduction in word error rate relative to the performance of the baseline SI recognizer.

In the table, the first adaptation result is from the application of standard MAP model translation to the SI recognizer. As expected this does not significantly improve the recognizer performance. However, RSW model translation does improve the performance of the system significantly despite the fact that its adaptation is guided by the error prone best path from the SI recognizer. This indicates that RSW model translation adaptation is far more robust to errors in the recognizer's best path than MAP model translation.

Next note that when RSW adaptation is performed on the GD cluster models, no significant improvement is observed. This could result from the fact that the GD models have a smaller variance than the SI models. When a model has a smaller variance, its likelihood estimates are affected more when its center of mass is altered than a model with larger variance. Thus, as the cluster models become more specific, model translation adaptation techniques become more sensitive to the noise in the center of mass estimation.

When consistency modeling is used with a scale factor of 0.2, the system's performance is almost universally improved regardless of the cluster models that are used. It should be noted that the relative improvements from consistency modeling decrease as the cluster models become more specific. This is expected because the contribution of the consistency model should decrease as the resemblance of the standard acoustic models to the true underlying speaker dependent models increases.

Next note that consistency modeling does not to improve the performance of the system when RSW adaptation is applied to the SI models. This result is puzzling at first considering the improvements that consistency modeling provides to the various different cluster models. However, this result can be traced to the sensitivity of the system's performance to the consistency model scale.

Adaptation Method	Total Errors	Word Error Rate	Error Rate Reduction
SI	882	8.6%	—
SI+MAP	875	8.5%	0.8%
SI+RSW	825	8.0%	6.5%
SI+CM	810	7.9%	8.2%
SI+RSW+CM	808	7.9%	8.4%
GD	789	7.7%	10.5%
GD+RSW	783	7.6%	11.2%
GD+CM	738	7.2%	16.3%
GRD	737	7.2%	16.4%
GRD+CM	715	6.9%	18.9%
SCW	715	6.9%	18.9%
SCW+CM	701	6.8%	20.5%

Table 7.1: Summary of recognition results using various instantaneous adaptation techniques. The consistency model scale is fixed at 0.2 in these experiments.

Figure 7.2 shows the performance of the system when consistency modeling is applied to various different adaptation techniques and the consistency model scale factor is varied. As can be seen, the scale factor of 0.2 which is optimal for the SI recognizer is not optimal for the other techniques. However, the differences between the performance of the system when using a scale factor of 0.2 versus using the optimal scale factor are relatively small in most cases. In the worst case, the SI+RSW+CM system exhibits an increase of only 14 errors over the 1200 test utterances when a scale factor of 0.2 is used instead of the optimal scale factor for this test set. Table 7.2 presents the recognition results for the various adaptation technique when the consistency model utilizes the optimal scale factor instead of the predetermined 0.2 scaling.

7.3.3 Statistical Significance

When comparing the differences which exist between the performances of different recognizers it is important to consider whether these differences are statisically significant. If the difference between the perfromance of two recognizers is not statistically significant then it is possible that a test result indicating that one recognizer outperforms another is simply the result of chance and not an indicator of true superiority. To evaluate the significance of the results presented in Table 7.2, the *matched pairs sentence segment word error test* is utilized [31]. This test measures the likelihood that the differences present during an evaluation between two recognizers are a result of chance as opposed to genuine differences in the performances of the recognizers.

Table 7.3 showns the significance values for the comparison of different pairs of the adaptation methods used in Table 7.2. The differences are considered significant if likelihood of the differences occuring due to chance is estimated to be .05 or less. In other words, the results are considered significant if there is a 95% chance or better that the difference in perfromance between two recognizer is a result of genuine differences in the recognizers. In the table, significant differences are indicated with *italics* while insignificant differences are indicated with **boldface**. Also, all results with a significance level less than .001 are simply listed as having a significance level of .001 in the table.

The table indicates that the improvements going from the SI system to the GD system and from the GD system to the GRD system are significant. However, the improvement of the SCW system over the GRD system only has a significance level of .159 and is therefore is not statiscally significant. Adding the consistency model to the SI, GD, and GRD system results in a statistically significant perfromance improvement. However, the performance improevemtn gained from applying the consistency model on top to the SCW system was not significant.

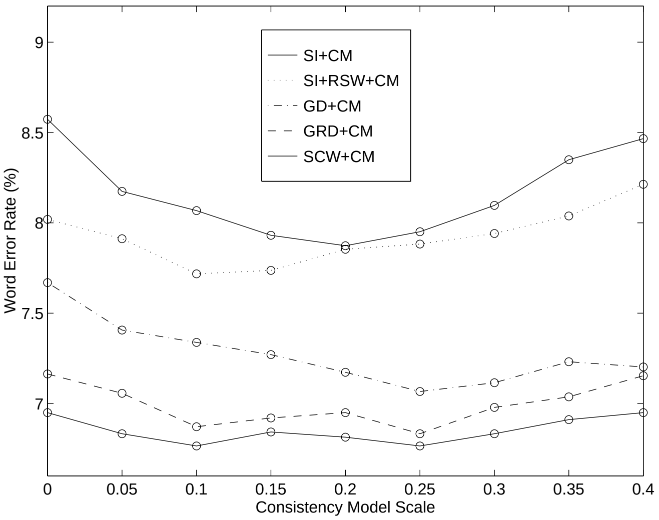


Figure 7.2: System performance using various different clustering techniques augmented with consistency modeling as the consistency model scale factor are varied.

Adaptation Method	CM Scale	Total Errors	Word Error Rate	Error Rate Reduction
SI	—	882	8.6%	—
SI+CM	0.20	810	7.9%	8.2%
SI+RSW+CM	0.10	794	7.7%	10.0%
GD	—	789	7.7%	10.5%
GD+CM	0.25	727	7.1%	17.6%
GRD	—	737	7.2%	16.4%
GRD+CM	0.25	703	6.8%	20.3%
SCW	—	715	6.9%	18.9%
SCW+CM	0.25	696	6.8%	21.1%

Table 7.2: Summary of recognition results using various instantaneous adaptation techniques. The optimal consistency scale is always used (when appropriate) and is shown in the CM scale column.

	SI		GD		GRD		SCW
	+CM	GD	+CM	GRD	+CM	SCW	+CM
SI	*.001*	*.001*	*.001*	*.001*	*.001*	*.001*	*.001*
SI+CM	—	**.459**	*.002*	*.006*	*.001*	*.001*	*.001*
GD	—	—	*.001*	*.002*	*.001*	*.001*	*.001*
GD+CM	—	—	—	**.589**	**.099**	**.484**	**.110**
GRD	—	—	—	—	*.032*	**.159**	**.060**
GRD+CM	—	—	—	—	—	**.617**	**.741**
SCW	—	—	—	—	—	—	**.342**

Table 7.3: Measure of statistical significance of differences between different instantaneous adaptation methods. Significant difference or shown in *italics* while insignificant differences are shown in **boldface**.

7.4 Context Dependent Experiments

7.4.1 Experimental Conditions

In the previous section, it was shown that instantaneous adaptation can reduce the error rate of the SUMMIT context independent recognizer by over 20%. However, this error rate reduction is not likely to be as large when the adaptation is performed on a system which is closer to state of the art in terms of accuracy. To determine the potential usefulness of the instantaneous adaptation techniques presented in this thesis on a system which is closer to state of the art, instantaneous adaptation can be performed on the context dependent (CD) version of SUMMIT. For full details on the CD version of SUMMIT refer to Appendix B.

The CD version of SUMMIT uses a set of 67 different context independent phonetic segment models as well as a set of 558 context dependent diphone boundary models. The segment models utilize a 77 dimension feature vector which is modeled with up to 100 Gaussian components per model. The anti-phone density function is modeled with 400 Gaussian mixture components. The diphone boundary models use a 50 dimension feature vector which is modeled with up to 50 Gaussian mixture components. To increase the robustness of the segment, boundary, and anti-phone models the final set of models are created by aggregating four different, independently trained sets of models. Thus, the final segment models actually contain up to 400 mixture components per model (100 components from each of four different training trials).

During training, the full 80 speaker training set and 40 speaker development set are utilized (as opposed to the 72 speaker training set and 37 speaker development set used in previous experiments). In these experiments, all system parameters are optimized by first training the system on only the 80 speaker training set and then optimizing the system parameters on recognition trials tested on the 40 speaker development set. Once the system's parameters are set, the acoustic models are retrained using the full 120 speakers in the combined training and development sets. The recognizer achieves a word accuracy of 95.8% when evaluated on the full test set. However, because of the large amount of computation required, this CD recognizer runs considerably slower than real time.

Three different adaptation experiments were run using the CD recognizer. First the recognizer was tested with consistency model rescoring. Second, the system was tested using gender dependent modeling. Finally, the system was tested using both gender dependent modeling and consistency modeling.

When performing consistency modeling, the scale factor and phone pair set were optimized on recognition experiments on the development set. A set of the 30 consistency model pairs were chosen based upon their ability to help improve the performance of the system on the development test set. Thus, pairs which did not improve the performance of the system on the development set were not used during final testing. The 30 pairs consist of 16 segment-segment pairs and 14 boundary-boundary pairs. The consistency pairs which improve the performance the most are pairs which contain segment or boundary models which occur in the most frequently misrecgonized words such as *and*, *of*, *what* and *was*. The consistency model scale was set to 0.4 based on the recognition experiments of the development set.

When performing gender dependent modeling, the system utilizes interpolated gender dependent models. The gender specific models are trained in the same fashion as the SI models and also utilize 4 aggregated training trials for each model. The interpolation factors were determined using the EM algorithm on the development data using models trained on the 80 speaker training set.

7.4.2 Experimental Results

Table 7.4 shows the results of the CD recognizer when using GD models and/or consistency modeling. As can be seen, the same general trend of improved results as seen in the CI system is apparent. However, the relative reductions in error rate are not as large as in the CI system. There are two main reasons for this. First, the CD boundary models introduce some speaker constraint because their measurements extend over the regions of two adjacent phones thus providing joint phone modeling capabilities to the system. The second reason is that many of the errors that remain after using the CD system are not easy to correct through improved acoustic modeling because they occur in reduced function words. Table 7.5 shows the most common confusions made by the CD system. These confusions are most easily corrected by the language model using surrounding context. Remember that the recognizer for this task does not utilize any statistics or syntax in its language model.

Table 7.6 presents the statiscal significance analysis comparing the different adaptation methods used within the CD recognizer. In all cases the performance improvements obtained with different adaptation methods are deemed insignificant over this test set. However, the differences between the SI system and the GD+CM system have a significance level of .097 which is deemed not significant by only a small margin. Had the GD+CM system corrected 2 more errors that were present in the SI system, the confidence level threshold of 0.05 would have been met.

Adaptation Method	Total Errors	Word Error Rate	Error Rate Reduction
SI	436	4.24%	—
SI+CM	422	4.10%	3.2%
GD	416	4.04%	4.6%
GD+CM	409	3.98%	6.2%

Table 7.4: Summary of recognition results using gender dependent modeling and consistency modeling on the CD version of SUMMIT.

Rank	Occurrences	Confusion
1	15	AND ⟶ IN
2	6	WAS ⟶ IS
3	5	GET ⟶ GIVE
4	5	CHOPPED ⟶ CHOP
5	5	IN ⟶ AND
6	4	WHEN ⟶ WHAT
7	4	OF ⟶ THE
8	4	HOW ⟶ HAVE
9	4	MANY ⟶ ANY
10	3	WHEN WILL ⟶ WHEN+LL
11	3	ARE IN ⟶ AREN+T
12	3	GIVE ⟶ GET
13	3	LAT AND ⟶ LAT-LON
14	3	IS ⟶ AS
15	3	DID ⟶ DO

Table 7.5: Top 15 confusions incurred by the SI-CD recognizer.

	SI + CM	GD	GD + CM
SI	**.254**	**.180**	**.097**
SI + CM	—	**.575**	**.352**
GD	—	—	**.430**

Table 7.6: Measure of statistical significance of differences in performance between different instantaneous adaptation methods used by the SI-CD recognizer.

7.5 Summary

The instantaneous adaptation experiments presented in this chapter demonstrate the effectiveness of acccounting for speaker constraint during the recognition process. In particular, this chapter has demonstrated how the various techniques for incorporating speaker contraint presented in this thesis can be combined to further improve the performance of a recognizer. This chapter has also demonstrated that the adaptive capabilities of the techniques presented in this thesis are robust in unsupervised conditions and don't require a perfect transcription in order to be effective.

One primary purpose of the experiments in this chapter was to compare the relative merits of the three techniques presented in this thesis. Of the three techniques, speaker clustering clearly proved to be the most effective. This technique achieved excellent results, required only basic training techniques, and was simple to implement. Consistency modeling also proved to be a useful technique, especially as a supplement to speaker clustering. However, consistency modeling also required accurate tuning of its parameters and its performance was sensitive to its scaling factor. Reference speaker weighting proved to be a useful technique when applied to the SI recognizer, but did not perform well when used in conjunction with speaker clustering.

Chapter 8

Conclusion

8.1 Summary

The main purpose of this dissertation, as defined in Chapter 1, has been to attack the standard assumption that different observations extracted from a speech signal can be considered statistically independent by a speech recognition system. This assumption ignores constraints imposed upon the speech signal by various sources including the speaker, the environment, the channel, etc. This dissertation has focused on examining the constraints imposed by the speaker and developing techniques to account for the correlations which exist within the speech of a single speaker.

In Chapter 2 various analyses of within-speaker correlation were conducted. These experiments demonstrated the extent of the correlations which exist between different sounds produced by the same speaker. These correlations allow for the possibility that an observation of the acoustic realization of one phone can be used to predict the likelihood of the acoustic realization of an unseen observation for a different phone from the same speaker. The root of some of the within-speaker correlations can be traced to properties of the speaker, such as gender and speaking rate, which impose systematic constraint on the acoustic signal. The chapter also demonstrated that within-speaker correlation exists across different levels of the speech hierarchy including the models accounting for acoustics, duration, and pronunciation variation.

In Chapter 3 the basic principles of speaker adaptation were presented. Adaptation is essentially an estimation problem, where the characteristics of an individual's speech production patterns must be learned. A successful adaptation routine must be able to reliably and robustly estimate a set of adaptation parameters from a given set of adaptation data. Ideally, prior information about the statistically properties of the parameters to be learned should be incorporated into the process. To illustrate these principles, a series of adaptation experiments were conducted which demonstrated the performance of several basic adaptation techniques as the amount of available adaptation data is varied. The chapter also presented brief descriptions of the most common current adaptation techniques.

In Chapter 4 the reference speaker weighting (RSW) approach to speaker adaptation was presented. This adaptation technique enforces the constraint that a particular set of model parameters used by a recognizer must be a weighted interpolation of parameter sets drawn from individual training speakers. By enforcing this constraint across different phonetic models, the within-speaker correlation information can be incorporated via the within-speaker *tying* of the different phonetic parameters. This technique allows the models of unseen phones to be adapted based upon the observation of other phones. Experiments revealed that RSW model translation is superior to standard MAP model translation when limited adaptation data is available.

In Chapter 5 two methods for speaker clustering were presented. The speaker clustering techniques are similar to the RSW technique in that parameter sets from different phone models are tied together. However, in speaker clustering the tying is performed within a set of speakers belonging to a particular speaker cluster instead of within individual speakers. Combining individual speakers into more general speaker clusters allows more robust acoustic models to be trained than the RSW approach allows. The two methods of speaker clustering that were examined were hierarchical speaker clustering and speaker cluster weighting. Both methods provided large improvements in recognition accuracy.

In Chapter 6 a novel recognition technique called *consistency modeling* was presented. This technique attempts to estimate the contribution of the *consistency ratio* introduced in Chapter 1. The consistency ratio is a likelihood ratio which accounts for the correlation information that is ignored when acoustic observations are considered independent. A derivation of the probabilistic framework of the consistency modeling approach was presented. In this derivation estimation of the consistency ratio was reduced to the problem of measuring the *mutual information* that exists between pairs of acoustic observations. Experiments showed that consistency modeling can provide significant improvements in recognition accuracy when proper modeling techniques are employed.

In Chapter 7 a set of instantaneous adaptation experiments using a combination of reference speaker weighting, speaker clustering and consistency modeling are presented. These experiments revealed two primary observations. First, the different techniques can be combined together to improve upon the performance gains observed when they are each run independently. In particular the combination of speaker cluster weighting and consistency modeling is able to reduce the error rate of the context independent recognizer by over 20%. Second, the techniques can be robustly applied in an unsupervised instantaneous fashion to improve the recognition accuracy of the system.

8.2 Contributions

The goal of this dissertation has been to attack the independence of observations assumption which is common is today's speech recognition systems. To this end, this dissertation has presented three techniques which account for within-speaker correlation information during the recognition process. All three techniques share the common idea that prior information capturing the within-speaker correlations which exist between different speech events should be accounted for in the models used by a speech recognition system. As discussed in Chapter 1, there are two primary ways in which the correlation information that is ignored by standard speaker independent recognizers can be accounted for. The first way is to perform speaker adaptation to create a set of models which are as close to the true speaker dependent models of the current speaker as possible. The second way is to account for the contribution of the consistency ratio. This dissertation contributes techniques for attacking the problem using the two different approaches discussed above.

To begin, this thesis introduces a novel adaptation algorithm called reference speaker weighting (RSW). This algorithm combines two important ideas for speaker adaptation. First, a speaker adaptation algorithm should incorporate prior knowledge about the parameters of speaker dependent models. Because the goal of speaker adaptation is to learn the underlying density functions belonging to the acoustic model of a particular speaker, it is useful to incorporate knowledge about the constraints imposed on the model's parameter space. This *a priori* knowledge about a speaker's parameter space is where within-speaker correlation information can be applied. RSW adaptation incorporates within-speaker correlation by tying together phones which are highly correlated into phone sets which are adapted simultaneously.

The second important idea used in the RSW adaptation approach, is the idea that the number of adaptation parameters can be adjusted to match the amount of available adaptation data. This is an important component in today's transformational adaptation approaches such as MLLR adaptation. In MLLR similar phones are tied together into sets which share the same transformation. The number of tied phone sets can be varied depending on the amount of available adaptation data. With more data the number of adaptation parameters which can be reliably trained increases, thus allowing more transformations and less phone tying. This idea is extended to RSW by increasing or decreasing the number phones which share the same tied weighting of reference speaker parameters.

Reference speaker weighting combines the two important adaptation ideas discussed above into one algorithm. This allows the system to rely heavily on the use of within-speaker correlation information when the amount of adaptation data is small, but also allows the adaptation training process to gradually reduce the constraints imposed by the *a prior* RSW parameters and become more focused on the specific details of individual phone models as more adaptation data becomes available.

The speaker clustering adaptation algorithms which were presented build on the other past speaker clustering research efforts. In particular, this thesis showed the usefulness of employing speaker clustering weighting (SCW) as an alternative to the standard hierarchical speaker clustering approach. In typical hierarchical speaker clustering approaches, a *hard decision* must be made about which cluster the current test speaker belongs to. In some situations, this decision may be too restrictive. A hard decision essentially forces the set of potential models to be a discrete predefined set. On the other hand, speaker cluster weighting allows the system to make a *soft decision* by permitting the final model set to be an interpolation of the model sets of the individual speaker clustered models. This allows the system to utilize any model set that falls into the continuum between the original discrete set of speaker clustered models.

Despite the relative successes of the RSW and SCW adaptation routines, the most significant contribution of this dissertation is the introduction of the consistency modeling approach to speech recognition. In Chapter 1, the consistency ratio was shown to capture all of the correlation information which is disregarded when the observations of an utterance are considered to be independent. Consistency modeling directly attacks the independence of observations assumption by attempting to estimate the contribution of the consistency ratio. This approach works by scoring the mutual information of pairs of phone observations in a hypothesis. Pairs of observations which are consistent with the assumption that they were produced by the same speaker will exhibit positive mutual information scores which will boost the overall score of the hypothesis they belong to. Likewise, pairs of observations which are inconsistent with the assumption that they were produced by the same speaker will exhibit negative mutual information scores thereby reducing the overall score of the hypothesis they belong to. The consistency model framework provides a mechanism for accounting for within-speaker correlation information without explicitly performing speaker adaptation.

The consistency modeling approach addresses one of the most troublesome assumptions made by in the frameworks of typical speech recognition systems. However, many engineering details still need to be ironed out if the approach is to become an integral part of the SUMMIT system. Also, it is important to note that this approach was designed as part of a segment-based system. It is not as well suited for use in a standard HMM system because of the potential amount of computation which could be incurred. In a frame-based approach the number of frame-based observation pairs that would need to be scored by the consistency model would be significantly higher than in a segment-based system.

The final contribution of this thesis is the successful combination of the techniques presented in this thesis into a single framework. By performing speaking clustering adaptation and consistency modeling simultaneously, the advantages of both can be accounted for. The combination of the different techniques provides improvements beyond those obtained when the techniques are utilized independently. These techniques reduced the error rate of the context independent SUMMIT system (which is typically used in our group's real time demonstration systems) by over 20%. These techniques also reduced the error rate of our state of the art context dependent system by over 6%.

8.3 Discussion of Final Results

In examining the final results reported in this thesis it is clear that providing speaker constraint to a speaker independent system can help improve the performance of a recognition system. This is demonstrated by the performance improvements obtained by the various methods introduced in this thesis. However, the comparative gains obtained by the various methods should be addressed. In particular it is desirable to understand why consistency modeling fails to perform as well as the various speaker clustering techniques. One would hope that consistency modeling should be able to perform at least as well as gender dependent modeling, especially considering that the acoustic differences between male and female speech are so pronounced. At this time, any answer to this question would only be speculation. Never the less, it seems likely that the consistency modeling approach may suffer from insufficient training data and/or insufficient modeling techniques.

It is important to consider the effects of the corpus on the results. The choice of the Resource Management corpus for the experiments presented in this thesis was based on several positive characteristics of the corpus. In particular the corpus contains two distinct data sets, one for speaker independent experiments and one for speaker dependent/adaptive experiments. Second the standard experiments for the task do not require the use of a statistical language model. This means that the word recognition performance of a recognizer is largely determined by the abilities of the recognizer's acoustic model. Finally, the corpus was modest in size allowing for faster turn-around time during the training and development stages of the experiments.

Despite its advantages, the RM corpus also contains one major shortcoming which may have adversely affected the results obtained from the consistency model; the amount of training data available in the Resource Management corpus is relatively small in comparison to the corpora collected more recently. In particular, the Wall Street Journal corpus contains more data while still containing many of the characteristics that made the RM corpus appealing. For instance, the Wall Street Journal (WSJ) corpus contains different data sets for speaker independent and speaker dependent experiments [67]. In retrospect, it is clear that the experiments conducted in this thesis would have required more effort to construct and more time to run had they been performed on the WSJ corpus. On the other hand, it is entirely possible that the relative performance gains observed by the various different modeling techniques, and the consistency modeling technique in particular, may have been greater had the system had the benefit of the additional training data. Unfortunately, this belief can not be confirmed at the time of the writing of this thesis.

A second major factor determining the effectiveness of the consistency modeling approach is the ability of the modeling techniques to capture the relevant correlation information available between the various speech events in the speech signal. In this thesis, the consistency model utilized techniques which have been developed and refined for the task of estimating phonetic likelihoods generated by the standard acoustic model. While the techniques for the standard acoustic model have been under development for years, these same techniques may not be the most effective techniques for consistency modeling. The purpose of the standard acoustic model is to discriminate between different phone hypotheses. The purpose of the consistency model, on the other hand, is to capture correlation information between different phones. Because of this difference, it is possible that new or different modeling techniques may be necessary to improve the capabilities of the consistency modeling approach.

Finally, the decision to use word error rate as the decision metric in this thesis should be discussed. While word error rate is the standard evaluation metric for many recognition tasks, it is not necessarily the most illuminating method of evaluation. In particular, it is difficult to utilize the performance on words to determine what types of errors exist on the phone level. Because the techniques introduced in this thesis are designed around the phonetic acoustic models, the phone error rate metric could be more useful for error analysis. Thus, it is reasonable to wonder whether or not an analysis of the techniques presented in this thesis on the task of phonetic recognition on the TIMIT corpus would have been fruitful [51]. In fact some preliminary experiments on TIMIT were conducted. However, these experiments were set aside because the preliminary results when using consistency modeling were not promising.

There are likely two reasons why the TIMIT database was not suitable for demonstrating the capabilities of the consistency modeling approach. First, the TIMIT corpus only contains eight utterances per training speaker. This shortage of data per speaker limits the ability of the consistency model to accurately account for the within-speaker correlation information. Second, it is possible that consistency modeling will not perform well when the error rates of the hypotheses are high and the correct answer is not likely to be found among the proposed hypotheses. In this case the consistency model may often be used to score phone pairs where the hypothesized identity of both phones is incorrect. In this case the estimate provided by the consistency model does not have any physical meaning and as such may only be contributing noise to the final consistency ratio estimate.

8.4 Future Extensions

In hindsight there is always the realization of different ideas or approaches that could have been pursued during the course of a research project. It is difficult to speculate on the relative usefulness of untested ideas, but it is worthwhile discussing potential new directions that this research could take none the less. Despite the successes of the research in this thesis, there is still much room for improvement for the adaptation and consistency modeling techniques presented in this thesis. Three areas where future research could be directed are: (1) improving the modeling and training techniques, (2) expanding to techniques to perform simultaneous speaker, environment, and channel adaptation, and (3) expanding the techniques to different models in the recognition system such as the pronunciation and language models.

In examining the reference speaker weighting technique employed in this thesis, it is obvious that the approach ignores a potential source of improvement. The technique currently adapts only the *location* of the acoustic models but not their *shape*. By performing model translation on the speaker independent model set, the system is successful in recentering the mass of the models to more appropriate locations, but it does not attempt to tighten the distribution of the model around its center of mass. To accomplish this, it may be possible to combine RSW adaptation with speaker adaptive training to produce a model with tighter variances that more closely resemble the shape of a speaker dependent model than the broad distribution of the full SI model. This would be especially useful in systems which have access to a small to moderate amount of adaptation data for a given speaker. In this case semi-reliable estimation of the acoustic model center of mass parameters may be possible but not enough data is available to provide estimates for the shape of the models.

In examining the speaker clustering techniques, it may be possible to improve the performance of these techniques by broadening the clustering criteria to account for different speaker properties. For example, the speakers could be clustered by major regional dialects. Examining unsupervised clustering techniques might also reveal new speaker properties that haven't yet been accounted for.

In examining the consistency modeling approach it is clear that improved training techniques are needed. The standard training techniques do not capture the within-speaker correlation information as efficiently as would be hoped. This is obvious from the fact that the speaker mixture training technique does as well as the aggregated standard training technique while being far more computationally efficient. Clearly, new alternative methods for creating the joint density functions that efficiently capture the correlation information are needed. The use of mixtures of full covariance Gaussians might be one potential improvement.

Another important future step is to expand the scope of this research to account for all within-utterance correlations. Thus the techniques should be expanded to account for effects from the environment and channel as well as the speaker. For example, if the consistency modeling approach is to be useful in a system such as the telephone-based JUPITER system, it must tighten its joint probability assumptions to account for joint observations occurring only within the same call or potentially only the same utterance. This is necessary because the environment or channel conditions may change from call to call even if the calls are from the same speaker. The speaker's own speech characteristics may change from call to call depending on the speaker's mood and state of health. The speaking style of the user could even change from utterance to utterance. This is often the case after a spoken language system makes a recognition error. The system's user will often react to the recognition error by slowing down and/or over-articulating his speech in the next utterance he provides to the system.

Finally, it is important to develop techniques which account for within-speaker correlations which exist in models other than the acoustic model. In Chapter 2, statistics were presented which demonstrate some of the correlations which exist in the pronunciation model. It is possible that performance improvements could be gained by accounting for the within-speaker correlations which exist in the duration, pronunciation and language models. The consistency modeling framework is easily extensible to these models. In fact, the consistency model framework is very similar to the trigger-based language modeling approach [55, 53]. In trigger-based language modeling, the likelihood of words are adjusted based on the observation of related *trigger* words. This allows the language model to adjust to the general topic being spoken in an utterance. In this respect, consistency language modeling could operate in essentially the same fashion; words common to a particular topic could be deemed consistent if they appeared in the same sentence while words from unrelated topics could be deemed inconsistent.

Appendix A

The HMM Recognizer

The hidden Markov model (HMM) approach is the most prevalent method for performing speech recognition today. This appendix provides a basic description of the derivation of the probabilistic framework used by typical HMM recognizers. To begin, HMM systems are provided with a frame-based sequence of observations, O, as input. These observations are typically vectors of acoustic parameters characterizing the spectral content of the speech waveform over equally spaced windows or *frames* in time. If a spoken utterance consists of N_f different frames the observation sequence can be represented as:

$$O = \{\vec{o}_1, \vec{o}_2, \ldots, \vec{o}_{N_f}\} \tag{A.1}$$

It is important to note that each phonetic speech event, or *phone* will usually last many frames in duration, depending on the absolute duration of the phone and the frame rate.

The goal of the HMM approach is to decode the observation sequence into the underlying state sequence Ψ which generated the acoustics observed in O. The sequence of underlying states determines the underlying phonetic content and word string of the spoken utterance. At every time frame the underlying state is assumed to have randomly generated an observation vector based on some likelihood density function. With an observation sequence containing N_f frames the underlying state sequence Ψ is represented as:

$$\Psi = \{\psi_1, \psi_2, \ldots, \psi_{N_f}\} \tag{A.2}$$

The single state sequence which is most likely to have generated O can be represented as Ψ'. The maximum *a posterior* probability expression for decoding O to find Ψ' is written as:

$$\Psi' = \arg\max_{\Psi} \mathrm{p}(\Psi|O) \tag{A.3}$$

Using Bayes Rule this can equivalently be written as:

$$\Psi' = \arg\max_{\Psi} \frac{\mathrm{p}(O|\Psi)\mathrm{p}(|\Psi)}{\mathrm{p}(O)} \tag{A.4}$$

Because $\mathrm{p}(O)$ is constant over all possible Ψ, this expression is further simplified to:

$$\Psi' = \arg\max_{\Psi} \mathrm{p}(O|\Psi)\mathrm{p}(\Psi) \tag{A.5}$$

In order to simplify the modeling and search techniques utilized to find the most likely path, the HMM approach makes two major simplifying assumptions. First, it is assumed that all observations are statistically independent of each other. This simplifies the probability density function of the observations as follows:

$$\mathrm{p}(O|\Psi) = \prod_{i=1}^{N_f} \mathrm{p}(\vec{o}_i|\Psi) \tag{A.6}$$

HMM's also assume context independence, i.e., an acoustic observation $\vec{o}_i$ is only dependent on the current state ψ_i and not on any past or future states. This assumption reduces the expression to:

$$\mathrm{p}(O|\Psi) = \prod_{i=1}^{N_f} \mathrm{p}(\vec{o}_i|\psi_i) \tag{A.7}$$

This assumption is not as severe as it may seem because context dependency can be encoded directly into the state identities. Encoding context dependency into the state identities would necessarily increase the number of states used in the model but would still allow the utilization of the context independent expression in (A.7).

The second major assumption made in the HMM is written directly into the model's moniker. The hidden Markov model derives its name from the application of the Markov assumption. With this assumption, the probability of transitioning into a state is only dependent on the identity of the last state the system occupied. The assumption allows the following equation to be written:

$$\mathrm{p}(\Psi) = \prod_{i=1}^{N_f} \mathrm{p}(\psi_i|\psi_{i-1}) \tag{A.8}$$

With the full set of assumptions described above, the context independent HMM can be expressed as:

$$\Psi' = \arg\max_{\Psi} \prod_{i=1}^{N_f} \mathrm{p}(\vec{o}_i|\psi_i)\mathrm{p}(\psi_i|\psi_{i-1}) \tag{A.9}$$

In (A.9) the $\mathrm{p}(\vec{o}_i|\psi_i)$ term is typically referred to as the *observation probability*. In speech recognition systems this term is also referred to as the *acoustic model*. The $\mathrm{p}(\psi_i|\psi_{i-1})$ term is typically referred to as the *state transition probability*.

The assumptions that are made in deriving the context-independent HMM allow the use of two efficient algorithms, the Baum-Welch algorithm (used to train the probability functions used by the HMM) and the Viterbi algorithm (used to search for the most likely state sequence) [70]. However, these assumptions disregard obvious correlations in the speech signal which may be useful when decoding the underlying phonetic content of the signal. In particular, it is the independence assumptions utilized in the acoustic model that are primarily examined in this dissertation. After obtaining an understanding of the flaws of the HMM, new speech recognition algorithms an be developed which which utilize the strong points of the HMM approach while correcting for the HMM's shortcomings.

Appendix B

The SUMMIT Recognizer

B.1 Probabilistic Framework

The SUMMIT system is a segment-based speech recognition engine developed in the Spoken Language Systems Group at MIT [32, 85, 86, 87, 88]. The SUMMIT system is based on a probabilistic framework describing the speech recognition problem. In this thesis SUMMIT is utilized to perform word recognition. The goal during recognition is to find the word sequence which is most likely given the acoustic information.

Typical speech recognition systems represent the acoustic information as a frame-based sequence of observations. Each observation is represented as a vector capturing information about the speech signal during a short window or *frame* in the speech waveform. The observation windows are usually equally spaced apart and slightly overlapping in time. The full set of frame-based observations will be called O and will be represented as:

$$O = \{\vec{o}_1, \vec{o}_2, \ldots, \vec{o}_{N_f}\} \tag{B.1}$$

In this expression, each $\vec{o}_n$ is a frame-based observation vector, and N_f is the total number of frames used to represent the waveform. The observation vectors usually capture spectral information using a spectral representation such as Mel frequency scale cepstral coefficients (MFCC's) [63]. In frame-based approaches, such as hidden Markov models (HMM's), recognition uses the frame-based observations directly in the probabilistic framework used for scoring and search. The goal is to find a hypothesized string of words, W', which is most likely given O. This is represented as:

$$W' = \arg\max_{W} \mathrm{p}(W|O) \tag{B.2}$$

In a segment-based system, such as SUMMIT, the recognition is performed using a segment network instead of a sequence of frames. Thus an intermediate step of transforming the sequence of frames, O, into a network of segment-based feature

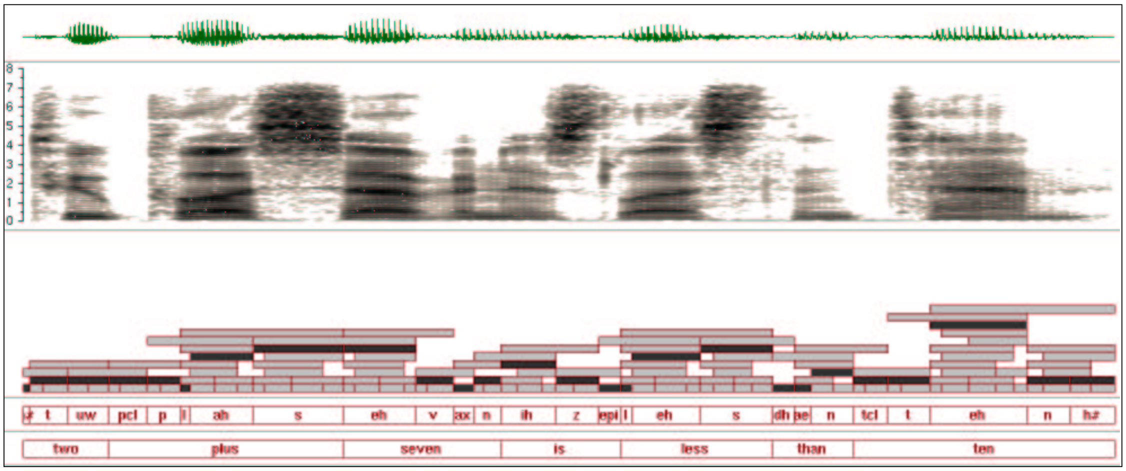


Figure B.1: Example utterance as displayed by SAPPHIRE. The display contains, from top to bottom, the waveform, the spectrogram, the segment network generated by SUMMIT, the phonetic transcription, and the orthographic transcription.

vectors, V, must be performed before the recognition process is begun. After the segment network is created, it replaces the observation sequence in the probabilistic framework. Thus, the generic probabilistic expression used to describe the recognition process performed by SUMMIT is represented as:

$$W' = \arg\max_{W} \mathrm{p}(W|V) \tag{B.3}$$

In SUMMIT each phonetic unit is presumed to occupy the space of one segmental unit. Thus, SUMMIT models a word as a sequence of phones each of which occupy one segment in the segment network. Figure B.1 demonstrates a segment network generated by SUMMIT as presented by the SAPPHIRE speech analysis and recognition tool [36]. In this figure, the waveform is shown on the top. The utterance's spectrogram is shown directly below the waveform. Below the spectrogram is the segment network. The time-aligned phone and word transcriptions are shown below the segment network. The string of segments corresponding to the underlying phonetic string is shown in the segment network in black. All of the hypothesized segments which are not part of the true path are shaded gray.

During the search for the best word sequence, W', SUMMIT simplifies the search procedure by simply choosing the single best path through the segment network (as opposed to summing the likelihoods of all paths that a specific word sequence could take through the network). In the process of maximizing the likelihood over the single best word sequence W', the search process also finds the single best path of segments through the segment network S' and the single best sequence of phonetic units P'. With these additions the probabilistic expression describing the recognition process becomes:

$$\{W', P', S'\} = \arg \max_{W,P,S} \mathrm{p}(W, P, S|V) \tag{B.4}$$

Using Bayes Rules, this probabilistic expression can be rewritten as:

$$\mathrm{p}(W, P, S|V) = \frac{\mathrm{p}(V|S, W, P)\mathrm{p}(S, W, P)}{\mathrm{p}(V)} \tag{B.5}$$

It can then be expanded to:

$$\mathrm{p}(W, P, S|V) = \frac{\mathrm{p}(V|S, W, P)\mathrm{p}(S|P, W)\mathrm{p}(P|W)\mathrm{p}(W)}{\mathrm{p}(V)} \tag{B.6}$$

Because, $\mathrm{p}(V)$ is constant over all W,P, and S the recognizer's decoding expression can be equivalently written as:

$$\{W', P', X'\} = \arg \max_{W,P,X} \mathrm{p}(V|S, W, P)\mathrm{p}(S|W, P)\mathrm{p}(P|W)\mathrm{p}(W) \tag{B.7}$$

Within this expression, the recognition framework contains four different probabilistic terms or *models.* These four models are referred to as:

- $\mathrm{p}(V|S, P, W) \Longrightarrow$ the acoustic model
- $\mathrm{p}(S|P, W) \Longrightarrow$ the duration model
- $\mathrm{p}(P|W) \Longrightarrow$ the pronunciation model
- $\mathrm{p}(W) \Longrightarrow$ the language model

B.2 Acoustic Modeling

B.2.1 Context Independent Framework

The acoustic model is represented by the expression, $\mathrm{p}(V|S,P,W)$. In SUMMIT, V represents acoustic feature vectors contained within a segment network. S represents a specific set of segments whose measurements are contained in V. Knowledge of S can be used to partition V into two subsets X and Y, where X contains the measurements of the segments in the path specified by S and Y contains the measurements of all of the segments of V that are not in S. Thus $X \cap Y = \emptyset$ and $X \cup Y = V$. Using these new definitions, the acoustic model can be equivalently expressed as:

$$\mathrm{p}(V|S,W,P) = \mathrm{p}(X,Y|S,W,P) = \mathrm{p}(X,Y|W,P) \tag{B.8}$$

Note that S can be left out of the final term $\mathrm{p}(X,Y|W,P)$ because X implies S.

In the expression $\mathrm{p}(X,Y|W,P)$, X represents all of the segments in the selected path while Y represents all of the segments which are not part of the path. In order to preserve the integrity of the probabilistic framework, it is necessary to account for all of the elements of both X and Y. Each element of X has a corresponding phonetic identity attached to it by the phonetic string P. To similarly account for the elements of Y, each element of Y can be mapped to a non-lexical unit called the *anti-phone*. This new segment class will be represent by the symbol $\bar{p}$.

In SUMMIT the elements of X and Y are considered independent. This allows the the following simplification:

$$\mathrm{p}(X,Y|W,P) = \mathrm{p}(X|W,P)\mathrm{p}(Y|W,P) \tag{B.9}$$

Because Y is known to include only anti-phone segments, the expression can be rewritten as:

$$\mathrm{p}(X,Y|W,P) = \mathrm{p}(X|W,P)\mathrm{p}(Y|\bar{p}) \tag{B.10}$$

The expression above implies that all segments in the network would have to be scored as either a phonetic unit or as the anti-phone unit in order to compute the acoustic score of any particular path. This can be avoided by recognizing that $\mathrm{p}(X,Y|\bar{p})$, which scores every segment in the segment network with the anti-phone model, is constant over all S. Using this information, (B.10) can be equivalently written as:

$$\mathrm{p}(X,Y|W,P) = \mathrm{p}(X|W,P)\mathrm{p}(Y|\bar{p})\frac{\mathrm{p}(X|\bar{p})}{\mathrm{p}(X|\bar{p})} = K\frac{\mathrm{p}(X|W,P)}{\mathrm{p}(X|\bar{p})} \tag{B.11}$$

Here, K represents the following expression:

$$K = \mathrm{p}(X,Y|\bar{p}) = \mathrm{p}(X|\bar{p})\mathrm{p}(Y|\bar{p}) \tag{B.12}$$

Note that the constant K can be ignored during actual scoring since it is applied to every path. Equation (B.11) demonstrates that the score for a path can be computed using only the segments in the path. The segments from the network which are not in the path are accounted for implicitly when the path scores are normalized by the anti-phone scores in $\mathrm{p}(X|\bar{p})$.

Like HMM's, SUMMIT also assumes that all segment observations are independent. Let the segment path X be represented as:

$$X = \{\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_N\} \tag{B.13}$$

Here, N is the number of segments in the path represented by X. If each segment is treated as independent then the acoustic model can be reduced to:

$$\mathrm{p}(X, Y|W, P) = K \prod_{n=1}^{N} \frac{\mathrm{p}(\vec{x}_n|W, P)}{\mathrm{p}(\vec{x}_n|\bar{p})} \tag{B.14}$$

In SUMMIT the segment scoring is dependent only on the phonetic string P and not on the word string W which P is drawn from. This allows the acoustic model expression to be further simplified to:

$$\mathrm{p}(X, Y|W, P) = K \prod_{n=1}^{N} \frac{\mathrm{p}(\vec{x}_n|P)}{\mathrm{p}(\vec{x}_n|\bar{p})} \tag{B.15}$$

Let the string of phones P corresponding to X be represented as:

$$P = \{p_1, p_2, \ldots, p_N\} \tag{B.16}$$

In this thesis, SUMMIT assumes context independence when scoring the segments. Using this approach the acoustic model finally reduces to:

$$\mathrm{p}(X, Y|W, P) = K \prod_{n=1}^{N} \frac{\mathrm{p}(\vec{x}_n|p_n)}{\mathrm{p}(\vec{x}_n|\bar{p})} \tag{B.17}$$

Recall that during the search the constant K can be ignored because it is applied equally to every path.

B.2.2 Context Dependent Framework

The previous section decribes a probabilistic framework for scoring a path of segment-based feature vectors extracted from a segment network given a hypothesized sequence of phones. The acoustic information contained in the segment network V can be augmented by a series of measurements made at acoustic landmarks in the utterance. The series of landmarks, which will be expressed as Z, represents the potential segment boundaries separating the segment elements in V. Acoustic measurements can be made which extend backwards and forwards from each landmark to capture the static and dynamic acoustic information which may be useful for classifying the landmark. Let, the sequence of landmarks be represented as:

$$Z = \{\vec{z_1}, \vec{z_2}, \ldots, \vec{z_L}\} \tag{B.18}$$

Here L is the total number of landmarks.

There are two main types of segment boundary landmarks. The first is a *transitional* boundary which exists between two adjacent segments in a path. The second is an *internal* boundary which falls inside the region of a segment in a path. For a given path X containing a total of N landmarks, there must be a total of $N-1$ transitional boundaries, one for every phonetic segment transition in the path. The remaining $L - N + 1$ landmarks must then be, by default, internal boundaries. Because the transitional boundaries contain acoustic measurements which extend over both the preceding and succeeding phones, the models capturing the acoustic information at the landmarks are called *diphone* models. These models provide context dependent modeling capabilities to SUMMIT.

The general expression for the acoustic model when both segment and diphone models are used is $\text{p}(V, Z|S, P, W)$. The measurements in V and Z are assumed to be independent by SUMMIT allowing the acoustic model to be expressed as:

$$\text{p}(V, Z|S, P, W) = \text{p}(V|S, P, W)\text{p}(Z|S, P, W) \tag{B.19}$$

From here $\text{p}(V|S, P, W)$ is expanded as described in the previous section. The diphone model also assumes that each landmark is considered independent and that the word string is not needed for scoring when given the phone string. These assumptions allow the diphone model to be expressed as:

$$\text{p}(Z|S, P, W) = \prod_{l=1}^{L} \text{p}(\vec{z_l}|S, P) \tag{B.20}$$

In SUMMIT diphone models are created to cover all possible internal and transitional boundaries. Because of sparse data problems for some boundaries, context sharing is used to ensure enough data exists for every model.

B.2.3 Density Function Modeling

All acoustic models in the SUMMIT system utilize mixtures of diagonal Gaussians to estimate the density function. These models are trained using the standard K-means and EM algorithms. The number of mixture components in each model is allowed to vary dependent on the size of the measurement vector and the number of available training vectors.

The measurement vectors used by the acoustic segment models are segment-based measurements primarily extracted from averages of frame-based MFCC's. In the context independent system each segment is represented with a 36 dimension vector containing segment-based measurements which were automatically selected based on their ability to perform phonetic discrimination [68]. The measurement vectors are all rotated using principal components analysis in order to remove global correlations before training. The boundary models used in the context dependent system are created in the same fashion using measurements extending forwards and backwards from each landmark.

B.3 The Duration Model

The duration model is represented by the expression, $\mathrm{p}(S|P, W)$. Although SUMMIT has the capability to incorporate a duration model, it currently does not use one because a duration feature is already included in the segment feature vectors used in V. A hierarchical duration model incorporating higher level information about the stress pattern, speaking rate, and word string is currently under development within our group [12].

B.4 Pronunciation Modeling

The pronunciation model is represented by the expression $\mathrm{p}(P|W)$. Every word in the vocabulary is initially represented with a phonetic baseform pronunciation and possible alternate pronunciations which are specific to that word. The baseform pronunciations are then expanded to account for alternative pronunciations caused by general phonological variations. The set of potential pronunciations for each word are represented by a phonetic network. To account for the fact that some phonological variations are more common than others, each arc in the network receives a score representing the likelihood of traversing that arc. The scores for each arc are determined using an error-correctiong training procedure.

B.5 Language Modeling

The language model is represented by the expression p(W). To account for the likelihood of observing a particular sequence of words, SUMMIT has the ability to incorporate various forms of N-gram language modeling into its search process. However, the experiments in this thesis do not utilize a statistical language model but rather just a word-pair grammar which specifies which words are allowed to follow any given word.

Appendix C

The Resource Management Corpus

The DARPA Resource Management (RM) Task Domain corpus was developed for the purpose of evaluating speech recognition performance for systems operating in a constrained domain [22, 69]. The corpus contains read utterances of 2800 different artificially generated sentences constrained by a predetermined grammar. The vocabulary size of the corpus is roughly 1000 words. The sentences themselves were designed to simulate database queries that might be spoken by naval personal when performing naval resource management tasks. Some example sentences are as follows:

```
What is Brooke's fuel level and fuel capacity?
Get me Kennedy's equipment readiness.
How many ships are currently in home port?
Show me chart with no speed data displayed.
Display tracks of any ships that are in Arctic Ocean.
```

The corpus contains two primary subsets of data. One subset was designed to evaluate speaker dependent (SD) recognition performance while the other was designed for speaker independent (SI) recognition.

The SD portion of the corpus contains utterances collected from 12 different speakers. Each speaker was recorded reading a total of 1012 utterances. Of these utterances 800 are standard sentences, 200 are utterances using spell-mode, 10 are specially designed rapid adaptation sentences, and the final 2 are specially designed dialect adaptation sentences. The standard utterances are subdivided into three independent sets: a training set, a development test set, and an evaluation test set. The training set for each speaker contains 600 utterances while the development and evaluation sets contain 100 utterances each.

The SI portion of the corpus contains utterances collected from 160 different speakers. These speakers were subdivided into three different subsets: a training set, a development test set, and an evaluation test set. The training set contains 80 speakers, the development set contains 40 speakers, and the evaluation set contains 40 speakers. Because some of the SI speakers overlap with the SD speakers, the SD speakers are typically excluded from the SI set to allow SI performance to be fairly evaluated on the SD speakers. With the SD speakers excluded, the training set contains 72 speakers, the development set contains 37 speakers and the evaluation set still contains 40 speakers. Within the SI training set each speaker read 40 standard sentences, 2 dialect sentences, and 15 spell-mode sentences. Within the SI development and evaluation sets each speaker read 30 standard sentences, 10 rapid adaptation sentences, 2 dialect sentences, and 15 spell-mode sentences.

For the experiments conducted in this thesis, only standard sentences were utilized. Thus, for SI models trained on the 109 speakers in the training and development sets, a total of 3990 utterances were available ($72 \times 40 = 2880$ from the training set and $37 \times 30 = 1110$ from the development set). The SI evaluation set contains a total of 1200 utterances.

In order for the utterances in the corpus to be used for training and analysis purposes, time-aligned word and phone transcriptions are required. These transcriptions are not provided with the corpus. For this thesis, these transcriptions are generated by the SUMMIT system using forced path alignment. Thus, it is possible that the transcriptions used during the experiments in this thesis contain segmentation errors introduced by the forced path alignment process.

Appendix D

Deleted Interpolation

D.1 Overview

In many speech recognition applications it is necessary to interpolate well-trained general models with more specific models which are not as well-trained. For example, in context dependent acoustic modeling specific context dependent models are smoothed with their more general context independent counterparts. In this thesis, the interpolation is performed between models for specific speaker types and more general models, such as the speaker independent (SI) model.

To provide an example, deleted interpolation can be performed between a general SI model and the more specific gender dependent model (GD). An interpolated gender dependent model (IGD) is the result. The density function for the IGD acoustic model for any given phone can be expressed as:

$$\mathrm{p}_{igd}(\vec{x}) = \lambda \mathrm{p}_{gd}(\vec{x}) + (1-\lambda)\mathrm{p}_{si}(\vec{x}) \tag{D.1}$$

In this expression, the density functions $\mathrm{p}_{gd}(\vec{x})$ and $\mathrm{p}_{si}(\vec{x})$ are trained in the standard fashion on the training data. However, the interpolation factor λ must also be determined. Ideally, this interpolation factor should be chosen so as to maximize the likelihood of *unseen* or *cross validation* data. There are two main issues which must be considered. First, a method for optimizing λ must be determined. The *expectation-maximization* (EM) algorithm fits this need. The EM algorithm is an iterative process which is guaranteed to adjust the parameters of a mixture density such that the total likelihood score produced over the cross-validation data is increased with every iteration [19]. Second, an adequate amount of cross-validation data must be produced in order to reliably determine the value for λ. The use of deleted interpolation is one means of addressing this problem.[1]

[1] This appendix draws freely from the explanation of deleted interpolation provided in [72].

D.2 The EM Algorithm

The EM algorithm is an iterative method for learning the parameters of a density function with the goal of maximizing the likelihood score it produces on a set of data. The algorithm is utilized when a closed form solution for determining the optimal set of density function parameters does not exist. For this thesis, the density function that is utilized is a mixture of fixed predetermined density functions. The parameters that must be learned are the weights applied to each of the fixed density functions contained in the mixture. The generic expression for this mixture of density functions is:

$$\mathrm{p}_{mix}(\vec{x}) = \sum_{i=1}^{F} \lambda_i \mathrm{p}_i(\vec{x}) \tag{D.2}$$

Here, each $\mathrm{p}_i(\vec{x})$ is a predetermined density function and λ_i is its corresponding weight. The total number of predetermined density functions is F.

Suppose that the set of all data available for training a model is represented as $\mathcal{T}$. It is a common practice for a set such as $\mathcal{T}$ to be divided into two independent subsets. One subset is typically used to train the parameters of the individual model density functions and is referred to as the *training* set. This set will be represented as $\mathcal{T}_T$. The second subset is typically used to optimize the free parameters which exist in the recognizer via cross-validation. This set is referred to as either the *cross-validation* set or the *development* set. This set will be represented as $\mathcal{T}_C$. Using this representation the following expressions hold:

$$\mathcal{T}_T \cup \mathcal{T}_C = \mathcal{T} \quad \text{and} \quad \mathcal{T}_T \cap \mathcal{T}_C = \emptyset \tag{D.3}$$

Let $\mathcal{T}_T$ be used to train each of the $\mathrm{p}_i(\vec{x})$ density functions and let $\mathcal{T}_C$ be used to optimize the λ_i weights via the EM algorithm. The EM algorithm performs its optimization using the following iterative process:

1. Initialize the λ_i values with guessed estimates.
2. Calculate updated values λ'_i for each λ_i using $\mathcal{T}_C$.
3. If $\lambda'_i \approx \lambda_i$ for all i then stop. Otherwise repeat Step 2.

The update equation used in Step 2 is written as:

$$\lambda'_i = \frac{1}{N_C} \sum_{n=1}^{N_c} \frac{\lambda_i \mathrm{p}_i(\vec{x}_n)}{\mathrm{p}_{mix}(\vec{x}_n)} = \frac{1}{N_C} \sum_{n=1}^{N_c} \frac{\lambda_i \mathrm{p}_i(\vec{x}_n)}{\sum_{j=1}^{F} \lambda_j \mathrm{p}_j(\vec{x}_n)} \tag{D.4}$$

Here, N_C is the number of observations in $\mathcal{T}_C$ that are used for determining the values of the λ_i weights of $\mathrm{p}_{mix}(\vec{x})$.

D.3 Deleted Interpolation

The disadvantage of using a simple cross-validation set for determining the interpolation weights of different models is that (a) the cross-validation set must be large enough to produce reliable estimates for the weights, and (b) the training set must be large enough to produce reliable estimates for the models. There is tension between points (a) and (b) because the cross validation set can not have any intersection with the training set. Because the cross validation data cannot be used to train the individual models used in the interpolated mixture, the process must essentially *steal* data from the training set in order to create the cross validation set. The more data that is taken from the training set for cross validation, the better the estimate of the interpolation weights will be, but the worse the estimates of the individual density functions will be. To alleviate this problem, deleted interpolation can be utilized [37].

The basic idea behind deleted interpolation is that multiple cross-validation data sets are created via a *jack-knifing* process applied to the training set. To illustrate this process, suppose the set $\mathcal{T}$ for a particular model can be partitioned into B different jack-knifed blocks of data as follows:

$$\mathcal{T} = \{\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_B\} \tag{D.5}$$

The sets obey the following constraints:

$$\mathcal{T}_1 \cup \mathcal{T}_2 \cup \cdots \cup \mathcal{T}_B = \mathcal{T} \tag{D.6}$$

$$\mathcal{T}_i \cap \mathcal{T}_j = \emptyset \text{ if } i \neq j \tag{D.7}$$

For each jack-knifed cross-validation set $\mathcal{T}_b$, a set of density functions can be trained from all training data not including the data in $\mathcal{T}_b$ (i.e., on $\mathcal{T} - \mathcal{T}_b$). Let $\mathrm{p}_j^{\mathcal{T}-\mathcal{T}_b}(\vec{x})$ be the j^{th} density function for the set of models training using all of the training data except $\mathcal{T}_b$. Also, let the observations contained in each block $\mathcal{T}_b$ be represented as:

$$\mathcal{T}_b = \{\vec{x}_1^{\,b}, \vec{x}_1^{\,b}, \ldots, \vec{x}_{N_b}^{\,b}\} \tag{D.8}$$

Here, N_b is the number observations in block $\mathcal{T}_b$.

By jack-knifing the training set into B different blocks, every data point in $\mathcal{T}$ can be used as unseen cross-validation data provided the appropriate set of models, $\mathrm{p}_j^{\mathcal{T}-\mathcal{T}_b}(\vec{x})$, is applied to it. This process of optimizing the weights using rotating blocks of jack-knifed data is called deleted interpolation. The process uses the same EM algorithm, but the new update equation that is used is as follows:

$$\lambda_i' = \frac{1}{|\mathcal{T}|} \sum_{b=1}^{B} \sum_{n=1}^{N_b} \frac{\lambda_i \mathrm{p}_i^{\mathcal{T}-\mathcal{T}_b}(\vec{x}_n^{\,b})}{\sum_{j=1}^{F} \lambda_j \mathrm{p}_j^{\mathcal{T}-\mathcal{T}_b}(\vec{x}_n^{\,b})} \tag{D.9}$$

Once the weights have been estimated, the final mixture density expression is:

$$\mathrm{p}_{mix}(\vec{x}) = \sum_{i=1}^{F} \lambda_i \mathrm{p}_i^{\mathcal{T}}(\vec{x}) \tag{D.10}$$

Here, each density function $\mathrm{p}_i^{\mathcal{T}}$ is trained over the entire set $\mathcal{T}$. The use of deleted interpolation has thus allowed the full set $\mathcal{T}$ to be used for both the training of the acoustic models and the optimization of the weights.

D.4 Incorporation into SUMMIT

The process described above details how deleted interpolation can be used to find a set of weights used in a mixture of density functions. The description details the steps needed to create one density function. In the SUMMIT system (or any other comparable recognition system) a density function is required for the acoustic model of every phone. In this case, the density function for a particular phone model m can be written as:

$$\mathrm{p}_{mix}(\vec{x}|\, p{=}m) = \sum_{i=1}^{F} \lambda_{m,i} \mathrm{p}_i^{\mathcal{T}}(\vec{x}|\, p{=}m) \tag{D.11}$$

Using this notation, it should be clear that each acoustic model $\mathrm{p}_{mix}(\vec{x}|\, p{=}m)$ will require it's own set of $\lambda_{m,i}$ weights. To accomplish this, the deleted interpolation process is executed independently for each acoustic model.

Appendix E

Model Aggregation

E.1 Problem Definition

The parameters of mixture Gaussian density functions are typically trained using an unsupervised hill-climbing algorithm which attempts to find the set of parameters which maximize the likelihood of the training data. A hill-climbing algorithm is used because no closed form solution for finding the globally optimal set of parameters exists for mixture Gaussian models. The algorithms typically used to determine the density function parameters, the K–means clustering algorithm and the Expectation-Maximization (EM) algorithm, do not guarantee a globally optimal solution. These algorithms often converge to a locally optimal solution, where the exact local optimum that is reached is highly dependent on the initialization of the parameters at the beginning of the training process. Thus, different initializations could result in markedly different sets of model parameters.

Because of the differences that can arise between models trained from different initializations, a common practice is to train multiple models from different initializations and then choose the one model which is most optimal based some predetermined criterion. One method for choosing a model is to pick the model with the highest likelihood on the training data. This method is acceptable if there is enough training data to ensure that the density function parameters are not over-fitting the training data. Unfortunately, how well a model will generalize to unseen data can not be determined solely on the likelihood score achieved on training data.

A second method of choosing a model from a set of training trials is to simply choose the model that performs the best on the development or cross-validation data. One problem with this strategy is that noise on the development data contributes a random component to the performance. As a result, better performance on the development set may not indicate models which are better matched to the true underlying distribution of the data unless the development set is very large. Instead, it may only indicate that the models are superficially better matched to the idiosyncrasies of the development set.

Thus, neither of the methods listed above is guaranteed to generate a model which is significantly better than a model which is drawn randomly from the set of estimated models. Both of the methods also suffer from the disadvantage that computation is wasted. The results of only one training trial are kept, while the models from the other trials are thrown away [9]. Ideally, a training routine should be able to utilize whatever is learned from all of the training trials and not just one selected trial.

To counter the problems discussed above, an algorithm is needed which produces a mixture density function which can be proven to yield better accuracy, on average, than any randomly initialized density function trained using standard techniques. Aggregation is a technique which meets this criterion [9]. Aggregation improves the performance of models which exhibit *uncertainty* or *instability* during their training phase. Aggregation has been applied to a variety of types of predictors and classifiers. For example, Breiman has shown the effectiveness of a specific type of aggregation known as *bagging* (or *bootstrap aggregating*) on linear regression predictors and on classification trees [10]. In this thesis, aggregation is utilized to improve the likelihood estimates generated by mixture Gaussian density functions.[1]

E.2 Theory

Aggregation of probabilistic density functions is performed by averaging the outputs of a set of independently trained models. The proof that follows will demonstrate that an aggregate density function is guaranteed to exhibit an error metric which is equal to or better than the average error metric of the individual density functions used to create the aggregate model. This proof is completely independent of the test data being presented to the classifier. Thus, the method is robust because it improves performance regardless of the test set being used.

To begin, assume that the true underlying density function for a particular model is represented as $\mathrm{p}(\vec{x})$. Next assume a set of N different density functions which attempt to estimate $\mathrm{p}(\vec{x})$ have been trained from a set of training data. Each individual density function can be represented as $\hat{\mathrm{p}}_n(\vec{x})$ where n is the index of the training trial. In this thesis, multiple density functions are generated from the same data set by using different random initializations in the K–means clustering prior to EM optimization of the mixture parameters. However, the proof does not depend in any way on how the classifiers are generated.

[1] This appendix draws freely from work conducted jointly with Andrew Halberstadt which is presented in [35].

In order to evaluate the accuracy of a density function, an appropriate error metric must be defined. Let the squared error of the likelihood be the error metric used to evaluate the accuracy of an estimated density function. For a particular observation $\vec{x}$, the squared error for the estimated model from training trial n is defined as:

$$e_n(\vec{x}) = (\mathrm{p}(\vec{x}) - \hat{\mathrm{p}}_n(\vec{x}))^2 \tag{E.1}$$

Using this error metric, the mean squared error for an input vector $\vec{x}$ averaged over the models from all N training trials is expressed as:

$$e(\vec{x}) = \frac{1}{N}\sum_{n=1}^{N} e_n(\vec{x}) = \frac{1}{N}\sum_{n=1}^{N}(\mathrm{p}(\vec{x}) - \hat{\mathrm{p}}_n(\vec{x}))^2 \tag{E.2}$$

The mean error of the N estimates expands to:

$$e(\vec{x}) = \frac{1}{N}\sum_{n=1}^{N} \mathrm{p}^2(\vec{x}) - 2\mathrm{p}(\vec{x})\hat{\mathrm{p}}_n(\vec{x}) + \hat{\mathrm{p}}_n^2(\vec{x}) \tag{E.3}$$

This can be rewritten as:

$$e(\vec{x}) = \mathrm{p}^2(\vec{x}) - \frac{2\mathrm{p}(\vec{x})}{N}\sum_{n=1}^{N} \hat{\mathrm{p}}_n(\vec{x}) + \frac{1}{N}\sum_{n=1}^{N} \hat{\mathrm{p}}_n^2(\vec{x}) \tag{E.4}$$

The aggregate density function simply averages the outputs of the density functions from the N different training trials. The aggregate density function is represented as $\hat{\mathrm{p}}_A(\vec{x})$ and is expressed as:

$$\hat{\mathrm{p}}_A(\vec{x}) = \frac{1}{N}\sum_{n=1}^{N} \hat{\mathrm{p}}_n(\vec{x}) \tag{E.5}$$

The error for the aggregate classifier model is expressed as:

$$e_A(\vec{x}) = (\mathrm{p}(\vec{x}) - \hat{\mathrm{p}}_A(\vec{x}))^2 = \mathrm{p}^2(\vec{x}) - 2\mathrm{p}(\vec{x})\hat{\mathrm{p}}_A(\vec{x}) + \hat{\mathrm{p}}_A^2(\vec{x}) \tag{E.6}$$

By substituting in the definition of $\hat{\mathrm{p}}_A(\vec{x})$ from (E.5), the error of the aggregate density can be rewritten as:

$$e_A(\vec{x}) = \mathrm{p}^2(\vec{x}) - \frac{2\mathrm{p}(\vec{x})}{N}\sum_{n=1}^{N} \hat{\mathrm{p}}_n(\vec{x}) + \left(\frac{1}{N}\sum_{n=1}^{N} \hat{\mathrm{p}}_n(\vec{x})\right)^2 \tag{E.7}$$

By comparing the expressions in (E.4) and (E.7), it can be seen that $e_A(\vec{x})$ will be less than or equal to $e(\vec{x})$ if:

$$\left(\frac{1}{N}\sum_{n=1}^{N} \hat{\mathrm{p}}_n(\vec{x})\right)^2 \leq \frac{1}{N}\sum_{n=1}^{N} \hat{\mathrm{p}}_n^2(\vec{x}) \tag{E.8}$$

In fact, this condition is always true for any arbitrary vector because it is a special case of the Cauchy–Schwarz inequality. Given any two vectors $\vec{a} = [a_1, a_2, \dots a_N]^T$ and $\vec{b} = [b_1, b_2, \dots b_N]^T$, the Cauchy–Schwartz inequality states:

$$\left|\sum_{n=1}^{N} a_n b_n\right|^2 \leq \left(\sum_{n=1}^{N} a_n^2\right)\left(\sum_{n=1}^{N} b_n^2\right) \tag{E.9}$$

Now let $b_n = 1$ for all n so that $\sum_{n=1}^{N} b_n^2 = N$ to obtain:

$$\left(\sum_{n=1}^{N} a_n\right)^2 \leq N \sum_{n=1}^{N} a_n^2 \tag{E.10}$$

This can be rewritten as follows:

$$\left(\frac{1}{N}\sum_{n=1}^{N} a_n\right)^2 \leq \frac{1}{N} \sum_{n=1}^{N} a_n^2 \tag{E.11}$$

This is the desired result which thereby proves that, for *any* input token $\vec{x}$, the error $e_A(\vec{x})$ of the aggregate density function is always equal to or smaller than the average error $e(\vec{x})$ of the N individual density functions which are used to create the aggregate model. Note that equality holds in (E.8) only if all N individual density functions are identical. Thus, in practical situations with density functions that produce different likelihoods, the inequality becomes strict.

E.3 Practical Considerations

In practice, creating an aggregate density function is simply a matter of combining all of the Gaussian components from each of the N different trials together into one large aggregate mixture with the Gaussian components in each individual mixture weighted by a multiplicative factor of $\frac{1}{N}$ to maintain probabilistic integrity. Thus, if each individual density function contains M different mixture components then the aggregate model contains a total of $N \times M$ different components. Thus, the gain in accuracy produced by aggregation comes at the cost of additional computation caused by the added number of parameters in the aggregate model. The computation issues and potential methods for computational savings are addressed in [35].

Bibliography

[1] M. Afify, Y. Gong, and J. Haton. Correlation based predictive adaptation of hidden Markov models. In *Proceedings of the 5th European Conference on Speech Communication and Technology*, pages 2111–2114, Rhodes, Greece, 1997.

[2] S. Ahadi and P. Woodland. Rapid speaker adaptation using model prediction. In *Proceedings of the 1995 International Conference on Acoustics, Speech and Signal Processing*, pages 684–687, Detroit, MI, 1995.

[3] T. Anastasakos, J. McDonough, and J. Makhoul. Speaker adaptive training: A maximum likelihood approach to speaker normalization. In *Proceedings of the 1997 International Conference on Acoustics, Speech and Signal Processing*, pages 1043–1046, Munich, Germany, 1997.

[4] T. Anastasakos, J. McDonough, and R. Schwartz. A compact model for speaker-adaptive training. In *Proceedings of the 1996 International Conference on Spoken Language Processing*, pages 1137–1140, Philadelphia, PA, 1996.

[5] L. Bahl, S. Balakrishnan-Aiyer, J. Bellegarda, M. Franz, P. Gopalakrishnan, D. Nahamoo, M. Novak, M. Padmanabhan, M. Picheny, and S. Roukos. Performance of the IBM large vocabulary continuous speech recognition system on the ARPA Wall Street Journal task. In *Proceedings of the 1995 International Conference on Acoustics, Speech and Signal Processing*, pages 41–44, Detroit, MI, 1995.

[6] L. Bahl, P. Brown, P. de Souza, R. Mercer, and D. Nahamoo. A fast algorithm for deleted interpolation. In *Proceedings of the 2nd European Conference on Speech Communication and Technology*, pages 1209–1212, Genova, Italy, 1991.

[7] L. Bahl, F. Jelinek, and R. Mercer. A maximum likelihood approach to continuous speech recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-5(2), pages 179–190, March 1984.

[8] J. Baker. *Stochastic Modeling as a Means of Automatic Speech Recognition.* PhD thesis, Carnegie Mellon University, April 1975.

[9] C. Bishop. *Neural Networks for Pattern Recognition.* Oxford University Press, New York, NY, 1995.

[10] L. Breiman. Bagging predictors. *Machine Learning*, 24(2), pages 123–140, 1996.

[11] S. Chen and P. DeSouza. Speaker adaptation by correlation (ABC). In *Proceedings of the 5th European Conference on Speech Communication and Technology*, pages 2111–2114, Rhodes, Greece, 1997.

[12] G. Chung and S. Seneff. Hierarchical duration modelling for speech recognition using the ANGIE framework. In *Proceedings of the 5th European Conference on Speech Communication and Technology*, pages 1475–1478, Rhodes, Greece, 1997.

[13] D. Van Compernolle, J. Smolders, P. Jaspers, and T. Hellemans. Speaker clustering for dialectic robustness in speaker independent recognition. In *Proceedings of the 2nd European Conference on Speech Communication and Technology*, pages 723–726, Genova, Italy, 1991.

[14] S. Cox. Speaker adaptation using a predictive model. In *Proceedings of the 3rd European Conference on Speech Communication and Technology*, pages 2283–2286, Berlin, Germany, 1993.

[15] J. Dalby. *Phonetic Structure of Fast Speech in American English.* PhD thesis, Indiana University, December 1984.

[16] V. Digalakis and L. Neumeyer. Speaker adaptation using combined transformation and Bayesian methods. In *Proceedings of the 1995 International Conference on Acoustics, Speech and Signal Processing*, pages 680–683, Detroit, MI, 1995.

[17] V. Digalakis, D. Rtischev, and L. Neumeyer. Speaker adaptation using constrained reestimation of gaussian mixtures. *IEEE Transactions on Speech and Audio Processing*, 3(5), pages 357–366, September 1994.

[18] Dragon naturally speaking. Published by Dragon Systems, Inc. Located at URL *http://www.dragonsystems.com/marketing/pcproducts.html*, 1997.

[19] R. Duda and P. Hart. *Pattern Classification and Scene Analysis.* John Wiley & Sons, New York, NY, 1973.

[20] E. Eide and H. Gish. A parametric approach to vocal tract length normalization. In *Proceedings of the 1996 International Conference on Acoustics, Speech and Signal Processing*, pages 346–349, Atlanta, GA, 1996.

[21] G. Fant. *Acoustic Theory of Speech Production.* Mouton, The Hague, The Netherlands, 1970.

[22] W. Fisher. The DARPA task domain speech recognition database. In *Proceedings of the DARPA Speech Recognition Workshop*, pages 105–109, San Diego, CA, March 1987.

[23] S. Furui. Unsupervised speaker adaptation method based on hierarchical spectral clustering. In *Proceedings of the 1989 International Conference on Acoustics, Speech and Signal Processing*, pages 286–289, Glasgow, Scotland, 1989.

[24] Y. Gao, M. Padmanabhan, and M. Picheny. Speaker adaptation based on preclustering training speakers. In *Proceedings of the 5th European Conference on Speech Communication and Technology*, pages 2091–2094, Rhodes, Greece, 1997.

[25] J. Gauvain, L. Lamel, and M. Adda-Decker. Developments in continuous speech dictation using the ARPA WSJ task. In *Proceedings of the 1995 International Conference on Acoustics, Speech and Signal Processing*, pages 65–68, Detroit, MI, 1995.

[26] J. Gauvain and C. Lee. Bayesian learning for hidden Markov model with Gaussian mixture state observation densities. In *Proceedings of the 2nd European Conference on Speech Communication and Technology*, pages 939–942, Genova, Italy, 1991.

[27] J. Gauvain and C. Lee. Bayesian learning of Gaussian mixture densities for hidden Markov models. In *Proceedings of the DARPA Speech and Natural Language Workshop*, pages 272–277, Pacific Grove, CA, February 1991.

[28] J. Gauvain and C. Lee. Improved acoustic modeling with Bayesian learning. In *Proceedings of the 1992 International Conference on Acoustics, Speech and Signal Processing*, volume I, pages 481–484, San Francisco, CA, 1992.

[29] J. Gauvain and C. Lee. MAP estimation of continuous density HMM: Theory and application. In *Proceedings of the DARPA Speech and Natural Language Workshop*, pages 185–190, Harriman, NY, February 1992.

[30] J. Gauvain and C. Lee. Maximum *a posteriori* estimation for multivariate Gaussian mixture observation of Markov chains. *IEEE Transactions on Speech and Audio Processing*, 2(2), pages 291–298, April 1994.

[31] L. Gillick and S. Cox. Some statistical issues in the comparison of speech recognition algorithms. In *Proceedings of the 1989 International Conference on Acoustics, Speech and Signal Processing*, pages 532–535, Glasgow, Scotland, 1989.

[32] J. Glass, J. Chang, and M. McCandless. A probabilistic framework for feature-based speech recognition. In *Proceedings of the 1996 International Conference on Spoken Language Processing*, pages 2277–2280, Philadelphia, PA, 1996.

[33] D. Goddeau, E. Brill, J. Glass, C. Pao, and M. Phillips. GALAXY: A human-language interface to on-line travel information. In *Proceedings of the 1994 International Conference on Spoken Language Processing*, pages 707–710, Yokohama, Japan, 1994.

[34] T. Hazen. Probabilistic transfer vector prediction for speaker adaptation. Technical Report TR-IT-0124, ATR, Kyoto, Japan, August 1995.

[35] T. Hazen and A. Halberstadt. Using aggregation to improve the performance of mixture Gaussian acoustic models. In *Proceedings of the The 1998 International Conference on Acoustics, Speech and Signal Processing*, Seattle, WA, 1998.

[36] L. Hetherington and M. McCandless. SAPPHIRE: An extensible speech analysis and recognition tool based on Tcl/Tk. In *Proceedings of the 1996 International Conference on Spoken Language Processing*, pages 1942–1945, Philadelphia, PA, 1996.

[37] X. Huang, M. Hwang, L. Jiang, and M. Mahajan. Deleted interpolation and density sharing for continuous hidden Markov models. In *Proceedings of the 1996 International Conference on Acoustics, Speech and Signal Processing*, pages 885–888, Atlanta, GA, 1996.

[38] Q. Huo and C. Chan. On-line Bayes adaptation of SCHMM parameters for speech recognition. In *Proceedings of the 1995 International Conference on Acoustics, Speech and Signal Processing*, pages 708–711, Detroit, MI, 1995.

[39] Q. Huo, C. Chan, and C. Lee. Bayesian learning of SCHMM parameters for speech recognition. In *Proceedings of the 1994 International Conference on Acoustics, Speech and Signal Processing*, volume I, pages 221–224, Adelaide, Australia, 1994.

[40] Q. Huo and C. Lee. On-line adaptive learning of the correlated continuous density hidden Markov models for speech recognition. In *Proceedings of the 1996 International Conference on Spoken Language Processing*, pages 985–988, Philadelphia, PA, 1996.

[41] Q. Huo and C. Lee. Combined on-line model adaptation and Bayesian predictive classification for robust speecg recognition. In *Proceedings of the 5th European Conference on Speech Communication and Technology*, pages 1847–1850, Rhodes, Greece, 1997.

[42] IBM ViaVoice. Published by IBM Corporation. Located at URL *http://www.software.ibm.com/is/voicetype/us_vv.html*, 1997.

[43] T. Kamm, A. Andreou, and J. Cohen. Vocal tract normalization in speech recognition: compensating for systematic speaker variability. In *Proceedings of the 15th Annual Speech Research Symposium*, Baltimore, MD, June 1997.

[44] A. Kannan and M. Ostendorf. Modeling dependency in adaptation of acoustic models using multiscale tree processes. In *Proceedings of the 5th European Conference on Speech Communication and Technology*, pages 1863–1866, Rhodes, Greece, 1997.

[45] P. Kenny, R. Hollan, V. Gupta, M. Lennig, P. Mermelstein, and D. O'Shaughnessy. A*-admissible heurustics for rapid lexical access. In *Proceedings of the 1991 International Conference on Acoustics, Speech and Signal Processing*, pages 689–692, Toronto, Canada, 1991.

[46] D. Klatt and L. Klatt. Analysis, synthesis, and perception of voice quality variations among female and male talkers. *Journal of the Acoustical Society of America*, 87(2), pages 820–857, February 1990.

[47] T. Kosaka, S. Matsunaga, and S. Sagayama. Tree-structured speaker clustering for speaker-independent continuous speech recognition. In *Proceedings of the 1994 International Conference on Spoken Language Processing*, pages 1375–1378, Yokohama, Japan, 1994.

[48] T. Kosaka and S. Sagayama. Tree-structured speaker clustering for fast speaker adaptation. In *Proceedings of the 1994 International Conference on Acoustics, Speech and Signal Processing*, volume I, pages 245–248, Adelaide, Australia, 1994.

[49] F. Kubala, H. Jin, S. Matsoukas, L. Nguyen, R. Schwartz, and J. Makhoul. Advances in transcription of broadcast news. In *Proceedings of the 5th European Conference on Speech Communication and Technology*, pages 927–930, Rhodes, Greece, 1997.

[50] H. Kuwabara. Acoustic and perceptual properties of phonemes in continuous speech as a function of speaking rate. In *Proceedings of the 5th European Conference on Speech Communication and Technology*, pages 1003–1006, Rhodes, Greece, 1997.

[51] L. Lamel, R. Kassel, and S. Seneff. Speech database development: Design and analysis of the acoustic-phonetic corpus. In *Proceedings of the DARPA Speech Recognition Workshop*, pages 100–109, Palo Alto, CA, February 1986.

[52] M. Lasry and R. Stern. *A posteriori* estimation of correlated jointly Gaussian mean vectors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(4), pages 530–535, July 1984.

[53] R. Lau. *Adaptive Statistical Language Modelling.* Master's thesis, Massachusetts Institute of Technology, May 1994.

[54] R. Lau, G. Flammia, C. Pao, and V. Zue. WebGalaxy – integrating spoken language and hypertext navigation. In *Proceedings of the 5th European Conference on Speech Communication and Technology*, pages 883–886, Rhodes, Greece, 1997.

[55] R. Lau, R. Rosenfeld, and S. Roukos. Trigger-based language models: A maximum entropy approach. In *Proceedings of the 1993 International Conference on Acoustics, Speech and Signal Processing*, volume II, pages 45–48, Minneapolis, MN, April 1993.

[56] C. Lee and J. Gauvain. Speaker adaptation based on MAP estimation of HMM parameters. In *Proceedings of the 1993 International Conference on Acoustics, Speech and Signal Processing*, volume II, pages 558–561, Minneapolis, MN, 1993.

[57] K. Lee. *Large Vocabulary Speaker-Independent Continuous Speech Recognition: The Development of the SPHINX System.* PhD thesis, Carnegie Mellon University, April 1988.

[58] L. Lee and R. Rose. Speaker normalization using efficient frequency warping procedures. In *Proceedings of the 1996 International Conference on Acoustics, Speech and Signal Processing*, pages 353–356, Atlanta, GA, 1996.

[59] C. Leggetter and P. Woodland. Speaker adaptation of continuous density HMMs using multivariate linear regression. In *Proceedings of the 1994 International Conference on Spoken Language Processing*, pages 451–454, Yokohama, Japan, 1994.

[60] C. Leggetter and P. Woodland. Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models. *Computer Speech and Language*, 9(2), pages 171–185, April 1995.

[61] A. Ljolje. Speaker clustering for improved speech recognition. In *Proceedings of the 3rd European Conference on Speech Communication and Technology*, pages 631–634, Berlin, Germany, 1993.

[62] L. Mathan and L. Miclet. Speaker hierarchical clustering for improving speaker-independent HMM word recognition. In *Proceedings of the 1990 International Conference on Acoustics, Speech and Signal Processing*, pages 149–152, Albuquerque, NM, 1990.

[63] P. Mermelstein and S. Davis. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 28(4), pages 357, August 1980.

[64] N. Morgan, E. Fosler, and N. Mirghafori. Speech recognition using on-line estimation of speaking rate. In *Proceedings of the 5th European Conference on Speech Communication and Technology*, pages 2079–2082, Rhodes, Greece, 1997.

[65] P. Niyogi. *Modeling Speaker Variability and Imposing Speaker Constraints in Phonetic Classification.* Master's thesis, Massachusetts Institute of Technology, December 1991.

[66] M. Padmanabhan, L. Bahl, D. Nahamoo, and M. Picheny. Speaker clustering and transformation for speaker adaptation in large-vocabulary speech recognition systems. In *Proceedings of the 1996 International Conference on Acoustics, Speech and Signal Processing*, pages 701–704, Atlanta, GA, 1996.

[67] D. Paul and J. Baker. The Design for the Wall Street Journal-based CSR corpus. In *Proceedings of the DARPA Speech and Natural Language Workshop*, pages 357–362, Harriman, NY, February 1992.

[68] M. Phillips and V. Zue. Automatic discovery of acoustic measurements for phonetic classification. In *Proceedings of the 1992 International Conference on Spoken Language Processing*, pages 795–798, Banff, Canada, 1992.

[69] P. Price, W. Fisher, J. Bernstein, and D. Pallett. The DARPA 1000-word Resource Management database for continuous speech recognition. In *Proceedings of the 1988 International Conference on Acoustics, Speech and Signal Processing*, pages 651–654, New York, NY, 1988.

[70] L. Rabiner and B. Juang. *Fundamentals of Speech Recognition.* PTR Prentice-Hall, Englewood Cliffs, NJ, 1993.

[71] A. Sankar, L. Neumeyer, and M. Weintraub. An experimental study of acoustic adaptation algorithms. In *Proceedings of the 1996 International Conference on Acoustics, Speech and Signal Processing*, pages 713–716, Atlanta, GA, 1996.

[72] B. Serridge. *Context-dependent Modeling in a Segment-based Speech Recognition System.* Master's thesis, Massachusetts Institute of Technology, August 1997.

[73] K. Shinoda and T. Watanabe. Speaker adaptation with autonomous model complexity control by MDL principle. In *Proceedings of the 1996 International Conference on Acoustics, Speech and Signal Processing*, pages 717–720, Atlanta, GA, 1996.

[74] M. Siegler and R. Stern. On the effects of speech rate in large vocabulary speech recognition systems. In *Proceedings of the 1995 International Conference on Acoustics, Speech and Signal Processing*, pages 612–615, Detroit, MI, 1995.

[75] M. Sondhi and B. Gopinath. Determination of vocal-tract shape from impulse responses at the lips. *Journal of the Acoustical Society of America*, 49(6), pages 1847–1873, June 1971.

[76] F. Soong and E. Huang. A tree-trellis based fast search for finding the N best sentence hypotheses in continuous speech recognition. In *Proceedings of the 1991 International Conference on Acoustics, Speech and Signal Processing*, pages 705–708, Toronto, Canada, 1991.

[77] R. Stern and M. Lasry. Dynamic speaker adaptation for feature-based isolated word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-35(6), pages 751–763, June 1987.

[78] J. Takahashi and S. Sagayama. Vector-field-smoothed Bayesian learning for incremental speaker adaptation. In *Proceedings of the 1995 International Conference on Acoustics, Speech and Signal Processing*, pages 696–699, Detroit, MI, 1995.

[79] M. Tonomura, T. Kosaka, and S. Matsunaga. Speaker adaptation based on transfer vector field smoothing using maximum a posteriori probability estimation. In *Proceedings of the 1995 International Conference on Acoustics, Speech and Signal Processing*, pages 688–691, Detroit, MI, 1995.

[80] B. Winer. *Statistical Principles in Experimental Design.* McGraw-Hill, New York, NY, 1971.

[81] G. Zavaliagkos, R. Schwartz, and J. Makhoul. Batch, incremental and instantaneous adaptation techniques for speech recognition. In *Proceedings of the 1995 International Conference on Acoustics, Speech and Signal Processing*, pages 676–679, Detroit, MI, 1995.

[82] G. Zavaliagkos, R. Schwartz, and John McDonough. Maximum a posteriori adaptation for large scale HMM recognizers. In *Proceedings of the 1996 International Conference on Acoustics, Speech and Signal Processing*, pages 725–728, Atlanta, GA, 1996.

[83] P. Zhan, M. Westphal, M. Finke, and A. Waibel. Speaker normalization and speaker adaptation - a combination for conversational speech recognition. In *Proceedings of the 5th European Conference on Speech Communication and Technology*, pages 2087–2090, Rhodes, Greece, 1997.

[84] V. Zue. Conversational interfaces: Advances and challenges. In *Proceedings of the 5th European Conference on Speech Communication and Technology*, pages KN9–KN18, Rhodes, Greece, 1997.

[85] V. Zue, J. Glass, D. Goodine, H. Leung, M. Phillips, J. Polifroni, and S. Seneff. Recent progress on the SUMMIT system. In *Proceedings of the DARPA Speech and Natural Language Workshop*, pages 380–384, Hidden Valley, PA, June 1990.

[86] V. Zue, J. Glass, D. Goodine, M. Phillips, and S. Seneff. The SUMMIT speech recognition system: Phonological modeling and lexical access. In *Proceedings of the 1990 International Conference on Acoustics, Speech and Signal Processing*, pages 49–52, Alburquerque, NM, 1990.

[87] V. Zue, J. Glass, M. Phillips, and S. Seneff. Acoustic segmentation and phonetic classification in the SUMMIT system. In *Proceedings of the 1989 International Conference on Acoustics, Speech and Signal Processing*, pages 389–392, Glasgow, Scotland, 1989.

[88] V. Zue, J. Glass, M. Phillips, and S. Seneff. The MIT SUMMIT speech recognition system: A progress report. In *Proceedings of the DARPA Speech and Natural Language Workshop*, pages 179–189, Philadelphia, PA, February 1989.

[89] V. Zue, S. Seneff, J. Glass, L. Hetherington, E. Hurley, H. Meng, C. Pao, J. Polifroni, R. Schloming, and P. Schmid. From interface to content: translingual access and delivery of on-line information. In *Proceedings of the 5th European Conference on Speech Communication and Technology*, pages 2227–2230, Rhodes, Greece, 1997.