REMAP: Recursive Estimation and Maximization of A Posteriori Probabilities in Transition-based Speech Recognition

by

Yochai Konig

B.S. (Technion, Israel Institute of Technology) 1990

A dissertation submitted in partial satisfaction of the
requirements for the degree of
Doctor of Philosophy

in

Computer Science

in the

GRADUATE DIVISION
of the
UNIVERSITY of CALIFORNIA at BERKELEY

Committee in charge:

Professor Nelson Morgan, Chair
Professor Jerome Feldman
Professor Charles Stone

1996

REMAP: Recursive Estimation and Maximization of A Posteriori Probabilities in Transition-based Speech Recognition

Abstract

REMAP: Recursive Estimation and Maximization of A Posteriori Probabilities in Transition-based Speech Recognition

by

Yochai Konig
Doctor of Philosophy in Computer Science

University of California at Berkeley

Professor Nelson Morgan, Chair

In this thesis we present a framework for training and modeling continuous speech recognition systems based on the theoretically optimal Maximum a Posteriori (MAP) criterion. In contrast, most state-of-the-art systems are trained according to the Maximum Likelihood (ML) criterion. Specifically, we introduce a discriminant training algorithm (REMAP) for statistical sequence classification which, for any given sentence, monotonically increases the posterior probability of the correct sentence while reducing the probabilities of all rival models.

Based on the studies described here, which show that explicitly modeling transitions between speech units can improve recognition performance, REMAP is developed in the context of a transition-based model (although it is also applicable to non-transition-based models). Furthermore, the model uses local transition probabilities (i.e., the posterior probability of the current state given the current acoustic vector and the previous state) to estimate global posteriors of sentences. Thus, it is a true recognition model, i.e., it directly maps from acoustic sequences to sentences, unlike Hidden Markov Models (HMMs) which model the inverse relation (the likelihood of producing an acoustic sequence given an assumed state sequence).

Experimental results support the proposed framework. In comparison to a baseline system, the results show an increase in the estimates of posterior probabilities of the correct sentences after training, and a significant decrease in error rate.

To my parents, Pnina and Aron König

Contents

List of Figures

Acknowledgements

First and foremost I would like to thank my advisor, Nelson Morgan for sharing with me his vision and enthusiasm for research and science. Morgan introduced me to the field of speech recognition, and has guided and collaborated with me along the long way that has culminated in this thesis. He set very high standards for everything starting with scientific work and ending with funny stories. I feel fortunate to be part of a research group led by Morgan, as he inspired spirit of cooperation, bonding, and fun, while doing ground-breaking research.

I want to thank my "second" advisor Hervé Bourlard for sharing his wisdom, knowledge, and Belgian sense of humor with me. Hervé is a role model for me in terms of his analytic and systematic approach to scientific research. My images of research, are discussing and arguing over equations written in green and blue on a whiteboard with Hervé and Morgan. Most of the research described in this thesis is a result of collaborating with Morgan and Hervé.

I would like to thank Hynek Hermansky for sharing his deep understanding of speech science with me, for stimulating discussions, and for his friendship. My thanks to Steve Greenberg for his patience, sound advice, and for making available to me his vast knowledge and wisdom. I am grateful to Jerry Feldman, for his insight in recommending that I work on speech recognition with Morgan, for serving as a member of my qualification and thesis committees, and for providing such a wonderful research environment at ICSI. My gratitude goes to John Wawrzynek and John Ohala for serving as members of my qualification committee, and to Charles Stone for being a member of my thesis committee.

I wish to thank my past and present peers at the realization group, here at ICSI. I feel really lucky to know, work, collaborate, and socialize with such a great group of people. In particular I would like to thank Chuck Wooters, Su-Lin Wu, Krstè Asanovic, Jeff Bilmes, Eric Fosler, Dan Jurafsky, David Johnson, Nikki Mirghafori, Brian Kingsbury, Mike Shire, Warner Warren, Andreas Stolcke, Gary Tajchman, Steve Renals, Chris Bregler, and Kristine Ma.

My life in Berkeley would not be the same without my friends. They made Berkeley not just a great school, but also a fun and exciting place to live. To mention a few, I would like to thank Amir Guttman, Kim Zetter, Shlomo Zilberstein, Gil Sudai, Eitan Reuveny, Adriana Helmer, Tina Perry, Chih-Po Wen, Roy Twersky, Michele Gill, Yizhaq Makovsky, and Benjamin Fuchs.

Last but not least, I want to thank my family, my sister Mickey, and my brother Dudy, for their encouragement, support, and visits. This thesis is dedicated to my parents, Pnina and Aron. Their values, education, warmth, support, and love, made it all possible.

Chapter 1

Introduction and Goals

1.1 Thesis Overview

Many pattern recognition problems that are of crucial importance today are inherently sequential in nature. Some examples include recognizing an utterance given a sequence of samples from a speech signal, or deciphering a hand written sentence given a digitized pen trajectory. Theoretically the optimal way to classify an input sequence is to choose the class with the highest posterior probability given this sequence (Duda & Hart 1973). Therefore, at training one wants to maximize the posterior probability of the correct model (sentence) given the evidence (sequence of acoustic vectors). An optimization criterion for parameter estimation that achieves this goal during training is the Maximum A Posteriori (MAP) criterion. Most state-of-the-art systems, however, are trained according to other criteria such as Maximum Likelihood (ML), which achieve this goal only under strong assumptions as discussed in Chapter 3. In this thesis we present a framework for training and modeling continuous speech recognition systems based on the MAP criterion. Specifically, we introduce a discriminant training algorithm for statistical sequence classification that monotonically increases the posterior probability of the correct sentence while reducing the probabilities of all rival models (sentences). Roughly speaking, instead of modeling the distribution of each class observation, the boundaries between classes are modeled. Thus, the correctness of the model is not assumed and during training one can optimize the overall goal of minimizing recognition errors.

REMAP can be used in a new form of hybrid Hidden Markov Models (HMM)/ Artificial Neural Network (ANN) which, in addition to the advantages of standard HMM/ANN hybrids, uses "full" posterior probabilities for training and recognition. Furthermore, in the new HMM/ANN hybrid, the ANN targets and weights are iteratively re-estimated, a process that guarantees an increase of the posterior probability of the correct model, hence reducing the error rate.

Based on the studies described here, which show that explicitly modeling transitions between speech units can improve recognition performance, a training algorithm is developed in the context of a transition-based model (the algorithm is general, and also applicable to other models). Our interest in transition-based models was motivated by perceptual and physiological evidence, e.g., (Lindblom & Studdert-Kennedy 1967; Furui 1986b; Kiang 1984; Ruggero 1994; Smith & Zwislocki 1975), which show that spectral transition information is crucial for human perception. Furthermore, the Discriminant HMM (DHMM) model, uses local transition probabilities (the posterior probability of the current state given the current acoustic vector and the previous state) to estimate global sentence posteriors. It is a true recognition model. It directly maps from acoustic sequences to sentences, unlike HMMs that model the inverse relation (the likelihood of producing an acoustic sequence given an assumed state sequence) (Levinson *et al.* 1983b; Jelinek 1976; Jelinek & Mercer 1980; Baker 1975).

Experimental results support the proposed framework. In comparison to a baseline system, the results show an increase in the estimates of posterior probabilities of the correct sentences after training, and a significant decrease in error rate. Thus, a posterior based approach may be a viable alternative to current paradigms.

In the following sections we expand on the goals and the motivations for this study.

1.2 Thesis Goals

1.2.1 A Recognition Model

In automatic speech recognition the following two steps are usually performed:

1. The first step is to transform the speech signal into a finite sequence of numbers hopefully without loss of relevant information for recognition. This step, usually called *feature extraction*, is constrained on the one hand by the need to reduce the dimensionality of the data and on the other hand by the need to have good information for recognizing the utterance.

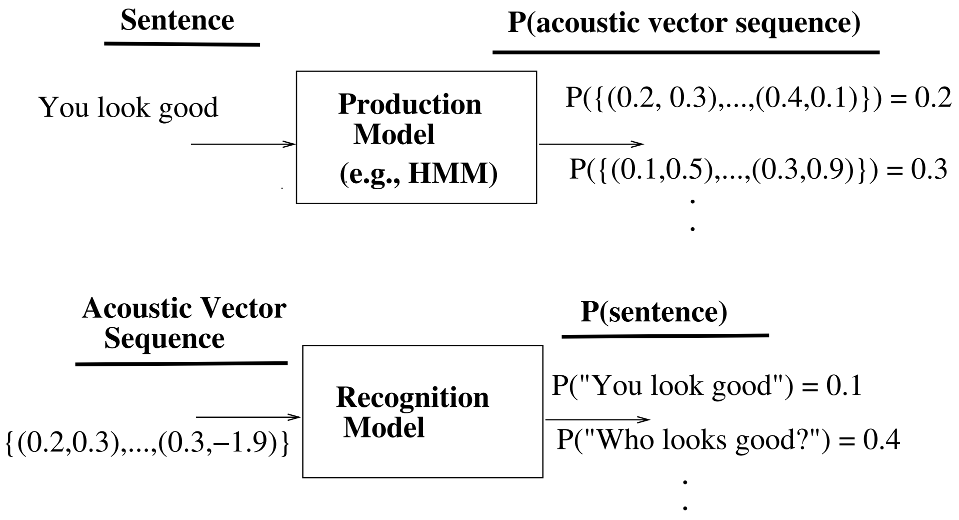

Figure 1.1: Goal - A Recognition Model

2. The second step is to construct a model that provides a mapping from the acoustic vector sequence into sentences, i.e., a *recognition model.*

In current state-of-the-art systems however, inverse modeling is usually used. Specifically, given a sentence, these systems model the likelihood of producing an acoustic sequence, a *production model.* This involves modeling all the possible ways that a given sentence can be spoken, taking into account differences between speakers, different speaking rates, styles, pronunciations, and variability due to environmental and channel noise.

An illustration of the difference between a production model and a recognition model is given in Figure 1.1. The illustration specifies the domain and range for production and recognition models. When recognizing an utterance, the input is a sequence of acoustic vectors and the output should be the recognized sentence, i.e., the domain and range are the same as in the recognition model. Hence, directly optimizing the recognition model parameters (during training) to increase the posterior probabilities of the correct models is more straightforward, compared with the production model with inverse range and domain.

1.2.2 Transition-based Modeling

One basic goal of this thesis is to explicitly model transitions between speech units[1]. Specifically, the goal is to estimate "soft boundaries" between speech units, i.e., to estimate

[1]The choice of the speech unit set determines the type of transitions that we model

the posterior probability of the current state given the current acoustic vector and the previous state, where *states* represent basic speech units such as phones. Roughly speaking, a "soft boundary" represents a probability of having a transition at a particular time point, i.e., values between zero and one. This is in contrast to a "hard boundary" that marks each time point as a transition, or non-transition point, i.e., values of ones and zeros only. This goal is motivated by our experimental results as described in Chapter 4, other related work (Goldenthal 1994; Deng 1992), and perceptual and physiological evidence (Lindblom & Studdert-Kennedy 1967; Furui 1986b; Kiang 1984; Ruggero 1994; Sachs *et al.* 1988; Smith & Zwislocki 1975; Seneff 1988) that show that spectral transition information is crucial for human perception.

The underlying reason for the "soft boundaries" is that human speech production is a continuous process, and modeling it as a sequence of discrete states is at best an approximation. A popular model of human speech production is that an utterance is organized as a succession of vocal-tract states, where each of these states represents a different configuration of the articulators. Further, at the level of articulatory performance, the gestures invoked to actualize these states are relatively slow. They merge spatially and temporally into a continuous process that usually only approximates the intended states and is highly sensitive to co-articulation (Deng & Sun 1994). Thus, there is not a single time point of transition between two states, but rather a "window" of transition which can be modeled with "soft boundaries." Similarly, in human speech perception, the point at which a new phonetic element could be perceived is distributed in time.

1.2.3 A Discriminant Training Algorithm

Statistical pattern recognition theory tells us that the optimal recognition procedure (in terms of minimum classification error) is the one that chooses the sentence with the highest posterior probability given the acoustic vectors and all other available sources of knowledge (Duda & Hart 1973). Consequently, a training algorithm should estimate the parameters of a classifier to increase the posterior probability of the correct sentence (known during the training phase) while reducing the posteriors of all rival models (sentences). An illustration of a discriminant algorithm is given in Figure 1.2.

Roughly speaking, instead of modeling the distribution of the observations of each class, the boundaries between classes are modeled. Theoretically, both of these model-

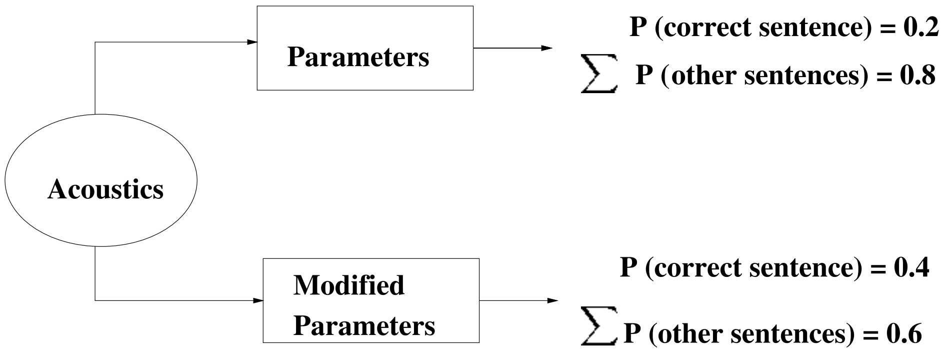

Figure 1.2: Goal - A Discriminant Training Algorithm

ing techniques can be used to achieve an optimal classifier. In practice, however, some assumptions have to be made in order to implement these approaches. In discriminant training model correctness is not assumed, and a training algorithm can directly minimize recognition errors. Generally, fewer free parameters are needed since the model describes boundaries, because there is no need to model the data everywhere (Lubensky *et al.* 1994; Renals *et al.* 1992).

1.3 Thesis Structure

Our starting point is to formulate the problem. This formulation serves both as a goal and an aid in evaluating existing solutions to this problem. It is followed by a description of the most popular models and algorithms used to solve this problem. These include Hidden Markov Models (HMMs) and hybrid systems of Artificial Neural Networks (ANNs) and HMMs, and their various training algorithms. In particular there is an emphasis on discriminant approaches such as Maximum Mutual Information (MMI) and Generalized Probabilistic Descent (GPD). The models are described in Chapter 2, and their training algorithms in Chapter 3.

In Chapter 4 the motivation for the use of transition-based models is given. Our early interest in modeling transitions was based on perceptual and physiological studies. Later, our own experiments reaffirmed the usefulness of it, and other recent related work strengthened this decision. In Chapter 5 we describe a particular model, Discriminant HMM (DHMM) that is a transition-based recognition model, and discusses its related mathematical theory. Experiments with this model motivated the need for a new training algorithm, REMAP, which is described in Chapter 6. The training algorithm, REMAP, is an approach for Recursively Estimating and Maximizing A posteriori Probabilities of transition-based Hidden Markov Models given input sequences. It is a discriminant training algorithm; REMAP maximizes the posterior probability of the correct sentence while reducing the posteriors of all rival models. Experimental results are reported in Chapter 7. These experiments were both on isolated and continuous speech tasks. The results show an increase in the estimates of the posterior probabilities of the correct sentences after training, and a significant decrease in error rates in comparison to a baseline system. The thesis ends with conclusions and discussion of the remaining open problems, and future research directions in Chapter 8. The text proper is followed by an Appendix giving the proofs for theorems used in REMAP.

Chapter 2

Problem Formulation and Existing Models

2.1 Introduction

2.1.1 Bayes Decision Theory

Bayes decision theory is a statistical approach to pattern classification. The approach is aimed at finding a *decision rule* that tells us which action to take for every possible observation. To simplify the discussion, let us assume a two-class classification problem wherein we have to choose between two classes M_1 and M_2. We assume some *a priori probability* $P(M_1)$ that the next observation belongs to M_1, and similarly for M_2. These a priori probabilities reflect our beliefs and prior knowledge about the models before seeing any evidence, e.g., if we needed to classify males and females based on their voices, we would assign equal prior probabilities[1] (Konig *et al.* 1993; Konig & Morgan 1993).

Each observation is a feature vector x, for instance (continuing the gender classification example) the fundamental frequency of the speaker and other speaker-dependent spectral information. The distribution of the feature vectors is class-dependent. Specifically, given a class say M_j, the distribution of the feature vectors is the likelihood function of the class, $p(x|M_j)$. For instance, given that the class is "male", we give higher probability to feature vectors with fundamental frequency around 120 Hz than to feature vectors with

[1] We assume equal number of males and females in the training set.

fundamental frequency around 210 Hz. If the class is "female" it would be the other way around.

Bayes rule specifies how observing the value of x transforms the a priori probability $P(M_j)$ into the *a posteriori* probability $P(M_j|x)$ (for the 2-class example):

$$P(M_j|x) = \frac{P(x|M_j)P(M_j)}{P(x)} \tag{2.1}$$

where

$$P(x) = \sum_{j=1}^{2} P(x|M_j)P(M_j) \tag{2.2}$$

Furthermore, according to Bayes decision theory, in order to minimize the average probability of error we should select the class with the highest a posterior probability $P(M_j|x)$ (Duda & Hart 1973). Hence the general decision rule for I classes is:

$$\text{Choose } M_j \text{ if } P(M_j|x) > P(M_i|x) \quad \forall i \neq j \tag{2.3}$$

Note that this would be equivalent to

$$\text{Choose } M_j \text{ if } P(x|M_j)P(M_j) > P(x|M_i)P(M_i) \quad \forall i \neq j \tag{2.4}$$

as $P(x)$ is fixed during recognition. Nonetheless, we are still left with the problem of choosing a model and estimating its parameters. The implication of Bayes decision rule is that we should optimize at training time the same measure that we use at recognition time, i.e., the *a posteriori* probability of the model given the observation.

2.1.2 Problem Formulation

Parameter Estimation

In statistical pattern classification as described above, it is known that a system leading to the minimum probability of error is the one that is trained to maximize the a posteriori probability of the correct class conditioned on the evidence (Duda & Hart 1973) and uses that same criterion during recognition. In real-life problems, however, we rarely have accurate knowledge about the structure of the probability functions. Given the typical high-dimensional input space and the limited training set, we can not evaluate the probability function directly from the training samples. Therefore, some parameterization of the probability function is needed, e.g., if we assume a normal density we only need to estimate its mean and covariance matrix.

In speech recognition two sources of knowledge are commonly used: "acoustic" and "language" knowledge. The acoustic knowledge provides the relation between the sound wave (or spectral patterns in the sound wave) and the linguistic identity of the utterance. The language information tells us about the phonemes and words, i.e., the pronunciation of each word in our vocabulary. In addition, the language model estimates the probability of a word given the hypothesized previous words.

Hence, these two sources of knowledge are used to parameterize the probability function that maps the acoustics of the utterance to the space of possible sentences. Specifically, if the speech signal is sampled at some fixed interval, and the acoustic vectors are concatenated, we can represent the input sequence to be classified with $X = \{x_1, \ldots, x_n, \ldots, x_N\}$. Additionally, we denote by M_i $(i = 1, \ldots, I)$ one from the set of all possible sentences. Then we parameterize $P(M_i|X)$ (the a posteriori probability of a sentence given a sequence of acoustic vectors) with L the parameter set which represents the language knowledge, (both a lexicon and a probabilistic grammar), and Θ, the parameter set that represents the acoustic information. Thus, we want to estimate in training and use in recognition the following probability function $P(M|X, L, \Theta)$.

Problem Formulation

Bayes Decision theory shows (as described above) that X will be optimally assigned to the sentence associated with model M_j if

$$M_j = \underset{M_i}{\text{argmax}}\ P(M_i|X, L, \Theta), \quad i = 1, \ldots, I \tag{2.5}$$

During training we should optimize the measure that we use in recognition. Thus, the ideal training algorithm should determine the set of parameters $(\hat{\Theta}, \hat{L})$ that will maximize $P(M_{w_j}|X_j, L, \Theta)$ for all training utterances[2] X_j $(j = 1, \ldots, J)$, associated with M_{w_j}[3], i.e.,

$$(\hat{\Theta}, \hat{L}) = \underset{(\Theta, L)}{\text{argmax}} \prod_{j=1}^{J} P(M_{w_j}|X_j, L, \Theta) \tag{2.6}$$

[2] Note that J is the number of training sentences in the training set, while N is the number of acoustic vectors in a particular acoustic vector sequence.

[3] M_{w_j} represents the correct model associated with the specific input sequence X_j that is known at training time. Strictly speaking, w_j is the index of the correct model for the acoustic vector sequence j.

with the following constraint:

$$\sum_{i=1}^{I} P(M_i|X_j, L, \Theta) \;=\; 1, \quad \forall j; \tag{2.7}$$

for every X, and where the sum over i represents the sum over all possible models. Note that this constraint makes the Maximum a Posteriori (MAP) criterion (2.6) discriminant. That is, when increasing the posterior probability of the correct model, the total probability mass assigned to all other models will automatically be reduced.

Preview

The problem formulation above does not specify the model, i.e., how to estimate the posterior probability of a sentence given the parameters, acoustics, and the language model. In the following sections we describe several solutions to this problem. The discussion is divided into existing models and training algorithms. Specifically, we discuss models such as the Hidden Markov Model (HMM), hybrid systems of Artificial Neural Networks (ANNs) and HMMs, and segment-based models. The next chapter includes a discussion of several training algorithms and approaches for the models mentioned above. In particular there is an emphasis on discriminant approaches, such as Maximum Mutual Information (MMI) and Generalized Probabilistic Descent (GPD). Overall, we show that these solutions do not maximize the MAP criterion. They either maximize other criteria or approximate them.

2.1.3 Definitions and Notation

To facilitate the following discussion notation and basic terms are defined:

- A set of states $\mathcal{Q} = \{q_1, \ldots, q_K\}$, that contains all the states from which phone and word models will be built. Each state class will be associated with a specific probability density function (PDF) or with specific statistical properties (see "conditional transition probabilities" in Section 5.2). For instance, if one wants to model the acoustic production in the beginning, middle, and end of a phone differently, each phone is modeled by three different states.

- $X = \{x_1, \ldots, x_N\}$ is a sequence of acoustic vectors that is associated with a specific utterance. A sub-sequence of acoustic vectors that is local to the current vector, extending c frames into the past and d frames into the future is expressed by $X_{n-c}^{n+d} = \{x_{n-c}, \ldots, x_n, \ldots, x_{n+d}\}$. Each acoustic vector is calculated from a short interval of speech, usually around 20 ms (shifted every 10 ms).

- M_i is defined for $i \in \mathcal{I} = \{1, 2, \ldots, \mathcal{I}\}$, the set of possible model indices; $\mathcal{I}$ is the number of possible models (in the case of continuous speech, I indicates the number of possible sentences allowed by the grammar, although this is generally infinite). In continuous speech recognition the models represent all possible sentences. M_i is a directed graph with C_i states each of which belongs to Q.

- M_{w_j}, $w_j \in \mathcal{I}$, is the "correct" model associated with a specific training sequence X_j, $j = 1, \ldots, J$. Strictly speaking, w_j is the index of the correct model for the training sequence X_j.

- The parameter set describing all models is defined as $\Theta = \{\lambda_1, \ldots, \lambda_i, \ldots, \lambda_I\}$, in which λ_i represents only the parameters present in M_i. Of course, the different M_i's, for $i = 1, \ldots, \mathcal{I}$, can share some common parameters. In the hybrid systems discussed in this study, all models share the same set of parameters Θ through a common neural network, which will be parameterized in terms of Θ.

- The set of parameters that are only present in M_{w_j} will be denoted Θ_{w_j}, which is a subset of Θ.

- q^n denotes the state at time n.

- q_k^n means that state q_k has occurred at time n. Strictly speaking, q_k^n denotes the assignment of the value k to the random variable q^n.

- Γ is a state sequence of length N. Each Γ is a realization of the stochastic state process, where the values assigned to the state process at each time step are taken from Q. Sometimes we write it explicitly as $\Gamma_j = \{q_{\Gamma_j}^1, \ldots, q_{\Gamma_j}^n, \ldots, q_{\Gamma_j}^N\}$. A state sequence which is legal in a given model M_i is also called a *path* in M_i.

- $P(\cdot)$ represents probabilities, while $p(\cdot)$ denotes probability density functions (PDFs) and likelihoods.

2.2 Hidden Markov Models (HMMs)

2.2.1 Introduction

This section is a short review of the classical HMM approach to speech recognition. For a complete explanation, see (Huang *et al.* 1990; Levinson *et al.* 1983a; Rabiner 1989; Baker 1975; Jelinek 1976; Jelinek & Mercer 1980). Currently, this approach is very popular and a number of large-vocabulary, speaker-independent, continuous speech recognition state-of-the-art systems have been based on this approach. An HMM used for speech is a production model, and for each sentence it models the inherent statistical variations in speaking rate, pronunciation, and the differences between speakers (in speaker-independent models). The main idea is that we can approximate a continuous process as a sequence of short steady states. Roughly speaking we can model human speaking as moving between different states such that each state has fixed properties in terms of its repertoire of sounds.

In order to implement practical systems based on HMMs, a number of simplifying assumptions are typically made about the signal. For instance, although speech is a non-stationary process, HMMs model the sequence of feature vectors as a piecewise stationary process. That is, an utterance $X = \{x_1, \ldots, x_n, \ldots, x_N\}$ is modeled as if it were produced by a succession L of discrete stationary states $q_\ell \in \mathcal{Q}$, with instantaneous transitions between these states. In this case, an HMM is defined (and represented) as a stochastic finite state automaton with a particular topology (usually strictly left-to-right, since speech is sequential). The approach defines two concurrent stochastic processes: the sequence of HMM states (modeling the temporal structure of speech), and a set of state output processes (modeling the [locally] stationary character of the speech signal). The HMM is called a "hidden" Markov model because there is an underlying stochastic process, the sequence of states, that is not observable but that affects the observed sequence of events. It is called "Markov" because the statistics of the current state are modeled as being dependent only on the current and the previous state (for the first-order Markov case).

Ideally, there should be an HMM for every possible utterance. However, this is clearly infeasible for all but extremely constrained tasks. Generally a hierarchical scheme must be adopted to reduce the number of possible models. First, a sentence is modeled as a sequence of words. To further reduce the number of parameters (and the required amount of training material) and to avoid the need of a new training each time a new word is added to the lexicon, sub-word units are usually preferred to word models. Although there are good linguistic arguments for choosing units such as syllables or demi-syllables (Fujimura 1975; Segui *et al.* 1980; Levelt & Wheeldon 1994), the unit most commonly used is the phone (or context-dependent versions such as the triphone). This is the unit that we have generally used in our work, resulting in a selection of between 50 and 70 sub-word models. In this case, word models consist of concatenations of phone models (constrained by pronunciations from a lexicon), and sentence models consist of concatenations of word models (constrained by a grammar).

In the following section we describe the assumptions of HMM modeling.

2.2.2 Assumptions

Traditionally, in HMM modeling, the probability estimation process is divided into two parts: (1) *language modeling*, which does not depend on the acoustic data, and (2) *acoustic modeling*. The goal of the language model is to estimate prior probabilities of sentence models $P(M_i|L)$. The acoustic modeling role is to estimate the model-dependent probability densities $p(X|M_i, \Theta)$.

Additionally, several additional assumptions are usually required to make the estimation of $p(X|M_i, \Theta_i)$ tractable:

- **Output-independence Assumption:** Acoustic vectors are not correlated (observational independence). The current acoustic vector x_n is assumed to be conditionally independent of the previous acoustic vectors (e.g., X_1^{n-1}). Furthermore, the assumption is that observations within the same speech segment (generated by the same state) are independent and identically distributed (i.i.d.), an unrealistic assumption given the non-stationary nature of speech.

  To limit the impact of these assumptions, acoustic vectors at time n are usually complemented by their first and second time derivatives (Furui 1986b; Poritz & Richter 1986) computed over a span of a few frames, allowing very limited acoustical context modeling. Another solution to limit these assumptions is to consider a few adjacent frames (typically 3-5 frames in total) on which linear discriminant analysis is performed to reduce the dimension of the acoustic features (Haeb-Umbach & Ney 1992).

- **Markov Assumption:** Markov models are generally first-order Markov chains. Explicitly, the probability that the Markov chain is in state q_ℓ at time n depends only

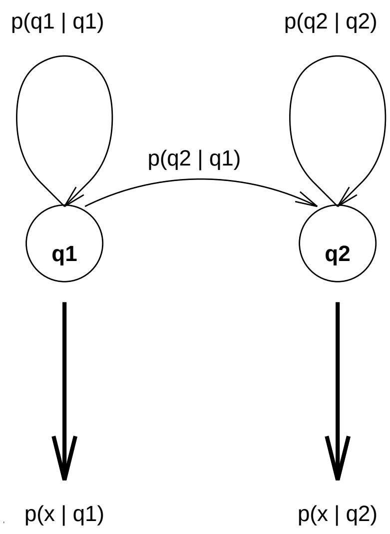

Figure 2.1: HMM - An Example

on the state of the Markov chain at time $n-1$, and is conditionally independent of the past (both the past acoustic vector sequence and the states before the previous one).

2.2.3 Definition of the Hidden Markov Model

In this section we formally describe HMMs. An HMM can be defined by:

- $A = \{a_{ij} | a_{ij} = P(q^{n+1} = j | q^n = i)\}$, a state transition probability distribution, where a_{ij} denotes the transition probability from state i to state j. Usually the assumption is that this probability distribution is the same for all time steps.

- $B = \{b_j(x_i) | b_j(x_i) = P(x_i | q = j)\}$, for each state, there is a corresponding output probability (a discrete probability distribution in the discrete case and a continuous probability density function in the continuous case). It refers to the probability of generating some discrete symbol x_i in state q_j. Usually these probabilities are called emission probabilities.

- $\Pi = \{\pi_i | \pi_i = P(q^1 = i)\}$, denotes the initial state distribution.

These definitions in addition to the relevant definitions in Section 2.1.3, represent an HMM, $\lambda = (A, B, \Pi)$. A very simplistic HMM is pictured in Figure 2.1.

2.3 Hybrid Systems

2.3.1 Multilayer Perceptrons (MLPs)

In this thesis, the discussion of neural networks for speech will be limited to the Multi-Layer Perceptron (MLP), a form of ANN that is commonly used for speech recognition. However, the analyses that follow are generally extensible to other kinds of ANN, e.g., a recurrent neural network (Robinson 1994), or a Time-Delay Neural Network (TDNN) (Waibel *et al.* 1989).

MLPs have a layered feed-forward architecture with an input layer, zero or more hidden layers, and an output layer. Each layer computes a set of linear discriminant functions (Duda & Hart 1973) (via a weight matrix) followed by a nonlinear function, which is often a sigmoid function

$$f(x) = \frac{1}{1 + \exp(-x)} \tag{2.8}$$

As discussed in (Bourlard & Morgan 1994), this nonlinear function performs a different role for the hidden and the output units. On the hidden units, it serves to generate high order moments of the input; this can be done effectively by many nonlinear functions, not only by sigmoids. On the output units, the nonlinearity can be viewed as a differentiable approximation to the decision threshold of a threshold logic unit or perceptron (Rumelhart *et al.* 1986), essentially to count errors. For this purpose, the output nonlinearity should be a sigmoid or sigmoid-like function. Alternatively, a function called *softmax* can be used. For an output layer of K units, this function is defined as

$$f(x_i) \;=\; \frac{\exp(x_i)}{\sum_{n=1}^{K} \exp(x_n)} \tag{2.9}$$

It can be proved that MLPs with enough hidden units can (in principle) provide arbitrary mappings $g(x)$ between input and output. The MLP parameter set Θ (the elements of the weight matrices) are trained to associate a "desired" output vector with an input vector. This is generally achieved via the Error Back-Propagation (EBP) algorithm (Rumelhart *et al.* 1986) that uses a steepest descent procedure to iteratively minimize a cost function in their parameter space. Since in our approach the HMMs will be described by the parameters of the neural network, we also denote the MLP parameter space by Θ.

Popular cost functions are, among others, the Mean Square Error (MSE) criterion:

$$E = \sum_{n=1}^{N} \| \, g(x_n, \Theta) - d(x_n) \, \|^2 \tag{2.10}$$

or the relative entropy criterion[4]:

$$E_e = \sum_{n=1}^{N} \sum_{k=1}^{K} d_k(x_n) \ln \frac{d_k(x_n)}{g_k(x_n, \Theta)} \tag{2.11}$$

where $g(x_n, \Theta) = (g_1(x_n, \Theta), \ldots, g_k(x_n, \Theta), \ldots, g_K(x_n, \Theta))^t$ represents the actual MLP output vector (depending on the current input vector x_n and the MLP parameters Θ), $d(x_n) = (d_1(x_n), \ldots, d_k(x_n), \ldots, d_K(x_n))^t$ represents the desired output vector (as given by the labeled training data), K the total number of classes, and N the total number of training patterns.

MLPs, as well as other neurally-inspired architectures, have been used for many speech-related tasks. For instance, for some problems the entire temporal acoustic sequence is processed as a spatial pattern by the MLP. For isolated word recognition, for instance, each word can be associated with an output of the network. However, this approach has not been useful for continuous speech recognition and will not be discussed further here (Lippmann 1989).

2.3.2 Motivations

ANNs have several advantages that make them particularly attractive for Automatic Speech Recognition (ASR), e.g.:

- They can provide discriminant learning between speech units or HMM states that are represented by ANN output classes. That is, when trained for classification (using common cost functions such as MSE or relative entropy), the parameters of the ANN output classes are trained to minimize the error rate while maximizing the discrimination between the correct output class and the rival ones. In other words, ANNs not only can train and optimize the parameters of each class on the data belonging to that class, but also can attempt to reject data belonging to the other (rival) classes. This is in contrast to the likelihood criterion that does not lead to minimization of the error rate.

[4]In a number of references, including (Bourlard & Morgan 1994), this criterion is defined differently. In particular, the desired outputs are sometimes assumed to be independent, binary random variables and as a result this criterion gets a different form (which is sometimes called the cross entropy (Richard & Lippmann 1991)). However, viewing the network outputs as a posterior distribution over the values of one random variable (class conditioned on acoustic data), a discrete version of the classical definition of relative entropy may be used, as given here.

- Because ANNs can incorporate multiple constraints and find optimal combinations of constraints for classification, feature vectors do not need to be assumed independent. More generally, there is no need for strong assumptions about the statistical distributions of the input features (as is usually required in standard HMMs).
- They have a very flexible architecture that easily accommodates contextual inputs and feedback, and both binary and continuous inputs.
- ANNs are typically highly parallel and regular structures, which makes them especially amenable to high-performance architectures and hardware implementations.

A general formulation of statistical ASR can be summarized simply by a question: how can an input sequence (e.g., a sequence of spectral vectors) be explained in terms of an output sequence (e.g., a sequence of phones or words) when the two sequences are not synchronous (since there are multiple acoustic vectors associated with each pronounced word or phone)? It is true that neural networks are able to learn complex mappings between two vector variables. However, a purely connectionist formalism is not very well suited to solve the sequence-mapping problem. Most early applications of ANNs to speech recognition have depended on severe simplifying assumptions, e.g., small vocabulary, isolated words, known word or phone boundaries (Lippmann 1989). We shall see here that further structure (beyond a simple MLP) is required to perform well on continuous speech recognition, and that HMMs provide one solution to this problem. First, the relation between ANNs and HMMs must be explored.

2.3.3 MLPs as Statistical Estimators

MLPs can be used to classify speech classes such as words. However, MLPs by themselves classifying complete temporal sequences have not been successful for continuous speech recognition (Lippmann 1989). In fact, used as spatial pattern classifiers, they are not likely to work well for continuous speech since the number of possible word sequences in an utterance is generally infinite. On the other hand, HMMs provide a reasonable structure for representing sequences of speech sounds or words. One good application for MLPs is to provide the local distance measure for HMMs, while alleviating typical drawbacks such as lack of discrimination and assumptions of no correlation between acoustic vectors.

For statistical recognition systems, the role of the local estimator is to approximate

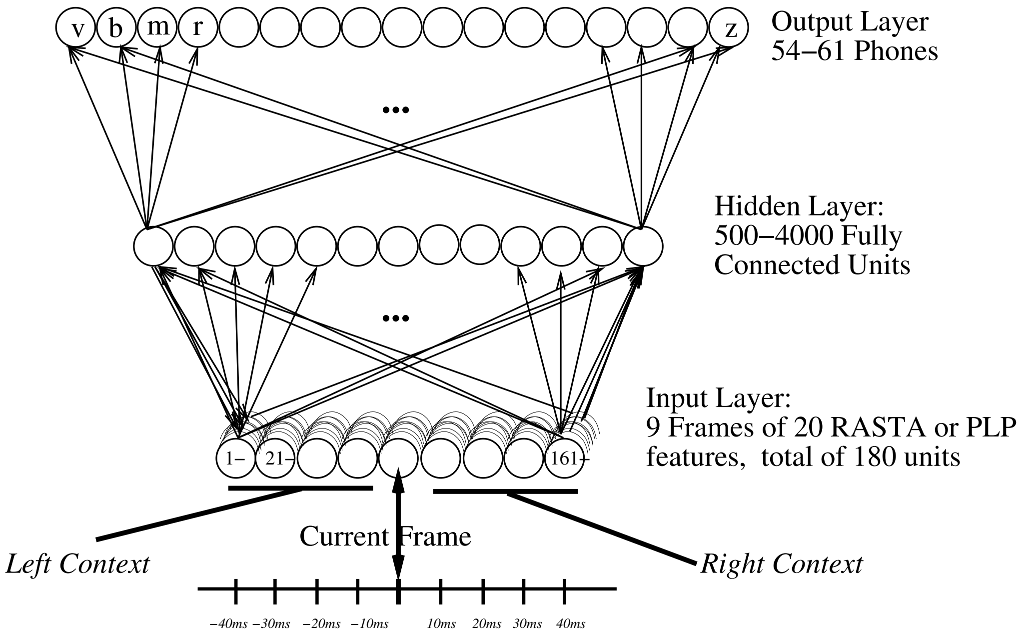

Figure 2.2: MLP - An Example

probabilities or probability density functions. In particular, given the basic HMM equations, we would like to estimate something like $p(x_n|q_k)$, which is the value of the probability density function (pdf) of the observed data vector given the hypothesized HMM state. The MLP can be trained to produce the *posterior* probability $P(q_k|x_n)$ of the HMM state given the acoustic data. This can be converted to emission probability density function values using Bayes' rule.

Several authors (Bourlard & Wellekens 1989a; Bourlard & Morgan 1994; Gish 1990; Richard & Lippmann 1991) have shown that ANNs can be trained to estimate *a posteriori* probabilities of output classes conditioned on the input pattern. Recently, this property has been successfully used in HMM systems, referred to as *hybrid HMM/ANN* systems, in which ANNs are trained to estimate local probabilities $P(q_k|x_n)$ of HMM states given the acoustic data (see, e.g., (Lubensky *et al.* 1994)).

Since MLPs require supervised training, all these systems have been used so far in the framework of Viterbi training, which provides the segmentation of the training sentences in terms of q_k's and, hence, MLP training targets. A typical MLP is pictured in Figure 2.2. The principle of these systems is briefly recalled here.

Let c_k, with $k = 1, \ldots, K$, be an output class of an MLP. Since we will use the

MLP for probability estimation associated with each HMM state q_k ($k = 1, \ldots, K$), there is a one-to-one equivalence between the q_k's and the c_k's that is associated with the discrete stationary states of Q. Also, we associate the parameter set Θ as defined for HMMs with the MLP parameter set.

The output activation of the k-th MLP output class for a given set of parameters Θ and an input x_n is denoted $g_k(x_n, \Theta)$. Since MLP training is supervised we will also assume the training set consists of a sequence of N acoustic vectors $\{x_1, x_2, \ldots, x_n, \ldots, x_N\}$ labeled in terms of q_k's. At time n, the input pattern of the MLP is acoustic vector x_n, and is associated with a state q_k^n.

For the popular MLP cost functions, it can be proved (as noted above) that the optimal MLP output values are estimates of the probability distribution over classes conditioned on the input $\hat{P}(q_k|x_n)$:

$$g_k(x_n, \Theta^{opt}) = \hat{P}(q_k|x_n) \tag{2.12}$$

if:

- the MLP contains "enough" parameters to be able to reasonably approximate the input/output mapping function,
- the network is not over-trained (which can be assured by stopping the training before the decline of generalization performance on an independent cross-validation set),
- the training does not get stuck at a local minimum.

In (2.12), Θ^{opt} represents the parameter set minimizing (2.10) or (2.11).

It has been experimentally observed that, for systems trained on a large speech corpus, the outputs of a properly trained MLP do in fact approximate posterior probabilities (Bourlard & Morgan 1994).

This conclusion can easily be extended to other cases. For example, if we provide the MLP input not only with the acoustic vector x_n at time n, but also with some acoustic context $X_{n-c}^{n+d} = \{x_{n-c}, \ldots, x_n, \ldots, x_{n+d}\}$, the output values of the MLP will estimate

$$g_\ell(x_n, \Theta^{opt}) = \hat{P}(q_\ell^n|X_{n-c}^{n+d}), \quad \forall \ell = 1, \ldots, K \tag{2.13}$$

This windowing over time has been used in the standard hybrid HMM/ANN system (briefly summarized later in this section) to account for correlation between acoustic vectors. If the previous class is also provided to the input layer (leading to a quasi-recurrent network), the MLP output values will be estimates of

$$g_\ell(x_n, \Theta^{opt}) = \hat{P}(q_\ell^n | X_{n-c}^{n+d}, q_k^{n-1}), \quad \forall k, \ell = 1, \ldots, K \tag{2.14}$$

It will be shown in Chapter 4 that this is a form of the local probability the hybrid HMM/MLP theory tells us to use. This will be referred to as the "conditional transition probability" and will be the major theme throughout this thesis.

In addition, this conclusion remains valid for other kinds of networks, given similar training conditions. For example, recurrent networks (Robinson 1994) and radial basis function networks (Renals *et al.* 1991) can also be used to estimate posterior probabilities.

There is another important generalization of this property that will be essential later in this thesis. If the ANNs are trained with an estimate of the posterior probabilities of the output states (as opposed to the "1-from-K" binary output targets used for a classification mode training), then (2.12) remains valid. In other words, if the targets come from some independent "expert", the net will learn to produce posterior probabilities as well.[5] Although this property is mentioned in (Bourlard & Wellekens 1989a; Bourlard & Morgan 1994; Richard & Lippmann 1991), it has not previously used in hybrid HMM/MLP systems because of the lack of a full algorithm for the convergence to better probabilities. Such an algorithm has now been developed, and will be presented in this thesis.

Estimating HMM Likelihoods with MLP

Since the network outputs approximate Bayesian probabilities, $g_k(x_n, \Theta)$ is an estimate of

$$P(q_k|x_n) = \frac{p(x_n|q_k)P(q_k)}{p(x_n)} \tag{2.15}$$

which implicitly contains the a priori class probability $P(q_k)$. It is thus possible to vary the class priors during classification without retraining, since these probabilities occur only as multiplicative terms in producing the network outputs. As a result, class probabilities can be adjusted during use of a classifier to compensate for training data with class probabilities that are not representative of actual use or test conditions (Richard & Lippmann 1991).

[5] Actually, it is easy to prove that, for the popular MLP cost functions, $g(x_n)$ will be an estimate of $E\{d(x_n)|x_n\}$, where E stands for the expected value.

Scaled likelihoods $p(x_n|q_k)$ for use as emission probabilities in standard HMMs can be obtained by dividing the network outputs $g_k(x_n)$ by the relative frequency of class q_k in the training set, which gives us an estimate of:

$$\frac{P(q_k|x_n)}{P(q_k)} = \frac{p(x_n|q_k)}{p(x_n)} \tag{2.16}$$

During recognition, the scaling factor $p(x_n)$ is a constant for all classes and will not change the classification. It could be argued that, when dividing by the priors, we are using a scaled likelihood, which is no longer a discriminant criterion. However, this need not be true, since the discriminant training has affected the parametric optimization for the system that is used during recognition. Thus, this permits the use of the standard HMM formalism, while taking advantage of ANN characteristics.

2.4 Non-stationary Modeling

This section includes a brief a number of models that have been proposed to remedy some of the shortcomings of HMMs. A quick solution might be to represent each unit of speech by enough different states to approximate its non-stationary nature in a stepwise fashion. For instance, a vowel could be represented by ten different states. This solution has two major limitations:

- There are too many free and independent model parameters. This necessitates more training data, and also might be more prone to capturing irrelevant sources of variance in the data than a simpler model.
- Such a model does not capture the correlation and dependence between the different states. For states with a short duration, this would be even more pronounced, since the change between two states in a sequence would correspond to only a small movement of articulators for a given speaker.

Several extensions to the basic HMM have been proposed in order to overcome some of these deficiencies. For example, autoregressive HMMs condition the emission probability of a given state on previous observations (Juang & Rabiner 1985). However, none of these extensions have explicitly modeled the emission in a given phone as a non-stationary process. In general this is too difficult to handle with a practical number of parameters.

A number of HMM alternatives model the sequence of frames emitted in a given sub-word unit as correlated and dependent on each other (Digalakis 1992; Ostendorf & Roukos 1989; Ghitza & Sondhi 1993; Deng 1992) (see also Section 4.3.2). The models differ in their assumptions about the nature of the correlation between the frames in the sequence. For instance, some assume that only consecutive frames are correlated, while others assume that all the frames in the sequence are dependent on each other. In general these models do not require the HMM assumption of independent and identically distributed observations. In the following section we survey segment-based approaches that are in this family.

2.4.1 Segment-Based Approaches

Introduction

In segment-based models the basic unit is a sequence of acoustic vectors emitted in a given speech unit (a "segment"), as opposed to a single acoustic vector as used for HMMs. The production of the acoustic vectors in a segment may be described as a three step procedure (Digalakis 1992):

1. Select the length of the segment according to $P(L|s_k)$, where L is the random variable that denotes the length of the segment, and s_k is a particular speech unit.

2. Generate a fixed length segment M according to the distribution $P(y_1, y_2, \ldots, y_M|s_k)$. The distribution models the trajectory of the sound in the feature vector space. M is chosen to be greater than all the possible values of L. $Y = y_1, y_2, \ldots, y_M$ is called the *hidden* sequence of acoustic vectors.

3. Down-sample Y using the time-warping transformation T_L and output the observed sequence of acoustic vectors $X = \{x_1, x_2, \ldots, x_L\}$. This transformation can be either linear or non-linear depending on the specific segmental model.

Segmental Models

These models differ in the form of the distribution $(Y|s_k)$ and in the time-warping transformation T_L. Ostendorf and Roukos (Ostendorf & Roukos 1989) have used (among a number of methods) linear time sampling in their study, i.e., sampling Y in equal intervals along the time axis as their time warping procedure. Their specific implementation had ten 14-dimensional vectors of cepstral coefficients. They used a multivariate Gaussian to represent the entire segment, which can require a 140 by 140 full covariance matrix for each phone (assuming that feature dependence is accounted for).

Ghitza and Sondhi developed a model (Ghitza & Sondhi 1993) that can also be viewed as a stochastic segment model with the following distinctions:

- Their warping procedure is a dynamic time warping technique, instead of the linear time warping method used by Ostendorf and Roukos.
- They used diphones as their sub-word units, as opposed to the phones in Ostendorf and Roukos' stochastic segment model (Ostendorf & Roukos 1989).
- They maintained the HMM framework and assumed a semi-hidden Markov chain, i.e., each state has an explicit duration distribution.

These stochastic segment models are not inherently subject to the constraints of the i.i.d. assumptions discussed earlier. However, there are some practical difficulties:

1. There are many free parameters that must be estimated reliably from the data, e.g., a large covariance matrix. As a result, independence assumptions are often made, leading to less powerful models.
2. These methods explicitly assume a particular parametric form for the observation distributions, e.g., multivariate Gaussian. This assumption is already faulty for standard HMMs, but may be even a worse approximation once observation interdependencies are taken into account. (Nonetheless, it is a sensible place to start.)
3. All the models assume a given segmentation, e.g., the knowledge of the boundaries between the basic speech units, which is difficult to obtain. One solution is to do an exhaustive search of all reasonable segmentations.
4. Warping the data to a fixed length segment may delete or obscure relevant information.

A Stochastic Dynamic System Approach

This model assumes a discrete-time, linear, stochastic dynamic system, with a state process as the source for the observation process. To model an underlying dynamic system, some assumptions are required. For example, Digalakis has proposed two possible model constraints:

1. *Trajectory invariance*: It is assumed that the unobserved trajectory of state vectors in the state space is the same for each speech segment length and is the source for all possible realizations of the speech segment. Given that the state vector at each time step is a vector random variable, this translates into a fixed sequence of state transition matrices. The observed speech segment is then a down-sampled version of the trajectory of the feature vectors created by the system. For a long realization (a long observation sequence) the underlying trajectory is sampled at shorter intervals than a short observation sequence. Consequently, long observation sequences have higher correlation between successive frames than short observation sequences.

2. *Correlation invariance*: It is assumed that the underlying trajectory in state space varies with the realization length, and the sequence of state-transition matrices for a particular realization depends on the realization length. It is assumed that the correlation between two observations depends only on the relative location of the observations in the segment. The correlation is invariant under the time-warping transformation. Roughly speaking the trajectory length is chosen according to the length of the realization. Thus, the state change rate is slower than under the trajectory invariance assumption, making it somehow a more realistic approach.

In his study, Digalakis assumes that the observed segment of speech is the output of a piecewise time-invariant linear dynamical system. He uses up to five invariant regions for each model. The models based on the *correlation invariance* assumption outperformed the models based on the *trajectory invariance* assumption for the task of phone classification. For more details see (Digalakis 1992; Digalakis *et al.* 1993). The stochastic dynamic system approach appears to have more modeling power than an HMM, and can potentially capture the dynamics of acoustic vectors within a segment of speech. However, there are still open issues about the structure of the dynamic system, such as the arguable assumptions of linearity and several types of invariance.

Chapter 3

Training Algorithms and Optimization Criteria

3.1 Likelihood Estimation and Training

3.1.1 Introduction

Theoretically the optimal way to classify an input sequence is to choose the class with the highest posterior probability given this sequence (Duda & Hart 1973). Therefore, at training we want to maximize the posterior probability of the correct model (sentence) given the evidence (sequence of acoustic vectors). An optimization criterion for parameter estimation that achieves this goal during training, is the Maximum a Posteriori (MAP) criterion. Most speech recognition systems however, are trained according to a maximum likelihood criterion that maximizes, in the parameter space, the likelihood of the data given some model. In this section, besides describing the ML criterion, we discuss under what assumptions it results in the best possible parameter set.

In HMMs, this likelihood can be represented as $P(X|M,\Theta)$. Training HMMs according to the ML criterion is aimed at finding the best set of parameters $\hat{\Theta}$ such that:

$$\hat{\Theta} = \underset{\Theta}{\operatorname{argmax}} \prod_{j=1}^{J} P(X_j|M_{w_j},\Theta) \tag{3.1}$$

Note that this criterion is different from the criterion that is specified in (2.6). The main difference is that ML is not discriminant; maximization of (3.1), the likelihood of the correct sentence producing the utterance, does not necessarily decrease the likelihood of all other models, unlike the MAP criterion that "ties" all models by (2.7).

3.1.2 The Relation between ML and MAP

Classically, the relation between likelihood estimation and MAP estimation is obtained by applying Bayes' rule:

$$P(M|X,L,\Theta) = \frac{P(X|M,L,\Theta)P(M|L,\Theta)}{P(X|L,\Theta)} \tag{3.2}$$

For practical reasons, it is assumed that:

1. The parameters Θ of the acoustic model are independent of the parameters L of the language model, yielding

$$\begin{aligned} P(X|M,L,\Theta) &\approx P(X|M,\Theta) \\ P(M|L,\Theta) &\approx P(M|L) \end{aligned} \tag{3.3}$$

2. Despite the fact that Θ and L vary during training, $P(X|L,\Theta)$ is assumed to be constant.

Incorporating these assumptions for a particular model M_{w_j} associated with the training utterance x_j we get,

$$P(M_{w_j}|X_j,L,\Theta) = \frac{p(X_j|M_{w_j},\Theta)P(M_{w_j}|L)}{p(X_j|L,\Theta)} \tag{3.4}$$

Since $p(X_j|L,\Theta)$ is not constant during training

$$\begin{aligned} &P(M_{w_j}|X_j,L,\Theta) \approx; \\ &\frac{p(X_j|M_{w_j},\Theta)P(M_{w_j}|L)}{\sum_{i\neq w_j} p(X_j|M_i,\Theta)P(M_i|L) + P(X_j|M_{w_j},\Theta)P(M_{w_j}|L)} \end{aligned} \tag{3.5}$$

Maximum likelihood training maximizes $p(X_j|M_{w_j},\Theta)$ which appears both in the numerator and the denominator of (3.5). However, it does not necessarily maximize the left side, which is our goal. Furthermore, besides increasing the probability of the correct model we might also be increasing the probabilities of the incorrect models.

3.1.3 Implementation - The EM Algorithm

The most popular approach to iterative maximization (3.1) has been described in a number of classic papers (Baum & Petrie 1966; Baum *et al.* 1970; Baum 1972; Liporace 1982). Starting from initial guesses Θ^0, the model parameters are iteratively updated according to the "Forward-Backward" algorithm [or, equivalently, the Expectation-Maximization (EM) algorithm (Dempster *et al.* 1977)] so that (3.1) is maximized at each iteration. This kind of training algorithm, often referred to as Baum-Welch training in the particular case of HMMs, can also be interpreted in terms of gradient techniques (Levinson *et al.* 1983a; Levinson 1985).

Below we describe the general idea behind the EM algorithm. Besides its usage in HMM training as described above, it has a role in the training algorithm that will be presented in Chapter 6.

The EM Algorithm

This section is a brief description of the EM algorithm. Its application to HMM parameter estimation is described in (Huang *et al.* 1990; Levinson *et al.* 1983b; Baker 1975; Jelinek 1976; Jelinek & Mercer 1980; Lee 1989). Roughly speaking, there is an optimization problem that would be greatly simplified by the knowledge of additional variables Y. For example, optimizing (3.1), the likelihood of producing the acoustics, would be easier if the state sequence that produced the data was known. So we *estimate* the missing data Y (state sequence) using the observed data (acoustic sequences) and the current set of parameters Θ_t. We *maximize* the function using the estimated Y and we get a new set of parameters Θ_{t+1} that lead to a new estimate of the missing data Y. The estimation and maximization steps are iterated until the guaranteed convergence to a local maximum (Dempster *et al.* 1977).

More precisely, following (Dempster *et al.* 1977), the goal is to maximize the following likelihood function $L(X;\Theta)$, where Θ is the set of the parameters of the function and X the observations. The function estimates the log-likelihood of producing the observations given Θ. The problem would be simplified by the knowledge of additional variables Y. In that case we will maximize $L_c(X \cup Y;\Theta)$, the log-likelihood of producing the complete data $X \cup Y$ given Θ, and usually called the complete likelihood. However, since Y is not observable, the EM algorithm relies on integrating over the distribution of Y, with the following auxiliary function:

$$Q(\Theta, \Theta_t) = \mathrm{E}_Y[L_c(X \cup \ Y; \Theta)|X, \Theta_t] \tag{3.6}$$

which is the expected value of the complete data likelihood, given the observed data X and the current estimate of the parameters Θ_t.

The main difference between the likelihood function and the auxiliary function is that the auxiliary function is a deterministic function and we can optimize it, while the likelihood function is a stochastic one (it depends on the unobserved data Y). The EM algorithm iterates between the following two steps:

- **Estimation:** Compute $Q(\Theta, \Theta_t)$ based on Θ_t.
- **Maximization:** Maximize the auxiliary function.

$$\Theta_{t+1} = \underset{\Theta}{\mathrm{argmax}} \ Q(\Theta, \Theta_t) \tag{3.7}$$

In the case that we are unable to maximize the auxiliary function as required by the EM algorithm $Q(\Theta, \Theta_t)$, but can only increase it, the algorithm is called Generalized EM (GEM), and is also guaranteed to converge to a local maximum.

3.1.4 Summary

The two main conceptual problems with the maximum likelihood approach are:

- It is implicitly assumed that the model (with all its assumptions relative to its topology and probability density functions) is accurate and reflects the structure of the data (although the data might not adhere to the constraints imposed by the HMMs). If we had enough training data, it might be preferable to infer all the parameters of the models (including topology and non-parametric probability density functions) directly from the data. This can be seen as implicitly using a Bayes or MAP criterion (maximizing $P(M|X)$) during training instead of ML. Since the posterior probability includes the effects of prior information, the language model would also be inferred from the training data. However, it appears that this would require a prohibitive amount of training data. Nonetheless, the role of domain-specific knowledge is irreplaceable. It has the role of reducing the search space, i.e., the search is constrained to all models that are feasible given our knowledge. In addition, the use of other knowledge sources might result in better generalization, as the dependency on a particular training set is weaker. Hence, a combination of data-driven and knowledge-based approaches is desired.

- By training with ML instead of MAP, we strongly reduce the discriminant properties of HMMs. Ideally, each HMM should be trained not only to generate high probabilities for its own class, but also to discriminate against rival models.

Both of these two points (but particularly the second one) are related to the discussion that follows on discriminant criteria for HMM training.

The ML criterion can lead to optimal recognition performance only if the model is an exact statistical model of the source and the amount of training data is infinite (Nadas 1983; Duda & Hart 1973). However, these conditions are rarely (if ever) satisfied in speech recognition.

3.2 Discriminant Approaches

3.2.1 Introduction

In this section we describe discriminant approaches that have been proposed for speech recognition in general, and for HMMs in particular. Roughly speaking, discriminant approaches try to model the boundaries between the classes, and not model the classes themselves. Thus, a priori knowledge of class models does not play the same role. With an inaccurate model, the best we can do is to optimize its ability to distinguish between the underlying classes, which is typically achieved by replacing the ML criterion by a discriminant one.

3.2.2 Maximum Mutual Information (MMI)

Initially introduced in (Bahl *et al.* 1986; Brown 1987), this method aims at maximizing the mutual information (Cover & Joy 1991) between a set of (sentence) models M_{w_j} and the associated sequences of acoustic vectors X_j. This mutual information is then defined as (Cover & Joy 1991):

$$I(M_{w_j}, X_j|L,\Theta) = \sum_{M_{w_j},X_j} p(M_{w_j}, X_j|L,\Theta) \log \frac{p(M_{w_j}, X_j|L,\Theta)}{P(M_{w_j}|L,\Theta)p(X_j|L,\Theta)} \qquad (3.8)$$

$$= E_{p(M_{w_j},X_j|L,\Theta)}\left\{\log \frac{p(M_{w_j},X_j|L,\Theta)}{P(M_{w_j}|L,\Theta)p(X_j|L,\Theta)}\right\} \tag{3.9}$$

where Θ is the whole parameter space (for all models) in which optimization is performed and the sum over $(M_{w_j},\ X_j)$ represents a sum over all training utterances. $E_{p(M_{w_j},X_j|L,\Theta)}$ stands for the expected value according to the mass function $p(M_{w_j},X_j|L,\Theta)$. For one particular (M_{w_j},X_j) set, we then have:

$$\begin{aligned} I(M_{w_j},X_j|L,\Theta) &= \log \frac{p(M_{w_j},X_j|L,\Theta)}{P(M_{w_j}|L,\Theta)p(X_j|L,\Theta)} \\ &= \log \frac{p(X_j|M_{w_j},L,\Theta)}{\sum_{i=1}^{I} p(X_j|M_i,L,\Theta)P(M_i|L,\Theta)} \end{aligned} \tag{3.10}$$

As already mentioned in Section 2.2, the language model parameters L are often assumed independent of the acoustic parameters Θ and are estimated independently from a (large) text copra. Furthermore, the likelihoods $p(X_j|M_i,\Theta)$ depend only on the parameters Θ_i present in M_i. As a consequence, (3.10) can be rewritten as

$$I(M_{w_j},X_j|L,\Theta) = \log \frac{p(X_j|M_{w_j},\Theta_{w_j})}{\sum_{i=1}^{I} p(X_j|M_i,\Theta_i)P(M_i|L)} \tag{3.11}$$

in which the contribution of each term in the denominator is weighted according to the prior probability of the associated sentence as given by the language model, independent of the acoustic training data.

It is obvious that (3.10) and (3.11) are discriminant criteria. In (Bahl *et al.* 1986) it is shown that it is possible to get some kind of re-estimation recursion of local probabilities but, unfortunately, there is no proof that the recursion converges and there is no guarantee that the new estimates of (e.g., transition) probabilities are positive. As a consequence, a local gradient ascent method is usually used for optimization and the standard (likelihood-based) forward-backward recurrences are used to estimate the gradient. This is similar to the Alpha-Nets presented in (Bridle 1990) in which the gradient of the mutual information criterion takes the form of the backward recurrence used in the Baum-Welch algorithm. In the framework of hybrid HMM/ANN systems, this MMI criterion has been used in (Bengio *et al.* 1992), in which the ANN generates the sequence of acoustic vectors for the HMM and is trained to optimize the (global) MMI. In that paper, it is shown that it is possible to compute the gradient of the HMM training criterion (MMI or ML) with respect to the parameters of the ANNs.

However, in addition to "theoretical" problems, this algorithm suffers from a "practical" problem for continuous speech recognition. Indeed, optimization of Θ to maximize (3.10) requires not only a forward recurrence for the numerator, but also many forward recurrences for the denominator to estimate the contribution of all possible rival models.

Several solutions have been proposed to alleviate this problem, including:

1. If phoneme models are trained, the use of a "looped" phonetic model, i.e., a word model that allows any possible phoneme sequence (Merialdo 1988). This model may generate all possible phoneme sequences and, by running the forward recursion through it, may provide the summed probability in the denominator of (3.10).

2. Estimating the denominator in (3.10) by running an N-best algorithm (Schwartz & Chow 1990; Schwartz & Austin 1991), that provides the N-best (rival) sentences through which we run the forward recursion.

With the algorithm proposed in this thesis, in addition to all the advantages of "standard" HMM/MLP hybrids (i.e., local nonlinear discrimination, time correlation and no significant assumptions about probability density functions as discussed below), multiple forward recursions are not needed. Also, all probabilities will always be estimates of actual (local and global) posterior probabilities, will be positive, and will sum to one.

3.2.3 Generalized Probabilistic Descent (GPD)

Generalized Probabilistic Descent (GPD) is another discriminant approach that is sometimes used to train speech recognition systems. GPD is actually very close in spirit to MMI, although it permits generalization to different kinds of training criteria (Katagiri *et al.* 1991).

The idea of GPD is simple and can be summarized as follows. Given the whole set of parameters[1] Θ, define a discriminant function associated with each (word or sentence) model M_i as $g_i(X;\Theta)$. This discriminant function can be any differentiable distance function or probability distribution. Several instances of this are discussed in (Katagiri *et al.* 1991), each of them leading to different interpretations (as is also the case for MMI and MAP training). However, often the discriminant function is defined as:

$$g_i(X;\Theta) = -\log p(X|M_i,\Theta) \tag{3.12}$$

[1] In the following discussion we omit the language model L

Here again, (3.12) can be considered as the "full" (word or sentence) likelihood, the best-path (Viterbi) approximation (referred to as "segmental GPD training") or any intermediate solution like a sum over the S-best matching path scores. Another solution could be to define $g_i(X;\Theta)$ as the MMI in (3.11). However, since this will then be used in a discriminant measure (as defined below) taking all the classes into account, it can be easily shown that using MMI or full likelihoods as discriminant functions results in the same misclassification measure.

Classification (i.e., recognition) will then be based on that discriminant function according to the (usual) rule

$$X \in M_j \quad \text{if} \quad j = \underset{i}{\operatorname{argmax}}\; g_i(X;\Theta) \tag{3.13}$$

Given this discriminant function, we can define a misclassification measure that will measure the "distance" between one specific class and all the others. Here again, several measures can be used, each of them leading to different interpretations. However, one of the most general ones given in (Katagiri *et al.* 1991) is:

$$d_j(X;\Theta) = g_j(X,\Theta) - \log \left[\frac{1}{I-1} \sum_{i \neq j} \exp(\eta g_i(X;\Theta)) \right]^{1/\eta} \tag{3.14}$$

in which I represents the total number of possible reference models.

It is easy to see that if $\eta = 1$, (3.14) is then similar to (3.11), assuming (3.12), as a weighted ratio of the likelihood of the correct model by the likelihoods all models.

The error measure (3.14) could be used as the criterion for optimization by a gradient-like procedure, which would result in something very similar to MMI training. However, the goal of GPD is to minimize the actual misclassification rate, which can be achieved by passing $d_j(X;\Theta)$ through a nonlinear, nondecreasing, differentiable function F (such as a sigmoid function) and then minimizing

$$E(\Theta) = \sum_j \sum_{X \in M_j} F(d_j(X;\Theta)) \tag{3.15}$$

Other functions can be used to approximate the error rate. For example, we can also assign zero cost when an input is correctly classified and a unit cost when it is not properly classified, which is then another formulation of the minimum Bayes risk.

As briefly shown above, this approach is certainly very general and includes several discriminant approaches as special cases. For some problems such as continuous speech recognition, however, this approach has the same potential difficulty as MMI, i.e., the need to estimate "scores" (whatever they might be) of both the correct model and all possible rival models.

Chapter 4

Transition-based Modeling

4.1 Introduction

In this section we describe our motivation for studying a transition-based modeling approach to speech recognition, a particular model (the time-index model), and our experiments with the model. Our early interest in modeling transitions between speech units was motivated by perceptual and physiological studies. Therefore, we decided to study the following question: given accurate transition information between speech units, can we significantly improve recognition performance? Specifically, given boundary information between speech units, we explicitly condition the emission probability of a state on the time index, where time index is defined as the number of frames between entering a state and the current frame. Experiments with the time-index model as described below established the necessary condition that accurate transition information can significantly improve recognition performance. These results were corroborated by a recent study by Goldenthal (Goldenthal 1994). Goldenthal found a consistent improvement in phone recognition results when enhancing his segment-based models with explicit transition models. He used a set of 200 *canonical* transitions that were created by clustering all the transitions in the training set. Each canonical transition modeled the trajectory of a fixed number of frames centered about the transition boundary.

4.2 Perceptual and Physiological Point of View

A popular model of human speech production is that an utterance is organized as a succession of vocal-tract states, where each of these states represents a different configuration of the articulators. Further, at the level of articulatory performance, the gestures invoked to actualize these states are relatively slow. They merge spatially and temporally into a continuous process that usually only approximates the intended states and is highly sensitive to co-articulation (Deng & Sun 1994). For example, when vowels are co-articulated with consonants, the spectral pattern of the speech signal varies such that the acoustic targets found in isolated vowels are never fully realized in the changing spectrum. This is usually called "undershoot" (Lindblom & Studdert-Kennedy 1967). Obviously, in humans this problem is normally coped with.

The question whether human phonemic recognition is context-free or context-dependent has been addressed by Lindblom and Studdert-Kennedy among others (Lindblom & Studdert-Kennedy 1967). In their experiments, they tested the role of formant transitions in vowel recognition. Specifically they tested whether vowel recognition is done by the steady-state of the formants, i.e., zero rate of change, or by the short-term acoustic context, such as the direction and rate of adjacent formant transition. In the experiment, American English listeners were asked to identify monosyllabic nonsense speech. Each consonant-vowel-consonant (CVC) syllable was a sequence of three elements: transition + pattern at point of closest approach to target + transition. The vowel-formant patterns assigned to the points of closest approach to the target were selected from a continuum ranging from [I] to [u]. The rate and direction of the adjacent transitions were varied by the choice of two consonantal frames: [w-w] and [j-j]. Their results showed that the identity of the vowel stimulus is determined not only by the formant pattern at the point of closest approach to the target, but also by the direction of the adjacent formant transitions. For instance the same vowel pattern: $F_1 = 350Hz, F_2 = 1578Hz, F_3 = 2604Hz$ was recognized by almost all the subjects as [I] in the context of [jVj] and was recognized by most of them as [U] in the context of [wVw]. In general, it was shown that the categorization of the continuum is adjusted in the different environments so as to compensate for an undershoot effect in the vowel stimuli.

In a more recent study Furui has suggested that sufficient information for both vowel and consonant identification is contained across the same initial part of each syllable

(Furui 1986a). This part includes the maximum spectral transition. He also verified that the steady-state portions of the formants are not the only key for either vowel or syllable perception.

Auditory physiologists have collected a vast amount of data describing the response of mammalian auditory-nerve fibers to simple signals (Kiang 1984; Ruggero 1994) as well as more complex signals such as synthetic speech (Sachs *et al.* 1988). From these data it is clear that some sort of frequency analysis is performed and the dynamics of the response to non-steady-state signals is an important aspect of the auditory processing. Specifically, during the initial 15 ms of acoustic stimulation, the discharge rate of auditory-nerve fibers is often significantly higher than during the steady-state level. The decrease in response rate is referred to as "adaptation." Usually there is a very rapid initial decay in rate immediately after the onset, followed by a slower decay to a steady-state level, about 50 ms after the onset (Smith & Zwislocki 1975).

4.3 The Time-Index Model

4.3.1 Introduction

Motivated by the studies mentioned above we decided to consider the following question: given accurate transition information between speech units can we significantly improve recognition performance? The model that we chose to answer this question is the time-index model. The main idea behind the time-index model is that all the trajectories of a phone in the acoustic vector space share a stochastic dependency on the time elapsed since the beginning of the phone. This dependency can be modeled by a parametric distribution as in Deng's model (Deng 1992) or by using an MLP in our model.

4.3.2 Deng's Trended HMM

Deng described a model that explicitly conditioned the emission probability of a state on the time index, i.e., on the number of frames between the current frame and the previous state transition. For example, if the Markov chain has two states A and B, and we assume a specific realization that alternates between the states, the time index for a given state is depicted in Figure 4.1 (the figure does not show all the "machinery" of the HMM). Deng has coined his model the "trended HMM" (Deng 1992). In this model, a sequence of

State Sequence :	**A**	**A**	**B**	**B**	**B**	**A**	**A**
Time Index:	**1**	**2**	**1**	**2**	**3**	**1**	**2**

Figure 4.1: Depiction of the time index along with the current state

observation vectors generated in a given state is a combination of a stationary process and a deterministic function of time, as illustrated in the following equation for the multivariate normal distribution:

$$p(x_t|state, ti) = \frac{exp(-(x_t - g_{state}(ti))^T(\Sigma)^{-1}(x_t - g_{state}(ti)))}{(2\pi)^{\frac{n}{2}}(det\Sigma)^{0.5}} \tag{4.1}$$

where ti is the time index as defined above, $g_{state}()$ is a deterministic function of the time index and has parameters that may differ from state to state. In this simplified example, $g_{state}()$ shifts the mean vector of distribution as a function of the time index, while the stationary part is the variance-covariance matrix Σ. In principle this model explicitly conditions the emission probability on the time index, and a sequence of observations emitted from a given state is no longer assumed i.i.d. We do not know however the optimal form of $g_{state}()$ for each unit of speech. For example, one would expect a different time index dependence for vowels and stops. In the following, the time-index idea has been incorporated into a connectionist context.

4.3.3 An Introduction to the Time-Index Model

We propose a time-index model that 1) differs from an HMM in that the observations emitted in a given phone are no longer identically distributed and that 2) differs from Deng's model and others by its use of posterior probabilities estimated by a connectionist network. In the time-index model, the realizations of the state process are no longer sequences of values taken from the phone set, but rather are chosen from a set of pairs consisting of a phone and a time index. The time index is defined as the number of frames between the current frame and the previous state transition. For this model, the probability of generating a sequence of observations $X = \{x_t, x_{t+1}, \ldots, x_{t+T}\}$ in a given phone q_j is:

$$P(X_t^{t+T}|q_j) = \prod_{i=t}^{t+T} P(x_i|(q_j, i - t + 1)) \tag{4.2}$$

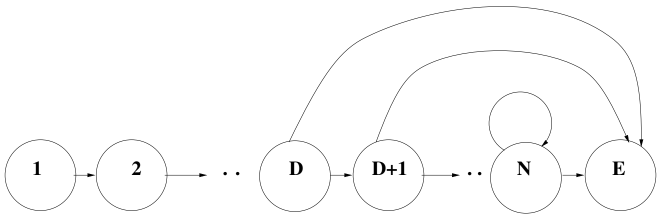

Figure 4.2: The topology of the time-index model

We can see that the $q's$ in the traditional HMM are replaced by a q and time index pair, as the state process is defined differently [1]. See also (Konig & Morgan 1994).

4.3.4 An Example

Figure 4.2 shows the topology of a basic unit of speech. In this case phones are the basic speech units. Only the last state in the model has a self loop. For states with indices smaller than the minimum duration, D, for that phone, only a transition to the next state (corresponding to a time index increment of one) is permitted. For all other states, transitions are permitted either to the next state or to the exit state. This model differs from a traditional HMM (assuming a similar representation for duration) primarily in that now the emission probability for each state (for each time associated with a phone or subphone unit type) is not constrained to be equal. Specifically, the emission probability of a state in the Markov chain is $P(x|(q_j, ti))$, where ti is the time index. Similar equations could be used for multi-state HMMs that are also commonly used, in which the basic speech unit is smaller than a phone. While certainly one could define a standard HMM with the kind of model shown in Figure 4.2, and with a separate emission probability for each state, the basic problems are how to reliably estimate so many free parameters and to model the correlation between these these states. One possible solution is to share parameters between the estimates for the separate densities. Another solution would be to assume a parametric form for the trajectory, as was done by Deng. Reported here is a multi-layer perceptron (MLP) approach which, in our previous work at ICSI (as discussed in Section 2.3.3), has proved useful for such estimates (Bourlard & Morgan 1994).

[1] Formally the values of the state process are ordered pairs of the phone and the time-index.

4.3.5 An Implementation of the Time-Index Model

In our model we define the emission probability of a state as $P(x|q_j, ti)$. While such a quantity can always be defined, the important question is how to estimate it. Now consider the following decomposition according to Bayes' law:

$$\frac{P(x|q_j, ti)}{P(x)} = \frac{P(q_j|ti, x)P(ti|x)}{P(ti, q_j)} \tag{4.3}$$

Where ti is the value of the time index, x is the acoustic vector, and q_j is a specific phone. Alternatively, this can be decomposed as follows:

$$\frac{P(x|q_j, ti)}{P(x)} = \frac{P(ti|q_j, x)P(q_j|x)}{P(ti, q_j)} \tag{4.4}$$

Each of the terms conditioned on x can be estimated by an MLP with an acoustic vector (or a local neighborhood of acoustic vectors) as input, as well as any additional conditioning terms as input (for instance, an additional input representing time index ti in order to estimate $P(q_j|ti, x)$). The targets correspond to a discrete binary coding of the class identity that is to the left of the condition bar (e.g., q_j for estimating $P(q_j|ti, x)$, or ti for estimating $P(ti|x)$). We have currently chosen to represent the ti inputs with a continuous-valued input as a smoother representation that requires fewer parameters. The first form of the equations given above requires the estimation of $P(q_j|ti, x)$, and this can be done with the MLP shown in Figure 4.3.

$P(ti, phone_j)$ can be estimated by counting the relative frequencies in the training set. Given that there is an accurate estimate of the boundaries between the phones we can calculate $P(ti|x)$; otherwise an estimation of this probability is a difficult problem (Glass 1988).

4.3.6 Experiments

The Resource Management (RM) speaker independent task (Price *et al.* 1988) and the TIMIT database were used for initial experiments. In the RM experiments our training data consisted of 3990 read, continuous-speech sentences, and the 300-sentence Feb89 test set for network cross-validation and testing. The time-index net (as shown in Figure 4.3) had 1000 hidden units and 61 outputs (the size of phone set). There were 235 inputs to the net, including 234 that consisted of 9 frames of 26 features each (PLP12 + log gain + delta features for each of these 13) (Hermansky 1990), and a final time-index input. With

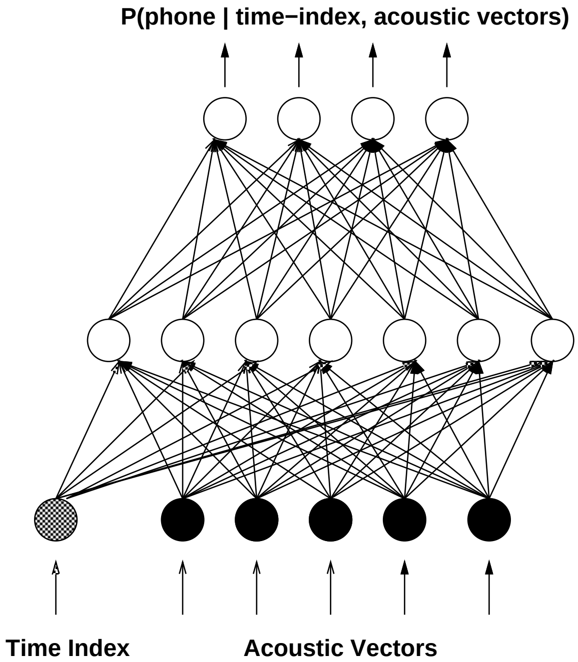

Figure 4.3: The time index net

the exception of this final input feature, this net was the same as the hybrid HMM/MLP system as described in (Bourlard & Morgan 1994).

During training we can find the boundaries between phones by running a Viterbi alignment on the known word strings and from these boundaries to calculate the time index as the distance of the current frame from the beginning of the phone as marked by the previous boundary. For the preliminary tests the boundaries between the phones were determined by an automatic alignment (Viterbi) procedure on the known word string (Viterbi 1967), also during recognition. This side information about the word sequence was used only to generate boundaries and no explicit phonetic information was preserved. Obviously during realistic recognition experiments the identity of the spoken sentence is not known. Therefore, one can expect little improvement over the boundary detection found by the Viterbi procedure with a known word sequence. Hence, these initial time-index results serve as a lower bound on the error. The results are summarized in Table 4.3.6.

The TIMIT corpus was chosen for the second set of experiments because it is phonetically balanced, and in addition there are time-aligned phonetic transcriptions of all the sentences in the database. The goals were to verify the potential of the model on

Task	*Baseline*	*Time-Index + Boundary Information*
RM	4.8%	1.1%
TIMIT	36.4%	25.0%

Table 4.1: Time-Index Results with Known Boundaries

a different test set and also to answer a potential criticism that the reduction of error is due to restricting the recognizer to utterances with the same number of phones as in the answers. By using the boundaries from the Viterbi alignment on the known word strings, we restrict the potential answers to have the same number of boundaries as in the answers, i.e., the same number of phones.

The experiments were done on a 200-sentence development set that was selected from the official training set and which was not used for the training. The size of the nets and the features were the same as in the RM task experiments. We used 3300 sentences for training and 396 sentences for cross-validation (the 200 sentence development set is a subset of the cross-validation set). No language model was used in these experiments. All results are on the full 61 TIMIT phone set. The standard system had 36.4% phone errors on this task, while the incorporation of the knowledge of phoneme boundaries in the time-index network reduced the error to 25.0%. See also Table 4.3.6. When we restricted our standard system to sentences that have the same number of phones as in the known answers, the error rate was still 36.4%, but with a different mix of insertions, deletions, and substitutions. Hence, the crucial information was the timing of the boundaries and not their number.

These results suggest that, given good information about the phoneme boundary locations, recognition error rate can be greatly reduced. This was a necessary result for the transition-based approach to be ultimately useful, but it is certainly not sufficient. There is still the difficult problem of either specifically locating boundaries, or getting smooth estimates of them.

Chapter 5

Discriminant HMM (DHMM)

5.1 Introduction

Motivated by the experimental results that indicate the importance of the timing of transitions between basic speech units as described above, we chose to study the Discriminant HMM (DHMM) model, that uses local transition probabilities to estimate posterior probabilities of word sequences. Furthermore it is a true recognition model, i.e., it directly maps from acoustic sequences to sentences, unlike HMMs that model the inverse relation (the likelihood of producing an acoustic sequence).

The original theory of DHMM was described in (Bourlard & Wellekens 1989b). However, in the years following the original theoretical formulation, simplified systems to reduce the dependence on distributional assumptions for the observation space, and to make the probability estimates more discriminant. These simplified approaches did not make use of the full power of the initial scheme. Nonetheless, in controlled tests they displayed significant strength (Lubensky *et al.* 1994; Renals *et al.* 1994; Robinson *et al.* 1993). The basic scheme consisted of training neural networks to estimate probabilities of HMM states, and then using simple functions of these probabilities to label the training data using Viterbi decoding (dynamic programming). This procedure was repeated iteratively to train the system. During recognition the Viterbi procedure was used with probabilities produced by trained networks.

The remainder of this section will describe the original theory, but with the benefit of hindsight because of more recent developments. In addition, we describe experiments incorporating the original theory. An analysis of these experimental results suggests a

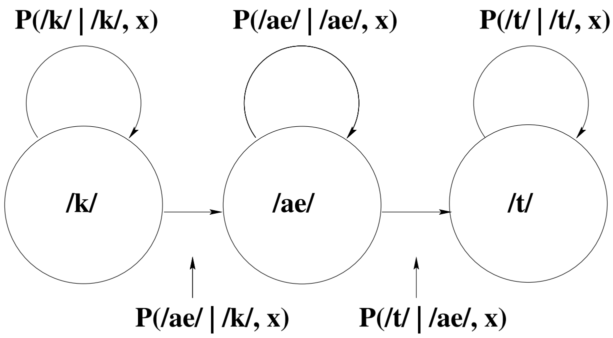

Figure 5.1: An example of a Discriminant HMM for the word "cat". The variable x refers to a specific acoustic observation x_n at time n.

partial explanation why the original theory did not initially work. In the next chapter we describe a new training algorithm that overcomes some of the problems.

5.2 Estimation of the Posterior Probability of Word Sequences

In (Bourlard & Morgan 1994) it was shown that it is possible to compute the global posterior probability $P(M|X, L, \Theta)$ of (2.5) and (2.6) as:

$$\begin{aligned} P(M|X, L, \Theta) &= \sum_{\Gamma_j} P(M, \Gamma_j|X, L, \Theta) && (5.1) \\ &= \sum_{\Gamma_j} P(M|\Gamma_j, X, L, \Theta) P(\Gamma_j|X, L, \Theta) && (5.2) \end{aligned}$$

in which "Γ_j" represents a legal state sequence in M, see Section 2.1.3. Considering the second factor of (5.2) as the acoustic model and assuming that it is independent of the language model parameters, i.e.,:

$$P(\Gamma|X, L, \Theta) \approx P(\Gamma|X, \Theta) \tag{5.3}$$

and writing $P(\Gamma|X, \Theta)$ explicitly as $P(q^1, \ldots, q^N|X, \Theta)$, allows us to factor it as follows:

$$\begin{aligned} P(q^1, \ldots, q^N|X, \Theta) &= P(q^1|X, \Theta)\ P(q^2|X, q^1, \Theta) \ldots \\ &\quad \ldots P(q^N|X, q^1, \ldots, q^{N-1}, \Theta) && (5.4) \\ &= \prod_{n=1}^{N} P(q^n|X, Q_1^{n-1}, \Theta) && (5.5) \end{aligned}$$

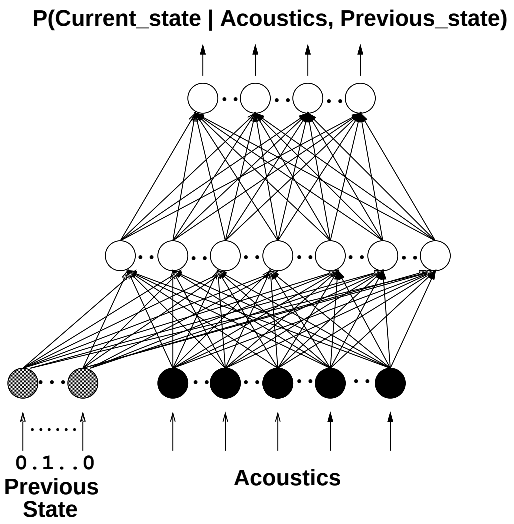

Figure 5.2: An example MLP that estimates local conditional probabilities.

where q^n represents the state observed at time n and Q_1^N the state sequence associated with X_1^N. Probabilities $P(q^1, \ldots, q^N|X, \Theta)$ can thus be calculated from "local" probabilities $P(q^n|X, Q_1^{n-1}, \Theta)$. These local probabilities may be simplified by relaxing the conditional constraints, by assuming dependency only on the previous state (first-order Markov model assumption) and on a temporal window X_{n-c}^{n+d} around the acoustic vector at time n (acoustic correlation limited to the contextual window). These local contributions can then be approximated by

$$P(q^n|X, q^1, \ldots, q^{n-1}, \Theta) \;\approx\; P(q^n|X_{n-c}^{n+d}, q^{n-1}, \Theta) \tag{5.6}$$

where input contextual information is taken into account. These probabilities can be estimated at the outputs of an MLP with contextual input and output feedback, as described in Figure 5.2.

Thus, using Bayes' law, we can then rewrite (5.4) for a particular path Γ_j as:

$$P(\Gamma_j|X, \Theta) \;\approx\; \prod_{n=1}^{N} P(q_{\Gamma_j}^n|q_{\Gamma_j}^{n-1}, X_{n-c}^{n+d}, \Theta) \tag{5.7}$$

A simple example of the model is given in Figure 5.2.

These new acoustic models, referred to as Discriminant HMMs (DHMM)[1], are now described in terms of conditional transition probabilities $P(q_n^\ell|q_{n-1}^k, x_n)$, in which q_n^ℓ

[1]It could be argued that these models are no longer HMMs but more like "stochastic finite state acceptors", a name suggested recently by John Bridle.

stands for the specific state q^ℓ of $\mathcal{Q}$ hypothesized at time n. As with traditional hybrid HMM/ANN systems, conditional transition probabilities can be estimated by an ANN (in our case a multi-layer perceptron) with K output units and in which the acoustic input x_n is[2] complemented by a set of additional input units representing the state q^ℓ hypothesized at the previous time step $n-1$. The conditional transition probabilities are also functions of Θ, the ANN parameter set, and can be written as $P(q_n^\ell | q_{n-1}^k, x_n, \Theta)$.

In estimating the first factor in (5.2) $P(M|\Gamma_j, X, L, \Theta)$ given the state sequence and the language model we can assume no dependence on the acoustic sequence:

$$P(M|\Gamma_j, X, L, \Theta) \approx P(M|\Gamma_j, L, \Theta) \tag{5.8}$$

Thus, (5.2) can be rewritten as:

$$P(M|X, \Theta, L) \approx \sum_{\Gamma_j} P(\Gamma_j|X, \Theta) P(M|\Gamma_j, L, \Theta) \tag{5.9}$$

in which $P(\Gamma_j|X, \Theta)$ is computed as described above.

5.2.1 Recognition

In the case that there are no homonyms, i.e., there is not a phone sequence that corresponds to more than one sentence, then the term $P(M|\Gamma_j, L, \Theta)$ serves as a 0/1-valued filter. Equation (5.9) can then be rewritten as follows:

$$P(M_i|X, \Theta, L) \approx \sum_{\Gamma_j \in M_i} P(\Gamma_j|X, \Theta) \tag{5.10}$$

And it can be estimated with the following efficient forward recursion[3].

Forward Recursion

We start with some definitions:

$$\alpha_n(k|M_j) = \sum_{\Gamma_n \in M_j} P(\Gamma_n, q_k^n | X_1^n, \Theta) \tag{5.11}$$

---

[2]As done with previous hybrid HMM/ANN systems, x_n will usually be replaced by $X_{n-c}^{n+d} = \{x_{n-c}, \ldots, x_n, \ldots, x_{n+d}\}$ to take some acoustic context into account.

[3]Note this forward recursion will be different from the forward recursion that is used in training time and will be defined in the next section.

summing up all the paths of length n in model M_j that end in state q_k at time n. We initialize the recursion with:

$$\alpha_1(k|M_j) = P(q_k^1|x_1, \Theta) \tag{5.12}$$

for the first k valid states in model M_j. Now for the dynamic programming step:

$$\begin{aligned} \alpha_{n+1}(\ell|M_j) &= \sum_{\Gamma_{n+1}\in M_j} P(\Gamma_{n+1}, q_\ell^{n+1}|X_1^{n+1}, \Theta) & (5.13) \\ &= \sum_{\Gamma_n \in M_j} P(\Gamma_n|X_1^{n+1}, \Theta) P(q_\ell^{n+1}|X_1^{n+1}, \Gamma_n, \Theta) \\ & \quad \text{assuming causality } : P(\Gamma_n|X_1^{n+1}, \Theta) = P(\Gamma_n|X_1^n, \Theta), \text{ denoting} \\ & \quad \text{the ith state of } \Gamma_n \text{ by } \; k \;, \text{ and assuming 1st order Markov model} \\ &= \sum_k \alpha_i(k|M_j) p(q_\ell^{n+1}|x_{n+1}, q_k^n, \Theta) & (5.14) \end{aligned}$$

Thus, equation (5.10) can be computed using the forward recursion:

$$P(M_i|X, L, \Theta) \approx \sum_{q_f} \alpha_N(q_f|M_i) \tag{5.15}$$

where q_f is a legal final state for the model M_i. The use of the language model (in the case that we have homonyms) is discussed in Section 6.4.

5.3 MAP Constraints

In the previous section we described how to estimate the global posterior probability of a Markov model given the acoustic sequence. In this section we show that it can be done while meeting the constraint specified in Equation (2.7), i.e.,

$$\sum_{i=1}^{I} P(M_i|X, L, \Theta) \;=\; 1 \tag{5.16}$$

where the sum over i represents the sum over all possible Markov models (Bourlard *et al.* 1994). Here lies the difference between an ML and an MAP criterion. Any modification of the parameters of a model M_i must be complemented by a modification of all the parameters of the other models so as to preserve this constraint, hence making the MAP procedure discriminant.

It is also important to show that, in this case, if the "local" constraint:

$$\sum_{k=1}^{K} p(q_k^n|x_n, q_\ell^{n-1}, \Theta) \;=\; 1 \tag{5.17}$$

is met (which will be the case, at least approximately, with sigmoidal MLP outputs[4]), the constraint (2.7) on the global MAP probabilities is also met. Indeed, summing over the set of all possible paths $\{q^1, \ldots, q^n, \ldots, q^N\}$ in all possible Markov models M_i, we have:

$$\begin{aligned}
\sum_i P(M_i|X, L, \Theta) &= \sum_{\Gamma_j} \sum_i P(M_i, \Gamma_j | X, L, \Theta) \qquad (5.18) \\
&= \sum_{\Gamma_j} \sum_i P(M_i | X, \Gamma_j, L, \Theta) P(\Gamma_j | X, L, \Theta) \\
&\quad \text{Assuming (5.3) and (5.8)} \\
&= \sum_{\Gamma_j} P(\Gamma_j | X, \Theta) \sum_i P(M_i | \Gamma_j, L, \Theta) \\
&\quad \text{and, assuming } \sum_i P(M_i|\Gamma_j, L, \Theta) = 1, \;\; \forall \Gamma_j : \\
&= \sum_{\ell_1=1}^{K} P(q_{\ell_1}^1 | x_1, \Theta) \ldots \\
&\quad \ldots \left(\sum_{\ell_2=1}^{K} P(q_{\ell_2}^2 | x_2, q_{\ell_1}^1, \Theta) \ldots \left(\sum_{\ell_N=1}^{K} P(q_{\ell_N}^N | x_N, q_{\ell_{N-1}}^{N-1}, \Theta) \right) \ldots \right) \\
&= 1
\end{aligned}$$

It is however important to remember that this property is valid only if one considers all possible paths through the models.

Besides the advantage of forcing discrimination, numerical problems that plague the classical HMM are avoided when using discriminant models: namely, the lack of balance between the transition probability values (which only depend on the topology of the model) and the emission probability values (which decrease with the dimension of the input feature space) (Brown 1987).

5.4 Early Experiments and Error Analysis

5.4.1 Early Experiments

Given the theoretical properties of the Discriminant HMM/MLP model described earlier, we felt that empirical evaluations of this model would be a good first step in improving our understanding of transition-based systems. In particular, we began to empirically

[4]This constraint is precisely met in the case of a softmax output layer, since the outputs are normalized to sum to 1.

evaluate conditional transition probabilities as used in Discriminant HMM/MLP systems on phoneme classification and phonemic frame classification tasks.

As presented in the initial theory (Bourlard & Morgan 1994) the paradigm for training (and recognition) was to use the Viterbi approximation, i.e., to consider only the most probable state sequence in assigning phonetic labels to acoustic frames. The local discriminant probabilities (2.14) were estimated by an MLP as represented in Figure 5.2. In this case, the previous state is coded as additional binary inputs, one for each possible previous state. For every hypothesized previous state we set the corresponding input to one and the rest to zero. The set of possible previous states (or the set of possible successor states for a given q_k at the input) will be given by the topology of the HMMs (and by the currently hypothesized states of the matching process).

In Viterbi training (as used so far) we know the correct previous state (again considering only the most probable state sequence), either by having a hand-segmented database such as TIMIT, or by running an automatic alignment (Forced-Viterbi) on the training data. During recognition however, the MLP outputs will have to be hypothesized for every possible previous state (possibly constrained by a particular HMM topology or a language model).

The TIMIT corpus (Garofolo 1988) was chosen for the experiments because it is phonetically balanced and in addition there are time-aligned phonetic transcriptions of all the sentences in the database. The experiments were done on a 200-sentence development set that was selected from the official training set and was not used for the training. We used 3300 sentences for training and 396 sentences for cross-validation (where the 200 sentence development set is a subset of the cross-validation set). No language model was used in these experiments. All the results were on the full TIMIT 61 phone set. Phone models were simple one-state-per-phone models.

The net that estimated the local discriminant probabilities (as shown in Figure 5.2) had 1000 hidden units, 61 outputs (the size of the phone set). There were 295 inputs to the net, including 234 that consisted of 9 frames of 26 features each (PLP12 + log gain + delta features for each of these 13) (Hermansky 1990), and 61 binary inputs that represented the possible previous state. With the exception of these binary inputs, this net was the same as the hybrid HMM/MLP system as described in (Bourlard & Morgan 1994). The baseline HMM/MLP system (Bourlard & Morgan 1994) had 36.3% phone error on this task. When evaluating the Discriminant HMM on this task the error rate was 40.4%. This was an intriguingly negative result; *increasing* the amount of input information led to a *decrease* in generalization performance. Why should this be so? Although it is difficult to draw firm conclusions from a negative result, it can at least inspire directions of inquiry. This result motivated the error analysis as described in the following section.

5.4.2 Error Analysis

As shown in the following, the error analysis suggested two potential reasons[5] for the observed performance loss: (1) poor transition detection, and (2) mismatch between the input space distribution of the MLP during training and recognition.

The first potential problem is missing transitions; implicitly the net is a transition detector because when it determines that the current state is different from the previous one it signals a transition, and transition detection between phonemes is known to be a hard problem (see (Glass 1988)). In order to test this assumption we compared the performance of the MLP described above on two kinds of acoustic frames: *transition frames* that start a new segment, i.e., their phonetic label is different from the previous frame, and all other frames, *self-loop frames*. *While presenting the correct previous state*, the frame level performance on the development set was:

1. Self-loop frames: 85.5% of correct phonemic classification.

2. Transition frames: 39.2% of correct transition detection and classification.

Transitions thus seems harder to detect and classify than "steady-state" frames. However, we suspect that this is not (only) due to the properties of transitions, but to two problems related to the training and testing procedure:

1. We have much less training data for transition frames than for "steady-state" frames (less than one-sixth). Thus the learning ability of the classifier will tend to focus on the steady-state phonetic classification.

2. Our training procedure assumes that a single frame is the transition and that its neighbors are not transitions[6] . This does not make sense in terms of the acoustic phonetics, since many spectral transitions are gradual. This makes a difficult classification function for a network to learn.

[5] Other than bugs.

[6] A transition frame is a frame whose target for the current state is different from the hypothesized previous state.

The second potential problem is the possible disparity between training and recognition input populations. During training we only present to the net "correct" pairs of acoustic vectors and the correct previous state, while in recognition we expect the net to generalize to all possible combinations of acoustic vectors and previous states. Some of these recognition inputs can be completely meaningless, e.g., like the combination of the acoustics of a middle of a vowel and a previous state that corresponds to a plosive. The net is not trained on anything close to these "impossible" pairs, but through the vagaries of interpolation these pairs could end up having the highest MLP outputs during recognition. This problem is often referred to as the "lack of negative training example" problem and sometimes can be partially overcome by presenting additional negative training examples to the net (Zavaliagkos *et al.* 1994).

In order to test this hypothesis we computed the frame level performance of the net on the development set for the following two cases:

1. When the correct previous state is presented as input, the highest probability output was correct 79.4% of the time.

2. Presenting all possible previous states and taking as the winner the output with the highest activity, i.e., taking for every frame the maximum of 61 by 61 probabilities (61 outputs for each possible previous state), and checking if it was the correct pair of previous state and current state. In this case the result was 15.9% correct, which was the weighted average of 18.3% correct on self-loop frames and 0.4% correct on transition frames.

These results seem to suggest that, even for "steady-state" frames there is a problem of mismatch between the space of training and testing for hypothesized inputs. Of course, performance is also hurt by the problems mentioned earlier.

All the problems identified here motivated the REMAP training and recognition algorithm for HMM/MLP hybrids that is presented below. Specifically, the first experiment showed that "hard" transitions are difficult to detect. As we will see, the full MAP training will provide the nets with soft targets and soft decisions, i.e., with conditional probabilities of transitions. Furthermore, by considering all possible paths and transitions, we will reduce the mismatch between training and recognition. A formalism will be introduced that automatically considers negative training examples without the need for explicit enumeration of impossible input combinations.

Chapter 6

REMAP Training of HMM/MLP Hybrids

6.1 Introduction

6.1.1 Motivations

The discriminant HMM/MLP theory as described above uses transition-based probabilities as the key building block for acoustic recognition. However, it is well known that estimating transitions accurately is a difficult problem (Glass 1988). Due to the inertia of the articulators, the boundaries between phones are blurred and overlapped in continuous speech (Deng & Sun 1994). It is also likely that some time variability exists in the human perception of the onset of a new phonetic region. Consequently, we would like to have a "window" of possible transitions instead of a single transition. Ideally the width of the transition window should depend on the specific bi-phone and on the speaker. Thus we need an automated way of estimating the transition windows to be used as targets in the MLP training.

In hybrid HMM/MLP systems, targets are typically obtained using an automatic alignment procedure incorporating a Viterbi approximation. However, this procedure yields rigid transition targets, and thus suffers from the problems mentioned above. Furthermore, our preliminary experiments with this procedure yielded poor transition detection performance.

Another related problem in our Viterbi-based MLP training procedure is a disparity between the training input space of the MLP and the input space used in recognition. Specifically, in training the network only processes inputs consisting of "correct" pairs of acoustic vectors and correct previous state, while in recognition we expect the net to generalize to all possible combinations of acoustic vectors and previous states.

One possible solution to these problems is to use a full MAP algorithm to find transition probabilities at each frame for all possible transitions with a forward-backward-like algorithm (Liporace 1982), which takes all possible paths into account. Furthermore a MAP algorithm would increase the a posteriori probability of the correct model and reduce the posterior probabilities of all other models. Thus, it might improve the approximation to an optimal Bayes classifier.

In the rest of this chapter, we describe a set of procedures that were derived in order to train and use the desired discriminant probability estimators in a full MAP framework.

6.1.2 Problem Formulation

Global MAP training of Discriminant HMMs should find the optimal acoustic parameter set Θ maximizing (2.6). In the following derivation the dependency on the language model L is omitted, in order to concentrate on the acoustic model parameters (the role of the language model will be discussed in Section 6.4). Although, in principle, we could use a generalized back-propagation-like gradient descent in Θ could maximize (2.6) (see, e.g., (Bengio *et al.* 1992)), an EM-like algorithm would have better convergence properties and would preserve the statistical interpretation of the ANN outputs. In this case, "full" MAP training of transition-based HMM/ANN hybrids requires a solution to the following problem: given a trained ANN at iteration t providing a parameter set Θ^t and, consequently, estimates of $P(q_n^\ell|x_n, q_{n-1}^k, \Theta^t)$, how can we determine new ANN targets that:

1. Will be smooth estimates of conditional transition probabilities, $\forall$ possible (k, ℓ) state transition pairs in M and $\forall n \in [1, N]$.

2. When used in training the ANN for iteration $t+1$, will lead to new estimates of Θ^{t+1} and $P(q_n^\ell|x_n, q_{n-1}^k, \Theta^{t+1})$ which are guaranteed to incrementally increase (2.6)?

6.2 Solution - REMAP

6.2.1 Introduction

In this section we describe a solution to the problem mentioned above. Specifically, we prove that for any training sentence X, an iteration consisting of

1. estimating new MLP training targets from a previously trained MLP via a backward recursion, and

2. training the MLP with the new targets

will increase the global MAP probability of the sentence model given the sequence of acoustic vectors[1] $P(M|X)$. It is easy to see that this proof can be generalized to several training sentences, since this is then simply equivalent to training on a long sentence built by concatenating all training sentences (with additional start and end point constraints). We describe below the main ideas and steps underlying the proof. Some of the technical details are discussed in Appendix A. The proof has three main steps:

1. Defining an auxiliary function such that maximizing this function is equivalent to maximizing the global posterior probability of the correct model and, since such probabilities must sum to 1 for the complete set of possible models, minimizing the posterior probabilities of the rival models.

2. Finding new targets for training the MLP that maximize the auxiliary function.

3. Showing that training the MLP with those new targets (using a weighted relative entropy error criterion) leads to an increase in the value of the auxiliary function.

Note that while this thesis has largely assumed the use of an MLP for the required probability estimation, other gradient-trained estimators (such as a recurrent network) could also be used.

[1]To simplify our notation, in all the following discussion only one training sentence X is associated with the Markov model M, but it is easy to see that the discussion remain valid in the case of several training sentences X_j associated with M_{w_j}, for $j = 1, \ldots, J$.

6.2.2 Definitions

Let us define an auxiliary function[2] $R(\upsilon_1, \upsilon_2)$ as a function on two probability sets υ_1, υ_2, where Υ is defined below:

$$\Upsilon = \{P_z(q_\ell^n | x_n, q_k^{n-1} : \forall k, \ell \in [1, \ldots, K],\ \forall n \in [1, \ldots, N], z \in \mathcal{Z}\} \tag{6.1}$$

Each set contains $K^2(N-1)$ possible conditional transition probabilities, where K is the number of states in the model, and N is the number of acoustic vectors, and $\mathcal{Z}$ the number of possible legal sets. Note that the probability sets can be a function of a probability estimator. In our study, these probabilities are estimated by an MLP with parameter (weight) set Θ, in which we denote the probabilities as $P(q_\ell^n | x_n, q_k^{n-1}, \Theta)$. In this case, the probability set υ also becomes a function of Θ and is then denoted by $\upsilon(\Theta)$.

The auxiliary function $R(\upsilon_1, \upsilon_2)$ is defined as

$$R(\upsilon_1, \upsilon_2) \ = \ \frac{1}{P(M|X, \upsilon_1)} \sum_{\Gamma} P(M, \Gamma | X, \upsilon_1) \ \log P(M, \Gamma | X, \upsilon_2) \tag{6.2}$$

where Γ is a legal path (state sequence) in model M and $P(M, \Gamma | X, \upsilon_i)$ represents the probability of a specific path Γ in M given a probability set υ_i.

6.2.3 Theorem 1

Theorem 1: *IF* $R(\upsilon_1, \upsilon_2) \geq R(\upsilon_1, \upsilon_1)$

THEN $P(M|X, \upsilon_2) \geq P(M|X, \upsilon_1)$.

In other words, if we can find a new set of probabilities υ_2 increasing R, the new set of probabilities will also increase the posterior probability of the model M. The proof is described in Appendix A.

6.2.4 Theorem 2

The question that arises from the first theorem is how to find a new set of probabilities υ_2 that increases the value of the auxiliary function $R(\cdot)$ and, consequently, the posterior probability of the correct model (and therefore also minimizes the posterior probability of the rival models).

[2]This auxiliary function is usually denoted $Q(\cdot)$ (Dempster *et al.* 1977). However, to avoid any possible ambiguity with an HMM state sequence, we denote it $R(\cdot)$ in this thesis.

Theorem 2:

Given υ_1 a fixed set of probabilities that is estimated by an MLP with a fixed set of weights Θ, we show that $R(\upsilon_1, \upsilon_2)$ attains its maximum value when the conditional transitional probabilities $P_{\upsilon_2}(q_\ell^n|x_n, q_k^{n-1}) \in \upsilon_2$ are defined as[3]

$$P_{\upsilon_2}(q_\ell^n|x_n, q_k^{n-1}) = P(q_\ell^n|X, q_k^{n-1}, \upsilon_1(\Theta), M) \tag{6.3}$$

Theorem 2 says the following: a trained MLP with a fixed set of parameters (MLP weights) Θ provides us with estimates of conditional transition probabilities $P(q_\ell^n|x_n, q_k^{n-1}, \upsilon_1(\Theta))$ (estimated on a given training data set $X = \{x_1, \ldots, x_n, \ldots, x_N\}$) $\forall n = 1, \ldots, N$ and $\forall k, \ell = 1, \ldots, K$. Given these estimates, obtained at the output of the MLP, it is possible to compute re-estimates of the conditional transition probabilities $P_{\upsilon_2}(q_\ell^n|x_n, q_k^{n-1}) = P(q_\ell^n|X, q_k^{n-1}, \upsilon_1(\Theta), M)$ by the backward recursion given in (6.14) to increase the global posterior probability of the correct model M over $P(M|X, \upsilon_1)$. The proof is described in Appendix A.

6.2.5 Theorem 3

As opposed to the "standard" EM algorithm (Baum *et al.* 1970; Baum 1972), Theorems 1 and 2 are not enough to prove convergence of the training process for two reasons:

1. The MLP training is usually minimizing a function (e.g., least mean square or relative entropy) that is different from the function optimized in Theorem 2. As a consequence, convergence must be proved through the same auxiliary function $R(\cdot)$.

2. Theorem 2 gives new ("optimal") values (MLP targets) for the conditional transition probabilities which are going to be used to train the MLP. If the cost function can be trained to reach its optimal minimum, the MLP will just "learn" the targets and we will have
$$g_\ell(x_n, q_k^{n-1}, \Theta) = p(q_\ell^n|X, q_k^{n-1}, M, \upsilon_1) \tag{6.4}$$
which, by Theorem 2, is known to increase $R(\cdot)$ and, consequently, $P(M|X)$. In this case, of course, we proved that MLP training is increasing $P(M|X)$ and we do not

[3] Of course, all x_n's in the following should be replaced by X_{n-c}^{n+d} if local contextual input is used, or X_1^n for a recurrent network.

need anything more. However, in general, the nets will not be trained to their optimal minimum because of

- "overlapping" of input patterns (e.g., two instances of the same pattern with two different classifications).
- use of cross-validation (early stopping) (Bourlard & Morgan 1994) to avoid overfitting and to get better estimates of actual probabilities.

Below we describe a training procedure for the MLP and a corresponding error criterion. We show that by minimizing this criterion we are maximizing the auxiliary function $R(\cdot)$. Thus by Theorem 1 the posterior probability of the correct sentence is increased. By this we show convergence (at least to a local minimum) on the training set.

Specifically, given a trained MLP with a set of weights Θ^t which provides a set of conditional transition probabilities $v_1(\Theta^t)$ and given a sequence of acoustic vectors X and a model M, the Discriminant HMM backward recursion can be used to compute a new set of probabilities

$$T \;=\; \{P_T(q_\ell^n|x_n, q_k^{n-1}) = P(q_\ell^n|X, q_k^{n-1}, v_1(\Theta^t), M); \forall k, l = 1, \ldots, K; \forall n = 1, \ldots, N\} \quad (6.5)$$

which will be used as targets to adapt further the MLP weights to a new set of weights Θ^{t+1} and, consequently, a new set of conditional transition probabilities $v_2(\Theta^{t+1})$.

In the following we prove this property in the case of a weighted relative entropy E_e, similar to a common cost function for MLP training.[4] In this case, given a sequence of acoustic vectors X and a model M and the current set of parameters Θ^t, the parameters Θ^{t+1} of the MLP are trained to minimize

$$E_e(\Theta^{t+1}) = \mathcal{E}_{P(x_n, q_k^{n-1}|M,X,\Theta^t)} \sum_{\ell=1}^{K} P_T(q_\ell^n|x_n, q_k^{n-1}) \log \frac{P_T(q_\ell^n|x_n, q_k^{n-1})}{g_\ell(x_n, q_k^{n-1}, \Theta^{t+1})} \quad (6.6)$$

where $g_\ell(x_n, q_k^{n-1}, \Theta^{t+1})$ is the ℓ-th output of the MLP given weight set Θ^{t+1} and inputs (x_n, q_k). Note that the expected value $\mathcal{E}_{P(x_n, q_k^{n-1}|M,X,\Theta^t)}$ is taken according to the distribution of the input variables that in the case of the Discriminant HMM are the concatenation

[4] Relative entropy is a particularly common error criterion for classification and probability estimation tasks, and we have used it for all of the speech training systems that we have developed over the last few years. The new criterion will actually only differ in that the expectation leading to its formulation will be taken with respect to the entire network input space, which includes a choice for the previous state.

of the acoustic input and the previous state. In this case (6.6) can also be expressed as:

$$
\begin{aligned}
E_e(\Theta^{t+1}) &= \sum_{x_n, q_k^{n-1}} P(x_n, q_k^{n-1}|M, X, \Theta^t) \sum_{\ell=1}^{K} P_T(q_\ell^n|x_n, q_k^{n-1}) \log \frac{P_T(q_\ell^n|x_n, q_k^{n-1})}{g_\ell(x_n, q_k^{n-1}, \Theta^{t+1})} \\
&= \sum_{n=1}^{N} \sum_{k=1}^{K} P(q_k^{n-1}|M, X, x_n, \Theta^t) P(x_n|X, M, \Theta^t) \\
&\qquad \left[\sum_{\ell=1}^{K} P_T(q_\ell^n|x_n, q_k^{n-1}) \log \frac{P_T(q_\ell^n|x_n, q_k^{n-1})}{g_\ell(x_n, q_k^{n-1}, \Theta^{t+1})} \right] \qquad (6.7)
\end{aligned}
$$

Given that $P(q_k^{n-1}|M, X, x_n, \Theta^t) = P(q_k^{n-1}|M, X, \Theta^t)$ and using the fact that $P(x_n|X, M, \Theta^t) \ = \ 1$,

$$
\begin{aligned}
E_e(\Theta^{t+1}) &= \sum_{n=1}^{N} \sum_{k=1}^{K} P(q_k^{n-1}|M, X, \Theta^t) \\
&\qquad \left[\sum_{\ell=1}^{K} P_T(q_\ell^n|x_n, q_k^{n-1}) \log \frac{P_T(q_\ell^n|x_n, q_k^{n-1})}{g_\ell(x_n, q_k^{n-1}, \Theta^{t+1})} \right] \\
&\quad \text{replacing } P_T(\cdot) \text{ with its definition (6.5) we get :} \\
&= \sum_{n=1}^{N} \sum_{k=1}^{K} P(q_k^{n-1}|M, X, \Theta^t) \\
&\qquad \left[\sum_{\ell=1}^{K} P(q_\ell^n|X, q_k^{n-1}, \upsilon_1(\Theta^t), M) \log \frac{P(q_\ell^n|X, q_k^{n-1}, \upsilon_1(\Theta^t), M)}{g_\ell(x_n, q_k^{n-1}, \Theta^{t+1})} \right] \qquad (6.8)
\end{aligned}
$$

It is easy to see that the above criterion will reach its global minimum when the outputs of the MLP are equal to the targets [5]. Note that the relative entropy between two probability mass functions is always greater or equal to zero (Cover & Joy 1991). Given that the targets are posterior probabilities, a network trained to the global minimum of error criterion (6.8) will estimate the posteriors.

An important point is that previous state q_k^{n-1} is not one of the features that are extracted from the speech waveform. Thus the scaling factor $P(q_k^{n-1}|X, M, \upsilon_1(\Theta^t))$ is needed to compute the expected value over the "extended" input space. There are several ways to implement this scaling, one being to choose the previous state uniformly and to scale the error signal that is back propagated by this factor. An alternative way in stochastic gradient descent training (online training) is to implement this criterion by first choosing

[5] In the case that both the target probability and the net output are zero, this still holds given $\lim_{\epsilon \to 0} \epsilon \log \frac{\epsilon}{\epsilon} = 0$

the acoustic frame x_n at random from the training test, and then choosing the previous state according to $P(q_k^{n-1}|X, M, v_1(\Theta^t))$.

<u>Theorem 3:</u>

Minimizing the weighted relative entropy criterion (6.6) with the target set T (which is calculated from a probability set v_1) maximizes the auxiliary function $R(\cdot)$. Specifically, the new set of probabilities v_2, implemented by the trained MLP, satisfies the following:

$$E_e(\Theta^{t+1}) \leq E_e(\Theta^t) \Longrightarrow R(v_1(\Theta^t), v_2(\Theta^{t+1})) \geq R(v_1(\Theta^t), v_1(\Theta^t)) \tag{6.9}$$

By proving Theorem 3 as described in Appendix A, we show that minimizing error criterion (6.8) is equivalent (within a scaling factor) to maximizing the auxiliary function. In combination with the previous Theorems, this confirms that a network trained using error criterion (6.8) and targets defined by Theorem 2 will increase the auxiliary function. This in turn means that the global probability of the correct model will be increased. In practice the change of the error measure on a cross-validation set is used to guide the training schedule of the MLP, e.g., for deciding the learning rate and the stopping point.

6.2.6 Summary and Discussion

Like the EM algorithm, REMAP training consists of two major steps: estimation (which in this case is estimating new targets for the MLP), and maximization (which here consists of adapting the MLP weights to maximize performance on the new set of targets). Here we have proved three theorems that together show that each iteration of REMAP training increases the posterior probability of the training sentence. It is assumed that the training set is a good sample of the overall input space, and cross-validation techniques are used to check that we have not over-fit to the training data, e.g., by computing the change of the posterior probability on an independent set after every iteration of the REMAP algorithm. In principle, REMAP should ultimately provide improved recognition accuracy for practical systems. However, as with all other gradient-based optimization techniques, we will be vulnerable to potential difficulties with local minima.

6.3 REMAP Training

6.3.1 Introduction

Since it is now well-known (Bourlard & Morgan 1994) how to train an MLP to lead to good estimates of posterior probabilities (whether the MLP targets are "1-from-K" binary vector or themselves estimates of posterior probabilities), the remaining problem is to find an efficient algorithm to express $P(q_\ell^n|X, q_k^{n-1}, M)$ in terms of $P(q_\ell^n|X_{n-c}^{n+d}, q_k^{n-1})$ so that the next iteration's targets can be found, and also how to estimate $P(q_k^{n-1}|X, M)$ in order to select the previous state in an adequate way. Note that from now on, to simplify notation, and only when there is no risk of confusion, we will drop the indices of M and X, keeping in mind that M will represent either a specific model associated with a specific training sentence X or one of the many possible hypothesized models during recognition of X.

6.3.2 Target Estimation

In this section we describe dynamic programming procedures for efficient estimation of the MLP targets. By simple statistical rules the desired targets may be expressed as:

$$P(q_\ell^n|X, q_k^{n-1}, M) = \frac{p(X, q_k^{n-1}, q_\ell^n, M)}{p(X, q_k^{n-1}, M)} \tag{6.10}$$

$$= \frac{p(X, q_k^{n-1}, q_\ell^n, M)}{\sum_\ell \ p(X, q_k^{n-1}, q_\ell^n, M)} \tag{6.11}$$

in which

$$p(X, q_k^{n-1}, q_\ell^n, M) = \sum_i p(X, \Gamma_i, q_k^{n-1}, q_\ell^n, M) \tag{6.12}$$

is equal to the sum of the probabilities of all possible paths Γ_i in a particular M visiting q_k at time $n-1$ and q_ℓ at time n. In a similar way, the denominator represents the sum of the probabilities of all possible paths in M visiting q_k at time $n-1$. Since we cannot afford to compute and memorize all possible paths in (6.12), to compute $P(q_\ell^n|X, q_k^{n-1}, M)$ we need to find efficient recursions in terms of local probabilities $P(q_\ell^n|X_{n-c}^{n+d}, q_k^{n-1})$ generated by the MLP. Denoting

- $\beta_n(k, \ell|M) = p(X_n^N, q_\ell^n|X_1^{n-1}, q_k^{n-1}, M)$ as the "backward" probability, defined as the probability of observing the rest of the sequence and starting from state q_ℓ at time

n given the previously observed X_1^{n-1}, and given state occupancy of class q_k at time $n-1$.

$\beta_n(k,\ell|M)$ is explicitly written with a conditional on M in the parentheses to remind us that the backward recurrences will be run through a specific Markov model M with a specific topology.

Backward Recursion

By using simple statistical rules, we have:

$$\begin{aligned}
\beta_n(j,k|M) &= p(X_n^N, q_k^n|X_1^{n-1}, q_j^{n-1}, M) \\
&= \sum_\ell p(x_n, X_{n+1}^N, q_k^n, q_\ell^{n+1}|X_1^{n-1}, q_j^{n-1}, M) \\
&= \sum_\ell p(X_{n+1}^N, q_\ell^{n+1}|X_1^n, q_j^{n-1}, q_k^n, M)\; p(x_n, q_k^n|X_1^{n-1}, q_j^{n-1}, M) \\
&\qquad \text{assuming 1st order HMMs :} \\
&\simeq \sum_\ell \beta_{n+1}(k,\ell|M)\; P(q_k^n|X_1^n, q_j^{n-1}, M)\; p(x_n|X_1^{n-1}, q_j^{n-1}, M) \quad (6.13)
\end{aligned}$$

As with standard HMMs, the conditional of "local" probabilities $P(q_\ell^{n+1}|X_1^{n+1}, q_k^n, M)$ will be restricted to a contextual acoustic window X_{n-c}^{n+d} and the local posterior (as generated by the neural network) will be assumed independent of M, leading to the following backward recursion:

$$\beta_n(j,k|M) \;=\; p(q_k^n|X_{n-c}^{n+d}, q_j^{n-1})\; c_n(j) \sum_\ell \beta_{n+1}(k,\ell|M) \qquad (6.14)$$

where $c_n(j)$ denotes $p(x_n|X_1^{n-1}, q_j^{n-1}, M)$.

Note that the last simplification, i.e., $P(q_k^n|X_1^n, q_j^{n-1}) \simeq p(q_k^n|X_{n-c}^{n+d}, q_j^{n-1})$ is not necessary in the case of a recurrent net for estimation of the conditional transition probabilities. However, in practice it might not make a difference if an MLP is used to estimate the conditional transition probabilities, i.e., to do the last simplification. Initialization of this backward recursion can be done according to

$$\beta_{N+1}(L,F|M) = p(X_{N+1}^N, q_F^{N+1}|X_1^N, q_L^N, M) = 1 \qquad (6.15)$$

in which q_F is the (non-emitting) final state and q_L represents any last emitting HMM state (associated with M, as imposed by the conditional).

MLP Output Targets Update

Equation (6.10) may now be expressed in terms of the backward recursion:

$$
\begin{aligned}
P(q_k^n|X, q_j^{n-1}, M) &= \frac{p(X, q_j^{n-1}, q_k^n|M)}{p(X, q_j^{n-1}|M)} \\
& \quad \text{dividing numerator and denominator by } P(X_1^{n-1}, q_j^{n-1})|M) \\
&= \frac{\beta_n(j,k|M)}{\sum_\ell \beta_n(j,\ell|M)} \\
& \quad \text{following (6.13)} \\
&= \frac{\sum_\ell \beta_{n+1}(k,\ell|M)\ P(q_k^n|X_{n-c}^{n+d}, q_j^{n-1})\ c_n(j|M)}{\sum_h \sum_\ell\ \beta_{n+1}(h,\ell|M)\ P(q_h^n|X_{n-c}^{n+d}, q_j^{n-1})\ c_n(j|M)} \\
&= \frac{\sum_\ell\ \beta_{n+1}(k,\ell|M)\ P(q_k^n|X_{n-c}^{n+d}, q_j^{n-1})}{\sum_h \sum_\ell\ \beta_{n+1}(h,\ell|M)\ P(q_h^n|X_{n-c}^{n+d}, q_j^{n-1})} \qquad (6.16)
\end{aligned}
$$

This final form of the equation shows that the probabilities required to determine MLP targets can be obtained from the previous MLP outputs and the beta recursions alone.

6.3.3 Forward Recursion

In order to efficiently calculate the posterior probability $P(q_k^{n-1}|X, M)$ we define a forward recursion[6]. We start with some definitions:

$$\alpha_n(k|M_j) = P(q_k^n|X_1^n, M_j, \Theta) \qquad (6.17)$$

i.e., the probability of being at q_k at time n given the acoustic vectors from the beginning of the utterance until time n. Also

$$\alpha_1(k|M_j) = P(q_k^1|x_1, M, \Theta) \qquad (6.18)$$

Now for the recursion step:

$$
\begin{aligned}
\alpha_{n+1}(\ell|M_j) &= \sum_{\Gamma_{n+1}} P(\Gamma_{n+1}, q_\ell^{n+1}|X_1^{n+1}, M, \Theta) \qquad (6.19) \\
&= \sum_{\Gamma_n} P(\Gamma_n|X_1^{n+1}, M, \Theta) P(q_\ell^{n+1}|X_1^{n+1}, \Gamma_n, M, \Theta) \\
& \quad \text{assuming causality}: \ P(\Gamma_n|X_1^{n+1}, M, \Theta) = P(\Gamma_n|X_1^n, M, \Theta)), \text{ denoting} \\
& \quad \text{the ith state of } \Gamma_n \text{ by } k, \text{ and assuming 1st order Markov model} \\
&= \sum_k \alpha_i(k|M_j) P(q_\ell^{n+1}|x_{n+1}, q_k^n, M, \Theta)
\end{aligned}
$$

[6]The forward recursion as described here is used at training time, i.e., when we know the correct model.

where the transition from q_k to q_ℓ is legal in model M_j. When the local probabilities are "tied" across models the recursion becomes:

$$\alpha_{n+1}(\ell|M_j) = \sum_k \alpha_i(k|M_j)P(q_\ell^{n+1}|x_{n+1}, q_k^n, \Theta) \tag{6.20}$$

6.3.4 Estimating the Previous State Distribution

We can compute $P(q_k^{n-1}|X, \Theta^t)$, the posterior probability of being in class q_k at time $n-1$ used in the training of the MLP as specified below, according to the following:

$$P(q_k^{n-1}|X, M, \Theta^t) = \frac{P(q_k^{n-1}, X|M, \Theta^t)}{P(X|M, \Theta^t)} \tag{6.21}$$

dividing numerator and denominator by $P(X_1^{n-1}|M, \Theta)$

$$= \frac{\alpha_{n-1}(k|M)\sum_\ell \beta_n(k, l|M)}{\sum_k[\alpha_{n-1}(k|M)\sum_\ell \beta_n(k, l|M)]} \tag{6.22}$$

6.3.5 REMAP Training Algorithm

The general scheme of the MAP Forward-Backward training of hybrid HMM/MLP systems can be summarized as follow:

1. Start from some initial net providing $P(q_\ell^n|X_{n-c}^{n+d}, q_k^{n-1}, \Theta^t)$, $t = 0$, for all possible (k, ℓ)-pairs[7].

2. Run backward recurrences to compute MLP targets $P(q_\ell^n|X_j, M_j, q_k^{n-1}, \Theta^t)$ according to (6.16), for all training sentences X_j associated with HMM M_j, for all possible (k, ℓ) state transition pairs in M_j and for all x_n, $n = 1, \ldots, N$ in X_j (see next point). Also as part of the forward and backward recurrences we compute $P(q_k^{n-1}|X, \Theta^t)$ (6.21), the posterior probability of being in class q_k at time $n-1$, conditioned on the acoustic vector sequence X, to be used in the training of the MLP as specified below.

3. For every x_n (or X_{n-c}^{n+d}) in the training set choose q_k^{n-1} according to $P(q_k^{n-1}|X, \Theta^t)$, train the MLP to minimize the relative entropy between the network outputs and targets which equal to $P(q_\ell^n|X, q_k^{n-1}, \Theta^t)$. (See Appendix A for a more theoretical explanation.) This provides us with a new set of parameters Θ^t, for $t = t + 1$.

4. Iterate from 2 until convergence.

[7]This can be done, for instance, by training up such a net from a hand-labeled database like TIMIT or from some initial forward-backward estimator of equivalent local probabilities (usually referred to as "gamma" probabilities in the Baum-Welch procedure).

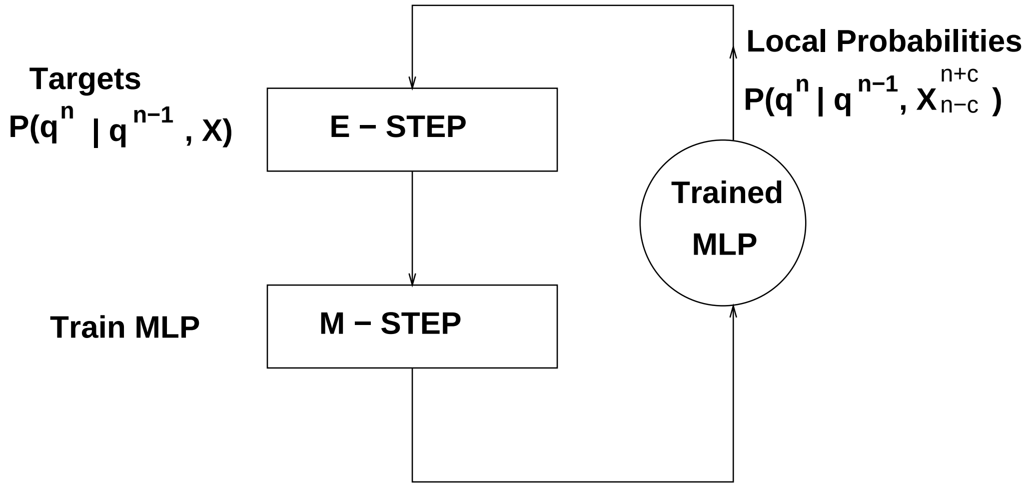

Figure 6.1: An illustration of REMAP

This procedure is composed of two steps: an Estimation (E) step, corresponding to step 2 above, and a Maximization (M) step, corresponding to step 3 above. In this regards, it is reminiscent of the Estimation-Maximization (EM) algorithm as discussed in (Dempster *et al.* 1977). The algorithm is illustrated in Figure 6.1. However, in the standard EM algorithm, the M step involves the actual maximization of the likelihood function. In a related approach, usually referred to as the Generalized EM (GEM) algorithm, the M step does not actually maximize the likelihood, but simply increases it (by using, e.g., a gradient procedure). Similarly, REMAP increases the global posterior function during the M step (in the direction of targets that actually maximize that global function), rather than actually maximizing it. Recently, a similar approach was suggested for mapping input sequences to output sequences (Bengio & Frasconi 1995).

6.3.6 Complexity Issues

It is important the computational cost of REMAP, particularly in comparison to the standard approach of training hybrid HMM/MLP systems (Bourlard & Morgan 1994). The computation of the MLP targets and the distribution of the previous state is done by running the forward and backward recurrences as described above. Denoting the number of states in a particular model M_i by K and the number of frames in a particular training sentence associated with M_i by N, the complexity of running the recurrences is $O(KN)$. This is explained by the fact that the number of operations per state per frame is a constant, assuming a bounded branching factor of the directed graph that represents the topology of[8] M_i. This is the same computational cost as running a forced alignment (Viterbi) to get targets for the MLP training as done in the standard HMM/MLP systems. The difference, though, is that to obtain the conditional transition probabilities that are the input for the recurrences, we need to perform more computations in the REMAP case. Specifically, we have to present every acoustic frame once for every possible previous state[9]. Hence, we have to do K feed-forward runs for every acoustic frame. This is in contrast to the standard HMM/MLP system training, which requires presenting each acoustic frame only once.

During training, for each acoustic frame, the previous state is selected by random according to the previous state distribution as computed by the forward and backward recurrences (in the current implementation). Thus, due to the extra input which represents the previous state, there are up to K times more training patterns. In practice most of the previous states have very low probability (< 0.05) and can be ignored, resulting in only two to three times more patterns. Therefore, the overall training time is two to three times longer than the standard HMM/MLP training, which has the standard back-propagation complexity of $O(T\mathcal{W})$, where T is the number of acoustic frames, and $\mathcal{W}$ is the number of the MLP weights.

6.4 The Role of the Language Model

So far, in the description of the suggested training algorithm, we have omitted the language model. In this section we describe some initial ideas about incorporating the language model into the REMAP framework. Current techniques of speech modeling assume a separation between the language model parameters and the acoustic model parameters as described in Section 3.1. Usually the language model estimates prior probabilities of sentences by computing n-grams (the probability of a word given the previous (n-1) words), by counting the relative frequencies in the training set or in a large text corpus. Acoustic models use a separate set of parameters, which are estimated independently from the language model parameters. In the posterior-based framework proposed here, the posterior probability of the sentence should be estimated given *both* the acoustics and high-level knowledge such as a language model (see Chapter 2). Furthermore, at training time we

[8]This is a reasonable assumption given that the transitions are strictly from left to right in our models.
[9]This computation cost could be reduced by pruning based on the probability of each previous state

want to optimize the measure that we use in recognition. Therefore, it would be desirable to maximize the posterior probability of the correct model given both the acoustics and the language model. However, the training algorithm that is proposed in this study, REMAP, increases the posterior probability of the correct model based only on acoustic information.

An alternative to the separation assumption is a unified model with one parameter set, e.g., a neural network for isolated word recognition, with one output unit for each possible word. It is not clear however, how to avoid the separation assumption in modeling continuous speech recognition, given the unavoidable hierarchical modeling. In hierarchical modeling the larger units (sentences) are built from smaller units (words and phone-like units) that are shared across the larger representation. Thus, there is usually one model for the relation between phones and acoustic vectors (e.g., emission and transition probabilities), one for the relation between phones and words (e.g., pronunciation models), and one for the relation between words and sentences (e.g., language model). Consequently, instead of directly estimating $P(M|X, L, \Theta)$, we decompose it into terms which can be estimated separately. Currently, the model M represents a sequence of words, i.e., there is one-to-one mapping between sentences and models[10]. Thus, going one level down in the modeling hierarchy, the model may be described in terms of state sequences:

$$P(M|X, L, \Theta) = \sum_{\Gamma_i} P(\Gamma_i|X, L, \Theta) P(M|\Gamma_i, X, L, \Theta) \tag{6.23}$$

Although acoustic information such as fundamental frequency, silences, and energy levels, might help to estimate the word sequence from the state sequence, our algorithm currently does not utilize it. Thus, assuming that given the state sequence and the language model, the dependence on the acoustic sequence can be dropped, we get:

$$P(M|X, L, \Theta) \approx \sum_{\Gamma_i} P(\Gamma_i|X, L, \Theta) P(M|\Gamma_i, L, \Theta) \tag{6.24}$$

Using Bayes' rule, we also have:

$$P(M|X, \Theta, L) \approx \sum_{\Gamma_i} P(\Gamma_i|X, \Theta, L) \frac{P(\Gamma_i|M, L, \Theta) P(M|L, \Theta)}{P(\Gamma_i|L, \Theta)}$$

Approximating that the effect of the language model can be ignored in the acoustic term[11],

---

[10] In contrast to *speech understanding* tasks where the model may represent an action (e.g., dialing a particular phone number) that can be expressed by several sentences.

[11] This approximation is especially inaccurate at word boundaries.

i.e., $P(\Gamma_i|X,L,\Theta) \approx P(\Gamma_i|X,\Theta)$, yields:

$$P(M|X,L,\Theta) \approx \sum_{\Gamma_i} P(\Gamma_i|X,\Theta)\frac{P(\Gamma_i|M,L,\Theta)P(M|L,\Theta)}{P(\Gamma_i|L,\Theta)} \tag{6.25}$$

in which $P(\Gamma_i|X,\Theta)$ is computed as described in Chapter 5. $P(M|L,\Theta)$ can be assumed independent of the acoustic model parameters and can be estimated using standard language modeling techniques. In principle $P(\Gamma_i|M,L,\Theta)$ and $P(\Gamma_i|L,\Theta)$ can be estimated during training by dynamic programming techniques similar to our α and β recurrences (Bourlard *et al.* 1994), and the ratio of these two terms represents the additional state transition information that is gained by knowing the specific word sequence.

Chapter 7

Experimental Results with REMAP

7.1 Isolated Speech Experiments

In this section we report on experiments with isolated speech, where recognition was based on acoustic information. The initial task was isolated speech recognition task on the Digits+ corpus in use at ICSI, which is a subset of a larger database recorded over a clean telephone line at Bellcore. It is composed of 200 speakers saying the words "zero" through "nine", "oh", "no", and "yes". The additive noise in these experiments was automotive sound that was recorded over a cellular telephone. Noise was randomly selected from this source and then added to the clean speech waveforms (10 dB S/N ratio). In order to better utilize the data we used a jackknife procedure (Efron 1982). For each of four experiments, three fourths of the data was used for training and cross-validation, and one fourth was used for testing. Specifically, in each experiment there were 1720 utterances for training, 230 for cross-validation and 650 (from 50 speakers) for testing. All nets had 214 inputs: 153 inputs for the acoustic features, and 61 to represent the previous state (one unit for every possible previous state). The acoustic features were combined from 9 frames with 17 features each (RASTA-PLP8 + delta features + delta log gain) computed with an analysis window of 25 ms computed every 12.5 ms (overlapping windows) and the sampling rate was 8 Khz. The nets had 200 hidden units and 61 outputs. All the initial weights for the nets were from a net that was trained on the NTIMIT database. The combined

System	cut 1	cut 2	cut 3	cut 4	Overall Error	Avg. Posterior
Baseline Hybrid	3.5%	2.3%	4.4%	3.4%	3.4%	-
DHMM, pre-REMAP	2.5%	2.6%	2.9%	2.5%	2.7%	0.1269
1 REMAP iteration	2.8%	2.0%	2.5%	2.8%	2.5%	0.1731
2 REMAP iterations	2.9%	2.0%	2.5%	2.6%	2.5%	0.1773

Table 7.1: Training and testing on noisy isolated digits.

results for all four cuts are summarized in Table 7.1. Note that the row entitled "Baseline Hybrid" refers to an ANN trained on targets of 1's and 0's that were obtained from a forced Viterbi procedure by our standard HMM/ANN system as described in (Bourlard & Morgan 1994); the row entitled "DHMM, pre-REMAP" means a Discriminant HMM using the same training approach, with hard targets determined by the first system, and additional inputs to represent the previous state. The rightmost column gives the average probability of the correct model over all test words as determined during recognition. The recognition rate after the first and second iterations of REMAP is significantly better (at $p < 0.05$ level) than the baseline hybrid system. Although the contribution of the REMAP step is small for this task, The overall improvement including the effect of using the transition-based, posterior framework as done in the Discriminant HMM, is significant. In figure 7.1 we illustrate the effect of REMAP iterations of the probability of transition, i.e., changing state for every frame in the utterance "one." Specifically, we calculated the probability that the current state is different from the previous state. As a result of REMAP iterations we get smoother transition probabilities as desired.

7.2 Continuous Speech Experiments

The next step was to test whether this improved performance can also be obtained with continuous speech. For this purpose we chose the Numbers'93 corpus. It is a continuous-speech database collected by CSLU at the Oregon Graduate Institute. It consists of numbers spoken naturally over telephone lines on the public-switched network (Cole *et al.* 1994). The Numbers'93 database consists of 2167 speech files of spoken numbers produced by 1132 callers. We used 877 of these utterances for training and 657 for cross-validation and testing (200 for cross-validation). There are 36 words in the vocabulary, namely *zero, oh, 1,*

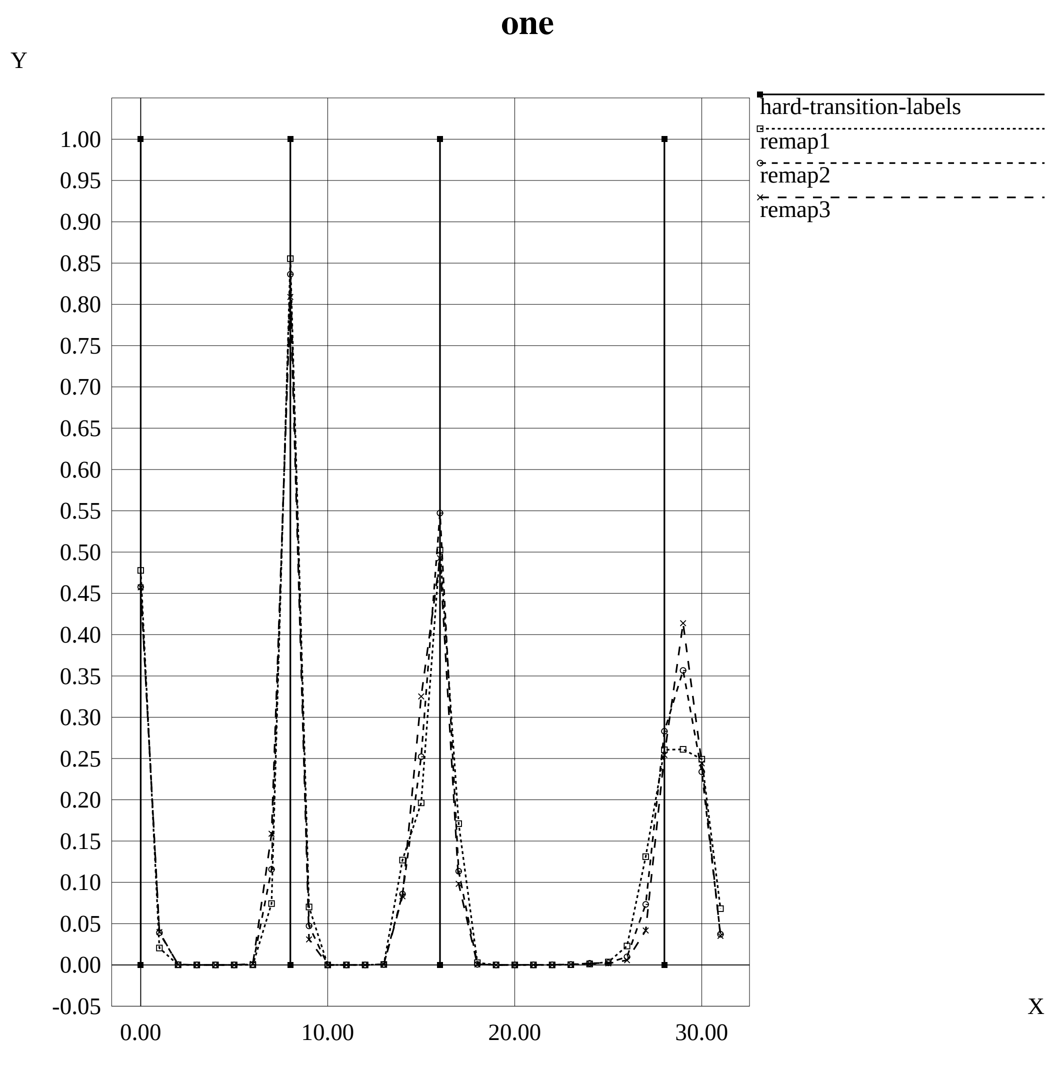

Figure 7.1: The probability of a transition (changing state) for every frame in the utterance "one." The Y-axis represents the probability of transition, and the X-axis the time step

System	Error Rate
DHMM, pre-REMAP	14.9%
1 REMAP iteration	13.6%
2 REMAP iterations	13.2%

Table 7.2: Training and testing on continuous numbers, no syntax, no durational models.

2, 3,...,20, 30, 40, 50,...,100, 1000, a, and, dash, hyphen, and *double*. As before, all the nets have 214 inputs: 153 inputs for the acoustic features, and 61 to represent the previous state (one unit for every possible previous state). All the initial weights for the nets were from a net that was trained on the NTIMIT database. The acoustic features were combined from 9 frames with 17 features each (RASTA-PLP8 + delta features + delta log gain) computed with an analysis window of 25 ms computed every 12.5 ms (overlapping windows) and the sampling rate was 8 Khz. The nets have 200 hidden units and 61 outputs. Our results are summarized in Table 7.2. Note that the row entitled "DHMM, pre-REMAP" means a Discriminant HMM using our standard training approach, with hard targets determined by the first system, and additional inputs to represent the previous state.

The improvement in the recognition rate as a result of REMAP iterations was significant at $p < 0.05$. However, all the experiments were done using acoustic information alone. Using our (baseline) hybrid system under equal conditions, i.e., no duration information and no language information, yielded 31.6% word error; adding the duration information back reduced the error rate to 12.4%.

7.3 Analysis and Discussion

7.3.1 Invalid State Sequences

The improvement in the recognition performance as a result of REMAP training was significant for the continuous speech experiment and not significant for the isolated speech experiment. Most of the improvement in the isolated speech task came from using the posterior-based model, DHMM. This is an intriguing result, especially given the significant increase in the posterior probabilities of the correct models in the DIGITS+ experiment. One explanation for this might be related to phone sequences that do not represent any valid sentence.

Essentially, REMAP discriminates between phone sequences (through the state sequences). In Section 5.3 we show that the MAP constraint is met by the DHMM model, i.e., the sum of all posterior probabilities of all state sequences is 1. However, some of these state sequences[1] are not valid in any particular model. They can be viewed as part of a big "garbage model" that includes all invalid state sequences. Further, the sum of the posterior probability of this "garbage model" and the posterior probabilities of all valid models is one. The question is how big is the share of the "garbage model" ? This share could be evaluated for the DIGITS+ case (given that we only have 13 legal models). As it turns out, the average posterior probability of the "garbage model" before the REMAP iterations is 0.87, and REMAP reduced it to 0.82. Therefore, it might be the case that most of the increase in the average posterior probability of the correct models was at the expense of the "garbage model" and not at the expense of other valid rival models. We tested this hypothesis by measuring the arithmetic average of the ratio of the posterior probability of the correct model and the sum of the posterior probabilities of all valid models, before and after REMAP. Before REMAP this ratio was 0.971, and REMAP increased it to 0.975. Explicitly, the average posterior probability of the correct models before REMAP was 0.1269 and the posterior probability of all other incorrect yet feasible models before REMAP sum to 0.00379, and REMAP increased them to 0.1773 and 0.00455 respectively. Therefore, REMAP only slightly increased the posterior probability of the correct model relative to the other valid models. In other words, for this task the model that had lower probabilities were primarily "illegal" models. This might be more useful in a word spotting task, where illegal sequences may be input.

The same measures could not be calculated for the NUMBERS'93 task given the very large number of valid models (continuous speech). However, the share of the probability mass dedicated to the "garbage model" may not be very big. The reason is that, given more valid models, more state sequences are encompassed. Therefore, there are fewer state sequences left for the "garbage model." This might explain the significant improvement in the continuous speech experiment. Hopefully, for large tasks (in terms of vocabulary and perplexity) the valid models will cover more possible sound sequences, resulting in a less influential "garbage model."

[1]Each state sequence represents one phone sequence, as each state belongs to a particular phone.

7.3.2 Hard Targets

Number of Transitions

In chapter 5 we observed that for a phone classification task training DHMM with "hard" targets resulted in worse performance than for the baseline system. In contrast, on both the DIGITS+ and the NUMBERS'93 experiments, we got better results with DHMM trained on "hard" targets than with the baseline system under equivalent training and testing conditions. One possible explanation is related to the number of phone transitions that the network has to learn. In the DIGITS+ corpus and the NUMBERS'93 task the number of possible transitions is small (less than a hundred) , in contrast to more than 1000 possible transitions in the TIMIT case[2]. Therefore, it might be possible to learn a small number of transitions with "hard" targets, but this approach may not be scalable to a task with a large number of transitions, or at the very least might require significantly more data for training.

Initialization

In order to start the REMAP iteration there is a need for a trained MLP to supply the initial conditional transition probabilities for the forward and backward recursions. In these experiments, these probabilities were estimated using a network trained on "hard targets." It might be that this initialization limits the training to networks that are too similar to the initial network. One way try to overcome this potential limitation might be to add noise (analogous to simulated annealing techniques) to the initial MLP weights. Alternatively we can blur the "hard" targets, e.g., by adding a Gaussian centered around each target. Both of these approaches might result in initial transition probabilities that are smoother than the ones that are estimated by the MLP trained on "hard targets."

[2]The exact number of possible transitions are function of the phone set in use

Chapter 8

Conclusions and Future Work

8.1 Conclusions

This thesis introduced a framework for training and modeling continuous speech recognition systems based on the theoretically optimal MAP criterion. In contrast, most current state-of-the-art systems are trained according to other criteria such as Maximum Likelihood (ML). Our proposed HMM/ANN paradigm is based on re-estimating ANN targets and weights to guarantee an increase in the posterior probability of the correct model (sentence). We have described the theory behind the new framework, included a convergence proof for the training algorithm, and reported on experimental results that support the proposed paradigm.

Our studies have shown that explicitly modeling transitions between speech units can improve recognition performance. Specifically, we have shown that accurate boundary information between phones can improve recognition performance significantly. We have studied a transition-based model that uses local transition probabilities (i.e., the posterior probability of the current state given the current acoustic vector and the previous state) to estimate global posterior of sentences. Therefore, it is a true recognition model, and it directly maps from acoustic sequences to sentences, unlike HMMs that model the inverse relation (the likelihood of producing an acoustic sequence)

Early experiments with the proposed model and the Viterbi-based training procedure showed the need for a new training algorithm. Specifically, the Viterbi-based training (when applied to the transition-based model) suffered from poor transition detection and a mismatch between the input space distribution during training and recognition. These reasons, combined with the goal of training HMM/ANN systems based on the theoretically optimal MAP criterion, led to a new discriminant training algorithm, REMAP.

REMAP is a method to estimate parameters and train systems that classify sequences according to the MAP criterion. This can be used in a new form of hybrid HMM/MLP that, in addition to the advantages of standard HMM/MLP hybrids, uses "full" posterior probabilities for training and recognition. We still use neural nets (in our case MLPs, though recurrent nets or TDNNs could be used) to estimate local posterior probabilities (conditional transition probabilities), but our nets are trained with probabilistic targets that are themselves estimates of local posterior probabilities (conditioned on the acoustic data and the previous state).

There is now a way, similar in spirit to the forward-backward recursions of the Baum-Welch algorithm, of estimating these optimal targets given a previously trained neural network. Additionally, we have a convergence proof that guarantees iterative increase of the global posterior probability. This method is valid for any hybrid HMM/ANN system but, in this thesis, was developed in the framework of "Discriminant HMMs" using conditional transition probabilities.

While our gains with this approach are still small, in many ways the experimental results support the proposed framework. Experimenting with an isolated speech recognition task resulted in an increase in the estimates of the posterior probabilities of the correct sentences after training. Extending these experiments to a continuous speech recognition task achieved a significant decrease in error rate in comparison to a baseline system. While learning a larger number of transitions may be difficult, it still may be the case that larger tasks will be better for REMAP, since less probability mass will be associated with "illegal" state sequences.

8.2 Future Work

8.2.1 Incorporating Language Information

This work and other current approaches to speech recognition modeling assume independent models for language and acoustic information. Usually the language model estimates prior probabilities of sentences by computing n-grams (the probability of a word given the previous (n-1) words), by counting the relative frequencies in the training test or in a large text corpus[1]. Acoustic models use a separate set of parameters, which are estimated independently from the language model parameters.

In the posterior-based framework proposed here, we should estimate the posterior probability of the sentence given *both* the acoustics and high-level knowledge such as a language model (see Chapter 2). Furthermore, during training we want to optimize the measure that we use in recognition. Therefore, we want to maximize the posterior probability of the correct model given both the acoustics and the language model. However, the training algorithm that is proposed in this study, REMAP, increases the posterior probability of the correct model based only on acoustic information.

One possible solution is described in Section 6.4. Specifically, Equation (6.25) describes a way to incorporate the language model into the estimation of the posterior probability of a particular model. So far, we have not not experimented with this approach. Another possible solution is to estimate "local" probabilities, e.g., local conditional posterior probabilities in this work, based on both the local acoustic and on language information. Specifically, we could take into account whether the transition between two phones occurs in the middle of a word or between two words, e.g., one phone is the last phone of the current word and the second is the first in the following word. A possible implementation of this idea is to add an additional input to the MLP described in Figure 5.2 to represent the words of the source and target phones. In the case of large vocabularies, the words can be clustered into classes as has been done in class grammar techniques. Unfortunately, this increases the dimensionality of the input space, resulting in a need for more training data and more computation.

Several researchers have worked on learning pronunciation models using both the linguistic and the acoustic training data (Stolcke & Omohundro 1993b; Stolcke & Omohundro 1993a; Wooters 1993). Extending this line of work towards learning probabilistic grammars based on both the linguistic and the acoustic training data seems like a direction worth pursuing. Eventually we might learn the models themselves (not just the parameters of the models) from the data. Complementing this data-driven direction as much domain-specific knowledge should be incorporated as possible. For instance, new experiments are now being done at ICSI using enforced minimum duration constraints with REMAP.

[1] Generally, smoothing techniques are used to get a better estimate of rare or non-occurring word transitions, e.g., averaging the bigram probabilities with the product of the corresponding unigram probabilities.

8.2.2 Extensions

A Smaller Phone Set

In section 7.3 we describe a potential problem regarding invalid state sequences. We showed that in a small isolated speech experiment, around 80% of the probability mass was dedicated to a "garbage model", which contains all phone sequences that are not part of any valid model. Furthermore, our speculation was that in the continuous speech task, this problem is not so severe, given the large number of legal models which encompass more phone sequences. Additionally, a potential partial solution is to use a smaller phone set, such that the phone sequences "covered" by the legal models will have a larger portion from all phone sequences.

Recurrent Nets

In all the mathematical development through this thesis we rely on the assumption that to estimate conditional transition probabilities, we can represent both the acoustic past and the previous state sequence by the most recent previous state and by a temporal window of several acoustic frames into the past. This simplifying assumption can be relaxed by replacing the MLP used in this study by a recurrent neural network (Robinson 1994). A recurrent neural network encodes the past in its state units. Hence, the estimation of the transition probabilities is conditioned on the captured history of the utterance.

Negative Training

In section 5.4.2 we describe a potential problem regarding a mismatch between the input space distribution during training and recognition. During training the net is only presented with "correct" pairs of acoustic vectors and the correct previous state, while in recognition the net is expected to generalize to all possible combinations of acoustic vectors and previous states. A possible solution might be to do explicit negative training. Specifically, we would train the network with all possible combinations of acoustic vectors and previous states, but for the "incorrect" previous states we would present uniform targets, i.e., all targets would have the same value. We can estimate the posterior probability of each previous state, by running the alpha and beta recursions in a fully-connected phone model. According to this estimated distribution of the previous state we would select the pairs of acoustic vectors and previous states as described in Section 6.3. In other work, Lyon and Yaeger recently showed that explicit training on negative examples can improve recognition for on-line hand-recognition with ANNs (Lyon & Yaeger 1996).

Modeling Extensions

In this study although we could increase the complexity we have used a simple 1st order Markov model. The extension to an M-th order model is described in (Bourlard *et al.* 1994). However, the increased complexity of the model requires more training data to reliably estimate the model parameters. Further, a more complex model has no guarantee to work better. Other modeling extensions might be to encode speaker-dependent information such as gender, speaking rate, and accent as extra inputs to the Discriminant HMM. Another possible extension is to apply REMAP to perceptually-motivated models such as SPAM (Morgan *et al.* 1994; Morgan *et al.* 1995).

8.3 Epilog

In experimental science such as speech recognition, a division between *data description* and *data modeling* can be drawn (Cohen 1995). Data description is taking an existing model and increasing its complexity, adding more exceptions to better fit the observed data; for instance, adding triphones to HMMs. In contrast, data modeling refers to finding models that are a better match to the underlying phenomena, such as a model based on a parameter for vocal tract length. Furthermore, these new models should be a simpler explanation of the observed data, with better generalization capabilities, and not just a more complex version of the existing models.

Currently, in speech recognition, the most popular models are HMMs, with a number of labs and companies using hybrid HMM/ANN in different forms. In this thesis we suggest a different framework than HMMs for speech recognition. The main difference between HMMs and our suggested framework is in the modeling objective. HMMs model a production system, i.e., all the different ways that a given sentence can be realized. Their input is a sentence and their output its acoustic realizations. Thus, a mismatch with observed data could be solved through a more detailed model of the production process, e.g., having different models for males and females. The framework proposed here models a recognition process, i.e., the input is an acoustic utterance and the output is a sentence.

We estimate a fixed number of classes (phones in our implementation) given the context. The context can represent the previous state (as done in this study), the speaking rate of the speaker, the gender of the speaker, the noise level, etc.

The long term usefulness of the proposed framework may be determined on the performance on larger and more challenging speech tasks (such as the Switchboard corpus (NIST 1992)) than tested here and on application of it to other domains that are also sequential in nature such as hand-writing recognition. For REMAP to become a popular approach for speech recognition, two main objectives must be achieved. The first objective is to incorporate more speech specific knowledge into this framework, such as duration constraints, speaking rate, etc. Initial experiments with enforcing minimum duration constraints are beginning to produce promising results. The second objective is to integrate the language information into the framework. We do believe that the framework presented here is a good start.

Appendix A

REMAP Convergence - Theorem Proof

A.1 Theorem 1

Theorem 1:

IF $R(v_1, v_2) \geq R(v_1, v_1)$
THEN $P(M|X, v_2) \geq P(M|X, v_1)$.

In other words, if we can find a new set of probabilities v_2 increasing R, the new set of probabilities will also increase the posterior probability of the model M.

Proof:

$$\begin{aligned}
&\log \frac{P(M|X, v_2)}{P(M|X, v_1)} \\
&= \log \left[\sum_{\Gamma} \frac{P(M, \Gamma|X, v_1)}{P(M|X, v_1)} \frac{P(M, \Gamma|X, v_2)}{P(M, \Gamma|X, v_1)} \right] \\
&\geq \sum_{\Gamma} \frac{P(M, \Gamma|X, v_1)}{P(M|X, v_1)} \log \left[\frac{P(M, \Gamma|X, v_2)}{P(M, \Gamma|X, v_1)} \right] \\
&\text{(because of Jensen}'\text{s inequality and concavity of log function)} \\
&= R(v_1, v_2) - R(v_1, v_1)
\end{aligned}$$

(Note that the random variable used for the Jensen's inequality is a $\frac{P(M,\Gamma|X,v_2)}{P(M,\Gamma|X,v_1)}$ which is a function of the random variable Γ). As a consequence, we have:

$$\log \frac{P(M|X, \upsilon_2)}{P(M|X, \upsilon_1)} \geq R(\upsilon_1, \upsilon_2) - R(\upsilon_1, \upsilon_1) \tag{A.1}$$

which proves the theorem. If a new set of probabilities υ_2 that makes the right-hand side of (A.1) positive can be found, then the model re-estimation algorithm can be guaranteed to increase the posterior probability of the model to $P(M|X, \upsilon_2)$.

A.2 Theorem 2

Theorem 2:

Given υ_1, a fixed set of probabilities that is estimated by an MLP with a fixed set of weights Θ, we show that $R(\upsilon_1, \upsilon_2)$ attains its local maximum value when the conditional transitional probabilities $P_{\upsilon_2}(q_\ell^n|x_n, q_k^{n-1})$ are defined as[1]

$$P_{\upsilon_2}(q_\ell^n|x_n, q_k^{n-1}) = P(q_\ell^n|X, q_k^{n-1}, \upsilon_1(\Theta), M) \tag{A.2}$$

Proof:

We now treat the conditional transitional probabilities $P_{\upsilon_2}(\cdot)$ as the variables for the optimization. Thus, we need to maximize $R(\cdot)$ in the space of transition-probability sets Υ, under the KN constraints

$$\sum_{j=1}^{K} P_{\upsilon_2}(q_j^n|x_n, q_k^{n-1}) = 1, \ \forall k = 1, \ldots, K; \forall n = 1, \ldots, N \tag{A.3}$$

Using Lagrange multipliers $\Lambda = (\lambda_{1,1}, \ldots, \lambda_{1,N}, \ldots, \lambda_{K,1}, \ldots, \lambda_{K,N})^t$, maximization of $R(\cdot)$ as defined in (6.2) under the constraints specified in (A.3) is then equivalent to maximization of

$$R^*(\upsilon_1, \upsilon_2, \Lambda) \ = \ R(\upsilon_1, \upsilon_2) + \sum_{k,n} \lambda_{k,n} \left(1 - \sum_{j=1}^{K} P_{\upsilon 2}(q_j^n|x_n, q_k^{n-1})\right) \tag{A.4}$$

So there are K^2N unknowns that are the conditional transition probabilities in υ_2 and KN unknowns that are the Lagrange multipliers. Fortunately, there are the same number of equations as we compute the partial derivative of $R^*(\cdot)$ relative to each unknown and

[1] Of course, all x_n's in the following should be replaced by X_{n-c}^{n+d} if local contextual input is used, or X_1^n for a recurrent network.

equalize it to zero. Furthermore, it turns out that we can solve each of the K^2N equations described above independently and find solutions that satisfy the KN constraints.

Considering a specific transition (q_k^{n-1}, q_ℓ^n), we then have:

$$\frac{\partial R^*(\cdot)}{\partial \lambda_{k,n}} = 0 \tag{A.5}$$

which returns the constraint (A.3). For the partial derivative of $R^*(\cdot)$ with respect to $P_{\upsilon_2}(\cdot)$, we first use the following decomposition:

$$P(M, \Gamma|X, \upsilon_2) = P(\Gamma|X, \upsilon_2)P(M|\Gamma, X, \upsilon_2) \tag{A.6}$$

According to (5.7), the first factor in (A.6) can be expressed as

$$P(\Gamma|X, \upsilon_2) \;=\; \prod_{n=1}^{N} P_{\upsilon_2}(q_\ell^n|x_n, q_k^{n-1}) \tag{A.7}$$

Also, the second factor in (A.6) can be assumed independent of the conditional transition probabilities (i.e., given a state sequence, the probability of the model does not depend on the transition probabilities), in which case we have:

$$P(M|\Gamma, X, \upsilon_2) = P(M|\Gamma, X) \tag{A.8}$$

Taking partial derivatives, then, the second term in (A.6) has no effect, since it can be assumed to have no dependence on $P_{\upsilon_2}(q_\ell^n|x_n, q_k^{n-1})$ and since it only appears as an additive term once the logarithmic function has been applied.

We then get

$$\begin{aligned} &\frac{\partial R^*(\cdot)}{\partial P_{\upsilon_2}(q_\ell^n|x_n, q_k^{n-1})} \\ &\quad = \frac{1}{P(M|X, \upsilon_1)} \sum_{\Gamma_{k,\ell,n}} \left[P(M, \Gamma_{k,\ell,n}|X, \upsilon_1) \frac{1}{P_{\upsilon_2}(q_\ell^n|x_n, q_k^{n-1})} \right] - \lambda_{k,n} \\ &\quad = 0 \end{aligned} \tag{A.9}$$

where $\Gamma_{k,l,n}$ stands for those paths containing the transition (q_k^{n-1}, q_ℓ^n). Solving (A.9) gives:

$$P_{\upsilon_2}(q_\ell^n|x_n, q_k^{n-1}) \;=\; \frac{1}{\lambda_{k,n}} \frac{\sum_{k,\ell,n} P(M, \Gamma_{k,\ell,n}|X, \upsilon_1)}{P(M|X, \upsilon_1)} \tag{A.10}$$

We now have to find the value (or "a" value) of $\lambda_{k,n}$ that guarantees that the new estimates of $P_{\upsilon_2}(q_\ell^n|x_n, q_k^{n-1})$ will meet the constraint. It is possible to find it without directly solving the set of equations. It is indeed easy to show that:

$$\begin{aligned}
\frac{\sum_{k,\ell,n} P(M,\Gamma_{k,\ell,n}|X,\upsilon_1)}{P(M|X,\upsilon_1)} &= \frac{\sum_{\forall\Gamma} P(M,\Gamma,q_\ell^n,q_k^{n-1}|X,\upsilon_1)}{P(M|X,\upsilon_1)} \\
&= \frac{P(M,q_\ell^n,q_k^{n-1}|X,\upsilon_1)}{P(M|X,\upsilon_1)} \\
&= P(q_\ell^n,q_k^{n-1}|X,M,\upsilon_1) \\
&= P(q_\ell^n|q_k^{n-1},X,M,\upsilon_1)P(q_k^{n-1}|X,M,\upsilon_1)
\end{aligned} \tag{A.11}$$

Since the second factor in (A.11) is a function of k and n only, we can set $\lambda_{k,n}$ to $P(q_k^{n-1}|X,M,\upsilon_1)$ which then gives us, according to (A.10):

$$P_{\upsilon_2}(q_\ell^n|x_n,q_k^{n-1}) = P(q_\ell^n|X,q_k^{n-1},M,\upsilon_1) \tag{A.12}$$

This is a valid solution since the constraint

$$\sum_{\ell=1}^{K} P(q_\ell^n|X,q_k^{n-1},M,\upsilon_1) = 1, \forall k \text{ and } \forall n$$

given in (A.3) is automatically met.[2] In order to verify that we have a local maximum point we have to compute the Hessian matrix. It is easy to see by looking at (A.9) that all the non diagonal elements are zero. In computing the diagonal elements we get

$$\begin{aligned}
&\frac{\partial^2 R^*(\cdot)}{\partial P_{\upsilon_2}^2(q_\ell^n|x_n,q_k^{n-1})} \\
&\quad = -\frac{1}{P(M|X,\upsilon_1)}\sum_{\Gamma_{k,\ell,n}}\left[P(M,\Gamma_{k,\ell,n}|X,\upsilon_1)\frac{1}{P_{\upsilon_2}(q_\ell^n|x_n,q_k^{n-1})}\right]
\end{aligned} \tag{A.13}$$

and it is obvious that for probabilities (i.e., positive numbers) we get negative diagonal elements. Thus, this is a local maximum point. This proves Theorem 2.

A.3 Theorem 3

Theorem 3:

When minimizing the weighted relative entropy criterion (20) with the target set T (which is calculated from a probability set υ_1), the auxiliary function $R(\cdot)$ is maximized. Specifically

[2] We cannot prove that this is a unique solution since most of the equations are nonlinear, but we know this is at least one valid solution.

the new set of probabilities v_2, implemented by the trained MLP satisfies the following:

$$E_e(\Theta^{t+1}) \leq E_e(\Theta^t) \Longrightarrow R(v_1(\Theta^t), v_2(\Theta^{t+1})) \geq R(v_1(\Theta^t), v_1(\Theta^t)) \qquad (A.14)$$

Proof:

$$\begin{aligned}
E_e(\Theta^{t+1}) - E_e(\Theta^t) &= \sum_{n=1}^{N}\sum_{k=1}^{K} P(q_k^{n-1}|X, M, v_1(\Theta^t)) \\
&\qquad \sum_{\ell=1}^{K} P(q_\ell^n|X, q_k^{n-1}, M, v_1(\Theta^t)) \log \frac{g_\ell(x_n, q_k^{n-1}, \Theta^t)}{g_\ell(x_n, q_k^{n-1}, \Theta^{t+1})} \\
&= \sum_{n=1}^{N}\sum_{k=1}^{K}\sum_{\ell=1}^{K} P(q_\ell^n, q_k^{n-1}|X, M, v_1(\Theta^t)) \log \frac{g_\ell(x_n, q_k^{n-1}, \Theta^t)}{g_\ell(x_n, q_k^{n-1}, \Theta^{t+1})}
\end{aligned} \qquad (A.15)$$

Below we show that the change in the auxiliary function $R(\cdot)$ has the same magnitude then E_e, but was with the opposite sign.

$$\begin{aligned}
&R(v_1(\Theta^t), v_2(\Theta^{t+1})) - R(v_1(\Theta^t), v_1(\Theta^t)) \\
&\quad = \sum_{\Gamma} \frac{P(M, \Gamma|X, v_1)}{P(M|X, v_1)} \log \left[\frac{P(M, \Gamma|X, v_2)}{P(M, \Gamma|X, v_1)}\right] \\
&\quad = \sum_{\Gamma} \frac{P(M, \Gamma|X, v_1)}{P(M|X, v_1)} \log \left[\frac{P(M|\Gamma, X, v_2)P(\Gamma|X, v_2)}{P(M|\Gamma, X, v_1)P(\Gamma|X, v_1)}\right]
\end{aligned} \qquad (A.16)$$

Given a state sequence, the probability of the model does not depend on the transition probabilities (v_1 and v_2). As stated in (A.8), we then get:

$$\begin{aligned}
&R(v_1(\Theta^t), v_2(\Theta^{t+1})) - R(v_1(\Theta^t), v_1(\Theta^t)) \\
&\quad = \sum_{\Gamma} \frac{P(M, \Gamma|X, v_1)}{P(M|X, v_1)} \log \left[\frac{P(\Gamma|X, v_2}{P(\Gamma|X, v_1)}\right] \\
&\quad = \sum_{\Gamma} \frac{P(M, \Gamma|X, v_1)}{P(M|X, v_1)} \log \left[\frac{\prod_{n=1}^{N} P(q_\ell^n|x_n, q_k^{n-1}, \Theta_{t+1})}{\prod_{n=1}^{N} P(q_\ell^n|x_n, q_k^{n-1}, \Theta_t)}\right], \quad \text{[as in (A.7)]} \\
&\quad = \sum_{\Gamma} P(\Gamma|M, X, v_1) \log \left[\frac{\prod_{n=1}^{N} P(q_\ell^n|x_n, q_k^{n-1}, \Theta_{t+1})}{\prod_{n=1}^{N} P(q_\ell^n|x_n, q_k^{n-1}, \Theta_t)}\right] \\
&\qquad \text{rearranging the terms in the summation :} \\
&\quad = \sum_{n=1}^{N}\sum_{k=1}^{K}\sum_{\ell=1}^{K}\sum_{\Gamma_{k,\ell,n}} P(\Gamma_{k,\ell,n}|M, X, v_1) \log \frac{g_\ell(x_n, q_k^{n-1}, \Theta^{t+1})}{g_\ell(x_n, q_k^{n-1}, \Theta^t)}
\end{aligned}$$

$$\begin{aligned} &= \sum_{n=1}^{N}\sum_{k=1}^{K}\sum_{\ell=1}^{K}\sum_{\Gamma} P(\Gamma, q_k^{n-1}, q_\ell^n | M, X, v_1(\Theta_t)) \log \frac{g_\ell(x_n, q_k^{n-1}, \Theta^{t+1})}{g_\ell(x_n, q_k^{n-1}, \Theta^t)} \\ &= \sum_{n=1}^{N}\sum_{k=1}^{K}\sum_{\ell=1}^{K} P(q_k^{n-1}, q_\ell^n | M, X, v_1(\Theta_t)) \log \frac{g_\ell(x_n, q_k^{n-1}, \Theta^{t+1})}{g_\ell(x_n, q_k^{n-1}, \Theta^t)} \end{aligned} \tag{A.17}$$

in which $\Gamma_{k,\ell,n}$ stands for those paths containing the transition (q_k^{n-1}, q_ℓ^n).

A closer look at the last equation shows the term that we got for the difference in the auxiliary function $R(\cdot)$ is with an opposite sign and the same magnitude as the difference in E_e in (A.15). Thus, minimizing the cost function E_e (as part of the MLP training) is equivalent to maximizing the auxiliary function $R(\cdot)$. Hence, we have proved Theorem 3, and in fact showed that minimizing the error criterion (6.8) is equivalent (within a scaling factor) to maximizing the auxiliary function.

Bibliography

Bahl, L. R., P. F. Brown, P. V de Souza, & R. L. Mercer. 1986. Maximum mutual information estimation of hidden Markov model parameters. In *Proc. Int. Conf. on Acoust., Speech, and Signal Processing*, 49–52, Tokyo, Japan.

Baker, J. K., 1975. *Stochastic Modeling as a Means of Automatic Speech Recognition*. Carnegie Mellon University dissertation.

Baum, L. E. 1972. An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes. *Inequalities* 3.1–8.

——, & T. Petrie. 1966. Statistical inference for probabilistic functions of finite state Markov chains. *The Annals of Mathematical Statistics* 36.1554–1563.

——, T. Pitrie, G. Soules, & N. Weiss. 1970. A maximization technique occurring in the statistical analysis of probabilistic functions in Markov chains. *The Annals of Mathematical Statistics* 41.164–171.

Bengio, Y., & P. Frasconi. 1995. An input output HMM architecture. In *Advances in Neural Information Processing Systems*, ed. by G. Tesauro, D. Touretzky, & T. Leen, volume 7. Cambridge: MIT press.

——, R. De Mori, G. Flammia, & R. Kompe. 1992. Global optimization of a neural network-hidden Markov model hybrid. *IEEE trans. on Neural Networks* 3.252–258.

Bourlard, H., Y. Konig, & N. Morgan. 1994. REMAP: Recursive estimation and maximization of a posteriori probabilities, application to transition-based connectionist speech recognition. Technical Report TR-94-064, International Computer Science Institute, Berkeley, CA.

——, & N. Morgan. 1994. *Connectionist Speech Recognition - A Hybrid Approach*. Kluwer Academic Publishers.

——, & C. J. Wellekens. 1989a. Links between Markov models and multilayer perceptrons. In *Advances in Neural Information Processing Systems 1*, ed. by D.J. Touretzky, 502–510, San Mateo. Morgan Kaufmann.

——, & ——. 1989b. Speech pattern discrimination and multilayer perceptrons. *Computer, Speech, and Language* 3.1–19.

Bridle, J. S. 1990. Alpha-nets: A recurrent "neural" network architecture with a hidden Markov model interpretation. *Speech Communication* 9.83–92.

Brown, P. F., 1987. *The Acoustic-Modelling Problem in Automatic Speech Recognition*. Pittsburgh, PA: CMU dissertation.

Cohen, J. R., 1995. Informal Communication.

Cole, R. A., M. Fanty, & T. Lander. 1994. Telephone speech corpus development at CSLU. In *Proceedings Int'l Conference on Spoken Language Processing*, Yokohama, Japan.

Cover, T. M., & A. T. Joy. 1991. *Elements of Information Theory*. New York: John Wiley and Sons, Inc.

Dempster, A. P., N. M. Laird, & D. B. Rubin. 1977. Maximum likelihood from incomplete data via the *EM* algorithm. *Journal of the Royal Statistical Society, Series B* 34.1–38.

Deng, L. 1992. A generalized hidden Markov model with state-conditioned trend functions of time for the speech signal. *Signal Processing* 27.65–78.

——, & D.X. Sun. 1994. A statistical approach to automatic speech recognition using the atomic speech units constructed from overlapping articulatory features. *J. Acoustical Society of America* 95.2702–2719.

Digalakis, V. V., 1992. *Segment-Based Stochastic Models of Spectral Dynamics for Continuous Speech Recognition*. Boston University dissertation.

——, J. R. Rohlicek, & M. Ostendorf. 1993. Segment-based stochastic models of spectral dynamics for continuous speech recognition. *IEEE trans. on Speech and Audio Processing* 1.431–442.

Duda, R. O., & P. E. Hart. 1973. *Pattern Classification and Scene Analysis*. John Wiley and Sons, Inc.

Efron, B. 1982. The jackknife, the bootstrap, and other resampling plans. Society for Industrial and Applied Mathematics, Philadelphia, Pa.

Fujimura, O. 1975. Syllable as a unit of speech recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing* ASSP-23.82–87.

Furui, S. 1986a. On the role of spectral transition for speech perception. *J. Acoustical Society of America* 80.1016–1025.

—— 1986b. Speaker independent isolated word recognizer using dynamic features of speech spectrum. *IEEE Trans. on Acoustics, Speech, and Signal Processing* 34.52–59.

Garofolo, J. S., 1988. *Getting Started with the DARPA TIMIT CD-ROM: an Acoustic Phonetic Continuous Speech Database*. National Institute of Standards and Technology (NIST), Gaithersburgh, Maryland.

Ghitza, O., & M. M. Sondhi. 1993. Hidden Markov models with templates as non-stationary states: an application to speech recognition. *Computer Speech and Language* 2.101–119.

Gish, H. 1990. A probabilistic approach to the understanding and training of neural network classifiers. In *Proc. Int. Conf. on Acoust., Speech, and Signal Processing*, 1361–1364, Albuquerque, NM.

Glass, J. R., 1988. *Finding Acoustic Regularities in Speech Applications to Phonetic Recognition*. M.I.T dissertation.

Goldenthal, W. D., 1994. *Statistical Trajectory Models for Phonetic Recognition*. M.I.T dissertation.

HAEB-UMBACH, R., & H. NEY. 1992. Linear discriminant analysis for improved large vocabulary continuous speech recognition. In *Proc. Int. Conf. on Acoust., Speech, and Signal Processing*, volume 1, 13–16, San Francisco, USA.

HERMANSKY, H. 1990. Perceptual linear predictive (PLP) analysis of speech. *J. Acoustical Society of America* 87.

HUANG, X. D., Y. ARIKI, & M. A. JACK. 1990. *Hidden Markov Models For Speech Recognition*. Edinburgh University Press.

JELINEK, F. 1976. Continuous speech recognition by statistical methods. *Proceedings of the IEEE* 64.532–556.

——, & R. L. MERCER. 1980. Interpolated estimation of Markov source parameters from sparse data. In *Pattern Recognition in Practice*, ed. by E. S. Gelsema & L. N. Kanal, 381–397. Amsterdam, The Netherlands: North-Holland Publishing Company.

JUANG, B. H., & L. R. RABINER. 1985. Mixture autoregressive hidden Markov models for speech signals. *IEEE ASSP Magazine* 6.1404–1413.

KATAGIRI, S., C. H. LEE, & JUANG B. H. 1991. New discriminative training algorithms based on the generalized probabilistic decent method. In *Proc. of the IEEE Workshop on Neural Netwroks for Signal Processing*, ed. by B.H. Juang, S.Y. Kung, & C.A. Kamm, 299–308.

KIANG, N. Y. S. 1984. Peripheral neural processing of auditory information. In *In Handbook of physiology - the nervous system, volume III, sensory processes, part 2*, ed. by J. M. Brookhart & V. Mountcastle, 639–674. Bethesda: American Physiological Society.

KONIG, Y., & N. MORGAN. 1993. Supervised and unsupervised clustering of the speaker space for connectionist speech recognition. In *Proc. Int. Conf. on Acoust., Speech, and Signal Processing*, Minneapolis, Minnesota. IEEE.

——, & ——. 1994. Modeling dynamics in connectionist speech recognition - the time index model. In *Proceedings Int'l Conference on Spoken Language Processing*, 1523–1526, Yokohama, Japan.

——, ——, C. Wooters, V. Abrash, M. Cohen, & H. Franco. 1993. Modeling consistency in a speaker independent continuous speech recognition system. In *Advances in Neural Information Processing Systems 5*, ed. by J.S. Hanson & J.D. Cowan znd C.L. Giles, volume 5, San Mateo. Morgan Kaufman.

Lee, K. F. 1989. *Automatic Speech Recognition: The Development of the SPHINX System.* Kluwer Academic Publishers.

Levelt, W. J. M., & L. Wheeldon. 1994. Do speakers have access to a mental syllabary? *Cognition* 50.239–269.

Levinson, S. E. 1985. Structural methods in automatic speech recognition. *Proceedings of the IEEE* 73.1625–1650.

——, L. R. Rabiner, & M. M. Sondhi. 1983a. An introduction to the application of the theory of probabilistic functions on a Markov process to automatic speech recognition. *Bell System Technical Journal* 62.243–272.

——, Lawrence R. Rabiner, & Man Mohan Sondhi. 1983b. An introduction to the application of the theory of probabilistic functions of a Markov process to automatic speech recognition. *Bell Syst. Tech. J.* 62.1035–1074.

Lindblom, B. E. F., & M. Studdert-Kennedy. 1967. On the role of formant transitions in vowel recognition. *J. Acoustical Society of America* 42.830–843.

Liporace, L. A. 1982. Maximum likelihood estimation for multivariate observations of markov sources. *IEEE Trans. on Information Theory* IT-28.729–734.

Lippmann, R. P. 1989. Review of neural networks for speech recognition. *Neural Computation* 1.1–38.

Lubensky, D. M., A. O. Asadi, & J. M. Naik. 1994. Connected digit recognition using connectionist probability estimators and mixture-gaussian densities. In *Proceedings Int'l Conference on Spoken Language Processing*, 295–298, Yokohama, Japan.

Lyon, R. F., & L. S. Yaeger. 1996. On-line hand-printing recognition with neural networks. In *MicroNeuro '96 – Fifth International Conference on Microelectronics for Neural Networks and Fuzzy Systems*, 201–212, Lausanne, Switzerland. IEEE.

MERIALDO, B. 1988. Phonetic recognition using hidden Markov models and maximum mutual information training. In *Proc. Int. Conf. on Acoust., Speech, and Signal Processing*, 111–114, New York.

MORGAN, N., H. BOURLARD, S. GREENBERG, & H. HERMANSKY. 1994. Stochastic perceptual auditory-event-based models for speech recognition. In *Proceedings Int'l Conference on Spoken Language Processing*, 1943–1946, Yokohama, Japan.

——, S. WU, & H. BOURLARD. 1995. Digit recognition with stochastic perceptual speech models. In *Proceedings of the Sixth European Conference on Speech Communication and Technology*, Madrid, Spain.

NADAS, A. 1983. A decision-theortic formulation of a training problem in speech recognition and a comparison of training by unconditional versus conditional maximum likelihood. *IEEE Transactions on Acoustics, Speech, and Signal Processing* 31.814–817.

NIST, 1992. Switchboard Corpus: Recorded Telephone Conversations. National Institute of Standards and Technology Speech Disc 9-1.1 to 9-29.1, Collected by Texas Instruments.

OSTENDORF, M., & S. ROUKOS. 1989. A stochastic segment model for phoneme-based continuous speech recognition. *IEEE ASSP trans.* 37.1857–1869.

PORITZ, A. B., & A. L. RICHTER. 1986. On hidden Markov models in isolated word recognition. In *Proc. Int. Conf. on Acoust., Speech, and Signal Processing*, volume 1 of *Tokyo, Japan*, 705–708.

PRICE, P., W. FISHER, J. BERNSTEIN, & D. PALLET. 1988. The darpa 1000-word resource management database for continuous spee ch recognition. In *Proceedings IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing*, volume 1, 651–654, New York. IEEE.

RABINER, L.R. 1989. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE* 77.257–285.

RENALS, S., N. MORGAN, & H. BOURLARD. 1991. Probability estimation by feed-forward networks in continuous speech recognition. Technical Report TR-91-030, International Computer Science Institute, 1947 Center St., Suite 600, Berkeley, CA 94704.

——, ——, ——, M. Cohen, & H. Franco. 1994. Connectionist probability estimators in HMM speech recognition. *IEEE Trans. on Speech and Audio Processing* 2.161–174.

——, N. Morgan, M. Cohen, H. Franco, & H. Bourlard. 1992. Connectionist probability estimation in the DECIPHER speech recognition system. In *Proc. Int. Conf. on Acoust., Speech, and Signal Processing*, 601–604, San Francisco, California. IEEE.

Richard, M. D., & R. P. Lippmann. 1991. Neural network classifiers estimate bayesian a posteriori probabilities. *Neural Computation* 3.461–483.

Robinson, A. J. 1994. An application of recurrent nets to phone probability estimation. *IEEE transactions on Neural Networks* 5.298–305.

——, L. Almeida, J.-M. Boite, H. Bourlard, F. Fallside, M. Hochberg, D. Kershaw, P. Kohn, Y. Konig, N. Morgan, J. P. Neto, S. Renals, M. Saerens, & C. Wooters. 1993. A neural network based, speaker independent, large vocabulary, continuous speech recognition system: The wernicke project. In *Proceedings European Conf. on Speech Communication and Technology. (EUROSPEECH)*, 1941–1944, Berlin, Germany.

Ruggero, M. 1994. Physiology and coding of sound in the auditory nerve. In *The Mammalian Auditory Pathway: Neurophysiology*, ed. by A. Popper & R. Fay, 34–93. New York, USA: Springer.

Rumelhart, D. E., G. E. Hinton, & R. J. Williams. 1986. Learning internal representations by error propagation. In *Parallel Distributed Processing. Explorations of the Microstructure of Cognition*, ed. by D. E. Rumelhart & J. L. McClelland, volume 1: Foundations. MIT Press.

Sachs, M. B., R. L. Winslow, & C. C. Blackburn. 1988. Representation of speech in the auditory periphery. In *Auditory Function: Neurobiological Bases of Hearing*, ed. by G. M. Edelman, W. E. Gall, & W. M. Cowan, 747–774. New York, USA: Wiley.

Schwartz, R., & S. Austin. 1991. A comparison of several approximate algorithms for finding multiple (N-Best) sentence hypotheses. In *Proc. Int. Conf. on Acoust., Speech, and Signal Processing*, Toronto, Canada.

——, & Y. Chow. 1990. The N-Best algorithm: An efficient and exact procedure for finding the n most likely sentence hyoptheses. In *Proc. Int. Conf. on Acoust., Speech, and Signal Processing*, New Mexico, USA.

Segui, J., E. Dupoux, & J. Mehler. 1980. The role of the syllable in speech segmentation, phoneme identification, and lexical access. In *Cognitive models of speech processing: Psycholinguistic and computational perspectives. ACL-MIT Press series in natural language processing*, ed. by G. T. M. Altmann, 263–280. Cambridge, MA, USA: MIT Press.

Seneff, S. 1988. A joint synchrony/mean-rate model of auditory processing. *Journal of Phonetics* 16.55–76.

Smith, R., & J. J. Zwislocki. 1975. Short-term adaptation and incremental responses of single auditory-nerve fibers. *Biological Cybernetics* 17.169–182.

Stolcke, A., & S. Omohundro. 1993a. Best-first model merging for Hidden Markov Model induction. Technical report, International Computer Science Institute, 1947 Center St. Suite 600, Berkeley, CA.

——, & ——. 1993b. Hidden Markov Model induction by Bayesian model merging. In *Advances in Neural Information Processing Systems 5*, 11–18. San Mateo, Ca.: Morgan Kaufman.

Viterbi, A. J. 1967. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Trans. on Information Theory* 13.260–269.

Waibel, A., Hanazawa T., Hinton G., S. Kiyohiro, & Lang K. J. 1989. Phoneme recognition using time-delay neural networks. *IEEE Trans. on Acoust., Speech, Signal Processing* 37.328–339.

Wooters, C., 1993. *Lexical Modeling in a Speaker Independent Speech Understanding System*. Berkeley, CA: University of California dissertation.

Zavaliagkos, G., Y. Zhao, R. Schwartz, & J. Makhoul. 1994. A hybrid segmental neural net/hidden Markov model system for continuous speech recognition. *IEEE Transactions on Speech and Audio Processing* 2.151–160.