Discriminative Training for Speaker Adaptation and Minimum Bayes Risk Estimation in Large Vocabulary Speech Recognition

Vlasios Doumpiotis

A dissertation submitted to the Johns Hopkins University in conformity with the requirements for the degree of Doctor of Philosophy.

Baltimore, Maryland
2004

Copyright © 2004 by Vlasios Doumpiotis,
All rights reserved.

Abstract

Stochastic acoustic models are an important component in Automatic Speech Recognition (ASR) systems. The model parameters in Hidden Markov Model (HMM) based speech recognition are normally estimated using Maximum Likelihood Estimation (MLE). If certain conditions hold, including model correctness, then MLE can be shown to be optimal. However, when estimating the parameters of HMM-based speech recognizers, the true data source is not an HMM and therefore other training objective functions, in particular those that involve discriminative training, are of interest. These discriminative training techniques attempt to optimize an information theoretic criterion which is related to the performance of the recognizer.

Our focus in the first part of this work is to develop procedures for the estimation of the Gaussian model parameters and the linear transforms (used for Speaker Adaptive Training) under the Maximum Mutual Information Estimation (MMIE) criterion. The integration of these discriminative linear transforms into MMI estimation of the HMM parameters leads to discriminative speaker adaptive training (DSAT) procedures. Experimental results show that MMIE/DSAT training can yield significant increases in recognition accuracy compared to our best models trained using Maximum Likelihood Estimation (MLE). However by applying MMIE/DSAT training in ASR systems, performance is optimized with respect to the Sentence Error Rate metric that is rarely used in evaluating these systems.

The second part of this thesis investigates how ASR systems can be trained using a task specific evaluation criterion such as the overall risk (Minimum Bayes Risk) over the training data. Minimum Bayes Risk (MBR) training is computationally expensive when applied to large vocabulary continuous speech recognition. A framework for efficient Minimum Bayes risk training is developed based on techniques used in MBR decoding. In particular lattice segmentation techniques are used to derive iterative estimation procedures that minimize empirical risk based on general loss functions such as the Levenshtein distance. Experimental results in one small and two large vocabulary speech recognition tasks, show that lattice segmentation and estimation techniques based on empirical risk minimization can be integrated with discriminative training to yield improved performance.

Advisor:	Dr. William J. Byrne, Professor
Second Reader: :	Dr. Gert Cauwenberghs, Professor
Thesis Committe:	Dr. Sanjeev Khundanpur, Professor
	Dr. Andreas Andreou, Professor

Acknowledgements

At this point, I would like to express my gratitude to all the people who supported and accompanied me during my studies.

I would first like to thank my advisor Bill Byrne for his guidance and encouragement. He has been a friend and a mentor, providing direction when necessary, while giving me the freedom to follow my interests. I am fortunate to have had the privilege of working with him.

I am grateful to the members of my dissertation committee, Professors Gert Cauwenberghs, Sanjeev Khundanpur and Andreas Andreou, for many useful comments and suggestions. I am also grateful to Professor Frederick Jelinek for the opportunity of working at the Center for Language and Speech Processing(CLSP). I would also like to thank Professor Vassilios Digalakis of TUC for introducing me into the very interesting area of speech recognition research.

My colleagues at CLSP, Ahmad Emami, Stavros Tsakalidis, Shankar Kumar, Veera Venkataramani, Yonggang Deng, and all the other people for the time we spent together. They made the center a stimulating and fun place to work. Their enjoyable company and patience to go through several versions of this thesis is greatly appreciated. I am indebted to the administrative staff of CLSP for providing a smooth research environment.

Finally, I thank my parents, Kaiti and Dimitrios Doumpiotis. They gave me constant support and encouragement to pursue my dreams. I am deeply grateful for their love, their patience, and their advise. These pages are dedicated to them.

To my parents.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Automatic Speech Recognition (ASR) is the automatic transcription of human speech by machines. These systems have been around for many years with numerous applications such as dictation, telephone directory assistance, call routing and human-computer interface. During this period a large body of knowledge and experience has been acquired that has resulted in significant improvement in the ability and performance of ASR systems for processing speech and natural language.

The progress achieved during the past years can be attributed mainly to advances in statistical modeling techniques used for acoustic, language modeling and the ability for learning from large speech and text corpora. Due to this progress ASR systems are now becoming a reality and in the past few years they have started to see greater usability on a wide-spread scale, primarily due to the availability of continuous-speech dictation systems for the personal computer (PC). Nevertheless the performance achieved is still not comparable to that achieved by humans. The goal of this thesis is to develop discriminative training procedures to improve performance of ASR systems.

1.1.1 Likelihood vs Discriminative based training

The model parameters in Hidden Markov Model (HMM) based speech recognition systems are normally estimated using Maximum Likelihood (ML) [13, 3]. HMMs have been used successfully in speech recognition for many years, in a large variety of tasks ranging from recognition of a few hundred words for small vocabulary tasks to large vocabulary conversational speech recognition. However, in many aspects the assumptions behind the HMM framework are poor. The limitations of the Maximum Likelihood Estimation (MLE) procedures used widely in (HMM) speech recognition systems are well known. One of the most commonly cited problems is the violation of the model correctness assumption [62],[84].

Parameterized models obtained via MLE [13, 3, 66] can be employed optimally for detection and classification in the large data case if the data encountered is generated by some distribution from the model family. In speech, various conditional independence assumptions are made so that HMMs can be implemented efficiently, but these surely lead to violations of the model correctness assumption. Given these assumptions, it is unlikely that the processes that actually generate speech can be closely modeled by HMMs. Therefore ML estimation of HMMs cannot be relied upon to yield models that are optimum for ASR.

During ML training, the Gaussian model parameters are adjusted to increase the likelihood of the word strings corresponding to the training utterances, but do so without taking account of the probability of other possible word strings. As an alternative to ML estimation, there are modified estimation procedures that directly attempt to optimize automatic speech recognition (ASR) performance criteria [2, 59, 61, 84] such as the sentence error rate (SER) in the training set leading to the Maximum Mutual Information (MMI) criterion [2, 61]. This Discriminative training scheme attempts to increase the *a posteriori* probability of the reference transcription.

Unfortunately discriminative re-estimation of the Gaussian model parameters under the Maximum Mutual Information (MMI) criterion is much more complex and requires substantially more computation than the corresponding ML case[62]. Nevertheless, lattice based MMI estimation techniques have recently been shown to be useful in improving the recognition performance in large vocabulary conversational speech recognition (LVCSR) tasks [77, 84]. Its success has triggered an interest to applying discriminative training to all aspects of the ASR system.

The next section provides an overview of previous research in the field and serves as an introduction to some of the issues tackled in this thesis. Finally an outline of the thesis is presented.

1.2 Proposed Work

This thesis focuses on two research areas, namely discriminative training of acoustic models for speaker adaptation and minimum Bayes risk estimation in large vocabulary continuous speech recognition.

1.2.1 Linear transforms in ASR

Speech recognition performance degrades significantly when there is a mismatch between training and testing conditions. In typical state-of-the-art large vocabulary conversational speech recognition (LVCSR) systems a single model is developed using data from a large number of speakers to cover the variance across dialects, speaking styles, etc. However there are speakers who are poorly modeled using this paradigm. Because it is difficult to estimate separate models for these speakers, linear transforms have been used extensively in the estimation of HMM-based ASR systems[25, 32, 43, 1], either for feature-space or for model-space transformation.

One application of linear transforms in feature space modeling is the decorrelation of the feature vector, typically termed Maximum Likelihood Linear Transformation (MLLT) [32, 26]. It is well known that explicit modeling of correlations between spectral parameters in speech recognition results in increased classification accuracy and improved descriptive power. However, computational, storage and robust estimation considerations make the use of unconstrained, full covariance matrices in HMM observation distributions impractical. The Maximum Likelihood Linear Transformation (MLLT) [32, 26] framework applies a linear transform to the acoustic features in an attempt to capture the correlation between the feature vector components.

A second application done in the model space is the constrained adaptation of the acoustic models to the speaker, the channel, or the task, and this is termed Maximum Likelihood Linear Regression (MLLR)[46, 47, 15]. In general, adaptation techniques are applied to well trained speaker independent model sets to enable them to better model the characteristics of particular speakers. Thus, it would be advantageous using a small amount of a test speaker's data to adapt the speaker independent (SI) model to the speaker. Speaker Adaptation has been shown to be effective in improving the performance of speaker independent (SI) LVCSR systems by adapting the system to the test set.

Adaptation can also be applied to the speakers in the training set, to produce matched conditions with the test set and this is termed Maximum Likelihood (ML) Speaker Adaptive Training (SAT) [1]. The goal of SAT is to reduce inter-speaker variability within the training set. SAT is an iterative procedure that generates one or more transforms to represent each training speaker and/or acoustic environment. Then a *canonical model* is trained given these speaker dependent transforms. SAT is a powerful technique for building speech recognition systems on non-homogeneous data.

1.2.2 Discriminative Linear Transforms

It is also possible to formulate Discriminative estimation procedures for these applications of linear transforms. When estimated in this manner they are called Discriminative Linear Transforms (DLT) [75]. One approach to the use of DLTs is Maximum Mutual Information Linear Regression (MMILR) which was introduced by Uebel and Woodland [75, 76], who showed that it can be used for supervised speaker adaptation. Gunawardana and Byrne [35] introduced the Conditional Maximum Likelihood Linear Regression (CMLLR) algorithm and showed that CMLLR can be used for unsupervised speaker adaptation.

Maximum likelihood linear transforms have also been incorporated with MMI training. McDonough et al. [54] combined SAT with MMI by estimating speaker dependent linear transforms under ML and subsequently used MMI for the estimation of the speaker independent HMM Gaussian parameters. Similarly, Ljolje [50] combined MLLT with the MMI estimation of HMM Gaussian parameters. These transforms were found using ML estimation techniques and were then fixed throughout the subsequent iterations of MMI model estimation. These are hybrid ML/MMI modeling approaches.

Discriminative criteria have also been combined with Linear Discriminant Analysis (LDA)[68]. In LDA [38, 36, 43] a transform is estimated by a class separability criterion and is used to select a compact subset of the original feature set which results in improved processing time and yields minimal decrease in the overall performance of the system. Linear MMI Analysis (LMA) [68], on the other hand, replaces the class separability criterion of LDA with a MMI criterion. As observed by Schlüter [68], although for single densities a relative improvement in word error rate could be observed for LMA in comparison to LDA, the prominence of LMA diminishes with increasing parameter numbers.

Until recently the most widely used estimation technique for MMIE training came from the Extended Baum-Welch Algorithm (EBW) by Gopalakrishnan et al [31]. Normandin [62] extended the EBW algorithm to HMMs with continuous Gaussian densities by using a sequence of discrete approximations.

1.2.3 Novel contributions in Speaker Adaptive Training

Here we propose training procedures that can be used both for MMIE estimation and for speaker adaptive training. In speaker adaptive training the conventional HMM parameter framework θ is extended to accomodate speaker specific transformations in order to produce matched conditions with the test set. They are based on the Conditional Maximum Likelihood (CML) criterion and estimate both the HMM gaussian parameters used in MMIE (4.4) and the linear transforms used in speaker adaptive training (5.7). Thus we obtain fully discriminative procedures for speaker adaptive training termed (DSAT)[73]. These procedures are derived by maximizing Gunawardana's Conditional Maximum Likelihood (CML) auxiliary function (equation 4,[34]) that does not require the quantization of the continuous Gaussian densities and can be applied to arbitrary continuous emission density HMMs.

1.2.4 Minimum Bayes Risk Discriminative Training

The training and decoding procedures of most current state-of-the-art Automatic Speech Recognition (ASR) systems are optimized with respect to the sentence error rate (SER) metric that is rarely used in evaluating these systems. Rather than using the (SER) metric as a training criterion we estimate the acoustic models under a criterion that is more closely related to the ASR recognition performance namely the word error rate (WER). The second part of this thesis investigates the use of discriminative training algorithms that estimate the Gaussian model parameters so as to reduce the overall risk over the training data. Risk minimization techniques have been applied successfully in many fields such as defense (war games), finance (equity investments), gambling (game theory) [64, 80, 74].

Prior research into the use of minimum Bayes-risk criteria for training speech recognizers were performed by Nadas [57, 58] and by Kaiser [41, 42]. The measurement of risk derives from a loss function that is appropriately chosen for the recognition task; for example, in ASR the Levenshtein distance [49] that measures the word error rate (WER) is most commonly used. Kaiser's approach which was applied on a small vocabulary task, is a generalization of MMIE and uses the Extended Baum Welch algorithm [31] for the estimation of the HMM model parameters.

While reducing expected loss on the training data is a desirable training criterion, these algorithms can be difficult to apply in large vocabulary continuous speech recognition systems. These systems typically have several million parameters and require many hours of training data. Unlike MMI estimation where efficient lattice based estimation techniques have been developed [77, 84], these algorithms require an explicit listing of the hypotheses to be considered and in complex problems such lists tend to be prohibitively large. The problem is that although lattice based structures used for large vocabulary tasks make likelihood estimation easy, they do not help with the computation of the risk.

1.2.5 Novel contributions

To overcome this difficulty, modeling techniques originally developed to improve search efficiency in Minimum Bayes Risk (MBR) decoding[28, 30] can be used to transform these estimation algorithms so that exact update, risk minimization procedures can be used for complex recognition problems. Minimum Bayes Risk Decoding is an alternative ASR search strategy that produces hypotheses in an attempt to minimize the empirical risk of speech recognition errors [71, 30]. MBR decoding has been found to consistently provide improved performance relative to straightforward *maximum-a-posteriori* (MAP) decoding procedures. This is usually credited to the integration of the task performance criterion (WER) directly into the decoding procedure. In our case we use lattice segmentation strategies discussed in section (6.3) that decompose a single large lattice into a sequence of smaller sub-lattices and obtain a "pinched" lattice. This approach can be thought of as identifying the recognition problems that remain after the initial recognition pass.

During training presented in section (6.4) we re-estimate the model parameters over these pinched lattices in order to minimize the empirical loss over the training data. We call this estimation strategy Pinched Lattice Minimum Bayes Risk Discriminative Training [17, 18, 16]. Experimental results in sections (9.2), (8.5) show that Minimum Bayes Risk Discriminative Training can yield improvement over MMI in the overall word error rate and in the distribution of individual word recognition errors (8.5.1).

1.3 Organization

The main body of this thesis is mostly concerned with discriminative training in large vocabulary conversational speech recognition(LVCSR). However before going into detail it is necessary to understand some of the basic principles of ASR systems and have some appreciation of how training and decoding is done.

Chapter 2 provides an overview of how continuous speech recognition systems are built today. Conventional speech recognition systems require a decoding criterion, a family of acoustic models, a language model and the basic performance criteria. The acoustic models incorporate knowledge extracted from the speech waveform and they are commonly based on Hidden Markov models (HMMs).

Chapter 3 provides a brief description of the Expectation Maximization (EM) algorithm, along with commentary about its strengths and weaknesses. In likelihood based training, via the Baum-Welch algorithm, given an initial HMM model and the corresponding observation sequence we generate new HMM model parameters that are more likely to have produced the observation sequence. Although it is quite popular, EM is not optimal in reducing the error rate of the ASR system.

Chapter 4 describes a popular form of discriminative training, maximum mutual information estimation (MMIE) that attempts to optimize an information-theoretic criterion which is related to the performance of the recognizer such as the sentence error rate (SER). Classification errors will hopefully be reduced since the likelihood of the correct hypothesis is increased relative to the likelihoods of the competing hypotheses. Parameter estimation is done by maximizing Gunawardana's Conditional Maximum Likelihood (CML) auxiliary function (equation 4,[34]). The efficient implementation of discriminative training in LVCSR systems is also discussed and modeling approaches are presented.

In Chapter 5 we formulate discriminative estimation procedures for the linear transforms used in Speaker Adaptive Training (SAT). Until recently SAT techniques have been based on the maximum likelihood (ML) parameter estimation framework. However discriminative optimization criteria can be more effective in reducing the word error rate than maximum likelihood estimation and hence are of interest. In this thesis both the linear transforms and the Gaussian model parameters are reestimated under MMIE criteria using the (CML) auxiliary function.

In Chapter 6 we present an extension to the standard discriminative training algorithms which focuses on reducing the overall risk over the training data. Therefore rather than using the *a posteriori* probability as a training criterion we estimate the acoustic models under a criterion that is more closely related to the ASR recognition performance. However these algorithms require an explicit listing of the hypotheses to be considered and in complex problems such lists tend to be prohibitively large. To overcome this difficulty, modeling techniques originally developed to improve search efficiency in Minimum Bayes Risk decoding[28, 30] are used during training.

In Chapter 7 we present experimental results on speech material from the *SWITCHBOARD Corpus* by applying discriminative training for speaker adaptation. As an alternative to ML estimation of the linear transforms we use the CML framework in order to obtain fully discriminative procedures. The results show that we can achieve improved recognition accuracy through discriminative training.

In Chapters 8 and 9 we develop estimation procedures based on minimum Bayes Risk criteria. The experiments are conducted on speech material from one small (*Alphadigits*) and two large vocabulary (*SWITCHBOARD, MALACH*) tasks. From these results we argue that it is beneficial to develop discriminative training procedures that are more closely related to the recognition performance criteria.

Finally Chapter 10 provides an overview of this thesis identifying specific research contributions and presents some suggestions for future work in this area.

Chapter 2

An overview of Automatic Speech Recognition

Optimal speech recognition performance depends on a number of factors such as the speech features used, the structure of the acoustic models, the type of output distributions (continuous or discrete), the language model and most importantly the training and decoding algorithms used. Furthermore the construction of the stochastic models (acoustic or language models) used in ASR systems, is dependent upon i) the availability of large corpora of transcribed speech material and ii) text specific to the language and the application we are interested in.

We begin with a brief introduction to the major components of Automatic Speech Recognition systems, by describing the decoding criterion, the language model, the use of HMMs in acoustic modeling and the basic performance criteria used. Readers who are familiar with these terms may skip this chapter.

2.1 Decoding criterion

Speech is the most natural means of human communication and therefore much effort has been spent in the automatic transcription by machines. The ASR problem can be described using the source-channel framework as shown in Fig. 2.1 [39]. In speech recognition we assume that the speech signal is a realization of some message encoded as a sequence W of one or more words. In general, an automatic speech recognition (ASR) system produces a transcription $\hat{W} = \{\hat{w}_1, \cdots, \hat{w}_K\}$ from a sequence of acoustic feature vectors(frames) $O = \{o_1, \cdots, o_R\}$.

ASR is a very difficult task mainly due to the following two reasons. Firstly, the mapping from words to speech is not one-to-one since different words can give rise to similar speech sounds. Secondly, there are large variations in the realised speech waveform due to speaker variability or speaking style.

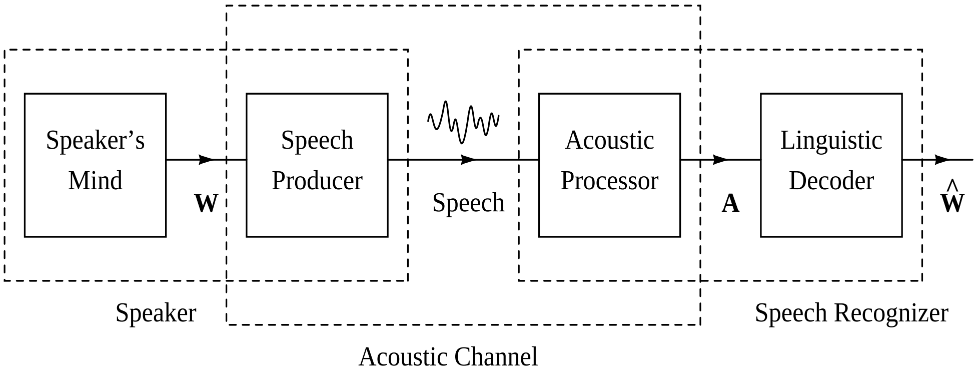


Figure 2.1: The Decoding Model

The probability of making an error is the most important factor in choosing the optimal decoder. This probability is given by:

$$P_e = 1 - \sum_{W'} P(W'|O)\delta(W', W). \tag{2.1}$$

$\delta(W', W)$ is an indicator(delta) function between a sentence hypothesis $W' = \{w'_1, \cdots, w'_N\}$ and the reference transcription W. The goal of the ASR system as shown in Fig 2.1, is to produce a transcription of the speaker's utterance with the least probability of error, thus the maximum a-posteriori(MAP) criterion is used. Given an utterance O, MAP generates a sentence hypothesis that is most likely to have produced the observed signal according to: [3, 65, 39]:

$$\hat{W} = \underset{W'}{\operatorname{argmax}} P(W'|O). \tag{2.2}$$

Computation of the *a posteriori* probability $P(W'|O)$ proceeds by an application of Bayes' rule, which rewrites this probability in terms of the conditional probability $P(O|W')$, that the observation sequence O was produced from the symbol sequence W' and is determined by the acoustic model. We then have

$$P(W'|O) = \frac{P(O|W')P(W')}{P(O)}. \tag{2.3}$$

The term $P(W')$ is the prior probability that the transcription consists of the sequence of symbols w_i'. This probability is usually provided by a language model. Since the signal representation remains constant for all frames, the term $P(O)$ is constant for a given observation sequence and it can be ignored.

2.2 Language Modeling

Speech is not a stochastic process where sounds are generated in an arbitrary sequence; it is rather a structured generation process. An utterance is more likely to contain valid words than nonsense, using structures such as verb, subject, object and prepositional phrases. All this structural information can be exploited when performing recognition and can be useful in improving recognition accuracy.

Current state of the art recognition systems form a stochastic model of word occurrence called the language model. The purpose of the language model is to estimate the likelihood $P(W')$ of a word sequence $W' = w_1'^N$, and is given by

$$P(W') = P(w_1', w_2', \ldots, w_N') = \prod_{i=1}^{N} P(w_i' \mid w_1', \ldots, w_{i-1}'). \tag{2.4}$$

Since it is unfeasible to calculate the probability of observing the entire word sequence, n-grams are used to estimate the likelihoods of smaller word strings within the sequence. Thus equation (2.4) becomes,

$$P(W') = P(w_1', w_2', \ldots, w_N') \simeq \prod_{i=1}^{N} P(w_i' \mid w_{i-1}', w_{i-2}', \ldots, w_{i-(n-1)}'), \tag{2.5}$$

after applying the Markov independence assumption.

Currently popular forms of n-grams are the unigram, which considers individual word frequencies, the bigram, which models word to word transition probabilities, and the trigram model, which computes the likelihood of any word, based on the two words immediately before it. Although the estimation of the language models itself is not considered in this thesis, language models need to be used during discriminative training of the acoustic models.

2.3 Acoustic Modeling

The acoustic model provides the connection between the acoustics and the lexical transcription of speech. The quality of the acoustic model used in speech recognition has a significant impact in determining the system's performance. Furthermore an inadequate acoustic model will limit the potential gains from other knowledge sources such as the language model. In the ideal case, the acoustic model obtained after training should yield the lowest possible recognition error rate.

2.3.1 HMMs in Speech Recognition

Hidden Markov models (HMMs) form an integral part of current state-of-the-art automatic speech recognition systems. They provide a robust and simple framework for speech modeling. HMMs are used to describe piecewise stationary signals. Although it can be argued that speech signals are not actually piecewise stationary, HMMs have many other desirable qualities that make them popular in ASR systems. Because their behavior can be described with simple formulas, HMMs provide a solid theoretical foundation from a probabilistic standpoint and the full power of mathematics and statistical theory can be brought into play on the speech recognition problem.

Furthermore HMMs are effective for ASR because the procedures used in their ML estimation, such as the Baum-Welch algorithm, are efficient and straightforward to implement. The re-estimated parameters are found so as to guarantee an improvement in the training data likelihood.

2.3.2 HMM parameters

An HMM has two types of parameters, the transition probabilities and the output distributions. The transition probabilities capture the time-varying nature of speech and the output distributions model the acoustic signal. In HMM based speech recognition, it is assumed that the sequence O of observed speech vectors corresponding to each utterance is generated by a Markov model as shown in Fig 2.2 borrowed from the HTK manual[86]. A Markov model is a finite state machine which changes state once every time unit and each time τ that a state j is entered, a speech vector o_τ is generated from the probability density $b_j(o_\tau)$. Furthermore, the transition from state i to state j is also probabilistic and is governed by the discrete probability a_{ij}.

Thus an HMM is completely specified by its state transition probabilities and the output probability distribution for each state. Fig 2.2 shows an example of this process where the model moves through the state sequence $S = 1, 2, 2, 3, 4, 4, 5, 6$ in order to generate the sequence o_1 to o_6. Notice that, the entry and exit states of the HMM are non-emitting. The model does not produce any output while in these states, but they are convenient when concatenating several models together to represent words of arbitrary length.

In ASR a separate HMM is trained for each fundamental recognition unit, which may be either whole words or sub-word units such as phonemes or triphones. Whole word models are well suited for small vocabulary tasks (generally less that a few hundred words). However for large vocabulary tasks, sub-word units are a better choice because of data sparsity. Sub-word units are used to facilitate the construction of composite models and therefore the acoustic models can be hierarchically decomposed into different levels. As a result whole sentences can share word models and word models can share sub-word models such as triphones.

In Fig 2.2, the arrows represent allowed transitions between states. At any discrete time index, the model occupies one and only one state and produces only one output. At each time increment, the model can follow one of the allowed transitions, according to a transition probability. The topology is left to right and the temporal characteristics of speech are captured by the transitions between the states of the

HMM.

2.3.3 Computations with HMMs

The joint probability that O is generated by the model M moving through the state sequence S is calculated simply as the product of the transition probabilities and the output probabilities. So for the state sequence S in Fig 2.2:

$$P(O,S|M) = a_{12}b_2(o_1)a_{22}b_2(o_2)a_{23}b_3(o_3)\dots \tag{2.6}$$

However, in practice, only the observation sequence O is known and the underlying state sequence S is hidden. Hence the term *Hidden Markov Model.* When estimating acoustic models, we typically use *training data* consisting of transcribed recorded speech$\{\bar{W},O\}$. Thus, we can observe $\bar{W}$ and O, but not the corresponding state sequence S.

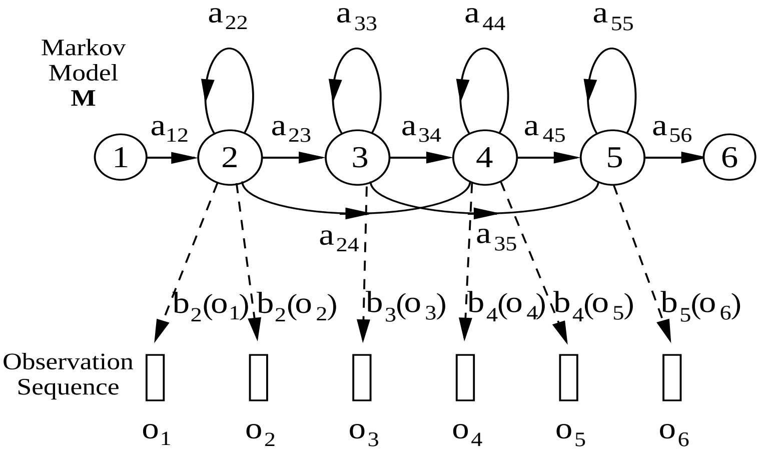


Figure 2.2: The Markov Generation Model

Given that S is unknown, the required likelihood is computed by summing over all possible state sequences $S = s(1), s(2), s(3), \dots, s(T)$, that is

$$P(O|M) = \sum_S P(O,S|M) = \sum_S a_{s(0)s(1)} \prod_{\tau=1}^{T} b_{s(\tau)}(o_\tau)a_{s(\tau)s(\tau+1)} \tag{2.7}$$

where $s(0)$ is constrained to be the model entry state and $s(T+1)$ is constrained to be the model exit state.

If we assume that the total number of states is N, then it can be seen that the computational complexity of the above calculation is $O(TN^T)$ based on the fact that there are N^T possible state sequences with $2T$ terms in each product. To ameliorate the computational complexity, recursive techniques exist that are more efficient than direct calculation. We will discuss them briefly in the next chapter.

As an alternative to equation (2.7), the likelihood can be approximated by only considering the most likely state sequence(*Viterbi path alignment*) [66], that is

$$P(O|M) = \max_{S} \left\{ a_{s(0)s(1)} \prod_{\tau=1}^{T} b_{s(\tau)}(o_\tau) a_{s(\tau)s(\tau+1)} \right\} . \tag{2.8}$$

This likelihood is computed using essentially the same algorithm as the forward probability calculation except that the summation is replaced by a maximum operation.

In the HMM framework the acoustic observations o_τ are assumed to be independent of each other therefore speech dynamics cannot be modeled directly. However, it is known that this is not a valid assumption for speech signals, which are by nature highly temporally correlated. Many approaches have been proposed to overcome this limitation of HMMs. The most successful example of these approaches is the use of dynamic parameters where the "static" features are augmented with the first and the second differentials. By using 12 mel-filterbanks, plus the energy estimate and their first and second order coefficients we end up with a 39 dimensional feature vector. The two most commonly used parameter forms in speech front-end processing, are the Mel Frequency Cepstral Coefficients (MFCC)[12] and the Perceptual Linear Prediction (PLP) coefficients[37].

2.3.4 Output emission densities

The output probability distributions $b_{s(\tau)}(o_\tau)$ may be either discrete or continuous depending on the observation space. For simplicity we express $b_{s(\tau)}(o_\tau) = q(o_\tau|s;\ \theta)$, which is the output emission probability of observing o in state s at time τ, as a single mixture multivariate Gaussian with mean vector μ_s and covariance matrix Σ_s. We use the compact representation for the model parameters $\theta = (\mu_s, \Sigma_s)$. Therefore the term $q(o_\tau|s;\ \theta)$ is reparametrized as

$$b_{s(\tau)}(o_\tau) = q(o_\tau|s;\ \theta) = \mathcal{N}(o_\tau;\ \mu_s, \Sigma_s) = \frac{1}{\sqrt{(2\pi)^n|\Sigma_s|}} e^{-\frac{1}{2}(o_\tau-\mu_s)'\Sigma_s^{-1}(o_\tau-\mu_s)}. \tag{2.9}$$

The covariances Σ_s are assumed diagonal(zero off-diagonal covariance terms) for reasons of efficient Gaussian evaluation, compact storage and robust parameter estimation given the limited available training data.

In order to build accurate acoustic models we use mixtures of Gaussian densities. Thus equation (2.9) becomes,

$$q(o_\tau|s;\ \theta) = \sum_{j=1}^{M} \frac{w_{s,j}}{\sqrt{(2\pi)^n|\Sigma_{s,j}|}} e^{-\frac{1}{2}(o_\tau-\mu_{s,j})'{\Sigma_{s,j}}^{-1}(o_\tau-\mu_{s,j})}. \tag{2.10}$$

The mixture weights(*a priori* probabilities for each of the mixture components) $w_{s,j}$ must satisfy

$$\sum_{j=1}^{M} w_{s,j} = 1, \quad 0 \leq w_{s,j} \leq 1. \tag{2.11}$$

The use of such densities has several advantages, most important of which is the ability of capturing the acoustic model variability across different samples of the same speech sound(better resolution). However the number of mixture components must be small enough to allow for reliable estimates of the Gaussian parameters given the limited amount of available training data. Usually the acoustic models are initialized by one single component density per state. Subsequently the mixture densities are then iteratively splitted up during the training process until the desired number(usually determined experimentally).

2.4 Decoding and Performance Evaluation

Assuming that the HMMs have been trained and that there exists a language model that can compute $P(W')$, then everything is in place to find the optimal word sequence $\hat{W}$ according to (2.2). The acoustic and language model scores, form a graph representing the search space. Transitions within HMMs are determined by the parameters θ of the acoustic model, and transitions within words are determined by the language model.

During recognition the Viterbi algorithm will return the most likely word sequence $\hat{W}$. By extending the original Viterbi algorithm(token passing algorithm), we can keep track of more than just the single best partial hypothesis during recognition, which may be used to generate a lattice of possible word hypotheses rather than only the individual best sequence. A lattice $\mathcal{W}$ as shown in Fig.2.3, consists of a set of nodes that correspond to particular instants in time, and arcs connecting these nodes to represent possible word hypotheses. Associated with each arc is an acoustic score (log likelihood) and a language model score.

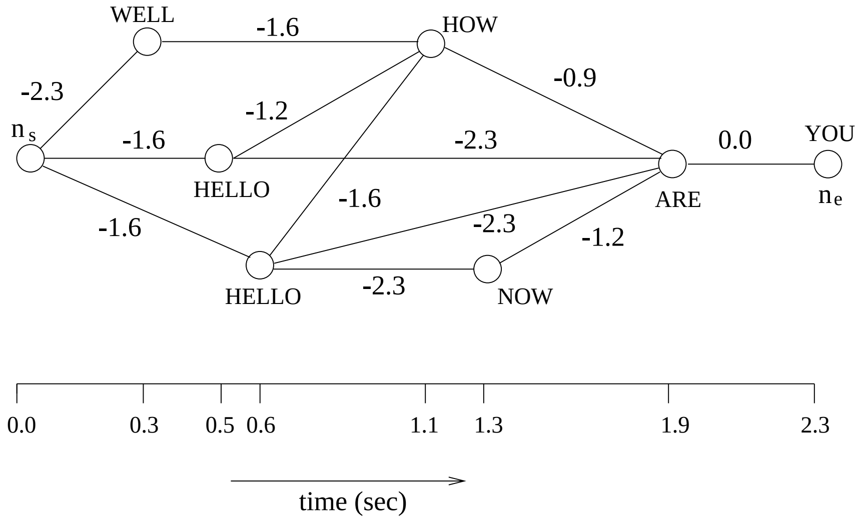


Figure 2.3: A lattice $\mathcal{W}$. The time marks correspond to the node times and the word ending times. The numbers on the edges are natural logarithms of acoustic plus language model scores.

2.4.1 Levenshtein distance and SER vs WER

Once the recognizer has been developed it is necessary to evaluate its performance. The two most commonly used measures of recognition accuracy are the WER and the SER. The WER metric measures accuracy at the word level rather than the sentence level as is the case for the SER. The SER is defined as:

$$SER = \frac{number\ of\ sentences\ with\ one\ or\ more\ errors}{total\ number\ of\ sentences} \tag{2.12}$$

In order to estimate the WER, a dynamic programming algorithm[67] that computes the Levenshtein distance [49] is used to align the top hypothesis $\hat{W}$ produced by the ASR system against the reference transcription W. The total error is obtained by counting the number of words that have been substituted, deleted, and inserted:

$$WER = 1 - \frac{N - S - D - I}{N} = \frac{S + D + I}{N} \tag{2.13}$$

where S is the number of substitutions, I number of insertions, D is the number of deletions needed to transform one word string into another and N is the number of words in the reference transcription.

The greater the Levenshtein distance, the more different the strings are. The metric is also sometimes called string edit distance because it measures the minimum number of string edit operations needed to transform one string into another. An example alignment is:

Reference :	UM	DID	YOU	TALK	ABOUT	–	ANN
Hypothesis :	–	DID	YOU	TALK	AGAIN	TO	ANN
Errors :	Deletion	–	–	–	Substitution	Insertion	–

There is a total error of three words in this example. The reference word UM is deleted, ABOUT is substituted by AGAIN and word TO is inserted. The word error rate (WER), the fraction of reference words in error, is 3/6 = 50%. In the example above, the entire reference sentence is not correctly transcribed, the sentence error rate (SER) is 100%. In the ideal case the WER should be 0% (no words in error). Otherwise, the word error rate for this sentence may vary anywhere between 17% (one word in error) to over 100% (six substitutions and some insertions), all of which would result in the same sentence error rate of 100%.

The sentence error rate may therefore not be a good indicator of performance on this task; it may be rather poorly correlated with the word error rate. We also note that the MAP decoding criterion (2.2) will minimize the sentence-level error rate

(SER) rather than the word error rate (WER). This observation is very important because current state-of-the-art ASR systems use either Maximum Likelihood or MMIE techniques for training and Maximum-A-Posteriori (MAP) techniques for decoding. These estimation criteria are optimal under the Sentence Error Rate metric which is rarely used in evaluating these systems. As an alternative the Minimum Bayes Risk framework attempts to minimize the cost of speech recognition errors and thus is more closely related to ASR performance criteria.

2.5 Summary

This chapter provided an overview of the basic components of an ASR system(acoustic and language models) along with the basic performance criteria used to evaluate different systems. HMMs have become the most popular parametric model used for speech recognition. They have been applied successfully in a large variety of speech recognition tasks ranging from recognition of a few hundred words for small vocabulary tasks to large vocabulary conversational speech recognition.

However HMMs are not the “correct” models of speech. They are based on assumptions such as independence assumption, Markov assumption, which are inaccurate for the speech generation process. Consequently Maximum Likelihood estimation of the HMM parameters θ can result in suboptimal performance. As an alternative to Maximum Likelihood estimation there are Discriminative training criteria that can yield significant improvement in speech recognition accuracy. In the next two chapters we will show how to estimate the acoustic model parameters θ, under the Maximum Likelihood and the Maximum Mutual Information Estimation criteria respectively.

Chapter 3

Expectation Maximization(EM) Algorithm

We have seen in the previous chapter, how to compute probabilities using a model M and the HMM parameter set $\theta = (\mu_s, \Sigma_s)$. However, nothing was said about how to estimate the acoustic model parameters θ themselves. In general, there are two types of training methods, likelihood-based and discriminative.

Maximum likelihood estimation (MLE), which was until recently, the most commonly used approach in estimating θ, is the topic of this chapter. Although the theory behind the Maximum likelihood estimation (MLE) is well documented elsewhere, the brief presentation of the parameter estimation formulae helps to serve as a reference for later chapters of this thesis. Several issues related to implementation are also discussed in order to provide an easier understanding of the experimental framework.

3.1 Estimation of Hidden Markov models (HMMs)

The estimation of Hidden Markov model (HMM) parameters for speech recognition [5, 13, 3, 66, 39] is an example of 'estimation from incomplete data'. Thus, we want to estimate statistical models when some of the random variables of interest are not directly observed.

There is a large body of work that deals with optimal methods of parameter estimation. We will focus on *maximum likelihood* [22, 6, 48] which attempts to choose parameters that maximize the likelihood assigned by the resulting models to the observations. It is known that such methods have desirable statistical properties such as sufficiency, consistency, and efficiency [22, 81, 48]. As mentioned above, we are interested in the case where only ‘incomplete’ or indirect observations of the variable of interest are available. In other words, we only have access to observations, because the underlying state sequence is unknown or “hidden”.

The solution of choice for problems of estimation from incomplete data in ASR is the *Expectation Maximization* (EM) algorithm of [13]. This is an iterative scheme which, given the parameters from the previous iteration, first forms an *auxiliary function* over the parameter set, and then chooses the next parameter set to be the maximizer of the auxiliary function under the current parameter set. The auxiliary function is defined to be the conditional expectation of the complete data log-likelihood given the observed data, evaluated at the current parameter.

Forming the auxiliary function is referred to as the expectation or E-step. The maximization of the auxiliary function is referred to as the maximization or M-step. The auxiliary function is defined so that this two step procedure ensures no decrease in the likelihood of the “incomplete data”. The EM algorithm is an iterative training procedure that guarantees local optimality, given an initial parameter set. For efficient implementation of the EM algorithm we either apply the Forward Backward(Baum-Welch) algorithm, or the Viterbi algorithm if the most likely state sequence is considered.

3.2 EM via Baum-Welch algorithm

The essential problem is to estimate the means and variances of a HMM, when the output distribution of each state s is a single component Gaussian given by equation (2.9). EM is a two-stage iterative procedure, the current values of the hidden data are calculated during the expectation step using the model parameters from the previous iteration and are then used in the maximization step to generate a new set of model parameters. Maximum likelihood estimation (MLE) is synonymous with the Baum-

Welch algorithm so we will use these terms interchangeably.

The MLE criterion increases the probability of the model sequence corresponding to the correct transcription. For R training observation sequences $\{O_1, \cdots, O_R\}$ with corresponding transcriptions $\{\bar{W}_1, \cdots, \bar{W}_R\}$, the MLE objective function, is given by

$$\theta^* = \underset{\theta}{\operatorname{argmax}} F(\theta) = \underset{\theta}{\operatorname{argmax}} \sum_{r=1}^{R} logP_\theta(O_r|M_{\bar{W}_r}) \tag{3.1}$$

Thus MLE tries to increase the probability of the r-th observation sequence, given the model $M_{\bar{W}_r}$ corresponding to the correct transcription. Increasing the likelihood of the training data is one technique that often leads to improved performance in the unseen test set. By observing (3.1), we see that the models from other classes(competing hypotheses) do not participate in the parameter re-estimation. As a result it is not obvious how the MLE objective function relates to the objective of reducing the error rate. In contrast, discriminative training methods have been developed which adjust the model parameters to increase not the likelihood of the data given the model, but rather increase the *a-posteriori* probability.

Because complex acoustic models typically employ thousands of parameters, in general it is not feasible to find a globally optimal θ^*. Instead the optimization algorithm starts from an initial value of θ and converges to a local optimum in the parameter space. At each iteration starting from θ we find $\bar{\theta}$ such that

$$\sum_{r=1}^{R} logP_{\bar{\theta}}(O_r|M_{\bar{W}_r}) \geq \sum_{r=1}^{R} logP_\theta(O_r|M_{\bar{W}_r}). \tag{3.2}$$

For simplicity we consider a single utterance $O = \{o_1, \cdots, o_T\}$ with a duration of T frames. The estimation procedure is done by maximizing the following auxiliary function:

$$Q(\bar{\theta}|\theta) = \sum_S P_\theta(O, S|M) logP_{\bar{\theta}}(O, S|M) \tag{3.3}$$

where S is a possible "hidden" state sequence. The log factor is:

$$\begin{aligned} logP_{\bar{\theta}}(O, S|M) &= \sum_{\tau=1}^{T} loga_{s(\tau)s(\tau+1)} + \sum_{\tau=1}^{T} logq(o_\tau|s_\tau;\ \bar{\theta}) + loga_{s(0)s(1)} \\ &= \sum_{\tau=1}^{T} loga_{s(\tau)s(\tau+1)} + \sum_{\tau=1}^{T} log\mathcal{N}(o_\tau;\ \bar{\mu}_s, \bar{\Sigma}_s) + loga_{s(0)s(1)} \end{aligned} \tag{3.4}$$

For the mean the auxiliary function becomes:

$$Q(\bar{\theta}|\theta) = -\frac{1}{2}\sum_{s\in S}\sum_{\tau=1}^{T}\gamma_s(\tau;\theta)\big[log|\Sigma_s| + (o_\tau - \bar{\mu}_s)^T\Sigma_s^{-1}(o_\tau - \bar{\mu}_s)\big] + C \qquad (3.5)$$

where C is a constant independent of θ coefficients, o_τ is the acoustic vector observation at time τ and $\gamma_s(\tau;\theta)$ denotes the probability of being in state s at time τ given the current parameter estimates θ. After we have estimated the new mean, the auxiliary function for the variance becomes:

$$Q(\bar{\theta}|\theta) = -\frac{1}{2}\sum_{s\in S}\sum_{\tau=1}^{T}\gamma_s(\tau;\theta)\big[log|\bar{\Sigma}_s| + (o_\tau - \bar{\mu}_s)^T\bar{\Sigma}_s^{-1}(o_\tau - \bar{\mu}_s)\big] + C \qquad (3.6)$$

The usefulness of $Q(\bar{\theta}|\theta)$ comes from the fact that $Q(\bar{\theta}|\theta) \geq Q(\theta|\theta)$ implies $F(\bar{\theta}) \geq F(\theta)$ which is (3.2).

3.2.1 Training

To determine the parameters of a HMM we can use two approaches. If the most likely state sequence(*Viterbi path alignment*) is available then the parameter estimation would be easy. The maximum likelihood estimates of μ_j and Σ_j (assuming single mixture gaussians) would be just the simple averages, that is

$$\bar{\mu}_j = \frac{\sum_{t=1}^{T}\delta(s(t),j)o_t}{\sum_{t=1}^{T}\delta(s(t),j)} \qquad (3.7)$$

and

$$\bar{\Sigma}_j = \frac{\sum_{t=1}^{T}\delta(s(t),j)(o_t - \bar{\mu}_j)(o_t - \bar{\mu}_j)'}{\sum_{t=1}^{T}\delta(s(t),j)} \qquad (3.8)$$

where $\delta(s(t),j)$ is an indicator function that at time t the state is j.

In the second approach there is no direct assignment of observation vectors to individual states because the underlying state sequence S is unknown. Since the full likelihood of each observation sequence is based on the summation of all possible state sequences, each observation vector o_t contributes to the computation of the maximum likelihood parameter values for each state j. In other words, instead of assigning each observation vector to a specific state as in the above approximation, each observation is assigned to every state in proportion to $\gamma_j(t;\theta)$, the probability of the model being in that state when the vector was observed.

By taking the gradient of $Q(\bar{\theta}|\theta)$ and setting it equal to zero, the equations (3.7) and (3.8) given above, become the following weighted averages

$$\bar{\mu}_j = \frac{\sum_{t=1}^{T} \gamma_j(t;\theta) o_t}{\sum_{t=1}^{T} \gamma_j(t;\theta)} \tag{3.9}$$

and

$$\bar{\Sigma}_j = \frac{\sum_{t=1}^{T} \gamma_j(t;\theta)(o_t - \bar{\mu}_j)(o_t - \bar{\mu}_j)'}{\sum_{t=1}^{T} \gamma_j(t;\theta)}. \tag{3.10}$$

Equations (3.9) and (3.10) are the Baum-Welch re-estimation formulae for the means and covariances of a HMM. A similar but slightly more complex formula can be derived for the transition probabilities. Of course, to apply equations (3.9) and (3.10), the probability of state occupation $\gamma_j(t;\theta)$ must be calculated. This is done efficiently using the so-called *Forward-Backward* algorithm.

3.2.2 Forward-Backward algorithm

To estimate the probability of state occupation $\gamma_j(t;\theta)$ we use the following quantities, the forward probability $\alpha_j(t)$ and the backward probability $\beta_j(t)$. To ameliorate the computational complexity, these quantities can be estimated efficiently using recursive techniques. We discuss them briefly in this section.

The forward probability $\alpha_j(t)$ for some model M with N states is defined as

$$\alpha_j(t) = P(o_1, \ldots, o_t, s(t) = j|M). \tag{3.11}$$

Thus, $\alpha_j(t)$ is the joint probability of observing the first t speech vectors and being in state j at time t. This forward probability can be efficiently calculated by the following recursion

$$\alpha_j(t) = \left[\sum_{i=2}^{N-1} \alpha_i(t-1) a_{ij} \right] b_j(o_t). \tag{3.12}$$

This recursion depends on the fact that the probability of being in state j at time t and observing o_t can be calculated by summing over all the forward probabilities for all possible predecessor states i, weighted by the transition probabilities a_{ij}. The initial conditions for the above recursion are

$$\alpha_1(1) = 1 \tag{3.13}$$

$$\alpha_j(1) = a_{1j} b_j(o_1) \tag{3.14}$$

for $1 < j < N$ and the final condition is given by

$$\alpha_N(T) = \sum_{i=2}^{N-1} \alpha_i(T) a_{iN}. \tag{3.15}$$

Notice here that from the definition of $\alpha_j(t)$,

$$P(O|M) = \alpha_N(T). \tag{3.16}$$

Hence, the calculation of the forward probability also yields the total likelihood $P(O|M)$ (2.7).

The backward probability $\beta_j(t)$ is defined as

$$\beta_j(t) = P(o_{t+1}, \ldots, o_T | s(t) = j, M). \tag{3.17}$$

As in the forward case, this backward probability can be computed efficiently using the following recursion

$$\beta_i(t) = \sum_{j=2}^{N-1} a_{ij} b_j(o_{t+1}) \beta_j(t+1) \tag{3.18}$$

with initial condition given by

$$\beta_i(T) = a_{iN} \tag{3.19}$$

for $1 < i < N$ and final condition given by

$$\beta_1(1) = \sum_{j=2}^{N-1} a_{1j} b_j(o_1) \beta_j(1). \tag{3.20}$$

Notice that in the definitions above, the forward probability is a joint probability whereas the backward probability is a conditional probability. The probability of state occupation is determined by taking the product of the two probabilities. From the definitions,

$$\alpha_j(t)\beta_j(t) = P(O, s(t) = j|M). \tag{3.21}$$

Hence,

$$\begin{aligned} \gamma_j(t;\theta) &= P(s(t) = j|O, M) \\ &= \frac{P(O, s(t) = j|M)}{P(O|M)} \\ &= \frac{1}{P}\alpha_j(t)\beta_j(t) \end{aligned} \tag{3.22}$$

where $P = P(O|M)$ is the total likelihood.

3.3 Practical issues & Overtraining

The Forward-Backward algorithm above, requires a set of phone-level transcriptions of the utterances in the training set. Given the previous description, the steps during the Baum-Welch training may be summarized as follows:

1. For each training utterance, the corresponding phone models are concatenated.

2. Calculate the forward and backward probabilities for all states j and times t.

3. For each state j and time t, use the probability $\gamma_j(t;\theta)$ and the current observation vector o_t to update the accumulators for that state.

4. When all of the training data has been processed, the accumulated statistics are used to calculate new parameter values for all of the HMMs according to (3.9) and (3.10).

5. If the data likelihood for this iteration is not higher than the value at the previous iteration then stop, otherwise repeat the above steps using the new re-estimated parameter values.

In practice, in order to get accurate acoustic models a large amount of training data is needed i.e several thousands of utterances are needed for speaker independent models. Given a set of training examples and their associated transcriptions a composite HMM is constructed for each one of them. Steps 1, 2 and 3 above are simply repeated for each distinct training sequence. Steps $1-5$ can all be repeated as many times as necessary to achieve the required convergence.

A good statistical model should accurately describe objects it has not yet encountered. In practice, it is usually observed that even though recognition rate on the unseen test set initially increases with each iteration, a maximum is quickly reached (often after as little as 3 or 4 iterations) after which performance goes down. This is called *overtraining* and is caused in part by differences between the training and test sets and in part because MLE does not necessarily decrease the error rate. Because of this behavior, training is usually stopped after a fixed number of iterations. Furthermore experimental results have shown that the initial θ has a strong influence on the system performance. This underscores the importance of a good initialization.

3.4 Summary

The MLE training criterion described here determines the parameters θ of the acoustic models such that the training data is optimally described. A major disadvantage with MLE is that it has no obvious relationship with the objective of minimizing the recognition error rate. To decrease classification error, one must instead optimize the posterior probability of a class given the data. This follows from Bayes decision theory which states that the minimum probability of error is achieved when a class is chosen that has the highest posterior probability.

Furthermore the rationale behind the use of MLE training relies on the assumption that the underlying models are the "true" models of speech and sufficient training data is available. However this is not always the case. Discriminatively training techniques discussed next remove the need for this assumption and directly attempt to improve recognition performance criteria.

Chapter 4

Discriminative Training

Discriminative training techniques attempt to produce statistical models that optimize the correct transcription relative to other rival hypotheses. This is done by using information theoretic criteria. Unlike likelihood-based training, they adjust the model to produce low probability scores for competing "rival" classes.

In this chapter, we discuss the maximum mutual information principle and its application to acoustic modeling. We show how the Conditional Maximum Likelihood (CML) auxiliary function can be applied to the re-estimation of the Gaussian model parameters. We also address implementation issues when using discriminative training in large vocabulary applications and how to get the most from a limited amount of training data.

4.1 MMIE Criterion

Maximum mutual information estimation(MMIE) was proposed in [2] as an alternative to MLE. It maximizes the mutual information between the training word sequences and the observation sequences. The criterion to be maximized is

$$I(\bar{W}, O;\ \theta) = \log \frac{P(\bar{w}_1^n, o_1^l;\ \theta)}{P(\bar{w}_1^n)\ P(o_1^l;\ \theta)} \tag{4.1}$$

Without loss of generality we consider the single observation case. $(\bar{W}, O)$ may also represent multiple training utterances. In that case with R training utterances equation (4.1) becomes:

$$I(\bar{W}, O;\ \theta) = \sum_{i=1}^{R} \log \frac{P((\bar{w}_1^n, o_1^l)^{(i)};\ \theta)}{P(\bar{w}_1^{n\,(i)})\ P(o_1^{l\,(i)};\ \theta)}.$$

Since the language model $P(\bar{w}_1^n)$ is independent of the acoustic model parameters θ, maximizing the mutual information estimate is equivalent to maximizing the conditional likelihood $P(\bar{w}_1^n | o_1^l;\ \theta)$. Thus the MMI estimate is equivalent to the *conditional maximum likelihood* (CML) estimate of [57], so we will use these terms interchangeably.

4.1.1 CML criterion

As mentioned before, by ignoring the language model, maximizing the mutual information is equivalent to maximizing the conditional likelihood $P(\bar{W}|O)$. For R training observation sequences $\{O_1, \cdots, O_R\}$ with corresponding transcriptions $\{\bar{W}_1, \cdots, \bar{W}_R\}$, the CML objective function, is given by:

$$\theta^* = \underset{\theta}{\operatorname{argmax}}\, F(\theta) = \underset{\theta}{\operatorname{argmax}} \sum_{r=1}^{R} logP(\bar{W}_r|O_r) = \underset{\theta}{\operatorname{argmax}} \sum_{r=1}^{R} log \frac{P_\theta(O_r|M_{\bar{W}_r})P(\bar{W}_r)}{\sum\limits_{W' \in \mathcal{W}} P_\theta(O_r|M_{W'})P(W')} \tag{4.2}$$

where $M_{W'}$ is the composite model corresponding to the word sequence W' and $P(W')$ is the probability of this sequence as determined by the language model.

Therefore CML/MMIE adjusts the model parameters by considering all models $M_{W'}$ simultaneously. In contrast to MLE, CML/MMIE optimizes the correct models at the expense of the set $\mathcal{W}$ of alternative word hypotheses. We call $\mathcal{W}$ the *evidence space* and it can be approximated either by N-best lists or lattices as shown in Fig.2.3. Because of the large size of $\mathcal{W}$, discriminative methods require substantially more computations than MLE. The efficient implementation of discriminative training in LVCSR systems will be discussed in the following sections. Since (4.2) is also the criterion used in MAP decoding(2.2), the relationship between MMIE and error rate is much more intuitive than it is with MLE.

4.2 Minimum Classification Error Criterion

There have been other discriminative training procedures proposed in the literature. An alternative discriminative training criterion, the Minimum Classification Error principle was introduced by Juang and Katagiri [40], has been successfully applied to speech recognition [10, 33, 7]. The MCE function attempts to approximate the error rate in the training data and its optimization by the generalized probabilistic descent(GPD) algorithm results in a classifier with minimum error. Juang and Katagiri initially used only one competing word sequence per utterance. This means that only the best incorrectly recognized word sequence is used as competition [40].

The MCE objective function measures the negative log distance between the correct transcription and the alternative hypotheses. The "cost" function gets a small value, if the distance becomes as negative as possible, it means that we have correct classifications, otherwise we have incorrect classifications. A simple zero-one cost function would measure this error rate perfectly, but it violates the constraint that the function should be continuously differentiable with respect to the Gaussian model parameters. Therefore the MCE criterion uses a smoothed version of the empirical error rate on the training data such as a sigmoid function, in order to obtain a continuous and differentiable objective function. For the MCE criterion, gradient descent is usually the optimization method of choice.

4.3 Background

Initially there were no algorithms proven to be convergent for discriminative training, and gradient descent techniques that converged slowly and gave little improvement were used. Then the most widely used estimation technique for MMIE training came from the Extended Baum-Welch Algorithm (EBW) by Gopalakrishnan et al [31], which extends the well known Baum-Eagon inequality [4] for optimizing rational functions. Gopalakrishnan et al. introduced the following reestimation formula for rational objective functions such as $R(\theta) = P(\bar{W}|O)$ associated with discrete

HMMs:

$$\bar{\theta} = \frac{\theta(\nabla_{\theta} logR(\theta) + D)}{\sum_{\theta'} \theta'(\nabla_{\theta'} logR(\theta') + D)} \tag{4.3}$$

The extended growth transformation above applies only to discrete HMMs and was used to estimate the model parameters under the MMI criterion. For this case, a proof of convergence was presented although the corresponding iteration constants lead to very low convergence rates. Normandin [62] extended the EBW(4.3) algorithm to HMMs with continuous Gaussian densities. He used a sequence of discrete approximations to the Gaussian density so that the original EBW algorithm of Gopalakrishnan can be applied. The iteration constants for which convergence could be proven in the discrete case, map to infinity in the continuous case and hence convergence cannot be easily proven.

These parameter estimation procedures under the maximum mutual information criterion had a computational cost that made them impractical for large scale applications. In particular the minimization of the denominator term in equation (4.2) involves a full recognition on all the training data for each iteration of MMIE training. For large vocabulary tasks this is computationally impracticable, therefore an approximation to the denominator term is required and lattices are used. This step dominates the computation and depends on the size of the vocabulary, the grammar and any contextual constraints. Valchev et al. [77, 84] implemented MMIE training of LVCSR systems through the use of lattices and used Normandin's update equations for the re-estimation of the Gaussian model parameters.

In order to limit the computational complexity arising from the need to find confusable speech segments in the large search space of alternative word hypotheses, Nornandin [61] used only one competing word sequence per utterance. This means that only the best recognized word sequence is used as competition against the spoken utterance, leading to “corrective training”.

4.4 The CML Algorithm

Here, we use a simplified derivation of the CML estimation algorithm of Normandin[61] that does not require the discrete approximation to the Gaussian density. Furthermore, it extends Normandin's algorithm to arbitrary continuous emission density HMMs. These procedures are derived by maximizing Gunawardana's Conditional Maximum Likelihood (CML) auxiliary function(equation 4,[34]).

The iterative CML algorithm of Gunawardana chooses $\theta^{(p+1)}$ given a parameter iterate $\theta^{(p)}$ according to:

$$\begin{aligned}\theta^{(p+1)} \in \operatorname*{arg\,max}_{\theta\in\Theta} \sum_{s_1^{\hat{l}}} \left[q(s_1^{\hat{l}}|\bar{w}_1^{\hat{n}}, o_1^{\hat{l}};\ \theta^{(p)}) - q(s_1^{\hat{l}}|o_1^{\hat{l}};\ \theta^{(p)})\right] \log q(o_1^{\hat{l}}|s_1^{\hat{l}};\ \theta) \\ + \sum_{s_1^{\hat{l}}} d'(s_1^{\hat{l}}) \int q(o_1^{\hat{l}}|s_1^{\hat{l}};\ \theta^{(p)}) \log q(o_1^{\hat{l}}|s_1^{\hat{l}};\ \theta)\ do_1^{\hat{l}}, \quad (4.4)\end{aligned}$$

where $d'\left(s_1^{\hat{l}}\right) = \frac{d\left(s_1^{\hat{l}}\right)}{q(\bar{w}_1^{\hat{n}}, o_1^{\hat{l}};\ \theta^{(p)})}$. Using calculus to do the maximization, we get the following update rule to be satisfied by the parameter estimation procedure. Given a parameter estimate $\theta = (\mu_s, \Sigma_s)$, a new estimate $\bar{\theta} = (\bar{\mu}_s, \bar{\Sigma}_s)$ is found so as to satisfy

$$\begin{aligned}\bar{\theta} : \sum_{s_1^{\hat{l}}} \left[q(s_1^{\hat{l}}|\bar{w}_1^{\hat{n}}, o_1^{\hat{l}};\ \theta) - q(s_1^{\hat{l}}|o_1^{\hat{l}};\ \theta)\right] \nabla_\theta \log q(o_1^{\hat{l}}|s_1^{\hat{l}};\ \bar{\theta}) \\ + \sum_{s_1^{\hat{l}}} d'\left(s_1^{\hat{l}}\right) \int q(o_1^{\hat{l}}|s_1^{\hat{l}};\ \theta) \nabla_\theta \log q(o_1^{\hat{l}}|s_1^{\hat{l}};\ \bar{\theta})\ do_1^{\hat{l}} = 0. \quad (4.5)\end{aligned}$$

Here, $O = o_1^{\hat{l}}$ is the acoustic observation vector sequence and $\bar{W} = \bar{w}_1^{\hat{n}}$ is the corresponding word sequence, i.e. the training data. The parameter $d'\left(s_1^{\hat{l}}\right)$ leads to the well-known MMI constant as $D_s = \sum_{s_1^{\hat{l}}:s(\tau)=s} d'(s_1^{\hat{l}})$.

4.5 Parameter estimation

Discriminative training involves an objective function (4.2) which is optimized by some sort of parameter update rule (4.5). The objective function measures how well the current Gaussian parameter set $\theta = (\mu_s, \Sigma_s)$ classifies the training data. The update algorithm alters the system parameters to incrementally improve the objective score. The calculation of the objective function and the re-estimation of the Gaussian parameters are repeated until the objective score converges to an optimum value.

Next we will show how to derive the update equations for the model parameters $\theta = (\mu_s, \Sigma_s)$, mean and the corresponding variance, (assuming diagonal covariance matrices) under the CML framework.

4.5.1 Mean reestimation derivation

The gradient of $\log q(o_1^l|s_1^l;\ \theta)$ with respect to the parameter component μ_s is given by (ignoring all terms independent of μ_s)

$$\begin{aligned}
\nabla_{\mu_s} \log q(o_1^l|s_1^l;\ \theta) &= \nabla_{\mu_s} \sum_{\tau=1}^{l} \left(-\frac{1}{2} \left(-{o_\tau}^T \Sigma_s^{-1} \mu_s - \mu_s^T \Sigma_s^{-1} o_\tau + \mu_s^T \Sigma_s^{-1} \mu_s \right) \right) 1_{\{s\}}(s(\tau)) \\
&= \sum_{\tau=1}^{l} \left(\Sigma_s^{-1} o_\tau - \Sigma_s^{-1} \mu_s \right) 1_{\{s\}}(s(\tau)) \\
&= \sum_{\tau=1}^{l} \left(\Sigma_s^{-1} (o_\tau - \mu_s) \right) 1_{\{s\}}(s(\tau))
\end{aligned}$$

Substituting into equation (4.5) gives

$$\begin{aligned}
&\left[\sum_{\tau=1}^{\hat{l}} \left(\gamma_s(\tau;\ \theta^{(p)}) - \gamma_s^g(\tau;\ \theta^{(p)}) \right) \right] \Sigma_s^{-1} \left(o_\tau - \bar{\mu}_s \right) \\
&\qquad\qquad\qquad + D_s \int q(o_1^{\hat{l}}|s_1^{\hat{l}};\ \theta^{(p)}) \Sigma_s^{-1} \left(o_\tau - \bar{\mu}_s \right) do_1^{\hat{l}} = 0.
\end{aligned}$$

We next define $\gamma_s'(\tau; \theta^{(p)}) = \gamma_s(\tau; \theta^{(p)}) - \gamma_s^g(\tau; \theta^{(p)})$. Here, $\gamma_s(\tau; \theta^{(p)}) = q_{s_\tau}(s|\bar{w}_1^{\hat{n}}, o_1^{\hat{l}}; \theta^{(p)})$ is the conditional occupancy probability of state s at time τ given the training acoustics and transcription, and $\gamma_s^g(\tau; \theta^{(p)}) = q_{s_\tau}(s|o_1^{\hat{l}}; \theta^{(p)})$ is the conditional occupancy probability of state s at time τ given only the training acoustic data.

Rearranging the equation above we get:

$$\left[\sum_{\tau=1}^{\hat{l}} \gamma_s'(\tau;\ \theta^{(p)})\right] \Sigma_s^{-1}\left(o_\tau - \bar{\mu}_s\right)$$
$$+ D_s\Sigma_s^{-1}\left(\int_o q(o_1^{\hat{l}}|s_1^{\hat{l}};\ \theta^{(p)})o_\tau do_1^{\hat{l}} - \int_o q(o_1^{\hat{l}}|s_1^{\hat{l}};\ \theta^{(p)})\bar{\mu}_s do_1^{\hat{l}}\right) = 0$$

$$\left[\sum_{\tau=1}^{\hat{l}} \gamma_s'(\tau;\ \theta^{(p)})\right] \Sigma_s^{-1}\left(o_\tau - \bar{\mu}_s\right) + D_s\Sigma_s^{-1}\left(\mu_s - \bar{\mu}_s\right) = 0 \qquad (4.6)$$

$$\left[\sum_{\tau=1}^{\hat{l}} \gamma_s'(\tau;\ \theta^{(p)})\Sigma_s^{-1}o_\tau + D_s\Sigma_s^{-1}\mu_s\right] = \left[\sum_{\tau=1}^{\hat{l}} \gamma_s'(\tau;\ \theta^{(p)}) + D_s\right]\Sigma_s^{-1}\bar{\mu}_s$$

Finally the estimate for $\bar{\mu}_s$ is given by

$$\bar{\mu}_s = \frac{\sum_\tau \gamma_s'(\tau;\ \theta^{(p)})o_\tau + D_s\mu_s}{\sum_\tau \gamma_s'(\tau;\ \theta^{(p)}) + D_s} \qquad (4.7)$$

4.5.2 Variance reestimation derivation

The gradient of $\log q(o_1^l|s_1^l;\ \theta)$ with respect to the parameter component Σ_s^{-1} is given by (ignoring all terms independent of Σ_s^{-1})

$$\begin{aligned}\nabla_{\Sigma_s^{-1}}\log q(o_1^l|s_1^l;\ \theta) &= \nabla_{\Sigma_s^{-1}}\sum_{\tau=1}^{l}\left(\log|\Sigma_s^{-1}| - \left(o_\tau - \mu_s\right)\Sigma_s^{-1}\left(o_\tau - \mu_s\right)^T\right)\\ &= \sum_{\tau=1}^{l}\left(\Sigma_s - \left(o_\tau - \mu_s\right)\left(o_\tau - \mu_s\right)^T\right)\end{aligned}$$

$$\sum_{\tau=1}^{\hat{l}} \gamma_s'(\tau;\ \theta^{(p)})\left(\bar{\Sigma}_s - \left(o_\tau - \bar{\mu}_s\right)\left(o_\tau - \bar{\mu}_s\right)^T\right)$$
$$+ D_s\int q(o_1^{\hat{l}}|s_1^{\hat{l}};\ \theta^{(p)})\left(\bar{\Sigma}_s - \left(o_\tau - \bar{\mu}_s\right)\left(o_\tau - \bar{\mu}_s\right)^T\right)do_1^{\hat{l}} = 0.$$

Calculating the integral in the previous equation gives us:

$$\begin{aligned}
\bar{\Sigma}_s - \int_o & q(o_1^{\hat{l}}|s_1^{\hat{l}};\theta^{(p)})\left(o_\tau - \bar{\mu}_s\right)\left(o_\tau - \bar{\mu}_s\right)^T do_1^{\hat{l}} \\
&= \bar{\Sigma}_s - \int_o q(o_1^{\hat{l}}|s_1^{\hat{l}};\theta^{(p)}) o_\tau o_\tau^T do_1^{\hat{l}} + 2\int_o q(o_1^{\hat{l}}|s_1^{\hat{l}};\theta^{(p)}) o_\tau \bar{\mu}_s{}^T do_1^{\hat{l}} - \bar{\mu}_s\bar{\mu}_s{}^T \\
&\qquad = \bar{\Sigma}_s - \left(\Sigma_s + \mu_s\mu_s{}^T\right) + \mu_s\bar{\mu}_s{}^T + \bar{\mu}_s\mu_s{}^T - \bar{\mu}_s\bar{\mu}_s{}^T
\end{aligned}$$

Substituting the integral into (4.5.2) yields:

$$\begin{aligned}
&\left[\sum_{\tau=1}^{\hat{l}} \gamma_s'(\tau;\ \theta^{(p)})\right]\left(\bar{\Sigma}_s - \left(o_\tau - \bar{\mu}_s\right)\left(o_\tau - \bar{\mu}_s\right)^T\right) \\
&\qquad\qquad + D_s\left(\bar{\Sigma}_s - \left(\Sigma_s + \mu_s\mu_s{}^T\right) + \mu_s\bar{\mu}_s{}^T + \bar{\mu}_s\mu_s{}^T - \bar{\mu}_s\bar{\mu}_s{}^T\right) = 0
\end{aligned}$$

Rearranging the previous equation gives us:

$$\begin{aligned}
&\left[\sum_{\tau=1}^{\hat{l}} \gamma_s'(\tau;\ \theta^{(p)}) + D_s\right]\bar{\Sigma}_s = \\
&= \sum_{\tau=1}^{\hat{l}} \gamma_s'(\tau;\ \theta^{(p)})\left(\left(o_\tau - \bar{\mu}_s\right)\left(o_\tau - \bar{\mu}_s\right)^T\right) + D_s\left(\Sigma_s + \mu_s\mu_s{}^T\right) - D_s\left(\mu_s\bar{\mu}_s{}^T + \bar{\mu}_s[\mu_s - \bar{\mu}_s]^T\right)
\end{aligned}$$

Finally the estimate for $\bar{\Sigma}_s$ is given by

$$\begin{aligned}
\bar{\Sigma}_s &= \frac{\sum_\tau \gamma_s'(\tau;\ \theta^{(p)})\left(\left(o_\tau - \bar{\mu}_s\right)\left(o_\tau - \bar{\mu}_s\right)^T\right) + D_s\left(\Sigma_s + \mu_s\mu_s{}^T\right) - D_s\left(\mu_s\bar{\mu}_s{}^T + \bar{\mu}_s[\mu_s - \bar{\mu}_s]^T\right)}{\sum_\tau \gamma_s'(\tau;\ \theta^{(p)}) + D_s} \\
&= \frac{\sum_\tau \gamma_s'(\tau;\ \theta^{(p)}) o_\tau o_\tau^T + D_s\left(\Sigma_s + \mu_s\mu_s{}^T\right) - \bar{\mu}_s\left(D_s\mu_s{}^T + \sum_\tau \gamma_s'(\tau;\ \theta^{(p)}) o_\tau^T\right)}{\sum_\tau \gamma_s'(\tau;\ \theta^{(p)}) + D_s} + \\
&\qquad + \frac{-\left(D_s\mu_s + \sum_\tau \gamma_s'(\tau;\ \theta^{(p)}) o_\tau\right)\bar{\mu}_s{}^T + \left(\sum_\tau \gamma_s'(\tau;\ \theta^{(p)}) + D_s\right)\bar{\mu}_s\bar{\mu}_s{}^T}{\sum_\tau \gamma_s'(\tau;\ \theta^{(p)}) + D_s}
\end{aligned}$$

Finally by noticing that

$$\bar{\mu}_s = \frac{\sum_\tau \gamma_s'(\tau;\ \theta^{(p)}) o_\tau + D_s\mu_s}{\sum_\tau \gamma_s'(\tau;\ \theta^{(p)}) + D_s}$$

we have

$$\bar{\Sigma}_s = \frac{\sum_\tau \gamma_s'(\tau;\ \theta^{(p)}) o_\tau o_\tau^T + D_s\left(\Sigma_s + \mu_s\mu_s{}^T\right)}{\sum_\tau \gamma_s'(\tau;\ \theta^{(p)}) + D_s} - \bar{\mu}_s\bar{\mu}_s{}^T. \tag{4.8}$$

The above equations (4.7) and (4.8) show that the conventional MMI gaussian parameter estimation algorithm proposed by Normandin using the EBW, can alternatively be derived as a maximization of the CML auxiliary function.

4.5.3 Calculating D_s

The speed of convergence of MMI using the update equations (4.7) and (4.8) is related to the value of the constant D_s. Small values of D_s result in a faster rate of convergence. Using a single global value of D_s can lead to very slow convergence. In practice a useful lower bound on D_s is the value which ensures that all variances remain positive in all dimensions as suggested by Woodland and Povey[84].

In our case a Gaussian specific value D_s was used. It was set at the maximum of i) twice the value necessary to ensure positive variance updates for all dimensions of the Gaussian; or ii) twice the denominator $\gamma_s^g(\theta^{(p)})$, numerator $\gamma_s(\theta^{(p)})$ state occupancies whichever is greatest.

By substituting (4.7) into (4.8), the condition of positive variances $\Sigma_s > 0$ [84], leads to inequalities which are quadratic of the form

$$\alpha {D_s}^2 + \beta D_s + \gamma > 0 \tag{4.9}$$

with

$\alpha = \Sigma_s$,

$\beta = \sum_\tau \gamma_s'(\tau;\ \theta^{(p)}) o_\tau^2 + \sum_\tau \gamma_s'(\tau;\ \theta^{(p)})(\Sigma_s + {\mu_s}^2) - 2\mu_s \sum_\tau \gamma_s'(\tau;\ \theta^{(p)}) o_\tau$ and

$\gamma = [\sum_\tau \gamma_s'(\tau;\ \theta^{(p)}) o_\tau^2] \sum_\tau \gamma_s'(\tau;\ \theta^{(p)}) - [\sum_\tau \gamma_s'(\tau;\ \theta^{(p)}) o_\tau]^2$

Since α is positive, the inequality above is valid when $D_s \in (-inf, D_1] \cup [D_2, +inf)$ where D_1 and D_2 are the roots of the quadratic equation $\alpha {D_s}^2 + \beta D_s + \gamma = 0$. These inequalities should be solved explicitly to give the largest constant D_s ensuring positive variance. Therefore an appropriate value of D_s should be found by solving the system of quadratic inequalities, such that all 39 elements of the diagonal variance vector are positive for each gaussian of the HMM model set.

4.6 Use of lattices in Discriminative training

Optimization of equation (4.2) requires the maximization of the numerator term, which is identical to the MLE objective function, while simultaneously minimising the denominator term which can be written as:

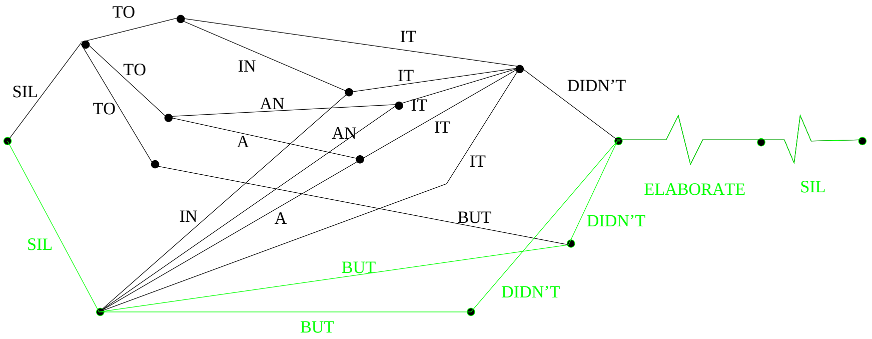
$$P_\theta(O_r|M_{\mathcal{W}}) = \sum_{W_r' \in \mathcal{W}} P_\theta(O_r|M_{W_r'})P(W_r') \tag{4.10}$$

This minimization is done over sets $\mathcal{W}$ of alternative word sequences representative of the recognition errors made by the machine. The set $\mathcal{W}$ of alternative word hypotheses W' can be approximated either by N-best lists or lattices. During parameter reestimation $\mathcal{W}$ is used in the probability $\gamma_s^g(\tau;\ \theta^{(p)})$ of being at state s at time τ given only the training acoustic data, which is explicitly given by:

$$\gamma_s^g(\tau;\ \theta^{(p)}) = \frac{\left\{\sum_{W' \in \mathcal{W}} P(W')P(O|W')\gamma_{s,W'}(\tau;\ \theta^{(p)})\right\}}{\left[\sum_{W'' \in \mathcal{W}} P(O|W'')P(W'')\right]} \tag{4.11}$$

By using N-best lists, time alignment and forward/backward estimation have to be carried out for every word sequence in the N-best list. For small vocabulary applications, this calculation(4.11) is feasible. On the other hand, for LVCSR tasks it would be very time consuming to perform forward/backward estimation of (4.11) during each training iteration for every word sequence in the N-best list. Because many word sequences differ only in few words much of the calculations done are redundant. This redundancy can be avoided by using lattices for discriminative training as introduced by Valchev [77].

The use of lattices as a constraining word graph [84, 75] forms the basis of most current state-of-the-art MMIE training algorithms, reducing the search space between iterations, .thus making discriminative training of LVCSR systems feasible. A word lattice forms a compact representation of many different sentence hypotheses and hence provides an efficient representation of the confusion data needed for discriminative training.



The lattice shown above contains the most likely word hypotheses as determined by the speech recognizer. The paths shown in green correspond to the correct transcription of the utterance and encode slightly different start/end times.

Lattices are generated as a by-product of the recognition process only once, and used for several iterations of MMIE training to speed up computations. We assume that lattice coverage does not change during parameter re-estimation, therefore there is no need to be regenerated again. For large vocabulary tasks, unconstrained recognition for the whole training set in every iteration of discriminative training is clearly unrealistic in terms of computation time.

The probability $\gamma_s^g(\tau;\ \theta^{(p)})$ can be estimated by summing over all paths in the lattice that pass through the triphone that state s belongs and normalizing by $P(O)$. This can be done by using the lattice forward/backward probabilities of the triphone that state s belongs.

4.7 MMIE implementation

Fig. 4.1 shows the MMIE implementation used in our experiments. The acoustic models used for discriminative training have the same complexity as those used for ML training, i.e same number of mixtures per state, therefore the number of parameters is the same for each training method. It involves the following steps:

1. Allocate and zero accumulators for all parameters of all HMMs. The numerator statistics for state s, are $\sum_\tau \gamma_s(\tau;\ \theta^{(p)})o_\tau$ and $\sum_\tau \gamma_s(\tau;\ \theta^{(p)})o_\tau^2$ for the mean and variance update equations. The denominator statistics are $\sum_\tau \gamma_s^g(\tau;\ \theta^{(p)})o_\tau$ and $\sum_\tau \gamma_s^g(\tau;\ \theta^{(p)})o_\tau^2$ respectively.

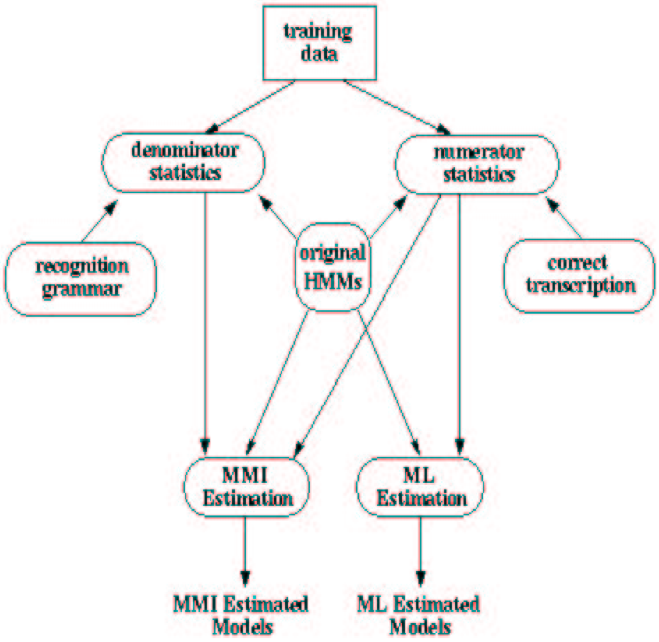


Figure 4.1: MMIE implementation

2. Get the next training utterance.

3. The numerator lattice is produced by aligning the acoustic data against a network of HMMs built according to the "correct" transcription. The numerator lattice may contain alternative pronunciations of the same word.

4. The denominator lattice corresponds to running an unconstrained recognition pass. As mentioned before, we assume that lattice coverage does not change, therefore there is no need to be regenerated again.

5. Use the forward and backward probabilities to compute the probabilities $\gamma_s(\tau;\theta^{(p)})$ and $\gamma_s^g(\tau;\ \theta^{(p)})$ of state occupation at each time frame. Collect numerator/denominator statistics.

6. Repeat from step 2 until all training utterances have been processed.

7. Use the accumulators to calculate new parameter estimates for all of the HMMs

according to equations (4.7), (4.8) and (4.9).

These steps (except for step 4) are repeated as many times as is necessary to achieve the required convergence.

4.8 Improving Generalization

Despite the discussion so far, there are a number of modeling issues that the basic theory does not address, but which are important when building discriminative models. An important issue in MMIE training is the ability to generalize to unseen test data. While MMIE training often greatly reduces training set error, the reduction in error rate on an independent test set is normally much less, the generalization performance is poorer. Furthermore, as with all statistical modeling approaches, the more complex the model, the poorer the generalization. We have considered two methods of improving generalization that both increase the amount of confusable data processed during training: weaker language models and acoustic model scaling.

4.8.1 Weaker language models

It was shown in[68, 69] that improved test-set performance could be obtained using a unigram LM during MMIE training, even though a trigram was used during recognition, which means that the language model used for discriminative training has to be less accurate than the optimal language model used for decoding/evaluation. The aim is to provide more focus on the discrimination provided by the acoustic model by loosening the language model constraints [68, 69]. In this way, more confusable data is generated which improves generalization.

4.8.2 Acoustic Scaling

When combining the likelihoods from an HMM-based acoustic model and the LM it is necessary to scale the LM log probability. This happens because, due to invalid modeling assumptions, the HMM underestimates the probability of acoustic vector sequences. An alternative to LM scaling is to multiply the acoustic model log likelihood values by the inverse of the LM scale factor (acoustic model scaling). While this produces the same effect as language model scaling when considering only a single word sequence as for Viterbi decoding, when likelihoods from different sequences are added, such as in the forward-backward algorithm or for the denominator of (4.2), the effects of LM and acoustic model scaling are very different.

If language model scaling is used, one particular state-sequence tends to dominate the likelihood at any point in time and hence dominates any sums using path likelihoods. However, if acoustic scaling is used, there will be several paths that have fairly similar likelihoods which make a non-negligible contribution to the summations. Therefore acoustic model scaling [77] tends to increase the confusable data set in training by broadening the posterior distribution of state occupation that is used in the update equations (4.7) and (4.8). This increase in confusable data also leads to improved generalization performance.

4.9 Summary

This chapter has described an implementation of MMIE for HMM gaussian model parameter estimation using Gunawardana's CML auxiliary function which does not require the discrete approximation to the Gaussian density of Normandin.

Since the MMIE criterion is also the criterion used in MAP decoding, MMIE is related to reducing the sentence error rate in the training set. Ideally, during recognition of the unseen test set the number of errors will be reduced thus improving recognition performance.

Finally a discriminative training framework based on the use of lattices was presented which will be used to perform MMIE training efficiently in our experiments.

Chapter 5

Speaker Adaptation

In the previous two chapters we discussed how to estimate the acoustic model parameters θ, under the Maximum Likelihood and the Maximum Mutual Information Estimation criteria. In this chapter, we discuss adaptation techniques used to compensate for differences between the training and testing conditions. We briefly present the popular adaptation techniques such as MLLR and SAT that are based on Maximum Likelihood estimation.

However discriminative optimization criteria can be more effective in reducing the word error rate than maximum likelihood estimation and hence are of interest. In the previous chapter we used the CML auxiliary function in order to estimate the gaussian model parameters. It is also possible to formulate discriminative estimation procedures for the estimation of the linear transforms used in speaker adaptive training. We will show how the CML auxiliary function can be applied to the re-estimation of the Gaussian model parameters and the linear transforms used in speaker adaptive training, therefore obtaining fully discriminative procedures.

5.1 Speaker Independent and Speaker Dependent Models

Typical state-of-the-art large vocabulary conversational speech recognition (LVCSR) systems use a single model, developed on data from a large number of speakers to cover the variance across dialects, speaking styles, etc. With this, we expect our systems to generalize well to any particular speaker. Although the training and recognition techniques described previously can produce high performance recognition systems, these systems can be improved upon by customizing the HMMs to the characteristics of a particular speaker.

The voice characteristics of different speakers vary widely due to many reasons such as: gender, age, height(vocal tract length), speaking style, emotional condition, or accent(native vs non-native). Speaker independent(SI) systems require complex acoustic models from a large amount of data(more training time) to capture these characteristics and model the inter-speaker variability. In training, by averaging statistics over a number of speakers, speaker independent models lose specificity and, along with it, discriminating capabilities. As a result there are speakers who are poorly modeled using this paradigm.

One solution is to use speaker-dependent(SD) models. Speaker dependent systems are trained to recognize speech from a single speaker, whereas speaker independent systems are capable of recognizing speech from any speaker. These systems achieve better recognition performance because of the limited variability in the speech signal coming from the same speaker. However, in order to achieve the desired level of accuracy, speaker-dependent systems require a large amount of training data from each individual speaker using the system. This is often undesirable, which is why *speaker adaptation* techniques have been developed.

5.2 Speaker adaptation

The purpose of speaker adaptation is to enable the speaker-independent models to capture the characteristics of particular speakers using a small amount of enrollment or adaptation data. The end result of this procedure is a speaker adapted system. Speaker Adaptation has been shown to be effective in improving the performance of speaker independent (SI) LVCSR systems by adapting the system to the test set. Many approaches have been developed which try to produce this effect.

Current speaker adaptation techniques for HMM-based speech recognition systems fall into two basic categories. The first of these employs methods which transform the input speech[32, 26, 43] of the new speaker to a vector space that is common with the training speech. These are known as spectral mapping techniques. Second are methods which transform the model parameters[46, 47, 15] to better match the characteristics of the adaptation data. These techniques are known as model mapping approaches and are the focus of this chapter.

5.2.1 Model Mapping Techniques

The most popular model-based adaptation techniques can be grouped into three families[82]: Maximum a posteriori (MAP) techniques[45, 27, 14], linear transformation techniques including MLLR [46, 47] and speaker clustering based techniques including CAT[23, 78] and eigenvoice [20, 19]. The most effective technique of adaptation will depend on the application.

Because it is difficult to reliably estimate a large number of parameters, linear transforms have been used extensively during adaptation of HMM-based ASR systems. By only estimating the transformation matrix or matrices on the adaptation data, the number of parameters that have to be estimated is very limited. This results in a robust estimation, while preserving the possibility of effectively adapting the HMM's distribution densities to the observed adaptation data.

One approach is the constrained adaptation of the acoustic models to the speaker, the channel, or the task, and this is termed Maximum Likelihood Linear Regression

(MLLR) [46, 47]. In MLLR, a transform is applied to the Gaussian model parameters in the estimation of the state independent observation distributions in order to best match the specific conditions of interest. These can be speech from an individual speaker or a particular acoustic environment, causing large variations in the realized speech waveform due to speaker variability, mood, environment, etc. MLLR finds the optimal affine transformation by maximizing the likelihood of the adaptation data.

Adaptation can also be applied to the speakers in the training set to produce matched conditions with the test set, and this is termed Maximum Likelihood (ML) Speaker Adaptive Training (SAT) [1]. SAT is a well-established technique aiming at reducing inter-speaker variability within the training set. SAT is an iterative procedure that generates a set of speaker independent (SI) Gaussian parameters along with matched speaker dependent transforms for all the speakers in the training set.

Another popular framework for doing speaker adaptation is the Bayesian-MAP (maximum a-posteriori) approach[45, 27, 14]. In MAP a prior density $\pi(\theta)$ on the parameter set θ is used to extend the EM auxiliary function. Thus if we know what the parameters of the model are likely to be using the prior knowledge, we may be able to get a decent estimate given the limited adaptation data available.

5.3 MLLR Adaptation

Maximum likelihood linear regression or MLLR[46, 47] computes a set of transformations that will reduce the mismatch between an initial model set and the adaptation data. MLLR estimates a set of linear transformations for the mean and/or variance parameters of a Gaussian HMM system.

The effect of these transformations is to shift the component means and alter the variances in the initial system so that each state in the HMM system is more likely to generate the adaptation data. The transformation matrix is usually applied to the mean vector of each Gaussian, because they define major characteristics of the distributions. Covariance adaptation is less commonly used and it is less effective than mean adaptation[24].

The new estimate of the adapted mean is given by:

$$\hat{\mu} = T\xi = A\mu + b, \tag{5.1}$$

where T is the $n \times (n+1)$ transformation matrix (where n is the dimensionality of the data) and ξ is the extended mean vector,

$$\xi = [1\ \mu_1\ \mu_2\ \dots\ \mu_n\ ]^T$$

Hence T can be decomposed into

$$T = [\ b\ A\ ] \tag{5.2}$$

where A represents an $n \times n$ transformation matrix and b represents a bias vector.

The transformation matrix T is obtained by solving a maximisation problem using the *Expectation-Maximization* (EM) technique(shown in section 5.5.2), and can be either a full unconstrained matrix or a block-diagonal one. By using full regression matrices we can model the correlation among the mean vector parameters more precisely and thus provide better description of speaker characteristics. However the large number of parameters ($n^2 + n$, in the full case) makes robust estimation of full regression matrices very difficult when the amount of adaptation data is small.

5.4 Regression Class Tree

MLLR adaptation can be applied in a very flexible manner, depending on the amount of adaptation data that is available. When performing speaker adaptation, the more data that is available, the more parameters we can reliably estimate. If a small amount of data is available then a *global* adaptation transform can be generated. A global transform (as its name suggests) is applied to every Gaussian component in the model set.

However as more adaptation data becomes available, improved adaptation is possible by increasing the number of transformations[47, 86]. Each transformation is now more specific and applied to certain groupings of Gaussian components. MLLR implementation in the HTK Toolkit [85] makes use of a *regression class tree* (Fig 5.1).

The regression class tree is constructed so as to cluster together gaussian mixture components that are close in the acoustic space and must be transformed in a similar way. The number of transformations to be estimated can be chosen according to the amount and type of adaptation data that is available. The tying of each transformation across a number of mixture components makes it possible to adapt distributions for which there were no observations at all. With this process all models can be adapted and the adaptation process is dynamically refined when more adaptation data becomes available.

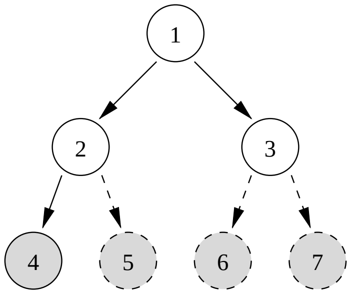


Figure 5.1: Regression Tree

5.5 Speaker Adaptive Training

Speaker Adaptive Training (SAT) [1] has been shown to be effective in improving the performance of speaker independent (SI) LVCSR systems by reducing the inter-speaker variability within the training set. The variability in SI acoustic models is attributed to both phonetic variation and variation among the speakers of the training population, that is independent of the information content of the speech signal. In SAT a transform is used to represent a speaker, both during training and testing. For each speaker k, a transform matrix $T_r^{(k)} = [b_r^{(k)} \; A_r^{(k)}]$ is applied to the (extended) mean vector $\xi_s = [1 \; \mu_s^T]^T$ according to

$$\bar{\mu}_s^{(k)} = T_r^{(k)} \xi_s = A_r^{(k)} \mu_s + b_r^{(k)}. \tag{5.3}$$

Here $A_r^{(k)}$ and $b_r^{(k)}$ are respectively the speaker-dependent transformation matrix and the additive bias associated with speaker k.

Due to (5.3), the emission density (2.9) of state s is reparametrized for each speaker $k = 1, 2, ..., K$ as

$$q(o_\tau|s, k; \theta) = \frac{1}{\sqrt{(2\pi)^n|\Sigma_s|}} e^{-\frac{1}{2}\left(o_\tau - T_r^{(k)}\xi_s\right)^T \Sigma_s^{-1}\left(o_\tau - T_r^{(k)}\xi_s\right)}.$$

To avoid introducing more parameters than can be reliably estimated, transformations are tied across sets of states. Thus the extended speaker dependent transformation matrix $T_r^{(k)}$ is associated with a group of states $S_r = \{s|\mathcal{R}(s) = r\}$ for classes $r = 1, \ldots, R$. The function S_r gives the set of mixtures belonging to the same regression class r.

Under this framework the augmented state dependent parameter set is defined as $\theta = (T_r^{(k)}, \mu_s, \Sigma_s)$, for all speakers k in the training set. Next we briefly show how to compute the speaker dependent transforms and speaker independent Gaussian parameters of the state dependent distributions under the ML criterion.

5.5.1 Maximum Likelihood Estimation

We first maximize the ML criterion with respect to the speaker dependent affine transforms while keeping the speaker independent means and variances fixed to their current values. Subsequently, we compute the speaker independent means and variances using the updated values of the speaker dependent affine transforms. All these estimation steps are done under the ML criterion.

The estimation procedure is done by maximizing the following auxiliary function:

$$Q(\bar{\theta}|\theta) = -\frac{1}{2}\sum_{k,r}\sum_{s\in S(r)}\sum_{\tau=1}^{T}\gamma_s(\tau;\theta)\left[log|\Sigma_s| + (o_\tau - \bar{\mu}_s^{(k)})^T\Sigma_s^{-1}(o_\tau - \bar{\mu}_s^{(k)})\right] + C \quad (5.4)$$

where C is a constant independent of θ coefficients and o_τ is the adaptation data. In SAT the training data are collected from a population of K speakers. Before training all utterances are partitioned according to speaker identity. To incorporate information about the speaker identities into the ML framework we denote by $\{\tau :$

$\hat{k}_\tau = k\}$, the sequence of feature vectors o_τ belonging to speaker k. The parameter update equation then becomes:

$$\bar{\theta} : \sum_{k,r} \sum_{s \in S_r} \sum_{\tau : \hat{k}(\tau) = k} \gamma_s(\tau;\theta) \nabla_\theta \log q(o_\tau | s, k;\ \bar{\theta}) = 0\ . \tag{5.5}$$

where $\gamma_s(\tau;\theta) = q_{s_\tau}(s | \hat{w}_1^{\hat{n}}, o_1^{\hat{l}}, k; \theta)$ is the conditional occupancy probability of state s at time τ given the training acoustics and the reference transcription.

5.5.2 Estimation of SAT Transforms

With the HMM parameters fixed, the parameter update relationship of equation (5.5) can be expressed as:

$$\bar{T}_r^{(k)} : \sum_{s \in S_r} \sum_{\tau : \hat{k}(\tau) = k} \gamma_s(\tau;\theta) \cdot \nabla_{T_r^{(k)}} \log q(o_\tau | s, k; \bar{T}_r^{(k)}, \mu_s, \Sigma_s) = 0\ . \tag{5.6}$$

The gradient of logarithm of the emission density q with respect to $T_r^{(k)}$ is found to be

$$\nabla_{T_r^{(k)}} \log q(o_\tau | s, k; \theta) = \Sigma_s^{-1} \left(o_\tau - T_r^{(k)} \xi_s \right) \xi_s^T$$

Substituting this into equation (5.6) it follows that the new transform estimates $\bar{T}_r^{(k)}$ should satisfy:

$$\sum_{s \in S_r} \Sigma_s^{-1} \sum_{\tau : \hat{k}(\tau) = k} \gamma_s(\tau;\theta) o_\tau \xi_s^T = \sum_{s \in S_r} \sum_{\tau : \hat{k}(\tau) = k} \gamma_s(\tau;\theta) \Sigma_s^{-1} \bar{T}_r^{(k)} \xi_s \xi_s^T \tag{5.7}$$

Here, the state occupancies $\gamma_s(\tau;\theta)$ are found via counts accumulated for each speaker under the initial parameters $(T_r^{(k)}, \mu_s, \Sigma_s)$.

5.5.3 Gaussian Parameter Estimation

The state independent Gaussian mean and variance parameters for ML-SAT are estimated under the ML criterion with the use of the parameter update relationship of equation (5.5) using the updated values of the speaker dependent affine transforms

$\bar{T}_r^{(k)}$. The parameter set is $\tilde{\theta} = (\bar{T}_r^{(k)}, \mu_s, \Sigma_s)$. The derivation of the update formulas involves the gradient of the reparametrized emission density with respect to μ_s and Σ_s^{-1} in equation (5.5). Subsequently, we solve for $\bar{\mu}_s$ and $\bar{\Sigma}_s$.

Mean estimation

From equation (5.5), the Gaussian means are found as:

$$\bar{\mu}_s : \sum_k \sum_{\tau:\hat{k}(\tau)=k} \gamma_s(\tau;\tilde{\theta}) \cdot \nabla_{\mu_s} \log q(o_\tau|s,k;\bar{T}_r^{(k)}, \bar{\mu}_s, \Sigma_s) = 0 \ . \tag{5.8}$$

In a similar fashion by taking the derivative with respect to the (SI) mean and taking into account all the training speakers we have

$$\nabla_{\mu_s} \log q(o_\tau|s;\ \tilde{\theta}) = \sum_k \bar{A}_r^{(k)T} \Sigma_s^{-1} \left(o_\tau - \bar{b}_r^{(k)} - \bar{A}_r^{(k)} \mu_s \right) \ . \tag{5.9}$$

Substituting the above expression for the gradient into the update rule of equation (5.8), speaker independent means are then reestimated as

$$\bar{\mu}_s = \left(\sum_k \sum_{\tau:\hat{k}(\tau)=k} \gamma_s(\tau;\tilde{\theta}) \bar{A}_r^{(k)T} \Sigma_s^{-1} \bar{A}_r^{(k)} \right)^{-1} \times \\ \sum_k \bar{A}_r^{(k)T} \Sigma_s^{-1} \sum_{\tau:\hat{k}(\tau)=k} \gamma_s(\tau;\tilde{\theta}) \left(o_\tau - \bar{b}_r^{(k)} \right) . \tag{5.10}$$

The term to be inverted need only be accumulated once for all speakers, thus making the parallel execution of ML-SAT algorithm across a network of machines feasible.

Variance estimation

From equation (5.5), the Gaussian variance is found as:

$$\bar{\Sigma}_s^{-1} : \sum_k \sum_{\tau:\hat{k}(\tau)=k} \gamma_s(\tau;\tilde{\theta}) \cdot \nabla_{\Sigma_s^{-1}} \log q(o_\tau|s;\bar{T}_r^{(k)}, \bar{\mu}_s, \bar{\Sigma}_s) = 0 \ . \tag{5.11}$$

By taking the derivative with respect to the speaker independent variance Σ_s^{-1} we have:

$$\nabla_{\Sigma_s^{-1}} \log q(o_\tau|s;\bar{T}_r^{(k)}, \bar{\mu}_s, \Sigma_s) = \Sigma_s - \left(o_\tau - \bar{\mu}_s^{(k)} \right) \left(o_\tau - \bar{\mu}_s^{(k)} \right)^T .$$

Substituting the above expression for the gradient into the update rule of equation (5.11), the speaker independent variances are then reestimated as

$$\bar{\Sigma}_s = \frac{\sum_k \sum_{\tau:\hat{k}(\tau)=k} \gamma_s(\tau;\tilde{\theta})(o_\tau^2 - 2o_\tau\bar{\mu}_s^{(k)} + \bar{\mu}_s^{(k)2})}{\sum_k \sum_{\tau:\hat{k}(\tau)=k} \gamma_s(\tau;\tilde{\theta})} \tag{5.12}$$

where $\bar{\mu}_s^{(k)} = \bar{A}_r^{(k)}\bar{\mu}_s + \bar{b}_r^{(k)}$, are the new speaker dependent means.

5.6 The ML-SAT Algorithm

The derivation described above is a two-stage iterative procedure as shown in Fig 5.2. It can be summarized in the following steps:

1. For each training utterance, the corresponding phone models are concatenated.
2. The forward and backward probabilities for all speakers k, and for all states s and times τ are calculated.
3. Initially, speaker dependent transforms are estimated via equation (5.7) while holding all other HMM parameters fixed at their current values.
4. The speaker dependent transforms $\bar{T}_r^{(k)}$ are applied to obtain the speaker dependent(SD) models. The forward and backward probabilities are calculated again for all speakers k, and for all states s and times τ.
5. For each speaker k and foreach state s and time τ, the probability of state occupation $\gamma_s(\tau;\tilde{\theta})$ and the current observation vector o_τ are used to update the mean and variance accumulators for that state.
6. When all of the training data has been processed, the accumulated statistics are used to calculate new parameter values for all of the HMMs according to (5.10) and (5.12).

These steps can then all be repeated as many times as is necessary to achieve the required convergence.

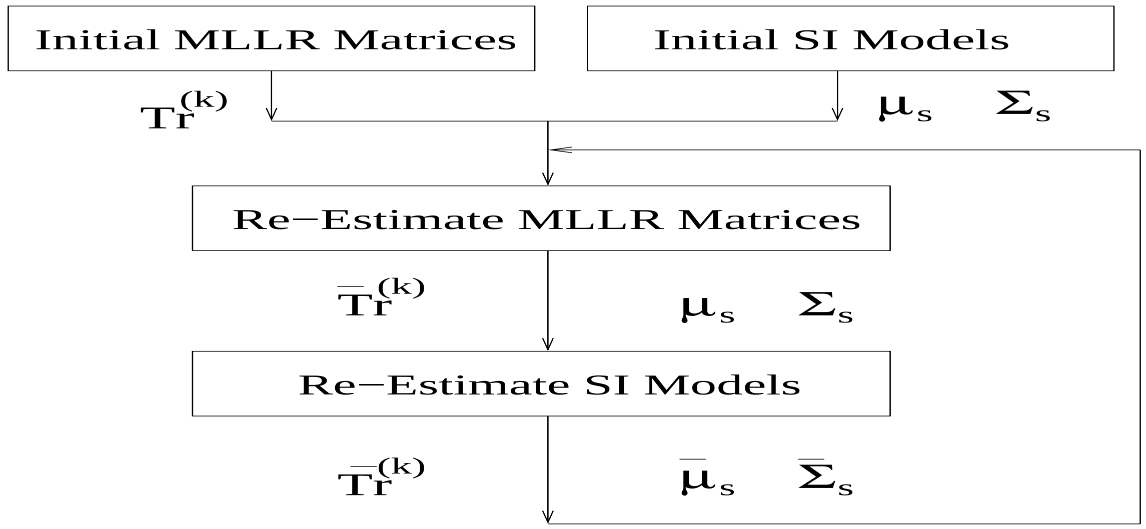

Initial MLLR Matrices
Initial SI Models
$\mathbf{Tr}^{(k)}$
μ_s Σ_s
Re−Estimate MLLR Matrices
$\overline{\mathbf{Tr}}^{(k)}$
μ_s Σ_s
Re−Estimate SI Models
$\overline{\mathbf{Tr}}^{(k)}$
$\overline{\mu}_s$ $\overline{\Sigma}_s$

Figure 5.2: SAT implementation

5.7 Discriminative Linear Transforms

Until recently adaptation techniques have been based on the maximum likelihood (ML) parameter estimation framework. However discriminative optimization criteria can be more effective in reducing the word error rate than maximum likelihood estimation and hence are of interest. As a result, discriminative procedures for the estimation of the linear transforms used in speaker adaptation have been developed. When estimated in this manner they are called Discriminative Linear Transforms (DLT) [75].

One approach to the use of DLTs is Maximum Mutual Information Linear Regression (MMILR) which was introduced by Uebel and Woodland [75, 76], who showed that it can be used for supervised speaker adaptation. Gunawardana and Byrne [35] introduced the Conditional Maximum Likelihood Linear Regression (CMLLR) algorithm and showed that CMLLR can be used for unsupervised speaker adaptation.

Maximum likelihood linear transforms have also been incorporated with MMI training. McDonough et al. [54] combined SAT with MMI by estimating speaker dependent linear transforms under ML and subsequently used MMI for the estimation of the speaker independent HMM Gaussian parameters. This is a hybrid ML/MMI modeling approach. In the next section both the linear transforms and the gaussian parameters used in speaker adaptive training are reestimated under MMIE criteria using the (CML) auxiliary function.

5.8 Discriminative Speaker Adaptive Training

Our objective is to compute both the speaker dependent transforms and the speaker independent parameters of the state dependent distributions used for speaker adaptive training under the CML criterion. As a result we obtain fully discriminative procedures. We call this Discriminative Speaker Adaptive Training (DSAT)[73].

We first maximize the CML criterion with respect to the speaker dependent affine transforms while keeping the speaker independent means and variances fixed to their current values. Subsequently, we compute the speaker independent means and variances using the updated values of the speaker dependent affine transforms. All these estimation steps are done under the CML criterion.

In SAT the training data are collected from a population of K speakers. To incorporate information about the speaker identities into the CML framework, we modify the observed random processes to include a sequence that labels each observation vector by the speaker who uttered it: $(o_1^{\hat{l}}, \hat{k}_1^{\hat{l}}, w_1^{\hat{n}})$. The training objective therefore becomes the maximization of $p(w_1^{\hat{n}}|\ o_1^{\hat{l}}, \hat{k}_1^{\hat{l}}; \theta)$. The parameter update relationship of equation (4.5) can be modified to include the speaker identity as follows:

$$\bar{\theta} : \sum_{s_1^{\hat{l}}} \left[q(s_1^{\hat{l}} | \hat{w}_1^{\hat{n}}, o_1^{\hat{l}}, \hat{k}_1^{\hat{l}};\ \theta) - q(s_1^{\hat{l}} | o_1^{\hat{l}}, \hat{k}_1^{\hat{l}};\ \theta) \right] \cdot \nabla_\theta \log q(o_1^{\hat{l}}, \hat{k}_1^{\hat{l}} | s_1^{\hat{l}};\ \bar{\theta})$$

$$+ \sum_{s_1^{\hat{l}}} d' \left(s_1^{\hat{l}} \right) \int q(o_1^{\hat{l}}, \hat{k}_1^{\hat{l}} | s_1^{\hat{l}};\ \theta) \cdot \nabla_\theta \log q(o_1^{\hat{l}}, \hat{k}_1^{\hat{l}} | s_1^{\hat{l}};\ \bar{\theta})\ do_1^{\hat{l}} = 0\ . \quad (5.13)$$

Using the Markov assumptions we can write $\log q(o_1^{\hat{l}}, \hat{k}_1^{\hat{l}} | s_1^{\hat{l}};\ \bar{\theta})$ as $\sum_{k,r,s} \sum_{\tau=1}^{\hat{l}} \log q(o_\tau | s, k; \bar{\theta}) 1_k(\hat{k}(\tau)) 1_s(s_\tau) 1_r(\mathcal{R}(s))$, where $1_k(\hat{k}(\tau)) = 1$ if $k = \hat{k}(\tau)$, 0 otherwise. Equation (5.13) then becomes:

$$\bar{\theta} : \sum_{k,r}\sum_{s\in S_r}\sum_{\tau:\hat{k}(\tau)=k} \gamma'_s(\tau;\theta)\nabla_\theta \log q(o_\tau|s,k;\ \bar{\theta})$$

$$+\sum_{k,r}\sum_{s\in S_r} D_s^{(k)} \int q(o|s,k;\ \theta)\nabla_\theta \log q(o|s,k;\ \bar{\theta})\ do = 0\ . \quad (5.14)$$

where $\gamma'_s(\tau;\theta) = \gamma_s(\tau;\theta) - \gamma_s^g(\tau;\theta)$ as defined in the previous chapter and $D_s^{(k)} = \sum_{\tau:\hat{k}(\tau)=k}\sum_{s_1^{\hat{l}}:s(\tau)=s} d'(s_1^{\hat{l}})$.

5.8.1 Estimation of DSAT Transforms

With the HMM parameters fixed, the parameter update relationship of equation (5.14) can be expressed as:

$$\bar{T}_r^{(k)} : \sum_{s\in S_r}\sum_{\tau:\hat{k}(\tau)=k} \gamma'_s(\tau;\theta)\cdot\nabla_{T_r^{(k)}} \log q(o_\tau|s,k;\bar{T}_r^{(k)},\mu_s,\Sigma_s)$$

$$+\sum_{s\in S_r} D_s^{(k)} \int q(o_\tau|s,k;T_r^{(k)},\mu_s,\Sigma_s)\nabla_{T_r^{(k)}} \log q(o_\tau|s,k;\bar{T}_r^{(k)},\mu_s,\Sigma_s)\ do = 0\ . \quad (5.15)$$

The gradient of logarithm of the emission density q with respect to $T_r^{(k)}$ can be found as

$$\begin{aligned}\nabla_{T_r^{(k)}} \log q(o_\tau|s,k;\theta) &= \frac{1}{2}\cdot\nabla_{T_r^{(k)}}\left(\left(o_\tau - T_r^{(k)}\xi_s\right)^T \Sigma_s^{-1}T_r^{(k)}\xi_s + \xi_s^T T_r^{(k)^T}\Sigma_s^{-1}o_\tau\right)\\ &= \Sigma_s^{-1}\left(o_\tau - T_r^{(k)}\xi_s\right)\xi_s^T\end{aligned}$$

Substituting this into equation (5.15) gives

$$\sum_{s\in S_r}\Sigma_s^{-1}\left[\sum_{\tau:\hat{k}(\tau)=k}\gamma'_s(\tau;\theta)\left(o_\tau - \bar{T}_r^{(k)}\xi_s\right)\xi_s^T + D_s^{(k)}\int q(o|s,k;T_r^{(k)})\left(o_\tau - \bar{T}_r^{(k)}\xi_s\right)\xi_s^T do\right] = 0$$

from which it follows that the new transform estimates $\bar{T}_r^{(k)}$ should satisfy:

$$\sum_{s\in S_r}\Sigma_s^{-1}\left[\sum_{\tau:\hat{k}(\tau)=k}\gamma'_s(\tau;\theta)o_\tau + D_s^{(k)}T_r^{(k)}\xi_s\right]\xi_s^T = \sum_{s\in S_r}\Sigma_s^{-1}\left[\sum_{\tau:\hat{k}(\tau)=k}\gamma'_s(\tau;\theta) + D_s^{(k)}\right]\bar{T}_r^{(k)}\xi_s\xi_s^T\ .$$

(5.16)

Here, the state occupancies are found via counts accumulated for each speaker under the initial parameters $(T_r^{(k)}, \mu_s, \Sigma_s)$.

5.8.2 Gaussian Parameter Estimation

The state independent Gaussian mean and variance parameters for DSAT are estimated under the CML criterion with the use of the parameter update relationship of equation (5.14). The derivation of the update formulas involves the gradient of the reparametrized emission density with respect to μ_s and Σ_s^{-1} and the calculation of the integral (5.14) . After these steps, we solve for μ_s and Σ_s. The parameter set is $\tilde{\theta} = (\bar{T}_r^{(k)}, \mu_s, \Sigma_s)$.

Mean estimation

From equation (5.14), the Gaussian means are found as:

$$\begin{aligned} \bar{\mu}_s : \sum_k \sum_{\tau:\hat{k}(\tau)=k} \gamma'_s(\tau;\tilde{\theta}) \cdot \nabla_{\mu_s} \log q(o_\tau | s, k; \bar{T}_r^{(k)}, \bar{\mu}_s, \Sigma_s) \\ + \sum_k D_s^{(k)} \int q(o_\tau | s, k; \bar{T}_r^{(k)}, \mu_s, \Sigma_s) \cdot \nabla_{\mu_s} \log q(o_\tau | s, k; \bar{T}_r^{(k)}, \bar{\mu}_s, \Sigma_s) do = 0 \ . \quad (5.17) \end{aligned}$$

In a similar fashion by taking the derivative with respect to the (SI) mean and regarding all the training speakers we have:

$$\begin{aligned} \nabla_{\mu_s} \log q(o_1^l | s_1^l;\ \theta) &= \nabla_{\mu_s} \sum_k \sum_{\tau=1}^{l} \left(-\frac{1}{2} \left(b_r^{(k)} - o_\tau^T \Sigma_s^{-1} A_r^{(k)} \mu_s \right) \right) \\ &+ \nabla_{\mu_s} \sum_k \sum_{\tau=1}^{l} \left(-\frac{1}{2} \left(\mu_s^T {A_r^{(k)}}^T \Sigma_s^{-1} (b_r^{(k)} - o_\tau + \mu_s^T {A_r^{(k)}}^T \Sigma_s^{-1} A_r^{(k)} \mu_s) 1_{\{s\}}(s(\tau)) \right) \right) \\ &= \sum_k \sum_{\tau=1}^{l} \left({A_r^{(k)}}^T \Sigma_s^{-1} \left(o_\tau - b_r^{(k)} - A_r^{(k)} \mu_s \right) \right) 1_{\{s\}}(s(\tau)) \ . \quad (5.18) \end{aligned}$$

Substituting the above expression for the gradient into the update rule of equation (5.17) gives

$$\sum_k \bar{A}_r^{(k)T}\Sigma_s^{-1}\left[\sum_{\tau:\hat{k}(\tau)=k}\gamma'_s(\tau;\tilde{\theta})\left(o_\tau-\bar{\mu}_s^{(k)}\right)+D_s^{(k)}\int q(o_\tau|s,k;\mu_s)\left(o_\tau-\bar{\mu}_s^{(k)}\right)do\right]=0$$

where $\bar{\mu}_s^{(k)}=\bar{A}_r^{(k)}\bar{\mu}_s+\bar{b}_r^{(k)}$, is defined as the new speaker dependent mean. Calculating the integral yields

$$\sum_k\sum_{\tau:\hat{k}(\tau)=k}\gamma'_s(\tau;\tilde{\theta})\bar{A}_r^{(k)T}\Sigma_s^{-1}\left(o_\tau-\bar{\mu}_s^{(k)}\right)+\sum_k D_s^{(k)}\bar{A}_r^{(k)T}\Sigma_s^{-1}\bar{A}_r^{(k)}\left(\mu_s-\bar{\mu}_s\right)do=0$$

Finally, given the new estimate of the speaker dependent transform $\bar{T}_r^{(k)}$, speaker independent means are then reestimated as

$$\bar{\mu}_s=\left(\sum_k\left(\sum_{\tau:\hat{k}(\tau)=k}\gamma'_s(\tau;\tilde{\theta})+D_s^{(k)}\right)\bar{A}_r^{(k)T}\Sigma_s^{-1}\bar{A}_r^{(k)}\right)^{-1}\times$$
$$\sum_k\bar{A}_r^{(k)T}\Sigma_s^{-1}\left(\sum_{\tau:\hat{k}(\tau)=k}\gamma'_s(\tau;\tilde{\theta})\left(o_\tau-\bar{b}_r^{(k)}\right)+D_s^{(k)}\bar{A}_r^{(k)}\mu_s\right)\ . \qquad (5.19)$$

Variance estimation

From equation (5.14), the Gaussian variance is found as:

$$\bar{\Sigma}_s^{-1}:\sum_k\sum_{\tau:\hat{k}(\tau)=k}\gamma'_s(\tau;\tilde{\theta})\cdot\nabla_{\Sigma_s^{-1}}\log q(o_\tau|s;\bar{T}_r^{(k)},\bar{\mu}_s,\bar{\Sigma}_s)$$
$$+\sum_k D_s^{(k)}\int q(o|s;\bar{T}_r^{(k)},\mu_s,\Sigma_s)\nabla_{\Sigma_s^{-1}}\log q(o|s;\bar{T}_r^{(k)},\bar{\mu}_s,\bar{\Sigma}_s)do=0\ . \qquad (5.20)$$

In a similar fashion by taking the derivative with respect to the speaker independent variance we have:

$$\nabla_{\Sigma_s^{-1}}\log q(o_\tau|s;\bar{T}_r^{(k)},\bar{\mu}_s,\Sigma_s)=\Sigma_s-\left(o_\tau-\bar{\mu}_s^{(k)}\right)\left(o_\tau-\bar{\mu}_s^{(k)}\right)^T$$

where $\bar{\mu}_s^{(k)}=\bar{A}_r^{(k)}\bar{\mu}_s+\bar{b}_r^{(k)}$, defined as the speaker dependent mean. Substituting the above expression for the gradient into the update rule of equation (5.20) gives

$$\sum_k \sum_{\tau:\hat{k}(\tau)=k} \gamma'_s(\tau;\tilde{\theta}) \left(\bar{\Sigma}_s - \left(o_\tau - \bar{\mu}_s^{(k)}\right)\left(o_\tau - \bar{\mu}_s^{(k)}\right)^T\right)$$

$$+ \sum_k D_s^{(k)} \int q(o|s;\mu_s) \left(\bar{\Sigma}_s - \left(o_\tau - \bar{\mu}_s^{(k)}\right)\left(o_\tau - \bar{\mu}_s^{(k)}\right)^T\right) do = 0\ .$$

Rearranging the previous equation and calculating the integral yields

$$\sum_k \left(\sum_{\tau:\hat{k}(\tau)=k} \gamma'_s(\tau;\tilde{\theta}) + D_s^{(k)} \right) \bar{\Sigma}_s = \sum_k \left(\sum_{\tau:\hat{k}(\tau)=k} \gamma'_s(\tau;\tilde{\theta}) \left(o_\tau - \bar{\mu}_s^{(k)}\right)\left(o_\tau - \bar{\mu}_s^{(k)}\right)^T \right)$$

$$+ \sum_k D_s^{(k)} \left[\Sigma_s + \left(\bar{A}_r^{(k)}\mu_s - \bar{A}_r^{(k)}\bar{\mu}_s\right)^2\right]$$

Finally, given the new estimate of the speaker dependent transform $\bar{T}_r^{(k)}$ and the new estimate of the speaker independent mean $\bar{\mu}_s$, the speaker independent variances are reestimated as

$$\bar{\Sigma}_s = \frac{\sum_k \left(\sum_{\tau:\hat{k}(\tau)=k} \gamma'_s(\tau;\tilde{\theta}) \left(o_\tau - \bar{\mu}_s^{(k)}\right)^2 \right) + D_s^{(k)} \left(\Sigma_s + \left(\bar{A}_r^{(k)}\mu_s - \bar{A}_r^{(k)}\bar{\mu}_s\right)^2 \right)}{\sum_k \left(\sum_{\tau:\hat{k}(\tau)=k} \gamma'_s(\tau;\tilde{\theta}) + D_s^{(k)} \right)}\ . \tag{5.21}$$

5.9 The DSAT Algorithm

The derivation described above is a two-stage iterative procedure that can be summarized in the following steps:

1. For each training utterance, create the numerator and denominator lattices.

2. Calculate the probabilities $\gamma_s(\tau;\theta)$ and $\gamma_s^g(\tau;\theta)$ of state occupation for all speakers k, and for all states s and times τ. Collect numerator/denominator statistics.

3. Initially, speaker dependent transforms are estimated via equation (5.16) while holding all other HMM parameters fixed at their current values.

4. Apply the speaker dependent transforms $\bar{T}_r^{(k)}$ to obtain the speaker dependent(SD) models. Calculate again the probabilities $\gamma_s(\tau;\tilde{\theta})$ and $\gamma_s^g(\tau;\tilde{\theta})$ of state occupation for all speakers k, and for all states s and times τ.

5. For each speaker k and foreach state s and time τ, use the probabilities $\gamma_s(\tau;\tilde{\theta})$ and $\gamma_s^g(\tau;\tilde{\theta})$ and the current observation vector o_τ to update the mean and variance accumulators for that state.

6. When all of the training data has been processed, the accumulated statistics are used to calculate new parameter values for all of the HMMs according to (5.19) and (5.21).

These steps can then all be repeated as many times as is necessary to achieve the required convergence.

5.10 Relationship between DSAT & MMIE

If we set $A^{(k)} = I$ and $b^{(k)} = 0$ in the update equations (5.19) and (5.21) we end up with the standard MMIE update equations (4.7) & (4.8). Indeed for the mean we get:

$$\bar{\mu}_s = \frac{\sum_k \sum_{\tau:\hat{k}(\tau)=k} \gamma'_s(\tau;\tilde{\theta})o_\tau + D_s^{(k)}\mu_{\mathbf{s}}}{\sum_k \sum_{\tau:\hat{k}(\tau)=k} \gamma'_s(\tau;\tilde{\theta}) + D_s^{(k)}}$$

or

$$\bar{\mu}_s = \frac{\sum_k \sum_{\tau:\hat{k}(\tau)=k} \gamma'_s(\tau;\tilde{\theta})o_\tau + D_s\mu_{\mathbf{s}}}{\sum_k \sum_{\tau:\hat{k}(\tau)=k} \gamma'_s(\tau;\tilde{\theta}) + D_s}$$

This last equation corresponds to standard MMIE mean update with

$$D_s = \sum_k D_s^{(k)}$$

Similarly the variance becomes

$$\bar{\Sigma}_s = \frac{\sum_k (\sum_{\tau:\hat{k}(\tau)=k} \gamma'_s(\tau;\tilde{\theta}) o_\tau^2 + D_s^{(k)} (\Sigma_s + \mu_s^2))}{\sum_k \left(\sum_{\tau:\hat{k}(\tau)=k} \gamma'_s(\tau;\tilde{\theta}) + D_s^{(k)} \right)} - \bar{\mu}_s^2$$

which is the standard MMIE variance.

5.11 Summary

This chapter has presented a speaker adaptive training (SAT) framework based on discriminative criteria. SAT is a powerful technique for building speech recognition systems on non-homogeneous data. A transform is generated for each speaker in the training set in order to produce matched conditions with the test set. These transformations are applied to the mean vectors before they enter the output distributions of the HMMs. We use a *regression class tree* to cluster together groups of output distributions that are to undergo the same transformation. The number of transforms is usually determined experimentally.

We briefly discussed the popular adaptation techniques such as MLLR and SAT that are based on Maximum Likelihood estimation. As an alternative to ML-SAT, the input transform and the gaussian model parameters are estimated using the CML auxiliary function. Thus we obtained fully discriminative procedures. Discriminative training is well suited for speaker adaptive training because of the very good performance on the training corpora.

Chapter 6

Minimum Bayes Risk Estimation

The discriminative modeling techniques MMI and DSAT discussed so far are the analog to the ML and ML-SAT maximum likelihood estimation procedures respectively. Although computationally expensive, they are efficient enough to be used for large vocabulary speech recognition tasks because of the existence of lattice based estimation procedures. However during MMIE/DSAT training the acoustic models are optimized with respect to the sentence error rate (SER) metric that is rarely used in evaluating these systems. The Minimum Bayes Risk modeling framework allows us to develop training procedures using a task specific evaluation criterion such as the word error rate (WER). Efficient Minimum Bayes Risk estimation for large vocabulary speech tasks is the topic of this chapter.

6.1 Risk Based Training and MMIE

MMIE increases the *a posteriori* probability of the correct transcription given the observed speech data. Since the ultimate goal is to reduce the number of words in error, estimation procedures that reduce the loss are more closely related to the ASR performance criteria. The Minimum Bayes Risk framework can compensate for the mismatch between the estimation and the evaluation criteria used in ASR systems.

Given a database ($\bar{W}$,O) we want to estimate model parameters to minimize the empirical loss

$$\theta^* = \operatorname*{argmin}_{\theta} R(\bar{W}, \mathcal{W};\ \theta) \tag{6.1}$$

where

$$R(\bar{W}, \mathcal{W};\ \theta) = \sum_{W' \in \mathcal{W}} l(\bar{W}, W') P(W'|O;\ \theta) \tag{6.2}$$

$\mathcal{W}$ is the set of hypotheses that are alternatives to the truth. The risk is measured under a loss function that is appropriately chosen for the recognition task; for example, in ASR the most commonly used loss function is the Levenshtein distance [49] that measures the word error rate(WER).

Therefore Empirical Risk Minimization estimates the model parameters so as to directly reduce the expected classification error on the training data. The interpretation is that the estimation procedure should attempt to minimize the empirical loss $R(\bar{W}, \mathcal{W};\ \theta)$ by assigning low conditional likelihood $P(W'|O; \theta)$ to the hypotheses W' that are far from the truth $\bar{W}$ in terms of $l(\bar{W}, W')$ and move probability mass towards those hypotheses that are close to the truth.

Under the 0/1 loss function between word strings $\bar{W}$,W'

$$l(\bar{W}, W') = \begin{cases} 0 \text{ if } \bar{W} = W' \\ 1 \text{ otherwise.} \end{cases} \tag{6.3}$$

the above equation (6.1) becomes

$$\theta^* = \operatorname*{argmin}_{\theta} \sum_{W' \in \mathcal{W}: W' \neq \bar{W}} P(W'|O;\ \theta) = \operatorname*{argmin}_{\theta} (1 - P(\bar{W}|O;\ \theta)) = \operatorname*{argmax}_{\theta} P(\bar{W}|O;\ \theta) \tag{6.4}$$

which is the MMIE objective function as shown in equation (4.2).

We note that MMI is a special case of risk minimization under the Sentence Error Rate loss function on the acoustic training set. This immediately suggests that beyond the usual difficulties of ensuring that performance on the training set generalizes to the test set, there may also be issues in generalization under different performance criteria.

The close relationship between MMI and minimum risk estimation is widely known, and MMI-variants for the training criterion of (6.1) have been developed

[41, 42, 83]. Normandin used the Extended Baum Welch (EBW) algorithm by Gopalakrishnan et al [31] for MMI estimation, while Kaiser et al [41, 42] also used (EBW) for risk minimization of (6.1).

6.1.1 Gaussian Parameter Estimation

Kaiser et al[41, 42] used the Extended Baum Welch algorithm by Gopalakrishnan et al [31] for risk minimization of (6.1) by observing that the overall risk $R(\bar{W}, \mathcal{W};\ \theta)$ is a rational function similar to $P(\bar{W}|O;\ \theta)$. We briefly present his estimation in the Appendix (A).

Following Kaiser we can express the derivative of $-Loss(\theta)$ in terms of $\nabla_\theta logP(O|W')$ as:

$$-\nabla_\theta Loss(\theta) = \sum_{W' \in \mathcal{W}} K(W', \mathcal{W}) \nabla_\theta logP(O|W') \tag{6.5}$$

where

$$K(W', \mathcal{W}) = [\sum_{W'' \in \mathcal{W}} P(W''|O) l(\bar{W}, W'') - l(\bar{W}, W')]\, P(W'|O)\,. \tag{6.6}$$

$K(W', \mathcal{W})$ determines the contribution of each hypothesis W' to the gradient of the loss. It plays the same role in the Extended Baum Welch update rule as the gradient of the likelihood does in the derivation of MMI.

Given the $K(W', \mathcal{W})$ and $\nabla_\theta logP(O|W';\ \theta)$, the new estimates for the means and variances can be estimated by the extension of the EBW as proposed by Normandin [62]. For the means, the estimate for $\bar{\mu_s}$ is given by

$$\bar{\mu_s} = \frac{\sum\limits_{W' \in \tilde{\mathcal{W}}} K(W', \mathcal{W}) \sum_\tau \gamma_s^{W'}(\tau) o_\tau + D_s \mu_s}{\sum\limits_{W' \in \tilde{\mathcal{W}}} K(W', \mathcal{W}) \sum_\tau \gamma_s^{W'}(\tau) + D_s} \tag{6.7}$$

For the variance we have

$$\bar{\Sigma_s} = \frac{\sum\limits_{W' \in \tilde{\mathcal{W}}} K(W', \mathcal{W}) \sum_\tau \gamma_s^{W'}(\tau) o_\tau^2 + D_s \left(\Sigma_s + \mu_s {\mu_s}^T\right)}{\sum\limits_{W' \in \tilde{\mathcal{W}}} K(W', \mathcal{W}) \sum_\tau \gamma_s^{W'}(\tau) + D_s} - \bar{\mu_s} {\bar{\mu_s}}^T \tag{6.8}$$

These estimates are such that $R(\bar{W}, \mathcal{W};\ \bar{\theta}) \leq R(\bar{W}, \mathcal{W};\ \theta)$.

However there are obvious difficulties with this approach. By observing equations (6.7) and (6.8) it follows that the estimation of the Gaussian model parameters to minimize the empirical risk differs from MMI in that it is necessary to compute the contribution of $K(W', \mathcal{W})$ in addition to the posterior likelihood $P(W'|O;\ \theta)$. This quantity (6.6) needs to be found over all paths W' in $\mathcal{W}$. The problem then is to choose the set $\mathcal{W}$. We refer to $\mathcal{W}$ as the "evidence space" since it determines the hypotheses over which the risk will be estimated.

6.1.2 Collecting Statistics over the Evidence Space

In large vocabulary speech recognition tasks, $\mathcal{W}$ is often a lattice generated by the ASR decoder. Lattices are used because the most likely hypotheses are so numerous that listing them explicitly is impractical, and probabilities such as $P(\bar{W}|O;\ \theta)$ can be found conveniently by summing over lattice paths so that procedures such as lattice-based MMI are feasible.

However finding $K(W', \mathcal{W})$, which must be computed and maintained for each path $W' \in \mathcal{W}$, is not as readily done over lattices. The source of the problem is the presence of $l(\bar{W}, W')$. For instance, this quantity requires finding the Levenshtein distance between the reference $\bar{W}$ and every other path W' in $\mathcal{W}$. These distances are not as easily computed as path likelihoods, since Levenshtein distance between two strings does not distribute over lattice arcs in the manner of path likelihoods. Beyond the computational difficulty in finding the $l(\bar{W}, W')$ over all paths $W' \in \mathcal{W}$, there are complications in accumulating the statistics for the mean and variance updates of Equations (6.7) and (6.8). The summation over $\mathcal{W}$ must be performed pathwise by explicitly enumerating W' so that the weighting terms $K(W', \mathcal{W})$ can be incorporated correctly into the statistics.

Given the framework described thus far, the only possibility for lattice based estimation is simply to expand first-pass ASR lattices into N-Best lists so that the string-to-string comparisons and the gathering of statistics (done via the Forward-Backward procedure) can be carried out explicitly. This was the approach proposed developed and validated by Kaiser on a small vocabulary task. While correct this approach is not feasible for large vocabulary continuous speech recognition. These N-Best lists would have to be extremely deep to contain a significant portion of the most likely hypotheses, and the computation of loss over them and the gathering of statistics needed to perform the parameter updates of Equations (6.7) and (6.8) would also be costly, leading to considerable higher training times than standard MMIE training.

An easy solution might be to use a subset of $\mathcal{W}$, although this subset must be representative of the errors that actually exist during training. While this approach avoids the high memory, computational and storage costs, the problem of estimating the loss $l(\bar{W}, W')$ for every W' still remains.

This problem of merging the computation of loss and likelihood also arises in the application of Minimum Bayes Risk decoding [28, 30], to large vocabulary ASR tasks. We now discuss how efficient techniques to compute risk over lattices can be used to obtain the statistics needed to implement the risk-based minimization criteria for parameter estimation in large vocabulary speech recognition tasks.

6.2 Minimum Bayes Risk Decoding

Minimum Bayes Risk(MBR) Decoding is an alternative ASR search strategy that produces hypotheses[71, 28, 30], with the least expected loss under a given task specific loss function. Let $l(W, W')$ be a real valued function that measures the loss incurred when an utterance W is mistranscribed as W'.

Under $l(W, W')$ between word strings W and W', the MBR recognizer seeks the optimal hypothesis given the acoustic data O as:

$$\hat{W} = \operatorname*{argmin}_{W \in \mathcal{W}} \sum_{W' \in \mathcal{W}} l(W, W') P(W'|O) \tag{6.9}$$

Thus the hypothesis with least expected loss is selected. MBR decoding has been found to consistently provide improved performance relative to straightforward maximum a posteriori (MAP) decoding procedures.

Prior work in MBR decoding has treated it essentially as a large search problem in which $\mathcal{W}$ are N-Best lists or lattices that incorporate $P(W'|O)$ as a posterior distribution on word strings obtained using an HMM acoustic model and an N-gram language model[71, 28]. Thus in implementing the MBR decoder there are conceptually two distinct steps:

Step 1 Compute the risk for each $W \in \mathcal{W}$

$$R(W, \mathcal{W}) = \sum_{W' \in \mathcal{W}} l(W, W')P(W'|O) \tag{6.10}$$

Step 2 Select the minimum risk hypothesis

$$\hat{W} = \operatorname*{argmin}_{W \in \mathcal{W}} R(W, \mathcal{W}) \tag{6.11}$$

A special case of MBR decoding is particularly useful; when $l(W, W')$ is the 0/1 valued identity function that measures the sentence error rate. Under these conditions the MBR recognizer of Equation(6.9) leads to the standard MAP rule.

Efficient algorithms have been developed to compute the risk $R(W; \mathcal{W})$ of a hypothesis W under the Levenshtein loss function[30]. Since it is straightforward to compute $P(W'|O)$ over lattices, the key is an efficient lattice-to-string alignment algorithm to find $l(W, W')$ for all W' in any lattice $\mathcal{W}$. Such an algorithm has been developed, and it yields the (nearly) optimum alignment of every W to W'. The lattice-to-string alignment algorithm is described in detail in Goel et al.[30]. For the purposes of this thesis, the algorithm will be described briefly in the next section.

6.3 Lattice-to-string alignment

The goal is to find $l(W, W')$ for all W' in a lattice $\mathcal{W}$. The algorithm described here yields an (nearly) optimum alignment of every W' to $\bar{W}$ called the lattice-to-string alignment. The top lattice in (Fig. 6.1), shows a lattice generated by an ASR system. The lattice arcs are labelled by word hypotheses and these arcs carry the negative log likelihood of each word. In this example, the lattice will be aligned to the reference string $\bar{W}$

$\bar{W}$: HELLO HOW ARE YOU ALL TODAY

it appears in the lattice marked in bold. The output of the lattice-to-string alignment algorithm is a lattice itself, as shown in the second lattice of Figure 6.1.

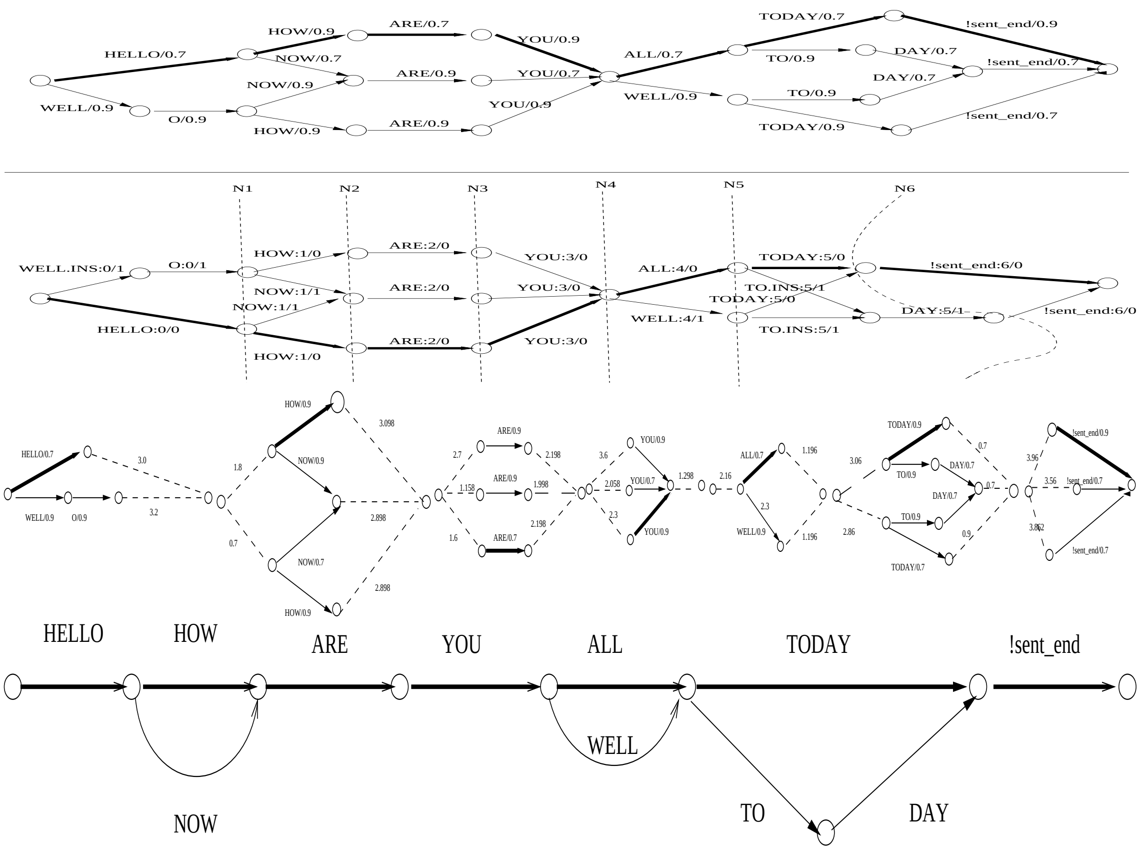


Figure 6.1: Lattice Segmentation for LVCSR Minimum Bayes Risk Estimation. *Top*: First-pass lattice of likely sentence hypotheses with reference path in bold; *Middle*: Alignment of lattice paths to reference with node cut sets and sublattices; *Bottom*: Pruned pinched lattice used for training.

The alignment of an arbitrary string from the first lattice, e.g.

W': WELL O NOW ARE YOU ALL TODAY

can be read from the corresponding string in the alignment lattice:

WELL.INS:0/1 O:0/1 NOW:1/1 ARE:2/0 YOU:3/0 ALL:4/0 TODAY:5/0

The notation WELL.INS:0/1 indicates that WELL is aligned as an insertion with cost 1 to the word at position 0 in the reference string (which is HELLO the alignment index starts at 0); similarly, O is aligned to HELLO and NOW is aligned to HOW, each with a substitution cost of 1. The overall alignment reads as

Word Index :	0	1	2	3	4	5
Reference :	HELLO	HOW	ARE	YOU	ALL	TODAY
Hypothesis :	WELL O	NOW	ARE	YOU	ALL	TODAY
Per Segment Cost:	2	1	0	0	0	0

with a total loss of $l(\bar{W}, W') = 3$. By tracing a path through the lattice and accumulating the Levenshtein alignment costs and weighting them by the arc likelihoods (which are copied from the original ASR output lattice), the risk $R(\bar{W}, \mathcal{W})$ of $\bar{W}$ can be computed. The connection between MBR decoding and Minimum Risk estimation becomes apparent when we note that a key quantity in risk based minimization training can be written in terms of the same lattice based risk needed for MBR decoding:

$$K(W', \mathcal{W}) = [R(\bar{W}, \mathcal{W}) - l(\bar{W}, W')] \, P(W'|O) \, . \tag{6.12}$$

6.3.1 Risk-Based Cutting of $\mathcal{W}$

Even with the aid of the lattice-to-string alignment algorithm, computing lattice-based risk can be computationally challenging for long or deep lattices. We have used risk-based lattice segmentation techniques that simplify MBR decoding and we now discuss how these methods can be applied to risk-based parameter estimation. Risk-based lattice segmentation proceeds by segmenting the lattice with respect to the reference string by following the lattice-to-string alignments. For the K words of the reference string, we identify K-1 node cut sets. To form the cut set $N_i, i = 1, ..., K-1$

- - identify all lattice subpaths that are aligned to the reference word W_{i-1}
- - the cut set N_i consists of the final lattice nodes of all these subpaths

The second panel of (Fig. 6.1) shows the cut sets for that lattice. The paths between adjacent cut sets are tied at their ends so that they form sublattices, and these are then concatenated to form a pinched lattice as shown in the third panel of (Fig. 6.1) and the second panel of Figure (Fig. 8.1). Each of these sublattices contains one word from the reference string and the other word sequences which aligned to it. The dashed arcs show the likelihood of the word hypotheses. For instance, $-logP(W_5 =$

$ALL, O) = 2.16 + 0.7 + 1.196$, is the log likelihood of all the paths whose fifth word is ALL.

The pinched lattice is a sequence of sublattices each of which is aligned to a single word in the reference string. These sublattices are called confusion sets because they contain likely and errorful hypothesis segments that the ASR system might confuse with the reference words. It is important to stress that all the sentence hypotheses from the original ASR lattice are preserved in creating the pinched lattice and that no paths are removed by pinching. In fact, pinching may actually introduce new paths by piecing together subpaths from the original lattice; however these new paths are insignificant from a modeling point of view, in that they should be of lower probability than any of the original lattice paths.

6.3.2 Pruning of $\mathcal{W}$

The evidence space is pruned in two steps. In the first step, the likelihood of each lattice arc is used to discard all paths through every confusion set so that only the most likely alternative to the reference word remains. This is illustrated in the transition from the second to the third panel of(Fig. 8.1). When the confusion sets are pruned to contain binary alternatives, we call them confusion pairs.

In the second pruning step, we simply count all the confusion pairs in the training set lattices, and if any pair has occurred fewer times than a set threshold, that pair is everywhere pruned back to the reference transcription. As an example, the bottom panel of (Fig. 8.1, *Bottom*), shows that some segment sets not in the final collection (e.g. OH+4) are discarded. The end result is a greatly reduced evidence space $\tilde{\mathcal{W}}$ derived from the original lattice $\mathcal{W}$. This reduction is controlled by the occurence threshold and we usually determine through experimentation what value gives a reasonable sized N-best list expansion. For example if we have 3 binary confusion pairs the N-best list depth is 2^3.

6.3.3 Induced Loss Function

Our original motivation was to speed up training, but this approach also allows us to redefine the string-to-string loss within $\tilde{\mathcal{W}}$. Suppose the reference string $\bar{W}$ has N words $\bar{W}_i, i = 1, 2, \ldots, N$. Another string W' is not allowed to be aligned arbitrarily to $\bar{W}$; it must follow the constraints of $\tilde{\mathcal{W}}$. We call this the "induced loss function"

$$l_I(\bar{W}, W') \simeq \sum_{i=1}^{N} l(\bar{W}_i, W'_i) \; . \tag{6.13}$$

Thus we assume that the loss function distributes over the segmentation. Ideally, we should satisfy the strong requirement that the loss function between any two word sequences $\bar{W}, W'$, is not affected by the lattice cutting, i.e. that $l(\bar{W}, W') = \sum_{i=1}^{K} l(\bar{W}_i, W'_i)$.

In summary, lattice segmentation produces both a reduced hypothesis space as well as an induced loss function. Once a lattice has been segmented, the original lattice $\mathcal{W}$ is approximated by the pinched lattice $\tilde{\mathcal{W}}$ and the distance between two strings in the lattice is constrained by the segmentation. We next discuss how to use these quantities to reduce the computational cost of minimum bayes risk discriminative training.

6.4 Pinched Lattice Minimum Bayes Risk Discriminative Training

In our approach to direct risk minimization we first incorporate lattice pinching, which produces both a reduced hypothesis space $\tilde{\mathcal{W}}$ as well as an induced loss function $l_I(\bar{W}, W')$, with the goal of focusing the estimation procedures on individual recognition errors. Because we apply discriminative training on the pinched lattice we term the procedure Pinched Lattice Minimum Bayes Risk Discriminative Training.

By approximating the original lattice $\mathcal{W}$ by the pinched lattice $\tilde{\mathcal{W}}$, and using equation (6.13) then the initial equation (6.1) becomes

$$\theta^* = \operatorname*{argmin}_{\theta} \sum_{W' \in \mathcal{W}} l(\bar{W}, W') P(W'|O;\ \theta) \simeq \operatorname*{argmin}_{\theta} \sum_{W' \in \tilde{\mathcal{W}}} \sum_{i=1}^{N} l(\bar{W}_i, W_i') P(W'|O;\ \theta) \tag{6.14}$$

Estimation is done with equations (6.7) and (6.8) with $\mathcal{W}$ replaced by $\tilde{\mathcal{W}}$ and taking

$$K(W', \tilde{\mathcal{W}}) = [\sum_{W'' \in \tilde{\mathcal{W}}} P(W''|O) l_I(\bar{W}, W'') - l_I(\bar{W}, W')]\, P(W'|O)\,. \tag{6.15}$$

which can be written as

$$K(W', \tilde{\mathcal{W}}) = [R_I(\bar{W}, \tilde{\mathcal{W}}; \theta) - l_I(\bar{W}, W')]\, P(W'|O)\,, \tag{6.16}$$

for all $W' \in \tilde{\mathcal{W}}$, where $R_I(\bar{W}, \tilde{\mathcal{W}}; \theta) = \sum_{W'' \in \tilde{\mathcal{W}}} P(W''|O) l_I(\bar{W}, W'')$ is the expected induced loss.

This leads to the following training algorithm.

6.4.1 The PLMBRDT Algorithm

Step 1 Generate lattices over the training set

Step 2 Align the training set lattices to the reference transcriptions $\bar{W}$

Step 3 Segment the lattices

Step 4 Prune the confusion sets to confusion pairs

Step 5 Discard infrequently occurring confusion pairs

Step 6 Expand each pinched lattice into an N-Best list, keeping $l_I(\bar{W}, W')$

Step 7 Compute $R_I(\bar{W}, \tilde{\mathcal{W}}; \theta)$ as defined above

Step 8 For all $W' \in \tilde{\mathcal{W}}$ compute $K(W', \tilde{\mathcal{W}})$ by equation (6.16)

Step 9 Perform Forward-Backward pass for all $W' \in \tilde{\mathcal{W}}$ using the weighting $K(W', \tilde{\mathcal{W}})$ to accumulate statistics

Step 10 Estimation is done by equations (6.7) and (6.8)

This implementation does expand lattices into N-Best lists of sentence hypotheses. However, it is the pinched lattices that are expanded, not the original lattices generated by the ASR decoder. The pinched lattices are much reduced relative to the original lattice and, since we have control over the degree of pinching and pruning, we can control the size of the N-Best lists. These pinched lattices are also representative of the errors that actually exist during training. In this way we can reduce the evidence space drastically so that the original formulation by Kaiser et al can be applied directly to large vocabulary ASR, albeit under the induced loss function.

Next we consider two simplifications to $\tilde{\mathcal{W}}$. The first variant is Pinched Lattice MMIE which is appropriate for small vocabulary ASR tasks based on whole-word models. The second variant is One-Worst Pinched Lattice MBRDT which is a form of corrective training against a competing hypothesis extracted from the pinched lattice.

6.5 Pinched Lattice MMIE

With lattice cutting, the hypothesis space $\mathcal{W}$ is segmented into N segments, $\tilde{\mathcal{W}}_1, \tilde{\mathcal{W}}_2, ..., \tilde{\mathcal{W}}_N$. In regions of low confidence, the search space contains portions of the MAP hypothesis(truth for training) along with confusable alternatives. In regions of high confidence, the search space is restricted to follow the MAP hypothesis itself (Fig. 8.1, *Bottom*). We can then express the empirical risk equation (6.14) above as:

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \sum_{W' \in \tilde{\mathcal{W}}} l_I(\bar{W}, W') P(W'|O;\theta) = \underset{\theta}{\operatorname{argmin}} \sum_{W'_1 W'_2 \ldots W'_N} \sum_{i=1}^{N} l(\bar{W}_i, W'_i) P(W'|O;\theta)$$

$$= \underset{\theta}{\operatorname{argmin}} \sum_{i=1}^{N} \sum_{w \in \tilde{\mathcal{W}}_i} \sum_{W'|W'_i = w} l(\bar{W}_i, w) P(W'|O;\theta) = \underset{\theta}{\operatorname{argmin}} \sum_{i=1}^{N} \sum_{w \in \tilde{\mathcal{W}}_i} l(\bar{W}_i, w) P_i(w|O;\theta) \tag{6.17}$$

In (6.17) $P_i(w|O;\theta)$ is the probability of observing the string w in the i^{th} segment set and is given by:

$$P_i(w|O;\theta) = \sum_{W' \in \tilde{\mathcal{W}} : W'_i = w} P(W'|O) \tag{6.18}$$

Next we introduce the global confusion class $C \subseteq \{1, \ldots, K\}$ to indicate the segment sets that permit alternatives to the MAP path, i.e. $i \in C$ implies that $\tilde{\mathcal{W}}_i$ contains at least one segment not in the MAP hypothesis. We can then write the objective as

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \sum_{i \in C} \sum_{w \in \tilde{\mathcal{W}}_i} l(\bar{W}_i, w) P_i(w|O, \tilde{\mathcal{W}};\theta). \tag{6.19}$$

Finally, we assume that we have a 0/1 loss function within the segment sets and arrive at the "pinched lattice" MMI objective function

$$\theta^* = \underset{\theta}{\operatorname{argmax}} \sum_{i \in C} P_i(\bar{W}_i|O, \tilde{\mathcal{W}};\theta). \tag{6.20}$$

Therefore the empirical risk is minimized by maximizing the likelihood of the correct hypothesis in the confusable segments. Initially lattice segmentation is used to segment lattices produced by an ASR system into sequences of separate sublattices involving small sets of confusable words. Then we train acoustic models specialized to discriminate between the competing words in these classes. We have developed the following strategy to integrate the estimation and decoding procedures described here.

6.5.1 The PL-MMIE Algorithm

Step 1 Generate lattices over the training set

Step 2 Align the training set lattices to the reference transcriptions $\bar{W}$

Step 3 Segment the lattices

Step 4 Prune the confusion sets to confusion pairs

Step 5 Discard infrequently occurring confusion pairs

Step 6 Tag word hypotheses in confusion pairs (Fig. 8.1, *Bottom*)

Step 7 Regenerate lattices over the training set using the tagged and pinched lattices to constrain recognition (in contrast to Step 1). If the task requires a grammar, compose the tagged and pinched lattices with the task grammar before lattice regeneration/rescoring. The grammar should be (trivially) extended to cover the tagged words.

Step 8 Perform lattice-based MMI [84] using the word boundary times obtained from the lattice. The procedure differs from regular MMI in that statistics needed in Equations (4.7) and (4.8) are gathered only over the tagged word hypotheses. Statistics from the un-tagged word hypotheses, which correspond to the high-confidence regions in the pinched lattice, are discarded.

In PLMMI the Levenshtein distance is not used explicitly in the reestimation procedure. It is used to create a pruned search space that contains only the confusable pairs identified by lattice segmentation. Statistics are compiled over these lattices as usual for lattice-based MMI, with the exception that statistics are gathered only for those word instances that appear in confusion sets. This modification forces the MMI procedure to focus on the low confidence regions identified by lattice pinching.

6.6 One Worst Pinched Lattice MBRDT

In this occasion we approximate the set of paths $W' \in \tilde{\mathcal{W}}$ by considering only the truth $\bar{W}$ and the worst alternative defined by:

$$W^* = \operatorname*{argmax}_{W' \in \tilde{\mathcal{W}}} l_I(\bar{W}, W'). \tag{6.21}$$

So that the set of hypothesis becomes $\tilde{\mathcal{W}} \simeq \{\bar{W}, W^*\}$. For the lattice shown in Figure (Fig. 8.1, *Bottom*) $l_I(\bar{W}, W^*) = 4$. Under this approximation the loss function (6.14) becomes

$$\theta^* = \operatorname*{argmin}_{\theta} l_I(\bar{W}, W^*) P(W^*|O;\theta) \tag{6.22}$$

That is we consider only 2 word sequences $\bar{W}$ and W^*, thus reducing the amount of computations significantly.

We are interested in applying the update rules of equations (6.7) and (6.8) based on this reduced loss function. By simple arithmetic in (6.6) it follows that:

$$R_I(\bar{W};\ \tilde{\mathcal{W}}) = P(W^*|O) l_I(\bar{W}, W^*)$$

$$K(\bar{W}, \tilde{\mathcal{W}}) = [R_I(\bar{W};\ \tilde{\mathcal{W}}) - l_I(\bar{W}, \bar{W})] P(\bar{W}|O) = P(\bar{W}|O) l_I(\bar{W}, W^*) P(W^*|O)$$
$$K(W^*, \tilde{\mathcal{W}}) = [R_I(\bar{W};\ \tilde{\mathcal{W}}) - l_I(\bar{W}, W^*)] P(W^*|O) = -P(\bar{W}|O) l_I(\bar{W}, W^*) P(W^*|O)$$

leading to $K(\bar{W}, \tilde{\mathcal{W}}) = -K(W^*, \tilde{\mathcal{W}})$. Note that in the above we restrict the acoustic likelihood $P(O)$ to the 2 word sequences $\bar{W}$ and W^*, that is $P(O) = P(O|\bar{W})P(\bar{W}) + P(O|W^*)P(W^*)$. Substituting in equations (6.7) and (6.8), the new estimates for the Gaussian model parameters $\bar{\theta} = (\bar{\mu}_s, \bar{\Sigma}_s)$ are:

$$\bar{\mu}_s = \frac{K(\bar{W}, \tilde{\mathcal{W}})(\sum_\tau \gamma_s(\tau;\ \bar{W}) o_\tau - \sum_\tau \gamma_s(\tau;\ W^*) o_\tau) + D_s \mu_s}{K(\bar{W}, \tilde{\mathcal{W}})(\gamma_s(\bar{W}) - \gamma_s(W^*)) + D_s} \tag{6.23}$$

$$\bar{\Sigma}_s = \frac{K(\bar{W}, \tilde{\mathcal{W}})(\sum_\tau \gamma_s(\tau;\ \bar{W}) o_\tau^2 - \sum_\tau \gamma_s(\tau;\ W^*) o_\tau^2) + D_s\,(\Sigma_s + {\mu_s}^2)}{K(\bar{W}, \tilde{\mathcal{W}})(\gamma_s(\bar{W}) - \gamma_s(W^*)) + D_s} - \bar{\mu_s}^2. \tag{6.24}$$

A futher approximation is to discard the terms $l_I(\bar{W}, W^*) P(\bar{W}|O) P(W^*|O)$. The update equations become:

$$\bar{\mu_s} = \frac{\sum_\tau \gamma_s(\tau;\ \bar{W}) o_\tau - \sum_\tau \gamma_s(\tau;\ W^*) o_\tau + D_s \mu_s}{\sum_\tau \gamma_s(\tau;\ \bar{W}) - \sum_\tau \gamma_s(\tau;\ W^*) + D_s} \tag{6.25}$$

For the variance we have

$$\bar{\Sigma}_s = \frac{\sum_\tau \gamma_s(\tau;\ \bar{W}) {o_\tau}^2 - \sum_\tau \gamma_s(\tau;\ W^*) {o_\tau}^2 + D_s\,(\Sigma_s + {\mu_s}^2)}{\sum_\tau \gamma_s(\tau;\ \bar{W}) - \sum_\tau \gamma_s(\tau;\ W^*) + D_s} - \bar{\mu_s}^2. \tag{6.26}$$

We term this procedure One Worst Pinched Lattice MBRDT. The algorithm can be summarized in the following steps.

6.6.1 The One Worst Pinched Lattice MBRDT Algorithm

Step 1 Generate lattices over the training set

Step 2 Align the training set lattices to the reference transcriptions $\bar{W}$

Step 3 Segment the lattices

Step 4 Prune the confusion sets to confusion pairs

Step 5 Discard infrequently occurring confusion pairs

Step 6 Extract the most errorful hypothesis W^*

Step 7 Perform Forward-Backward pass with respect to $\bar{W}$ and W^* to accumulate statistics

Step 8 Estimation is done by equations (6.25) and (6.26)

This approach is closely related to MCE [40] where we consider the truth $\bar{W}$ and the best incorrect hypothesis W'. What distinguishes this approach from other forms of corrective training is not the update procedure itself, but rather the way in which the competing hypothesis W^* is obtained.

6.7 Summary

We have demonstrated how techniques developed for Minimum Bayes Risk Decoding make it possible to apply risk-based parameter estimation algorithms to large vocabulary speech recognition tasks. Our approach starts with the original derivations of Kaiser et al.[41, 42], which show how the Extended Baum Welch algorithm can be used to derive a parameter estimation procedure to reduce expected loss over the training data.

However this estimation procedure is computationally expensive when applied to large vocabulary continuous speech recognition. It requires the explicit enumeration of competing hypotheses representative of the recognition errors made by the machine in order to estimate the loss. These competing word hypotheses are produced either by expanding N-Best lists or lattices. For large vocabulary ASR tasks, these word lattices or N-best lists are large and the MBR training is computationally expensive.

To alleviate the computational complexity, lattice segmentation techniques initially developed for MBR search over large lattices are used to derive iterative estimation procedures that minimize empirical risk based on general loss functions such as the Levenshtein distance. We use the formulation of Kaiser et al.[41, 42], and replace the Levenstein distance with the induced loss function. Through this approximation, we are able to efficiently compute the statistics needed to apply the risk-based parameter estimation algorithm over large vocabulary speech recognition tasks.

Chapter 7

Speaker Adaptive Training Results on SWITCHBOARD

The recent advances in continuous speech recognition have resulted in high recognition accuracy for simple recognition tasks such as scripted speech and small vocabulary tasks. Attention has therefore shifted to more challenging and realistic problems posed by the spontaneous conversational speech. The recognition performance of an ASR system is affected by the confusability of the vocabulary. As a result LVCSR tasks achieve lower accuracy than small vocabulary tasks. The experimental results in this chapter are performed on the *SWITCHBOARD* Corpus[8].

7.1 System Description

The *SWITCHBOARD Corpus* is a database of spontaneous dialogue, among English speakers. It is collected over standard telephone lines and the speech is extemporaneous and not scripted. It contains disfluencies, pauses, non-grammatical usage and channel distortion, all of which have major impact on the recognition performance. Our ASR system is a speaker independent continuous mixture density, tied state, cross-word, gender-independent, triphone HMM system. The baseline acoustic models used as seed models for our experiments, were built using HTK [86] from 16.4 hours of Switchboard-1 and 0.5 hour of Callhome English data. This collection defined the development training set for the 2001 JHU LVCSR system [8]. The speech was parameterized into 39-dimensional PLP cepstral coefficients with delta and acceleration coefficients [37].

Cepstral mean and variance normalization was performed over each conversation side to remove the channel effect of the features. The acoustic models used cross-word triphones with decision tree clustered states [86], where questions about phonetic context as well as word boundaries were used for clustering. There were 4000 unique triphone states with 6 Gaussian components per state. Lattice rescoring experiments were performed using the AT&T Large Vocabulary Decoder [56], with a 33k-word trigram language model provided by SRI [70].

The recognition tests were carried out on a subset of the 2000 Hub-5 Switchboard-1 evaluation set (SWBD1) [53] and the 1998 Hub-5 Switchboard-2 evaluation set (SWBD2) [52]. The SWBD1 test set was composed of 866 utterances consisting of 10260 words from 22 conversation sides, and the SWBD2 test set was composed of 913 utterances consisting of 10643 words from 20 conversation sides. The total test set was 2 hours of speech.

Discriminative training requires alternate word sequences that are representative of the recognition errors made by the decoder. These are obtained via triphone lattices generated on the training data. Our approach is based on the MMI training procedure developed by Woodland and Povey [84]. However, rather than accumulating statistics via the Forward-Backward procedure at the word level, we use the Viterbi procedure over triphone segments. These triphone segments are fixed throughout MMI training. The Baseline SI acoustic models yield a word error rate (WER) of 41.1% & 51.1%. These models were used to generate hypotheses in a compact lattice representation [63], which were then rescored with discriminative trained acoustic models in the subsequent experiments.

7.1.1 Conventional MMIE

This section describes a series of MMIE training experiments, shown in (Table 7.1). We start with a well trained ML-system(41.1%/51.1%). We update the model parameters $\theta = (\mu_s, \Sigma_s)$, mean and the corresponding variance, under the CML framework using equation (4.7) and equation (4.8). The learning rate constants D_s for the mean and variance parameters are set on a per Gaussian basis and quarantee that the variance remains positive as suggested by Woodland and Povey[84].

To validate our approach we calculated the value of (4.2) as a function of the iteration number. These results shown in the second column of Table 7.1 confirm that the MMI objective function is increasing under the estimation procedure. Since the denominator includes all possible word sequences(including the correct one) the objective function(4.2) has a maximum value of zero.

Table 7.1: Results on the SWBD training set, SWBD test set

MMIE	SWBD Training set		SWBD Test set	
	WER	OBJECTIVE $F(\theta)$	SWBD1	SWBD2
ML Baseline	29.42	-2.37E05	41.1	51.1
1 iteration	27.55	-2.05E05	40.6	50.5
2 iteration	26.24	-1.81E05	40.5	50.0
3 iteration	25.62	-1.647E05	39.9	49.7
4 iteration	25.66	-1.53E05	40.2	50.5

Next we present results after MMIE training on the SWBD training set and on the SWBD test set. From the results in Table 7.1 we see that significant improvement over the baseline can be obtained by MMI. Thus MLE, although widely used, is not optimal in reducing the error rate of the ASR system. It can be seen that the best WER is obtained after 3 iterations(after that performance degrades: we have overtraining). Experimental results show that the proposed method is very effective in reducing the word error rate on the training set, a reduction of 4.0% absolute is achieved. While MMIE training greatly reduces training set error from an MLE baseline, the reduction in error rate on an independent test set is normally much less(39.9%/49.7%, 1.3% absolute reduction), so the generalization performance is poorer.

In the next experiment we compare MMIE training versus conventional MLE training recognition performance on the test set. This experiment (Table 7.2) shows that the accuracy of a speech recognition system trained with Maximum Likelihood Estimation (MLE) can be further improved using discriminative training. From the experimental results we argue that MMIE training is more appropriate than MLE for reducing the error rate.

Table 7.2: Results comparing MMIE versus MLE training evaluated on Swbd1 and Swbd2 test sets. Both systems were initialized by ML trained models.

Models	MMIE		MLE	
Test Set	Swbd1	Swbd2	Swbd1	Swbd2
Baseline	41.1	51.1	41.1	51.1
1 iteration	40.6	50.5	41.0	51.2
2 iteration	40.5	50.3	40.7	51.1
3 iteration	39.9	49.7	40.7	51.2
4 iteration	40.2	50.5	40.5	51.0

7.2 Speaker Adaptation Results

The next section presents a speaker adaptive training framework based on discriminative criteria as discussed in Chapter 5. Until recently the popular adaptation techniques such as MLLR and SAT were based on Maximum Likelihood estimation. Nevertheless from the results in (Table 7.1) we see that discriminative optimization criteria can be more effective in reducing the word error rate than maximum likelihood estimation. As an alternative to ML-SAT, the input transform and the gaussian model parameters can be estimated using the CML auxiliary function. Thus we obtain fully discriminative training procedures termed (DSAT).

7.2.1 Optimal number of Regression Classes

Initially we conducted a series of experiments to compare MLLR with different number of transforms to determine the optimal number of regression classes. These experiments are shown in Table 7.3. We get the best result with 2 regression classes.

In MLLR the characteristics of a speaker are estimated from the test data itself and not from some transcribed enrollment data. Although we can estimate a large number of transforms for any of the training speakers; since in training we have the correct transcription and adequate amount of data, this is not the case for test speakers (unsupervised adaptation with few data). Therefore the number of transforms that can be reliably estimated is limited, because it is necessary to match the test set transforms to the training set transforms.

Using multiple regression classes resulted in sub-optimal performance which is not surprising given the unsupervised nature of the adaptation, the high word error rate and the large number of parameters that have to be estimated given our limited training data. In parentheses the number of parameters estimated for each test speaker is shown under different number of transforms.

MLLR Adaptation		SWBD Test set	
#Trans	#Params	Swbd1	Swbd2
2	(2*1600)	36.1	46.8
4	(4*1600)	36.9	47.5
6	(6*1600)	37.9	48.9

Table 7.3: Word Error Rate (%) of systems with test set MLLR adaptation, for various regression classes. All systems were initialized by MMI trained models.

7.2.2 DSAT Results

We then conducted a series of experiments to compare DSAT to ML-SAT estimation as described in sections (5.7) and (5.5) respectively. Throughout these experiments we used a fixed set of 2 regression classes corresponding to speech and non-speech states based on the results in Table 7.3. During test set recognition on the first pass(5 iterations) global adaptation is performed. We then use the global transform to generate better frame/state alignments which are then used to estimate a set of more specific transforms, using a regression class tree.

Table 7.4 shows the performance of the ML-SAT and DSAT estimation procedures. ML-SAT was performed starting with the best MMIE trained model indicated at iteration 0(35.9%/47.0%). In this implementation of ML-SAT, the transformation parameters and the Gaussian mean and variance parameters, are estimated at each iteration via Baum-Welch over the transcribed training data. In the DSAT experiments only the mean and the transformation parameters are reestimated under the

CML criterion. The variance is not updated between iterations; we keep the variance value estimated at ML-SAT iteration 5. Furthermore the lattice link posteriors used in DSAT are those obtained using the ML baseline model (41.1%/51.1%). Our goal is to show that DSAT can improve over ML-SAT through improved estimation of the speaker dependent models. We expect that further gains could be obtained by optimizing variances as well.

We performed multiple iterations of ML-SAT on the training set. DSAT was initialized by a well-trained ML-SAT system found at iteration 5. A comparison between DSAT (as described in Section 5.7) and ML-SAT is presented in the columns DSAT-2 and ML-SAT of Table 7.4. The DSAT-2 mean and transformation parameters were reestimated at each iteration under the CML criterion. The best DSAT-2 result was obtained after 5 iterations (33.4%/44.2%). For comparison we present results with further iterations of ML-SAT (34.1%/44.9%). These results show that discriminative estimation improves over ML estimation of speaker dependent transforms and speaker independent mean parameters. Since we start from a well trained MMIE system, the gains obtained from DSAT-2 are due to the fact that we incorporate speaker adaptive training into MMIE parameter estimation.

While DSAT-2 was found superior to ML-SAT, performing ML-SAT subsequent to MMI is needed for the best initialization of DSAT. In the DSAT-1 column of Table 7.4 the performance of DSAT initialized with MMIE is presented for a fair comparison with ML-SAT. Experimental results suggest that DSAT should be applied following several iterations of ML-SAT.

Finally, we compare DSAT with MMI-SAT. The previously developed MMI-SAT procedure by McDonough et al.[54] proceeds by fixing the ML-SAT transforms prior to subsequent iterations of MMIE estimation. A comparison between DSAT and MMI-SAT is presented in the columns DSAT-2 and MMI-SAT of Table 7.4. The experimental results show significant improvement over ML-SAT. Also DSAT gives slightly better results after 5 iterations, an absolute difference of 0.2%/0.4%, which is attributed to the discriminative calculation of the transformation matrices.

	ML-SAT		DSAT-1		DSAT-2		MMI-SAT	
	SWBD1	SWBD2	SWBD1	SWBD2	SWBD1	SWBD2	SWBD1	SWBD2
0	35.9	47.0	35.9	47.0	*	*	*	*
1	35.4	46.2	36.1	46.5	34.1	44.7	34.1	44.8
2	35.2	45.6	36.5	46.5	33.8	44.6	33.8	44.6
3	34.8	45.1	36.5	46.7	33.6	44.5	33.7	44.4
4	34.7	45.2	-	-	33.5	44.4	33.5	44.4
5*	34.5	44.8	-	-	33.4	44.2	33.6	44.6
6	34.6	45.0						
7	34.3	45.0						
8	34.3	44.7						

Table 7.4: Word Error Rate (%) of systems trained with ML-SAT, MMI-SAT and DSAT estimation and evaluated on Swbd1 and Swbd2 test sets. The ML-SAT and DSAT-1 models were initialized by MMI trained models. The MMI-SAT and DSAT-2 models were seeded from models found after 5 ML-SAT iterations. Results include unsupervised MLLR test speaker adaptation.

7.2.3 Summary of DSAT Results

In speaker adaptive training the conventional HMM parameter framework is extended to accomodate speaker specific transformations in order to produce matched conditions with the test set. As an alternative to ML estimation we used the CML framework in order to obtain fully discriminative procedures. We conducted a series of experiments to compare DSAT to ML-SAT estimation. We have found that discriminative versions of speaker adaptive training outperform ML training. These new training procedures were evaluated on the Switchboard corpus and gave approximately (1.1%,0.6%) absolute Word Error Rate improvement over the ML estimation procedures.

Chapter 8

Minimum Bayes Risk Estimation on Whole Word Models

In this chapter we present experimental results based on minimum Bayes Risk criteria on speech material from a small (*Alphadigits*) vocabulary task. The system was based on word level HMMs. Word models are the most natural form of speech and they form the output of the ASR systems. They are the obvious choice for small vocabulary applications. Because our system uses whole word models we can use the PL-MMIE Algorithm described in (6.5). Analysis of its performance, shows that it does indeed reduce the individual types of word errors in a way that MMI does not.

8.1 System Description

To develop the basic estimation and decoding mechanisms, we present results on the OGI Alpha-Digits task [60]. This is a fairly challenging small vocabulary task on which we still encounter a relatively high baseline WER (approx. 10%). This ensures that we have a significant number of errors to identify and correct. We begin by presenting the MMI baseline system and analyzing its performance and the errors it makes. The baseline system is built using the HTK Toolkit [85]. The acoustic data is parameterized as 13 element MFCC vectors with first and second order differences. The training set consists of 46,730 utterances. The test set consists of 3,112 utterances.

The baseline maximum likelihood models contain 12 mixtures per state, estimated according to the usual HTK training procedure. Starting from these models, several iterations of MMI estimation were performed. The AT&T Large Vocabulary Decoder [55] was used to generate lattices for the training set where are then transformed into word posteriors based on the lattice total acoustic score. MMI training is then performed at the word level using the word time boundaries taken from the lattices.

8.2 MMIE Results

Using the lattices obtained by the AT&T Decoder above, word level posteriors were then estimated based on the lattice total acoustic score. MMIE was then performed at the word level using the word time boundaries taken from the lattices. The Gaussian model parameters $\theta = (\mu_s, \Sigma_s)$, means and variances are updated by equations (4.7) and (4.8).

The Alpha-Digits task does not have a specific language model, thus recognition both for MMI lattice generation and test set decoding is performed using an unweighted word loop over the vocabulary. Table 8.3, Row 1 shows that significant improvement over the baseline can be obtained by MMI: the initial ML performance of 10.42% WER is reduced to 8.41% before overtraining is observed in the WER. We performed the 'sanity check' of rescoring the pinched lattices with the MMI-5 models: performance was identical to unconstrained rescoring. This verifies that the search space refinement introduces no new errors. Pinching does reduce the lattice search space substantially, however. The Lattice Word Error Rate of the original lattices is 1.27%, which increases to 3.11% after pinching.

8.3 Lattice Cutting and Search Space Refinements

As mentioned before MBR training is computationally expensive and lattice segmentation techniques initially developed for MBR search over large lattices are used to derive efficient iterative estimation procedures. Here we segment the lattice word strings by aligning each path in the lattice to the MAP sentence hypothesis, although in training, the segmentation is found relative to the correct transcription [29, 44]. This segmentation procedure is performed carefully so as to retain the structure of the original lattice in regions of low confidence [44]. No reordering of links from the original lattice is allowed. The process starts by identifying the MAP path in a first-pass ASR lattice (Fig. 8.1, *Top*). Period-1 risk-based lattice cutting is used to reduce the lattice to a sequence of segment sets. In some regions only the MAP path remains (Fig. 8.1, *Middle*).

The pinched lattice can be further pruned so that each low confidence region contains a certain number of competing hypothesis. Thus we further simplify the problem by restricting the segment sets to contain only two competing word sequences. We note that except for pruning, no path in the original lattice is excluded from pinching(Fig. 8.1, *Bottom*). Segment sets that occur less than ten times are discarded.

We then perform the same process on the training set to obtain a collection of segment sets representative of recognition errors found in the training data. We use these two collections to identify the 50 test segment sets that were also observed most frequently in training. In this way we identify a final collection of segment sets that are likely to contain recognition errors and that also occur frequently in the training set. The final step in the search space refinement is to restrict the segment sets initially identified in the test set to the final 50 that also occur frequently in the training set (Fig. 8.1, *Bottom*). Some segment sets not in the final collection (e.g. OH+4) are discarded.

8.4 Unsupervised Selection of Segment Sets

The effectiveness of the approach depends on the unsupervised selection of segment sets and the reliability with which they can be associated with ASR errors. We need to establish first that lattice cutting finds segment sets that are similar to the dominant confusion pairs observed in MMI decoding. We also need to establish that the segment sets identified in the test set are also found consistently in the training set. If these

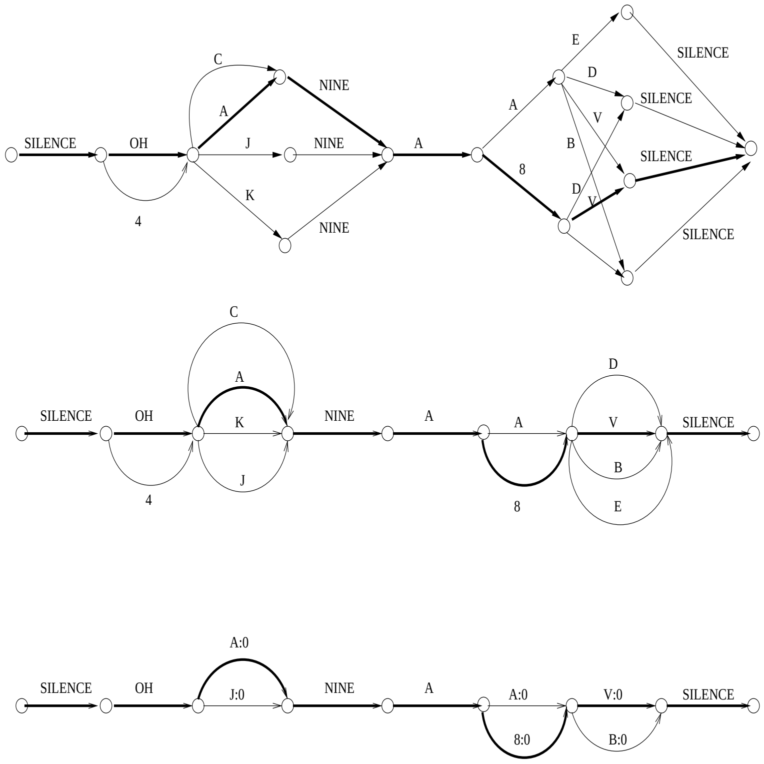


Figure 8.1: Lattice Segmentation for Estimation and Search. *Top*: First-pass lattice of likely sentence hypotheses with MAP path in bold; *Middle*: Alignment of lattice paths to MAP path; *Bottom*: Refined search space $\tilde{\mathcal{W}}$ consisting of tagged segment sets selected for Pinched Lattice MMIE.

two conditions hold, there is the possibility of training discriminative models on the segment sets in the training data and applying them to the test data to resolve the dominant errors remaining after MMI training.

We establish the first point by comparing the dominant MMI confusion pairs in Table 8.2 with the test set segment sets found in Table 8.1 by lattice cutting. There is good agreement among the top eight sets identified in each case, after which there is some divergence. A similar relationship holds between the segment sets identified in test and training reported in Table 8.1. We briefly present the changes in errors as MMI training proceeds on the Alphadigits Task [60].

Table 8.2 presents the most frequently confused words (‘*confusion pairs*’) observed after five iterations of MMI estimation. Iteration 5 is chosen because MMI perfor-

Test Set		Count	Training Set		Count
1	F+S	699	1	F+S	15130
4	P+T	660	4	P+T	10652
6	8+H	650	6	8+H	10706
3	M+N	584	3	M+N	9677
2	V+Z	493	2	V+Z	8005
10	B+D	344	10	B+D	5774
7	L+OH	300	7	L+OH	5289
5	B+V	319	5	B+V	5398
-	A+K	238	-	5+I	4014
-	5+I	236	-	J+K	3628

Table 8.1: Frequent confusion pairs found by lattice cutting. Indices provided for pairs in the dominant MMI confusable pairs.

mance is nearly optimal at that point. We tabulate errors over each word in each class. The notation $\bar{c}_0^{(5)}(1) = 35$ indicates that there are 35 instances in which 'F' is incorrectly recognized as 'S', and $\bar{c}_1^{(5)}(1) = 89$ indicates that there are 89 instances in which 'S' is incorrectly recognized as 'F'. The superscript indicates the MMI iteration.

The process is unsupervised in that no information is required for the test set other than what can be derived from the ASR system. Therefore lattice segmentation can be used both to identify potential errors in the MAP hypothesis and to derive a new search space for the subsequent decoding passes. Because the structure of the original lattice is retained whenever we consider alternatives to the MAP hypothesis, we can perform acoustic rescoring over this pinched lattice.

8.5 Pinched Lattice MMIE Results

Models trained after five MMI iterations (MMI-5) were used to initialize the pinched lattice MMI training procedure described in section (6.5). These models are refined using the training set segments identified for each $\mathcal{W}_i$, as described in the previous sections. After lattice cutting given a particular error pattern found in the test set, we can use training data associated with similar errors to train a discriminative model. PLMMI, is a modified version of MMI for whole word acoustic models

Rank	Error Pair	$\bar{c}_0^{(5)}$	$\bar{c}_1^{(5)}$	Occurences of Each Pair
1.	F+S	35	89	124
2.	V+Z	51	42	93
3.	M+N	24	56	80
4.	P+T	28	39	67
5.	B+V	30	37	67
6.	8+H	15	32	47
7.	L+OH	10	30	40
8.	A+8	20	18	38
9.	C+V	15	16	31
10.	B+D	11	17	28

Table 8.2: Dominant Error Pairs in Unconstrained Recognition after Five MMI Iterations. There are a total of 615 errors identified as belonging to one of these pairs out of a total of 1571 errors.

that is performed over pinched lattices with binary confusion pairs. The training objective for each set of distributions is to maximize (6.20), which is done using MMI over the appropriate training set segments.

We finally apply these models in a full acoustic rescoring of the pinched lattice by applying each $P_i(W|O)$ in decoding over the appropriate segment set. The Period-1 cutting used to identify the segment sets in training also simplifies the MBR decoding procedure of Equation (6.9). The minimum risk decoder is therefore the MAP decoder, and empirical risk is minimized by maximizing the likelihood of the correct hypothesis. When the search space is constrained to follow the MAP hypothesis, the MMI-5 models are used. In regions of the search space corresponding to a segment set $\mathcal{W}_i$, models $P_i(O|W)$ are used.

We observe in the second row of Table 8.3 that pinched lattice MMI estimation (PL-MMI) can yield continued improvement in WER(7.63%). This is in sharp contrast to "regular lattice" MMI which shows evidence of overtraining beyond the fifth iteration. This is done as a fair comparison between pinched lattice and regular MMI, in that the systems being compared are of equal complexity and have the same number of parameters. The improved performance can therefore be attributed to the use of lattice pinching in MMI estimation to refine the space of competing hypotheses.

Iteration	0	1	2	3	4	5	6	7
MMI	10.42	9.75	9.18	8.75	8.53	8.41$*$	8.8	–
PL-MMI	$*$	8.11	7.98	7.82	7.8	7.74	7.68	7.63

Table 8.3: Decoding performance in WER(%) using MMIE vs. Pinched lattice MMIE. PLMMIE models are initialized with MMIE models obtained after 5 iterations.

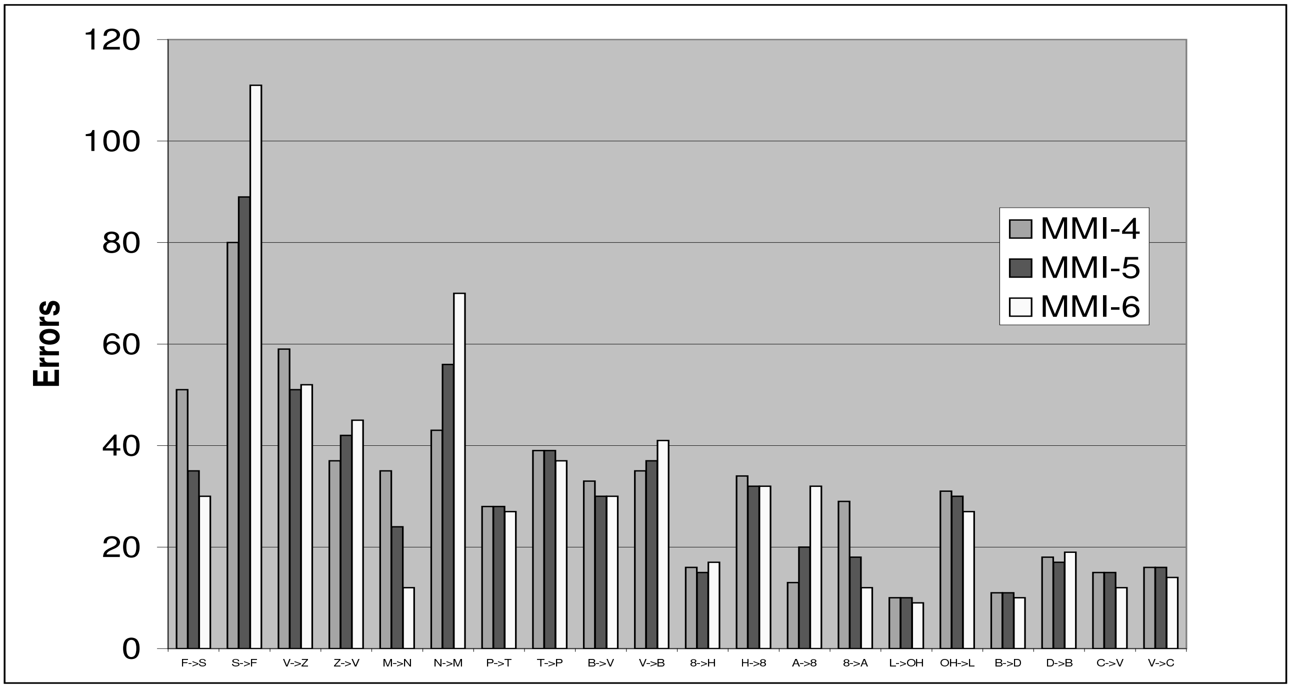
8.5.1 Within-Class Error Analysis

Next we analyze the behavior of the substitution errors made in rescoring with models trained with the MMI and pinched lattice MMIE procedures across 3 consecutive iterations. Eact confusion pair has two types of errors for example 'F+S', 'F' can be misrecognized as 'S' ($F \rightarrow S$) or 'S' can be misrecognized as 'F' ($S \rightarrow F$). Ideally both types of errors should decrease over each of the training iterations shown. However, as can be seen in Fig. 8.2, despite the overall reduction in WER achieved by MMI training, error types are not reduced uniformly as training proceeds. For example, the decrease in $F \rightarrow S$ indicates that the number of times 'F' is incorrectly recognized as 'S' decreases sharply over the three MMI iterations. While this is good in itself, the complementary value of $S \rightarrow F$ indicates that it is gained at the cost of introducing errors in which 'S' is recognized as 'F'. We find that this undesirable behavior is less evident with the Pinched Lattice MMI models in (Fig. 8.3), where the types of errors over each class are more balanced.

8.6 Summary

In this chapter we have described discriminative training procedures suited for small vocabulary tasks. Pinched Lattice MMI, was derived and applied to a whole word recognition task(*Alphadigits*). We considered the 50 most frequent confusion pairs. Analysis of its performance, shows that it does indeed reduce the individual types of word errors in a way that MMI does not. Many types of acoustic errors are excluded from this small number of confusion pairs and as a consequence these errors are not addressed by training. However, the value of this conservative approach is that it allows us to control and study the behavior of the estimation algorithms over

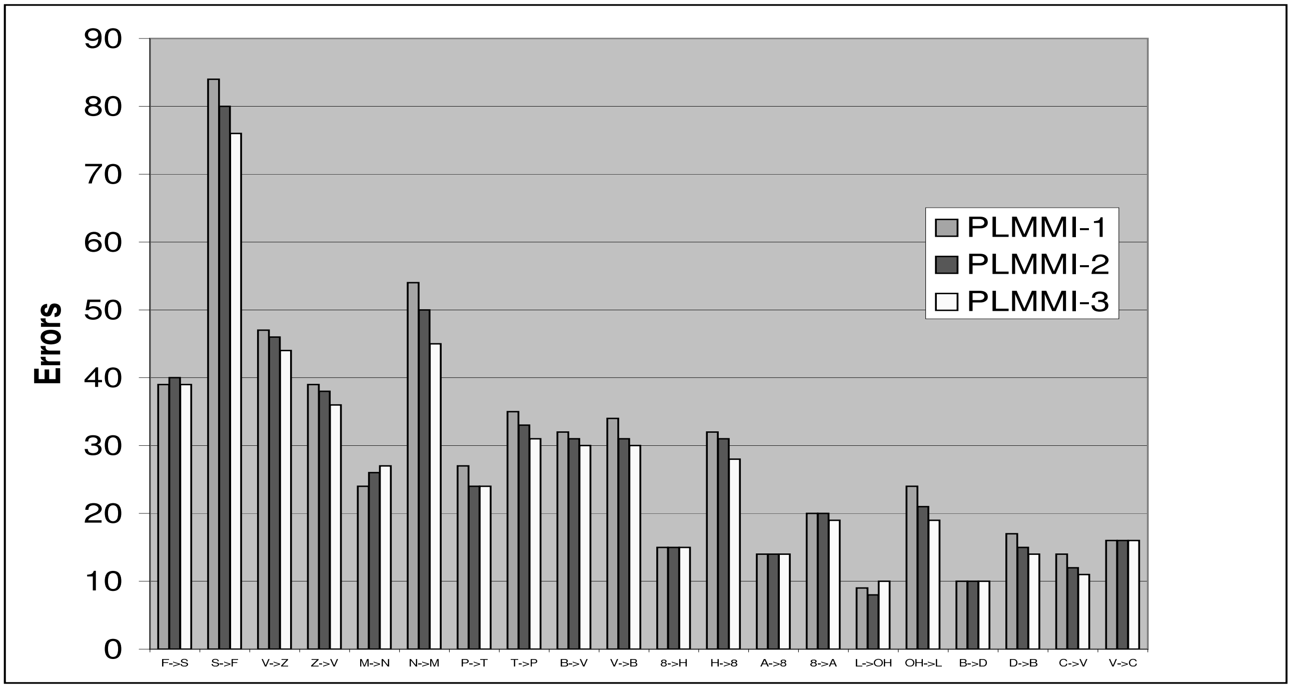
Figure 8.2: Error analysis using MMIE models across 3 iterations for the 10 most dominant confusion pairs shown in Table 8.2.



a manageable number of word pairs. The concept of focusing on confusable pairs for discriminative training using generative models could also be discussed in the context of large margin methods (such as SVMs) [79]. In that case the loss function of SVMs also focuses on the strongest competitor to define the 'margin'.

In contrast to small vocabulary speech recognition where word level models are used and each unit is observed with sufficient frequency in the training corpus, in large vocabulary tasks word level models are not an attractive solution and instead triphones are used. Large vocabulary tasks are likely to contain more acoustically confusable words, thus a large number of word models will require more training data. Sharing model parameters is the obvious solution to improve trainability. As a result Pinched Lattice MMIE can not be efficiently implemented in LVCSR systems were we use triphone models rather than word level models, and time boundaries are sometimes missing or vary in lattices. In the next chapter we will consider discriminative

Figure 8.3: Error analysis using PL-MMI models across 3 iterations for the 10 most dominant confusion pairs shown in Table 8.2.



training algorithms suited for large vocabulary tasks.

Chapter 9

LVCSR Performance with MBRDT Acoustic Models

As mentioned before the training and decoding procedures of most current state-of-the-art Automatic Speech Recognition (ASR) systems are optimized with respect to the sentence error rate (SER) metric that is rarely used in evaluating these systems. Rather than using the (SER) metric as a training criterion we estimate the acoustic models under a criterion that is more closely related to the ASR recognition performance namely the word error rate (WER). We next evaluate discriminative training algorithms that focus on reducing the overall risk over the training data from two large vocabulary tasks, the *SWITCHBOARD* and the *MALACH Corpus.*

9.1 Summary of Minimum Bayes Risk Algorithms

We have used the induced loss functions and pinched and pruned lattices that can be derived from lattice segmentation to simplify the implementation of Minimum Bayes Risk Discriminative Training for large vocabulary ASR systems. The first algorithm, PLMBRDT described in (6.4), is a direct application of the minimum risk estimation procedure of Kaiser et al.[41, 42] under the induced loss function. The other procedure, "One Worst" described in (6.6) is a simple form of corrective training in which the MMI-variant improves the likelihood of the reference hypothesis relative to the worst competing candidate found in the pinched lattice.

Both the PLMBRDT and the “One Worst” training procedures are carefully constructed so that they can be applied to large vocabulary ASR tasks with sub-word acoustic models. Once the pinched and pruned evidence space is expanded into an N-Best list of sentence hypotheses, the Forward-Backward algorithm is performed with respect to each hypothesis to generate the statistics needed for minimum risk reestimation. In this way we do not need to keep track of the word or subword model boundary times found in the initial lattice generation. Applying the estimation procedure to a large vocabulary task is as straightforward as performing Forward-Backward passes with respect to the transcriptions in the N-Best list extracted from the pruned evidence space and weighting the resulting statistics by the factor $K(W', \tilde{\mathcal{W}})$.

9.1.1 MALACH System Description

The MALACH Czech[9] baseline acoustic models were built from 62 hours of data with 24065 utterances. The speech was parameterized into 39-dimensional, MFCC coefficients, with delta and acceleration coefficients. The test set consisted of 954 utterances selected from held-out speakers (approx. 2 hours of speech). The MALACH language model was a back-off bigram with a 83K word vocabulary.

Lattice-based MMI was performed in each domain. The SWITCHBOARD lattices were generated once and the link posteriors were fixed for three iterations of MMI. In MALACH, the link posteriors were reestimated after each of six MMI iterations.

9.2 Minimum Bayes Risk Discriminative Training Steps

Following an initial lattice generation decoding pass over the training set, we use lattice cutting with respect to the correct hypothesis to produce pinched lattices that identify low-confidence segments W_i that are likely to contain recognition errors(Fig. 8.1, *Middle*). In regions of low confidence, the search space contains portions of the MAP hypothesis along with confusable alternatives. In regions of high confidence, the search space is restricted to follow the MAP hypothesis itself.

This produces a very large number of “confusable” pairs and in these experiments we focused only on the most frequently observed pairs. We identify those confusion pairs that are observed more than 75 & 5 times in the SWITCHBOARD training data and more than 100 times in MALACH. The less frequently occurring pairs are discarded. As an example, suppose that the pair HOW, NOW was observed less than 75 times in the pinched training set lattices. In each observed instance, the link corresponding to the incorrect word hypotheses (NOW) would be discarded and only the single link corresponding to the correct word (HOW) would be retained. This reduces the number of different types of binary confusions in SWITCHBOARD from 31467 to 159 and from 25847 to 117 in MALACH. This corresponds to a rate of 0.2 and 0.13 confusion pair per correct word in SWITCHBOARD and MALACH, resp;

Only those utterances that contain the confusion pairs in the final collection are selected for discriminative training while the rest are thrown out. We observed that due to this aggressive filtering, many training set lattices are reduced to a single word sequence, i.e. the reference transcription. These utterances do not contribute to the overall training criterion and they are therefore removed from the PLMBRDT training data. The MALACH training set is reduced from 24,065 to 15,436 utterances, and the SWITCHBOARD training set is reduced from 22,580 to 15,741 utterances.

We found that after filtering the average number of binary confusion pairs in each pinched training set lattice is 2.14 in SWITCHBOARD and 3.12 in MALACH. Hypothesis lists are then generated from these pinched lattices, resulting in an average transcription list depth of 13.1 in SWITCHBOARD and 36.5 in MALACH. The PLMBRDT calculations of Equations (6.7) and (6.8) are carried out over these lists of hypotheses, and the hypothesis needed for the “One Worst” algorithm is also extracted from them.

9.2.1 Minimum Bayes Risk Performance on SWITCHBOARD

MMIE does not directly measure the number of classification errors (WER) on the training data, but rather the sentence error rate(SER). Table 9.2 presents results

	SWITCHBOARD		MALACH
(hours /utterances)	16.9/22580		62.4/24065
Initial confusion pairs (types/ tokens)	25948 /99199		31467 /120695
Occurence threshold used	5	75	100
Segment sets retained after discarding infrequent confusion pairs	2139 /66349	159/33821	117/48302
Avg conf pairs (per word/ per uuterance)	0.35/3.37	0.2/ 2.14	0.13 / 3.12
Reduced training set acoustic data (hours/utterances	15.0 /19687	13.0/ 15741	52.4 / 15436
Avg depth of N-Best lists from pinched lattices	48.8	13.1	36.5

Table 9.1: Training sets statistics before and after lattice cutting for different pruning thresholds. Material comes from the *SWITCHBOARD* and the *MALACH Corpus.*

Table 9.2: Minimum Bayes Risk Results for those confusion pairs with more than 5 & 75 counts. We compare the PLMBRDT and 'One Worst" training procedures on the SWBD test set.

Occurence threshold	5		75	
Iteration	SwitchBoard1 MMIE baseline it3 39.9			
	PL-MBRDT	One Worst	PL-MBRDT	One Worst
1	39.6(0.082)	39.3(0.01)	39.6(0.05)	39.7(0.08)
2	39.3(0.018)	39.4(0.11)	39.5(0.103)	39.2(0.011)
3	39.5(0.23)	—	39.4(0.112)	39.8(0.667)
Iteration	SwitchBoard2 MMIE baseline it3 49.7			
	PL-MBRDT	One Worst	PL-MBRDT	One Worst
1	49.7(0.826)	49.5(0.23)	49.7(0.826)	49.7(0.726)
2	49.5(0.36)	49.6(0.52)	49.4(0.16)	49.2(0.184)
3	49.4(0.23)	—	49.7(0.112)	49.8(0.928)

with the most frequently confused words that have more than 75 counts. Again a useful lower bound on D_s is the value which ensures that all variances remain positive and a Gaussian specific value was used for our experiments. In PL-MBRDT, the Gaussian parameters are calculated by equations (6.7) and (6.8). We see that PL-MBRDT gives a modest (0.4%/0.3%) but significant improvement over MMIE models. We also apply discriminative training on the pinched lattice by considering only two hypotheses (One Worst). The Gaussian parameters are calculated by (6.25) and (6.26) giving an improvement of(0.7%/0.3%) over MMIE models. The reduction on (SWBD2) is less than (SWBD1) because the training set is overwelmingly from (SWBD1) (16.4 hours) and therefore the confusion pairs in the final collection are mostly from that set.

In parentheses the p-value is shown [11], under the significance test between each system and the MMIE system, with the null hypothesis that there is no performance difference between the two systems. We get a (0.7%/0.3%) improvement over conventionally trained MMIE models. From these results we argue that it is useful to develop training procedures that are more closely related to ASR evaluation criteria.

9.2.2 Minimum Bayes Risk Training on MALACH

We start with a well trained ML-system(44.3%) to seed the MMIE training. These results are shown in Table 9.3. We get the best MMIE result after 6 iterations(41.5%), we also re-estimate the lattice link posteriors at each iteration.

Table 9.3: MMIE results on MALACH-Cz

	MALACH-Cz MMIE
ML baseline	44.3
1	43.4
2	42.4
3	42.1
4	41.9
5	41.6
6 ($*$)	41.5

We then apply our MBR training procedure as described before. We select those confusion pairs that have more than 100 counts in the training set Experimental results are shown in Table 9.4. In parentheses the p-value is shown [11], under the significance test between each system and the MMIE system, with the null hypothesis that there is no performance difference between the two systems. PL-MBRDT results are shown in the first column of Table 9.4 with an improvement of (0.5%). By considering only two hypotheses (One-Worst) where the Gaussian parameters are calculated by (6.25) and (6.26) we get an improvement of(0.5%). The performance of the OneWorst approach in particular suggests that, even though sparse, the sets of competing hypotheses identified by lattice pinching can be used for discriminative training.

9.2.3 Contribution of the Loss Function

In most studies on discriminative training criteria, the recognition performance of discriminative training is usually compared to ML training. In particular, there is very limited information on direct comparison between different discriminative training criteria. However in the last column of Table 9.4 (PLMBRDT 0/1) we use PLMBRDT with a 0/1 loss function for calculating the $K(W', \tilde{\mathcal{W}})$ (which corresponds to doing MMIE over the pinched lattice), rather than using the Levenshtein distance.

This experiment shows the importance of using the WER as a training criterion rather than the SER. Loosely speaking, we conclude that the loss function contributes as much to the PLMBRDT gains as does the refinement of the evidence space. This is also consistent with the performance of the One Worst approach, which is constructed to pick the most errorful hypothesis from the refined search space. We conclude that it is beneficial to incorporate both the refined search space and the relative costs of the competing hypotheses in PLMBRDT.

9.3 Analysis of Minimum Bayes Risk Results

In the previous two chapters a framework for efficient discriminative training based on Minimum Bayes risk criteria was presented for both small and large vocabulary

Table 9.4: Minimum Bayes Risk with threshold = 100 seeded from MMIE after 6 iterations

	MALACH-Cz PLMBRDT		
MMIE baseline (∗)	41.5		
Loss	Levenstein	0/1 loss	One-Worst
1	41.4(0.114)	41.4(0.134)	41.3(0.107)
2	41.3(0.038)	41.3(0.129)	41.2(0.042)
3	41.3(0.112)	41.3(0.08)	41.0(0.003)
4	41.3(0.001)	41.3(0.197)	41.1(0.052)
5	41.1(0.031)	41.4(0.522)	—
6	41.0(0.013)	41.5(0.478)	—

continuous speech recognition systems. In these initial experiments we have focused on the most simple lattice pinching and pruning procedures. Each lattice path is aligned word-by-word against the reference transcription, and binary word confusion pairs are identified. These confusion pairs define the errors that the system will be trained to fix.

We have shown that discriminative training methods based on Minimum Bayes risk criteria(Pinched Lattice MMIE) can yield improvement both in the overall WER and in the distribution of individual word errors in a small vocabulary task such as *Alphadigits*. The same lattice pinching and pruning procedures can be applied to large vocabulary speech recognition. As in the small vocabulary case, we find that these PLMBRDT algorithms can be used to extend the gains obtained by MMI. These results are given on two large vocabulary recognition tasks, the conversational English SWITCHBOARD corpus(Table 9.2), and the spontaneous Czech MALACH corpus(Table 9.4). By varying the definition of the estimation algorithms, we find evidence that the improvement beyond MMI comes from both the inclusion of loss into estimation and from reducing the likelihood of the errorful hypotheses that are identified by pinching and pruning. As mentioned earlier, MMI is a particular instance of risk-based estimation.

From the view of minimizing risk, MMI is better matched to Sentence Error Rate than to Word Error Rate. This is clearly not a fatal shortcoming, in that MMI can be very effective in reducing Word Error Rate. However we find that MMI can be improved by using discriminative training procedures that are matched to the task metric, and we conclude that matching the estimation criterion to the task performance metric is beneficial for speech recognition performance. Minimum Bayes Risk discriminative training consistently gave better results than MMI training. From these results we argue that it is beneficial to develop discriminative training procedures that are more closely related to the recognition performance criteria such as the WER rather than the SER.

Chapter 10

Conclusions & Future Work

10.1 Thesis Summary

The recognition accuracy of large vocabulary ASR systems can be improved in many ways, i.e by optimizing the front-end, increasing the complexity of the language model. In this thesis we have placed emphasis on improving the discriminative capabilities of the acoustic model based on different performance criteria with the aim of improving the overall recognition performance of the system. Discriminative training approaches seem attractive because they directly optimize performance criteria such as Sentence Error Rate in MMIE training and Word Error Rate in MBR training.

The first part of this work has described an implementation of MMIE discriminative training using the Conditional Maximum Likelihood (CML) auxiliary function. It was motivated by the fact that Maximum Likelihood estimation does not consider competing hypotheses. Discriminative training criteria, as opposed to the standard maximum likelihood approaches, directly take into account the connection between the underlying models and the recognition performance of the ASR system. The use of lattices makes it feasible to apply MMIE training as shown in (Table 7.1) to very large HMM-based recognition systems. The re-estimation formulae used give good convergence on large systems. We have shown that MMIE training can yield 1.3% absolute reduction in word error rate on the SWITCHBOARD corpus over MLE. An improvement of 2.8% absolute was achieved on the MALACH corpus (Table 9.3) which shows the importance of reestimating the link posteriors after each iteration. MMIE training provides larger improvements in performance for small vocabulary tasks (Table 8.3) (*Alphadigits*) than when applied to large vocabulary tasks.

We have also described the integration of Discriminative Linear Transforms into MMI estimation for Large Vocabulary Speech Recognition shown in Chapter 5. During adaptation the conventional MMIE framework is enhanced to incorporate linear transforms that alter the means of the output distributions during training. We have developed estimation procedures that find DLTs jointly with MMI for speaker adaptive training (SAT). Thus we obtain fully discriminative training procedures termed (DSAT). This new training procedure was based on the Conditional Maximum Likelihood (CML) auxiliary function. It was evaluated on the SWITCHBOARD corpus and gave approximately (1.1%,0.6%) absolute Word Error Rate improvement over the ML estimation procedures (Table 7.4).

In the second part of this thesis shown in Chapter 6 we have presented an ASR modeling framework that incorporates discriminative training with empirical risk minimization techniques. When performing minimum Bayes risk discriminative training in large vocabulary tasks, a crucial problem is the computation and processing of the alternative word hypotheses. Risk minimization requires explicit enumeration of the alternative word hypotheses, and therefore lattice based estimation procedures are not readily available.

Motivated by efficient MBR decoding techniques that incorporate lattice segmentation strategies, we suggested a novel estimation method that attempts to minimize the empirical loss over the training set. Lattice segmentation decomposes the single large lattice into a sequence of smaller sub-lattices. This sequence is termed "pinched" lattice. It is significantly smaller that the original lattice yet still representative of the speech recognition errors that occur in training. Ideally the risk of each word string in the original lattice is unchanged after segmentation. During training each sentence in the pinched lattice is given a different weighting based on the number of errors and the acoustic likelihood. Lattice cutting is used first to identify distinct regions in the search space that are likely to contain errors, and then used in rescoring with models trained specifically to resolve these errors. This work focuses on efficiently incorporating the Levenstein distance into parameter estimation. However the formulation is very general and also supports other types of string-to-string loss functions.

A PLMBRDT variant, Pinched Lattice MMI, was derived and applied to a whole word recognition task(*Alphadigits*). Analysis of the performance (Table 8.3), shows that it does indeed reduce the individual types of word errors in a way that MMI does not. The lattice segmentation framework used in *Alphadigits* has also formed the basis for other novel estimation and classification procedures [79]. The same lattice pinching and pruning procedures can be applied to large vocabulary speech recognition tasks. As in the small vocabulary case, we find that these PLMBRDT algorithms can be used to extend the gains obtained by MMI (Table 9.2), (Table 9.4). From the experimental results, we argue that lattice segmentation and estimation techniques based on empirical risk minimization can be integrated with discriminative training to yield improved performance.

10.1.1 Suggestions For Future Work

Casting the ASR problem as a Minimum Bayes-Risk decision problem provides a rigorous framework for the integration of discriminative search and estimation procedures based on the Word Error Rate rather that the Sentence Error Rate. Lattice segmentation techniques were used to focus more attention on the recognition of certain confusable pairs. Due to the great diversity of ASR errors in large vocabulary tasks, we expect the primary challenge to be robust estimation of discriminative models from sparse training data. We expect that constrained, discriminative estimation procedures will prove useful in these problems [72].

Within the minimum Bayes risk discriminative training framework presented here, much work could be done on refining the selection of confusable pairs and the choice of segment sets. Many extensions on the techniques reported here are possible which could improve the effectiveness of the procedure. For example the minimum bayes risk acoustic models developed in this thesis can be used specifically in MBR decoding. This yields matched conditions during both training and decoding. MBR decoding has been found to consistently provide improved performance relative to straightforward maximum likelihood (ML) decoding procedures. This is usually credited to the integration of the task performance criterion(WER) directly into the decoding procedure[51, 21, 28].

The aim of this work was to build up a framework for efficient discriminative training based on different performance criteria, so as to improve both small and large vocabulary continuous speech recognition. Experimental results show that discriminative training schemes such as MMIE and MBR training can yield better estimates of the HMM gaussian model parameters thus improving recognition performance. We hope that this work would increase understanding of discriminative training and add further insight into the speech recognition problem, since the performance of current state of the art recognition systems is still far worse than that of humans.

Appendix A

Minimum Bayes Risk Estimation

We want to estimate model parameters θ to minimize the empirical loss

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \sum_{W' \in \mathcal{W}} l(\bar{W}, W') P(W'|O;\ \theta) \tag{A.1}$$

Since we want to minimize equation (A.1) we have to reverse the sign of the objective function. In this occasion, it is necessary to compute the derivative of some HMM-related probability with respect to the paramerer set θ. Such derivatives are fairly easy to compute following Kaiser et al[41, 42].

- Property: The partial derivatives of $Loss(\theta)$ with respect to the gaussian parameters are given by:

$$-\nabla_\theta Loss(\theta) = \sum_{W' \in \mathcal{W}} K_{W'} \nabla_\theta P(O|W') \tag{A.2}$$

where

$$K_{W'} = -\frac{\left\{ [\sum_{W'' \in \mathcal{W}} P(O|W'')P(W'')] l(\bar{W}, W') - [\sum_{W'' \in \mathcal{W}} P(O|W'')P(W'') l(\bar{W}, W'')] \right\} P(W')}{[\sum_{W'' \in \mathcal{W}} P(O|W'')P(W'')]^2} \tag{A.3}$$

- Proof: This follows Kaiser[41]. By applying the Bayes rule we get:

$$Loss(\theta) = \sum_{W' \in \mathcal{W}} l(\bar{W}, W')P(W'|O;\ \theta) = \sum_{W' \in \mathcal{W}} l(\bar{W}, W')\frac{P(O|W')P(W')}{P(O)} \tag{A.4}$$

where $P(O) = \sum_{W'' \in \mathcal{W}} P(O|W'')P(W'')$. Therefore

$$\begin{aligned}
-\nabla_\theta Loss(\theta) &= -\nabla_\theta \left\{ \sum_{W' \in \mathcal{W}} l(\bar{W}, W')\frac{P(O|W')P(W')}{P(O)} \right\} = \\
&- \nabla_\theta \left\{ \frac{\sum_{W' \in \mathcal{W}} l(\bar{W}, W')P(O|W')P(W')}{\sum_{W'' \in \mathcal{W}} P(O|W'')P(W'')} \right\} = \\
&- \frac{\sum_{W' \in \mathcal{W}} l(\bar{W}, W')\nabla_\theta\{P(O|W')P(W')\}\,[\sum_{W'' \in \mathcal{W}} P(O|W'')P(W'')]}{[\sum_{W'' \in \mathcal{W}} P(O|W'')P(W'')]^2} \\
&+ \frac{\sum_{W' \in \mathcal{W}} l(\bar{W}, W')P(O|W')P(W')\nabla_\theta \left\{ \sum_{W'' \in \mathcal{W}} P(O|W'')P(W'') \right\}}{[\sum_{W'' \in \mathcal{W}} P(O|W'')P(W'')]^2}
\end{aligned} \tag{A.5}$$

The above equation can be written as

$$\begin{aligned}
-\nabla_\theta Loss(\theta) &= -\frac{[\sum_{W'' \in \mathcal{W}} P(O|W'')P(W'')] \sum_{W' \in \mathcal{W}} l(\bar{W}, W')P(W')\nabla_\theta P(O|W')}{[\sum_{W'' \in \mathcal{W}} P(O|W'')P(W'')]^2} \\
&+ \frac{\sum_{W' \in \mathcal{W}} l(\bar{W}, W')P(O|W')P(W') \left\{ \sum_{W'' \in \mathcal{W}} P(W'')\nabla_\theta P(O|W'') \right\}}{[\sum_{W'' \in \mathcal{W}} P(O|W'')P(W'')]^2}
\end{aligned} \tag{A.6}$$

If we exchange the variables W', W'' in the second term then we have

$$\begin{aligned}
&\sum_{W' \in \mathcal{W}} l(\bar{W}, W')P(O|W')P(W') \left\{ \sum_{W'' \in \mathcal{W}} P(W'')\nabla_\theta P(O|W'') \right\} = \\
&\qquad \sum_{W'' \in \mathcal{W}} l(\bar{W}, W'')P(O|W'')P(W'') \left\{ \sum_{W' \in \mathcal{W}} P(W')\nabla_\theta P(O|W') \right\}
\end{aligned} \tag{A.7}$$

If we substitute (A.7) into (A.6) and rearrange with respect to the terms that contain $\nabla_\theta P(O|W')$ we get

$$- \nabla_\theta Loss(\theta) = \sum_{W' \in \mathcal{W}} \frac{[- \sum_{W'' \in \mathcal{W}} P(O|W'')P(W'')] l(\bar{W}, W') P(W') \nabla_\theta P(O|W')}{[\sum_{W'' \in \mathcal{W}} P(O|W'')P(W'')]^2}$$

$$+ \sum_{W' \in \mathcal{W}} \frac{\left\{ \sum_{W'' \in \mathcal{W}} P(O|W'')P(W'') l(\bar{W}, W'') \right\} P(W') \nabla_\theta P(O|W')}{[\sum_{W'' \in \mathcal{W}} P(O|W'')P(W'')]^2} \quad \text{(A.8)}$$

Or equivalently by using $K_{W'}$ as we have defined them above(A.3) we get

$$-\nabla_\theta Loss(\theta) = \sum_{W' \in \mathcal{W}} K_{W'} \nabla_\theta P(O|W') = \sum_{W' \in \mathcal{W}} K_{W'} P(O|W') \nabla_\theta log(P(O|W')) \quad \text{(A.9)}$$

by using the fact that $\nabla_\theta P(O|W';\ \theta) = P(O|W';\ \theta) \nabla_\theta log P(O|W';\ \theta)$. E.O.P

$K_{W'}$ is the term provided by Kaiser. We then introduce $K(W', \mathcal{W}) = K_{W'} P(O|W')$. Then the gradient of the loss becomes

$$-\nabla_\theta Loss(\theta) = \sum_{W' \in \mathcal{W}} K(W', \mathcal{W}) \nabla_\theta log(P(O|W')) \quad \text{(A.10)}$$

After some arithmetic we have

$$K(W', \mathcal{W}) = [\sum_{W'' \in \mathcal{W}} P(W''|O) l(\bar{W}, W'') - l(\bar{W}, W')]\, P(W'|O)\,. \quad \text{(A.11)}$$

Or equivalently we can write

$$K(W', \mathcal{W}) = [R(\bar{W}, \mathcal{W}; \theta) - l(\bar{W}, W')]\, P(W'|O)\,, \quad \text{(A.12)}$$

for all $W' \in \mathcal{W}$, where $R(\bar{W}, \mathcal{W}; \theta) = \sum_{W'' \in \mathcal{W}} P(W''|O) l(\bar{W}, W'')$ is the expected loss.

Bibliography

[1] T. Anastasakos, J. McDonough, R. Schwartz, and J. Makhoul. A compact model for speaker-adaptive training. In *International Conference on Spoken Language Processing*, pages 1137–1140, 1996.

[2] L. R. Bahl, P.F. Brown, P. V. de Souza, and R.L. Mercer. Maximum mutual information estimation of hidden markov models parameters for speech recognition. In *International Conference on Acoustics, Speech, and Signal Processing*, pages 49–52. IEEE, 1986.

[3] L. R. Bahl, F. Jelinek, and R. L. Mercer. A maximum likelihood approach to continuous speech recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-5(2):179–190, March 1983.

[4] L. E. Baum and J. A. Eagon. An inequality with applications to statistical estimation for probabilistic functions of Markov processes and to a model for ecology. *Bulletin of the American Mathematical Society*, 73:360–363, 1967.

[5] L. E. Baum and T. Petrie. Statistical inference for probabilistic functions of finite state Markov chains. *Annals of Mathematical Statistics*, 37(6):1554–1563, December 1966.

[6] P. J. Bickel and K. A. Docksum. *Mathematical Statistics: Basic Ideas and Selected Topics*. Holden-Day Series in Probability and Statistics. Holden-Day, Inc., Oakland, California, 1977.

[7] A. Biem and S. Katagiri. Feature extraction based on minimum classification

error/generalized probabilistic descent method. In *IEEE Proc. 1993 Int. Conf. Acoust. Speech Signal Process Minneapolis*, pp. 275-278.

[8] W. Byrne. The JHU March 2001 Hub-5 Conversational Speech Transcription System. In *Proceedings of the NIST LVCSR Workshop.* NIST, 2001.

[9] W. Byrne, D. Doermann, M. Franz, S. Gustman, J. Hajic, D. Oard, M. Picheny, J. Psutka B., Ramabhadran, D. Soergel, T. Ward, and W.-J. Zhu. Automatic recognition of spontaneous speech for access to multilingual oral history archives. *IEEE Transactions on Speech and Audio Processing,Special Issue on Spontaneous Speech Processing July*, 2004.

[10] W. Chou, C.-H. Lee, B.-H. Juang, and F. K. Soong. A minimum error rate pattern recognition approach to speech recognition. *International Journal of Pattern Recognition and Artificial Intelligence*, 8(1):5–31, 1994.

[11] P. D, Fisher W, and Fiscus J. Tools for the analysis of benchmark speech recognition tests. In *International Conference on Acoustics, Speech and Signal Processing*, volume 1, pages 97–100, 1990.

[12] S. B. Davis and P. Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech and Signal Processing*, ASSP-28(4):357–366, August 1980.

[13] A. P. Dempster, A. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38, 1977.

[14] V. Digalakis and L. G. Neumeyer. Speaker adaptation using combined transformation and Bayesian methods. *IEEE Transactions on Speech and Audio Processing*, 4(4):294–300, July 1996.

[15] V. Digalakis, D. Rtischev, and L. G. Neumeyer. Speaker adaptation using constrained estimation of Gaussian mixtures. *IEEE Transactions on Speech and Audio Processing*, 3(5):357–366, September 1995.

[16] V. Doumpiotis and W. Byrne. Pinched Lattice Minimum Bayes Risk Discriminative Training for Large Vocabulary Continuous Speech Recognition. In *Proc. of ICSLP04 South Korea*, 2004.

[17] V. Doumpiotis, S. Tsakalidis, and W. Byrne. Discriminative training for segmental minimum Bayes-risk decoding. In *IEEE Conference on Acoustics, Speech and Signal Processing*. IEEE, 2003.

[18] V. Doumpiotis, S. Tsakalidis, and W. Byrne. Lattice Segmentation and Minimum Bayes Risk Discriminative Training. In *Proc. of EUROSPEECH, Geneva Switzerland*, 2003.

[19] K. Chen et al. Fast speaker adaptation using eigenspace-based maximum likelihood linear regression. In *Proc. ICSLP*, Beijing, Oct. 2000.

[20] R. Kuhn et. al. Eigenvoices for speaker adaptation. In *Proc. ICSLP'98*, pages 1771–1774, Sydney, Australia, 1998.

[21] J. Fiscus. A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (ROVER). In *ASRU 1997*, pages 347–354, 1997.

[22] R. A. Fisher. On the mathematical foundations of theoretical statistics. *Philosophical Transactions of the Royal Society A*, 222:309–768, 1922.

[23] M. Gales. Cluster adaptive training for speech recognition. In *Proc. ICSLP'98*, pages 1783–1786, Sydney, Australia, 1998.

[24] M. Gales and P.C. Woodland. Variance compensation within the mllr framework. In *Technical Report CUED/F-INFENT/TR242*. University of Cambridge Engineering Department, Cambridge, UK, February 1996.

[25] M. J. F. Gales. Maximum likelihood linear transformations for hmm-based speech recognition. *Computer Speech and Language*, 12, 1998.

[26] M. J. F. Gales. Semi-tied covariance matrices for hidden markov models. *IEEE Transactions on Speech and Audio Processing*, 7(3), 1999.

[27] J.L Gauvain and C.H Lee. Bayesian learning for hidden Markov models with Gaussian mixture state observation densities. *Speech Communication*, 11:205–213, 1992.

[28] V. Goel and W. Byrne. Minimum Bayes-Risk automatic speech recognition. *Computer Speech and Language*, 14(2):115–135, 2000.

[29] V. Goel, S. Kumar, and W. Byrne. Confidence based lattice segmentation and minimum bayes-risk decoding of lattice segments. In *European Conference on Speech Communication and Technology*, 2001. Submitted.

[30] V. Goel, S. Kumar, and W. Byrne. Segmental minimum Bayes-risk decoding for automatic speech recognition. *IEEE Transactions on Speech and Audio Processing*, 2003.

[31] P. S. Gopalakrishnan, D. Kanevsky, A. Nádas, and D. Nahamoo. An inequality for rational functions with applications to some statistical estimation problems. *IEEE Transactions on Information Theory*, 37(1):107–113, January 1991.

[32] R. A. Gopinath. Maximum likelihood modeling with gaussian distributions for classification. In *International Conference on Acoustics, Speech, and Signal Processing*. IEEE, 1998.

[33] T.Pfau G.Ruske, R.Faltlhauser and A.Sankar. Extended linear discriminant analysis(elda) for speech recognition. In ICSLP Int Conf on Spoken Language Processing 98, pages pp 1095–1098, Sydney Australia.

[34] A. Gunawardana. Maximum mutual information estimation of acoustic hmm emission densities. Technical Report CLSP Research Note No. 40, CLSP, The

Johns Hopkins Univerisity, 3400 N. Charles St., Baltimore, MD 21218, USA, 2001.

[35] A. Gunawardana and W. Byrne. Discriminative speaker adaptation with conditional maximum likelihood linear regression. In *European Conference on Speech Communication and Technology*, 2001.

[36] R. Haeb-Umbach and H. Ney. Linear discriminant analysis for improved large vocabulary continuous speech recognition. In *International Conference on Acoustics, Speech, and Signal Processing*, pages 13–16. IEEE, 1992.

[37] H. Hermansky. Perceptual linear predictive (PLP) analysis of speech. *Journal of the Acoustical Society of America*, 87(4):1738–1752, April 1990.

[38] M.J. Hunt and C. Lefèbvre. A comparison of several acoustic representations for speech recognition with degraded and undegraded speech. In *International Conference on Acoustics, Speech, and Signal Processing*. IEEE, 1989.

[39] F. Jelinek. *Statistical Methods for Speech Recognition*. MIT Press, 1998.

[40] B.H Juang and S. Katagiri. Discriminative learning for minimum error classification. *IEEE Transactions on Signal Processing*, 40(12):3043–3054, December 1992.

[41] J. Kaiser, B. Horvat, and Z. Kacic. A novel loss function for the overall risk criterion based discriminative training of hmm models. In *ICSLP*. Beijing,China, 2000.

[42] J. Kaiser, B. Horvat, and Z. Kacic. Overall risk criterion estimation of hidden markov model parameters. *Speech Communication*, 38(3-4):383–398, 2002.

[43] N. Kumar. *Investigation of silicon-auditory models and generalization of linear discriminant analysis for improved speech recognition*. PhD thesis, The Johns Hopkins University, 1997.

[44] S. Kumar and W. Byrne. Risk based lattice cutting for segmental minimum Bayes-risk decoding. In *ICSLP 2002*, pages 373–376, Denver, CO, USA, 2002.

[45] C.H Lee, C.H Lin, and B.H Juang. A study on speaker adaptation of the parameters of continuous density hidden Markov models. *IEEE Transactions on Signal Processing*, 39(4):806–814, April 1991.

[46] C. J. Leggetter and P. C. Woodland. Speaker adaptation of continuous density HMMs using multivariate linear regression. *International Conference on Spoken Language Processing*, pages 451–454, 1994.

[47] C. J. Leggetter and P. C. Woodland. Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models. *Computer Speech and Language*, 9:171–185, April 1995.

[48] E. L. Lehmann. Efficient likelihood estimators. *The American Statistician*, 34(4):233–235, November 1980.

[49] V. I. Levenshtein. Binary codes capable of correcting spurious insertions and deletions of ones. *Problems of Information transmission*, 1(1):8–17, 1965.

[50] A. Ljolje. The AT&T LVCSR-2001 system. In *Proceedings of the NIST LVCSR Workshop*. NIST, 2001.

[51] L. Mangu, E. Brill, and A. Stolcke. Finding consensus in speech recognition: word error minimization an d other applications of confusion networks. *Computer Speech and Language*, 14(4):373–400, 2000.

[52] A. Martin, J. Fiscus, M. Przybocki, and B. Fisher. The evaluation: Word error rates and confidence analysis. In *Hub-5 Workshop*, Linthicum Heights, Maryland, 1998. NIST. [Online]. Available: http://www.nist.gov/speech/tests/ctr/hub5e_98/hub5e_98.htm.

[53] A. Martin, M. Przybocki, J. Fiscus, and D. Pallett. The 2000 NIST evaluation for recognition of conversational speech over the telephone. In *Proceeding of the Speech Transcription Workshop*. NIST, 2000.

[54] J. McDonough, T. Schaaf, and A. Waibel. On maximum mutual information speaker-adapted training. In *International Conference on Acoustics, Speech, and Signal Processing*. IEEE, 2002.

[55] M. Mohri, F. Pereira, and M. Riley. *ATT General-purpose finite-state machine software tools*, 2001. http://www.research.att.com/sw/tools/fsm/.

[56] M. Mohri and M. Riley. Integrated context-dependent networks in very large vocabulary speech recognition. In *European Conference on Speech Communication and Technology*, 1999.

[57] A. Nádas. A decision theoretic formulation of a training problem in speech recognition and a comparison of training by unconditional maximum likelihood. *IEEE Transactions on Acoustics, Speech and Signal Processing*, ASSP-31(4):814–817, August 1983.

[58] A. Nadas. Optimal Solution of a Training Problem in Speech Recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-33(1):326–329, 1985.

[59] A. Nadas, D. Nahamoo, and M.A. Picheny. On a model-robust training algorithm for speech recognition. *IEEE Transaction Acoustics, Speech and Signal Processing*, 36:1432–1435, 1988.

[60] M. Noel. Alphadigits. In *CSLU*. OGI, 1997.

[61] Y. Normandin. *Hidden Markov Models, Maximum Mutual Information, and the Speech Recognition Problem.* PhD thesis, McGill University, Montreal, 1991.

[62] Y. Normandin. Maximum mutual information estimation of hidden Markov models. In Chin-Hui Lee, Frank K. Soong, and Kuldip K. Paliwal, editors, *Automatic Speech and Speaker Recognition: Advanced Topics*, chapter 3, pages 57–81. Kluwer, 1996.

[63] J.J Odell. *The Use of Context in Large Vocabulary Speech Recognition.* PhD thesis, Queens' College, University of Cambridge, Cambridge University Engineering Department, Trumpington Street, Cambridge, CB2 1PZ, England, March 1995.

[64] W. Poundstone. Prisoner's dilemma/john von neumann, game theory and the puzzle of the bomb. 1993.

[65] L. Rabiner and B.H Juang. *Fundamentals of Speech Recognition.* Prentice-Hall, 1993.

[66] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, February 1989.

[67] D. Sankoff and J. Kruskal (Eds). *Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison.* Addison-Welsey, 1983.

[68] R. Schlüter. *Investigations on Discriminative Training Criteria.* PhD thesis, RWTH Aachen - University of Technology, 2000.

[69] R. Schlüter, B. Muller, F. Wessel, and H. Ney. Interdependence of language models and discriminative training. In *ASRU 1999*, pages 119–122, 1999.

[70] A. Stolcke, H. Bratt, J. Butzberger, H. Franco, V. R. Rao Gadde, M. Plauché, C. Richey, E. Shriberg, K. S onmez, F. Weng, and J. Zheng. The SRI march 200 Hub-5 conversational speech transcription system. In *Proceeding of the Speech Transcription Workshop.* NIST, 2000.

[71] A. Stolcke, Y. Konig, and M. Weintraub. Explicit word error minimization in n-best list rescoring. In *Eurospeech.* Rhodes, Greece, 1997.

[72] S. Tsakalidis, V. Doumpiotis, and W. Byrne. Discriminative Linear Transforms for Feature Normalization and Speaker Adaptation in HMM Estimation. In *International Conference on Spoken Language Processing*, 2002.

[73] S. Tsakalidis, V. Doumpiotis, and W. Byrne. Discriminative Linear Transforms for Feature Normalization and Speaker Adaptation in HMM Estimation. In *IEEE Transactions on Speech and Audio Processing*, 2003. *To Appear.*

[74] A. W. Tucker. The prisoners' dilemma. 1950.

[75] L. F. Uebel and P. C. Woodland. Discriminative linear transforms for speaker adaptation. In *Proceedings of the Tutorial and Research Workshop on Automatic Speech Recognition*. ISCA, 2001.

[76] L. F. Uebel and P. C. Woodland. Improvements in linear transforms based speaker adaptation. In *International Conference on Acoustics, Speech, and Signal Processing*. IEEE, 2001.

[77] V. Valtchev, J.J. Odell, P.C. Woodland, and S.J. Young. Mmie training of large vocabulary speech recognition systems. In *Speech Communication*, 1997. Vol. 22, pp. 303-314.

[78] V.Doumpiotis and Y.Deng. Eigenspace-based mllr with speaker adaptive training in large vocabulary conversational speech recognition. In *Proc. ICASSP*, Montreal Canada, May. 2004.

[79] V. Venkataramani, S. Chakrabartty, and W. Byrne. Support vector machines for segmental minimum bayes risk decoding of continuous speech. In *ASRU 2003*, 2003.

[80] John von Neumann. The theory of games and economic behavior. 1944.

[81] A. Wald. Note on the consistency of the maximum likelihood estimate. *Annals of Mathematical Statistics*, 20, 1949.

[82] P. C. Woodland. Speaker adaptation: Techniques and challenges. In *Proc. IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 85–90, 2000.

[83] P. C. Woodland and D.Povey. Minimum phone error and i-smoothing for improved discriminative training. In *ICASSP*. IEEE, 2002.

[84] P. C. Woodland and D. Povey. Large scale discriminative training for speech recognition. In *Proceedings of the Tutorial and Research Workshop on Automatic Speech Recognition*. ISCA, 2000.

[85] S. Young, D. Kershaw, J. Odell, D. Ollason, V. Valtchev, and P. Woodland. *The HTK Book, Version 3.0*, July 2000.

[86] Steve Young, Dan Kershaw, Julian Odell, Dave Ollason, Valtcho Valtchev, and Phil Woodland. *The HTK Book, Version 2.2*, January 1999.

Vita

Vlasios Doumpiotis was born on June 28, 1975 in Larisa, Greece. In 1998, he received the Bachelor of Science degree in Electrical and Computer Engineering from Technical University of Crete. He received the degree of Master of Science in Engineering from the Johns Hopkins University in Baltimore, Maryland in 2000. Since then, he has been pursuing his Ph.D. in Electrical and Computer Engineering at the Center for Language and Speech Processing at the Johns Hopkins University. He completed his Ph.D. in September 2004.

In 1999 he won the third award from Ericsson Hellas for his thesis in Technical University of Crete. During 1998-2001 he received award from the Greek Ministry of Education for outstanding academic performance. During the summers of 1999 and 2000 he worked at Lernout & Hauspie(currently Scansoft) as a research scientist. He is currently with Escription(Needham-MA)

His research interests include large vocabulary conversational speech recognition focusing on acoustic modeling, statistical methods and information theory, as well as their application to problems in large vocabulary speech recognition.