Discriminative Training and Acoustic Modeling for Automatic Speech Recognition

Von der Fakultät für Mathematik, Informatik und Naturwissenschaften der RWTH Aachen University zur Erlangung des akademischen Grades eines Doktors der Naturwissenschaften genehmigte Dissertation

vorgelegt von

Diplom–Informatiker Wolfgang Macherey

aus

Düren-Birkesdorf

Berichter: Universitätsprofessor Dr.-Ing. Hermann Ney
Universitätsprofessor Dr.-Ing. Gerhard Rigoll

Tag der mündlichen Prüfung: 9. März 2010

Diese Dissertation ist auf den Internetseiten der Hochschulbibliothek online verfügbar.

Poets say science takes away from the beauty of the stars – mere globs of gas atoms. I, too, can see the stars on a desert night, and feel them. The vastness of the heavens stretches my imagination – stuck on this carousel my little eye can catch one-million-year-old light. It does not hurt the mystery to know a little about it.

Richard P. Feynman
Nobel Prize Laureate in Physics, 1965

Acknowledgments

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

First I would like to thank my advisor Univ.-Prof. Dr. Ing. Hermann Ney, head of the Chair of Computer Science 6 at RWTH Aachen University. This thesis would not have been possible without his advice, continuous interest and support.

I would also like to thank my second reader Univ.-Prof. Dr. Ing. habil. Gerhard Rigoll for agreeing to review this thesis and for his great interest in this work.

I am also very grateful to my mentor Dr. rer. nat. Ralf Schlüter who introduced me to the complex and fascinating field of discriminative training during my time as an undergraduate at RWTH Aachen University.

All the people at the Chair of Computer Science 6 deserve my gratitude for many fruitful discussions, helpful feedback, and for the very good working atmosphere. I want to thank all those who helped me while I was writing this thesis by proofreading it and requesting clarifications. Furthermore, I would like to thank the secretaries, and the system administrators for their continuous support.

Großer Dank gilt meinen Eltern Ria und Peter Macherey, die mir das Studium der Informatik ermöglicht haben. Desweiteren möchte ich mich bei meiner Familie und Freunden für den angenehmen Ausgleich zum Arbeitsleben bedanken.

To Ria and Peter Macherey

Abstract

Discriminative training has become an important means for estimating model parameters in many statistical pattern recognition tasks. While standard learning methods based on the *Maximum Likelihood* criterion aim at optimizing model parameters only class individually, discriminative approaches benefit from taking all competing classes into account, thus leading to enhanced class separability which is often accompanied by reduced error rates and improved system performance. Motivated by learning algorithms evolved from neural networks, discriminative methods established as training methods for classification problems such as complex as *automatic speech recognition*.

In this thesis, an *extended unifying approach* for a class of discriminative training criteria is suggested that, in addition to the *Maximum Mutual Information* (MMI) criterion and the *Minimum Classification Error* (MCE) criterion, also captures other criteria more recently proposed as, for example, the *Minimum Word Error* (MWE) criterion and the closely related *Minimum Phone Error* (MPE) criterion. The new approach allows for investigating a large number of different training criteria within a single framework and thus to yield consistent analytical and experimental results about their training behavior and recognition performance.

This thesis also presents the first successful implementation of a large scale, lattice-based MCE training. Experiments conducted on several speech recognition corpora show that the MCE criterion yields recognition results that are similar to or even outperform the performance gains obtained with both the MWE and the MPE criterion.

The parameter optimization problem is discussed for Gaussian mixture models where the covariance matrices can be subject to arbitrary tying schemes. The re-estimation equations as well as the choice of the iteration constants for controlling the convergence rate are discussed for the case that *full* or *diagonal* covariance matrices are used. In case of full covariance matrices, the problem of choosing the iteration constants in the *Extended Baum* (EB) algorithm is shown to result in the solution of a *quadratic eigenvalue problem*. Two novel methods on setting the iteration constants are proposed that provide faster convergence rates across different variance tying schemes.

This thesis also suggests a novel framework that models the posterior distribution directly as a log-linear model. The direct model follows the principle of *Maximum Entropy* and can effectively be trained using the *Generalized Iterative Scaling* (GIS) algorithm. Both the direct model and its optimization via the GIS algorithm are compared analytically and experimentally with the MMI criterion and the EB algorithm.

Finally, this thesis presents a novel algorithm to efficiently compute and represent the exact and unsmoothed error surface over all sentence hypotheses that are encoded in a word lattice if all parameter settings of a log-linear model are considered that lie along an arbitrary line in the parameter space. While the number of sentence hypotheses encoded in a word lattice is exponential in the lattice size, the complexity of the error surface is shown to be always linearly bounded in the number of lattice arcs. This bound is independent of the underlying error metric.

Experiments were conducted on several standardized speech recognition tasks that capture different levels of difficulty, ranging from elementary digit recognition (SieTill) over read speech (Wall Street Journal and North American Business news texts) up to broadcast news transcription tasks (Hub-4). Questions pursued in this context address the effect that different variance tying schemes have on the recognition performance and to what extent increasing the model complexity affects the performance gain of the discriminative training procedure. All experiments were carried out in the extended, unifying approach for a large number of different training criteria.

Zusammenfassung

Diskriminative Lernverfahren haben sich zu einem wichtigen Instrument der Parameterschätzung in vielen Mustererkennungsaufgaben entwickelt. Während konventionelle, auf dem *Maximum Likelihood* Prinzip basierende Verfahren die Modellparameter nur klassenindividuell schätzen, berücksichtigen diskriminative Verfahren auch klassenfremde Trainingsdaten und führen so zu einer verbesserten Klassentrennbarkeit, was sich oftmals in einer niedrigeren Fehlerrate niederschlägt. Motiviert durch Lernverfahren, die in dem Bereich der neuronalen Netze entwickelt worden sind, haben sich diskriminative Methoden inzwischen als Trainingsverfahren in komplexen Klassifikationsproblemen wie die *automatische Spracherkennung* etabliert.

In dieser Arbeit wird ein *erweiterter, vereinheitlichender Ansatz* für eine Klasse diskriminativer Trainingskriterien vorgestellt, der neben dem *Maximum Mutual Information* Kriterium und dem *Minimum Classification Error* Kriterium weitere Kriterien wie zum Beispiel das *Minimum Word Error* Kriterium oder das hiermit nah verwandte *Minimum Phone Error* Kriterium umfaßt. Der neue Ansatz ermöglicht es, die genannten sowie zahlreiche weitere Kriterien in einer einheitlichen Theorie darzustellen und somit zu klaren Aussagen in der theoretischen sowie experimentellen Performanzanalyse zu kommen.

In dieser Arbeit wird ferner die erste erfolgreiche Implementierung eines rein Wortgraph-basierten MCE Trainings für großes Vokabular vorgestellt. Experimente, die auf zahlreichen Spracherkennugskorpora durchgeführt wurden, zeigen, daß die mit Hilfe des MCE Kriteriums erzielten Performanzen in derselben Größenordnung liegen wie die mit dem MWE und dem MPE Kriterium erzielten Fehlerraten, beziehungsweise diese sogar zu übertreffen vermögen.

Das Parameteroptimierungsproblem wird für *Hidden Markov Modelle* mit Gaußschen Mischverteilungsdichten formuliert, wobei die Reestimationsgleichungen als auch die Wahl der Iterationskonstanten für die Fälle diskutiert werden, daß entweder *voll besetzte* oder *diagonale* Kovarianzmatrizen verwendet werden. Die Kovarianzmatrizen können hierbei als gemeinsamer Parameter in die Mischverteilungsdichten beliebiger Zustände eingehen (sogenannte Varianz *tying* Schemata). Speziell für den Fall voll besetzer Kovarianzmatrizen wird gezeigt, daß die Wahl der Iterationskonstanten im *erweiterten Baum* (EB) Algorithmus auf die Lösung eines *quadratischen Eigenwertproblemes* zurückgeführt werden kann. Zwei neue Methoden zur Wahl der Iterationskonstanten werden vorgeschlagen, die unabhängig vom verwendeten Varianz *tying* Schema zu einer schnelleren Konvergenzrate führen als dies beispielsweise mit dem traditionellen EB Algorithmus möglich ist.

In dieser Arbeit wird darüber hinaus ein neuer Ansatz vorgestellt, der eine direkte Beschreibung der Posterior-Verteilung mittels eines log-linearen Modells ermöglicht. Es wird gezeigt, daß das direkte Modell dem Prinzip der *Maximalen Entropie* folgt und mit Hilfe des *Generalized Iterative Scaling* (GIS) Algorithmus effektiv trainiert werden kann. Das direkte Modell sowie die Optimierung mittels des GIS Algorithmus werden analytisch und experimentell mit dem MMI Kriterium sowie dem EB Algorithmus verglichen.

Schließlich wird in dieser Arbeit ein neuer Algorithmus vorgestellt, mit dessen Hilfe sich die exakte und ungeglättete Fehleroberfläche sämtlicher in einem Wortgraphen repräsentierten Satzhypothesen berechnen und speichern läßt, falls alle möglichen Parameterkonstellationen eines log-linearen Modells, die auf einer beliebigen Geraden im Parameterraum liegen, betrachtet werden. Während die Anzahl der in einem Wortgraphen repräsentierten Satzhypothesen exponentiell in der Größe des Wortgraphen ist, wird gezeigt, daß die Komplexität einer solchen Fehleroberfläche stets linear in der Anzahl der Kanten des Wortgraphen beschränkt ist. Diese Schranke ist unabhängig vom betrachteten Fehlermaß.

Experimente wurden auf verschiedenen, standardisierten Spracherkennungskorpora mit unterschiedlichem Schwierigkeitsgrad durchgeführt, die von elementarer Verbundziffernkettenerkennung (SieTill) über gelesene Meldungen aus Writschaftsfachblättern (Wall Street Journal und North American Business News) bis hin zur automatischen Verschriftung von Rundfunk-

nachrichten (HUB-4) reichen. Die hierbei untersuchten Fragestellugen betreffen den Einfluss verschiedener Varianzmodelle auf die Erkennungsperformanz sowie den Effekt, den eine Zunahme der Modellkomplexität auf die zu erwartende Erkennugsfehlerrate ausübt. Sämtliche Untersuchungen wurden im erweiterten, vereinheitlichenden Ansatz für eine Reihe verschiedener diskriminativer Kriterien durchgeführt.

Contents

1 Introduction **1**

2 Automatic Speech Recognition **3**
2.1 Signal Analysis . . . 4
2.2 Acoustic Model . . . 6
2.3 Language Model . . . 9
2.4 Global Search . . . 9
2.5 Acoustic Model Training . . . 10
2.6 The Discriminative Approach . . . 15
2.7 Summary . . . 17

3 State-of-the-Art and Scientific Goals **19**
3.1 Discriminative Training Criteria . . . 19
3.2 Parameter Optimization . . . 22
3.3 Discriminative Acoustic Modeling . . . 23
3.4 Direct Models for Automatic Speech Recognition . . . 24
3.5 Discriminative Model Combination . . . 25
3.6 Scientific Goals . . . 25
3.7 Summary . . . 27

4 Discriminative Training **29**
4.1 A General View of Discriminative Training Criteria . . . 29
4.2 Parameter Optimization . . . 37
4.3 Convergence Control . . . 43
4.4 Optimization in the Extended Unifying Approach . . . 48
4.5 Growth Transformations and Auxiliary Functions . . . 51
4.6 Summary . . . 52

5 Estimating Discriminative Statistics **53**
5.1 Modeling Correct and Competing Classes . . . 54
5.2 Forward-Backward Probabilities on Word Lattices . . . 61
5.3 Forward-Backward Accuracies on Word Lattices . . . 67
5.4 Word-Lattice Based MCE Training . . . 70
5.5 Methods for Accelerating the Training Procedure . . . 73
5.6 Summary . . . 75

6 Experimental Results **77**
6.1 Evaluation Metrics and Significance Tests . . . 77
6.2 Speech Corpora and Baseline Systems . . . 78
6.3 Experiments . . . 86
6.4 Conclusions . . . 126

7 Maximum Entropy and Discriminative Training **127**
7.1 The Maximum Entropy Principle . . . 128
7.2 Maximum Entropy in Automatic Speech Recognition . . . 131
7.3 Experiments . . . 133
7.4 Recent Developments . . . 134
7.5 Conclusions . . . 134

8 Minimum Error Rate Training **137**
8.1 Introduction . . . 137
8.2 Minimum Error Rate Training on N-best Lists . . . 138
8.3 Minimum Error Rate Training on Lattices . . . 141
8.4 Upper Bound for Size of Envelopes . . . 144
8.5 Summary . . . 145

9 Discriminative Linear Transformations **147**
9.1 Maximum Likelihood Based Linear Feature Transformations . . . 148
9.2 A Robust Optimization Procedure . . . 152
9.3 Discriminative Estimation of Linear Transformations . . . 153
9.4 Experiments . . . 156
9.5 Conclusions . . . 157

10 Scientific Contributions **159**
10.1 Conclusions . . . 161
10.2 Outlook . . . 161

A Symbols and Acronyms **163**
A.1 Mathematical Symbols . . . 163
A.2 Acronyms . . . 166

B Detailed Calculations **169**
B.1 Detailed Calculations for Chapter 2 . . . 169
B.2 Detailed Calculations for Chapter 4 . . . 171
B.3 Detailed Calculations for Chapter 9 . . . 181

Bibliography **187**

List of Tables

4.1 A class of discriminative training criteria 31
4.2 Smoothing functions and their first-order derivatives 50

5.1 Sizes of HMM networks for different speech corpora 56

6.1 SieTill: corpus statistics 78
6.2 SieTill: baseline results 79
6.3 SieTill: confidence intervals 79
6.4 Wsj-5k: corpus statistics 80
6.5 Wsj-5k: within-word baseline results 81
6.6 Wsj-5k: confidence intervals 81
6.7 Nab-20k and Nab-65k: corpus statistics 82
6.8 Nab-20k and Nab-65k: within-word model baseline results 82
6.9 Nab-20k and Nab-65k: across-word model baseline results 83
6.10 Nab-20k and Nab-65k: confidence intervals 83
6.11 Hub-4: corpus statistics 84
6.12 Hub-4: across-word baseline results with a single, globally pooled variance 84
6.13 Hub-4: across-word baseline results for density-specific variances 85
6.14 Hub-4: baseline results broken down by condition (mult. vars.) 85
6.15 Hub-4: confidence intervals 86
6.16 Wsj-5k: exact match vs. constrained search 87
6.17 Wsj-5k and Nab-20k: acoustic models applied to the Nab-20k test corpus 89
6.18 Nab-20k: improved ML-trained baseline system 90
6.19 Nab-20k: MMI training on improved baseline system 90
6.20 Optimization methods used in discriminative Training 92
6.21 Nab-20k: baseline for fast method 92
6.22 Nab-20k: fast discriminative training with the MMI criterion 92
6.23 Nab-65k: improved within-word baseline results 94
6.24 Nab-65k: MMI training on within-word system 94
6.25 Nab-20k: across word system optimized with various discriminative training criteria 95
6.26 Nab-65k: across word system optimized with various discriminative training criteria 95
6.27 Nab-65k: MMI training with within-word and across-word system 96
6.28 Nab-65k: comparison of the RWTH and the CU-HTK system 98
6.29 SieTill: single, globally pooled variance and density-specific variances (MMI, MLLT) 99
6.30 Hub-4: increasing graph densities (MMI) 101
6.31 Hub-4: increasing the step size (MMI, pooled var.) 103
6.32 Hub-4: effect of different acoustic scaling factor (MMI, pooled var.) 105
6.33 Hub-4: MMI training with and without variance updates 107
6.34 Hub-4: setting the iteration constants if variances are kept fixed (MMI) 110
6.35 Hub-4: setting the iteration constants if variances are re-estimated (MMI) 110
6.36 Wsj-5k: comparison of different methods to control convergence speed 110
6.37 Wsj-5k: changing the slope parameter of the sigmoid function in the MCE criterion 116
6.38 Hub-4: changing the slope parameter in the sigmoid function of the MCE criterion 116

6.39 HUB-4: comparison of discriminative training criteria 118
6.40 WSJ-5k: comparison of discriminative training criteria 118
6.41 HUB-4: MMI training with 1g, 2g, and 3g decoded lattices (male speaker portion) 122
6.42 HUB-4: MMI and MCE training with 1g decoded lattices (male speaker portion) . 123
6.43 HUB-4: MMI and MCE training with 1g decoded lattices (female speaker portion) 123
6.44 HUB-4: error rates obtained with I-smoothing and single densities (MMI) 124
6.45 HUB-4: MMI training broken down by number of mixture components and iteration index (mult. vars.) 124
6.46 HUB-4: MMI training broken down by number of mixture components and recording conditions (mult. vars.) 125
6.47 HUB-4: comparison of the RWTH and the CU-HTK system 125

7.1 SIETILL: corpus statistics 133
7.2 SIETILL: Generalized Iterative Scaling and Extended Baum Algorithm (WER) . . 133

9.1 HUB-4: error rates obtained with the MLLT and the DLT 156

List of Figures

2.1 Architecture of a statistical automatic speech recognition system. 4
2.2 Mel frequency cepstral coefficient feature extraction. 5
2.3 6-state Hidden Markov Model in Bakis topology. 8
2.4 Maximum approximation for Gaussian mixture densities. 11
2.5 Decomposition into forward and backward probabilities. 14
2.6 Maximum Likelihood estimates for a two-class problem. 16
2.7 Maximum Mutual Information estimates for a two-class problem. 17

4.1 Shannon Entropy and Gini Index as function of the Bayes error rate. 32
4.2 Jeffreys' criterion and logistic regression as a function of the Bayes error rate. . . . 33
4.3 Chernoff affinity for and MWE/MPE criterion as function of the Bayes error rate. 34
4.4 Course of the logistic function and its derivative. 36

5.1 Example of a hidden Markov model network for the numerator model. 55
5.2 HMM network for the competing model. 56
5.3 Language model look ahead tree. 59
5.4 Graph transformations to convert HMM networks into LM-LA trees. 59
5.5 Word lattices for the numerator and the denominator model 62
5.6 Merged lattice for the denominator model 62
5.7 Relative temporal overlap between a reference and a hypothesis. 68
5.8 Raw word accuracies on word lattice. 70
5.9 Forward-backward accuracies estimated on a word lattice. 71

6.1 WSJ-5k: exact match vs. constrained search (MMI) 88
6.2 WSJ-5k: exact match vs. constrained search (WER) 88
6.3 NAB-20k: improved ML baseline (MMI) 90
6.4 NAB-20k: improved ML baseline (WER) 91
6.5 NAB-20k: evolution of the number of densities 91
6.6 NAB-20k: fast discriminative training (MMI) 93
6.7 NAB-20k: fast discriminative training (WER) 93
6.8 WSJ tasks: learning curve for increasing model complexities (abs. improvements) . 97
6.9 SIETILL: tied covariance matrices (MMI) 100
6.10 SIETILL: tied covariance matrices (WER) 100
6.11 HUB-4: effect of changing the graph density (MMI) 102
6.12 HUB-4: effect of changing the graph density (WER) 102
6.13 HUB-4: effect of changing the step size (MMI) 104
6.14 HUB-4: effect of changing the step size (WER) 104
6.15 HUB-4: effect of changing the acoustic scaling factor (MMI) 106
6.16 HUB-4: effect of changing the acoustic scaling factor (WER) 106
6.17 HUB-4: training with and without fixing the variance parameters (MMI) 108
6.18 HUB-4: training with and without fixing the variance parameters (WER) 108
6.19 HUB-4: training with and without fixing the variance parameters (norm. factors) . 109
6.20 HUB-4: adjusting the step size with fixed variance parameters (MMI, mult. vars.) . 111

6.21 HUB-4: adjusting the step size with fixed variance parameters (WER, mult. vars.) 111
6.22 HUB-4: adjusting the step size with variance re-estimation (MMI, mult. vars.) . . . 112
6.23 HUB-4: adjusting the step size with variance re-estimation (WER, mult. vars.) . . 112
6.24 HUB-4: adjusting the step size with fixed variance parameters (MMI, pooled var.) 113
6.25 HUB-4: adjusting the step size with fixed variance parameters (WER, pooled var.) 113
6.26 HUB-4: adjusting the step size with variance re-estimation (MMI, pooled var.) . . 114
6.27 HUB-4: adjusting the step size with variance re-estimation (WER, pooled var.) . . 114
6.28 WSJ-5k: adjusting the step size with variance re-estimation (MMI, pooled var.) . . 115
6.29 WSJ-5k: adjusting the step size with variance re-estimation (WER, pooled var.) . 115
6.30 HUB-4: comparison of a class of discriminative training criteria ($\mathcal{F}_{\mathrm{DT}}$, pooled var.) 119
6.31 HUB-4: comparison of a class of discriminative training criteria (WER, pooled var.) 119
6.32 HUB-4: comparison of a class of discriminative training criteria ($\mathcal{F}_{\mathrm{DT}}$, mult. var.) . 120
6.33 HUB-4: comparison of a class of discriminative training criteria (WER, mult. var.) 120
6.34 WSJ-5k: comparison of a class of discriminative training criteria ($\mathcal{F}_{\mathrm{DT}}$) 121
6.35 WSJ-5k: comparison of a class of discriminative training criteria (WER) 121

7.1 ME objective function and word error rates 134

8.1 Upper envelope for a set of lines. 139
8.2 Global error surface. 140

9.1 Different decisions made by the HLDA and the LDA transformation 150
9.2 Effect of the LDA transformation 151
9.3 Effect of the DLT transformation 155
9.4 HUB-4: MLLT and DLT optimization 157

List of Algorithms

1 Forward-Backward Probabilities on Word Lattices 65
2 Discriminative Training Procedure 67
3 Forward-Backward Accuracies on Word Lattices 69

4 Generalized Iterative Scaling 129

5 SweepLine 141
6 Lattice Envelope 143

7 Maximum Likelihood Linear Transform 153

Chapter 1

Introduction

Automatic speech recognition (ASR) belongs to the important technical innovations in the field of man-machine interaction and forms an integral part in many applications such as *spoken document retrieval* [Macherey & Viechtbauer$^+$ 03], *speech-to-speech translation* [Matusov & Kanthak$^+$ 05], or *spoken dialog systems* [Macherey & Bender$^+$ 09]. Often, the performance of these applications depends substantially on the quality of the speech recognition component, and enhancing the recognition accuracy can lead to a significant improvement of the quality of the respective application. Since research in automatic speech recognition began several decades ago, the statistical approach based on *Hidden Markov Models* has prevailed over all other techniques and is nowadays the *de facto* standard.

The basic principle of the statistical paradigm is *Bayes' decision rule*: given a sequence of acoustic observations as the constituent features of a spoken utterance, Bayes' decision rule decides for that word sequence which provides the best explanation for the observations. If the *true* probability distributions are used, Bayes' decision rule can be shown to be optimal among all decision rules. In practice, however, the true probability distributions are typically not known and have to be replaced with appropriate model distributions. The recognition rates thus obtained are decisively determined by the choice of the distribution parameters of the underlying statistical model. Hence, one of the most important tasks in automatic speech recognition is to estimate the model parameters such that the recognized word sequence matches the spoken word sequence as often as possible.

The parameters of the underlying statistical model are estimated on large sets of training samples which are considered to be representative for the aimed application. Commonly used criteria for estimating the model parameters are based on the well known *Maximum Likelihood* approach. Training algorithms that utilize this estimator train, however, only class individually and leave the overlap between the classes out of account. A remedy are the so-called *discriminative training* methods which, in contrast to the classical Maximum Likelihood approach, also consider class extraneous data. Thus, discriminative methods focus on enhancing class separability which often leads to improved recognition performance.

Discriminative training methods established as an important means for estimating model parameters in many statistical pattern recognition tasks. Evolved from learning algorithms developed for neural networks, discriminative methods are nowadays applied to classification problems such as complex as automatic speech recognition [Schlüter & Macherey$^+$ 01, Woodland & Povey 02, Macherey & Haferkamp$^+$ 05] and statistical machine translation [Och 03, Macherey & Och$^+$ 08]. An implementation is, however, often accompanied by numerous problems and challenges: besides an increased amount of computing time, several modeling techniques which proved to be effective in a Maximum Likelihood framework are not always appropriate for discriminative training. The work presented in this thesis therefore addresses two major issues: first, the practical aspects of a discriminative training procedure for *large vocabulary continuous speech recognition* are investigated; second, new methods for discriminative acoustic modeling are developed and evaluated. In particular, the following items are addressed:

- Chapter 2 provides an introduction to automatic speech recognition and briefly describes the four main components of a speech recognizer which comprise the *signal analysis*, the *acoustic model*, the *language model*, and the *global search*. Furthermore, the ML-based training of an acoustic model is described and, based hereupon, the discriminative approach is motivated.
- Chapter 3 gives an overview of the state-of-the-art in discriminative training. In particular, major achievements in discriminative training criteria, methods for parameter optimization and discriminative acoustic modeling are addressed as well as a direct model to automatic speech recognition. Based on the state-of-the-art, the scientific goals for this thesis are defined.
- Chapter 4 presents an *extended unifying approach* for a class of discriminative training criteria. Besides the *Maximum Mutual Information* criterion and the *Minimum Classification Error* criterion, the new approach also captures criteria more recently proposed, including the *Minimum Word Error* criterion and the *Minimum Phone Error* criterion. Both the re-estimation equations for Gaussian mixture distributions as well as the choice of the iteration constants for controlling the convergence rate in a discriminative training environment are presented for the case that *full* covariance matrices with *arbitrary* tying schemes are used. Two novel methods on setting the iteration constants are proposed that turn out to be more robust while providing faster convergence rates if tied variances are used. The chapter ends with a discussion on growth transformations and auxiliary functions.
- Chapter 5 contains details on estimating discriminative statistics and suggests to model the correct and the competing classes using *Hidden Markov Model* (HMM) *networks*. HMM networks are state graphs which can be considered as a hybrid approach between *lexical prefix trees* and *Finite State Transducers*. Two forward-backward algorithms for estimating word posterior probabilities and word accuracies, respectively, are discussed in detail. The chapter also presents details on the first successful implementation of a purely lattice-based *Minimum Classification Error* training for large vocabulary continuous speech recognition. Various techniques to accelerate the training procedure including a client server model for distributed computing on inhomogeneous clusters are described.
- Chapter 6 presents experimental results that address different aspects of a discriminative training for automatic speech recognition including model complexity, variance modeling, comparison of a vast number of training criteria, and optimization procedures. The experiments were conducted on several speech corpora with different levels of difficulty.
- Chapter 7 suggests a log-linear model to automatic speech recognition which replaces the generative model traditionally used in automatic speech recognition with a direct model. The parameter optimization is performed using the *Generalized Iterative Scaling* algorithm and aims at maximizing an entropy criterion. Experimental results are presented on a small vocabulary speech recognition task.
- Chapter 8 presents a novel algorithm that allows for efficiently constructing, representing, and exploring the exact and unsmoothed error surface for *all* sentence hypotheses encoded in a word lattice under virtually any automated error criterion used in natural language processing. A line minimization method is described that can be used to train the parameters of a linear model such that the error rate is directly minimized on training data.
- Chapter 9 presents a novel framework for estimating discriminative linear feature transformations. As a case study, a discriminative version of the *Maximum Likelihood Linear Transform* is derived which aims at decorrelating the feature space when Gaussian mixture models with heteroscedastic covariance matrices are used.
- Chapter 10 summarizes the results and scientific contributions of this thesis. The chapter ends with an outlook on further research directions.

Chapter 2

Automatic Speech Recognition

The statistical approach to automatic speech recognition aims at modeling the stochastic relation between a speech signal and the spoken word sequence with the objective of minimizing the expected error rate of a classifier. The statistical paradigm is governed by *Bayes' decision* rule: given a sequence of acoustic observations $x_1^T = x_1, \ldots, x_T$ as the constituent features of a spoken utterance, Bayes' decision rule decides for that word sequence $w_1^N = w_1, \ldots, w_N$ which maximizes the class posterior probability $p(w_1^N|x_1^T)$:

$$\left[w_1^N\right]_{\text{opt}} = \underset{w_1^N}{\text{argmax}} \left\{p(w_1^N|x_1^T)\right\}. \tag{2.1}$$

Provided that the *true* probability distribution is used, Bayes' decision rule is optimal among all decision rules, that is, on average it guarantees the lowest possible classification error rate [Duda & Hart 73, p. 12]. However, for most pattern recognition tasks – and for automatic speech recognition in particular – the true probability distribution is usually not known but has to be replaced with an appropriate model distribution. In automatic speech recognition, the generative model, which decomposes the class posterior probability into a product of two independent stochastic knowledge sources, became widely accepted[1]:

$$p(w_1^N|x_1^T) = \frac{p(w_1^N) \cdot p(x_1^T|w_1^N)}{p(x_1^T)}. \tag{2.2}$$

The denominator $p(x_1^T)$ in Eq. 2.2 is assumed to be independent of the word sequence w_1^N and hence, the decision rule is equivalent to:

$$\left[w_1^N\right]_{\text{opt}} = \underset{w_1^N}{\text{argmax}} \left\{p(w_1^N) \cdot p(x_1^T|w_1^N)\right\}. \tag{2.3}$$

The word sequence $\left[w_1^N\right]_{\text{opt}}$ which maximizes the posterior probability is determined by searching for that word sequence which maximizes the product of the following two stochastic knowledge sources:

- The *acoustic model* $p(x_1^T|w_1^N)$ which captures the probability of observing a sequence of acoustic observations x_1^T given a word sequence w_1^N.
- The *language model* $p(w_1^N)$ which provides a prior probability for the word sequence w_1^N.

A statistical speech recognizer evaluates and combines both models through generating and scoring a large number of alternative word sequences (so-called *hypotheses*) during a complex search process. Figure 2.1 illustrates the basic architecture of a statistical automatic speech recognition system. Conceptually, a speech recognizer consists of four components: the *signal analysis*, the

[1] In Chapter 7, the generative approach is replaced with a direct model that aims to capture the posterior probability distribution using a log-linear model.

acoustic model, the *language model*, and the *global search*. Each component is briefly described in the following sections.

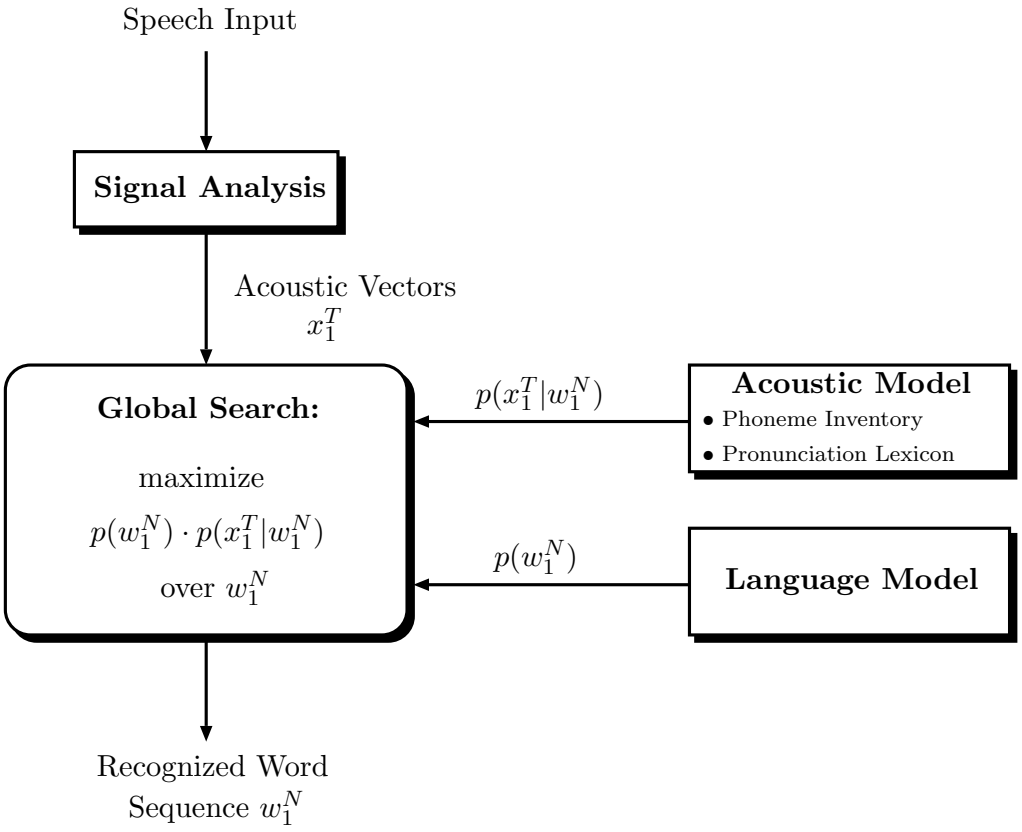


Figure 2.1: Basic architecture of a statistical automatic speech recognition system. The system comprises four components: the *signal analysis*, the *acoustic model*, the *language model*, and the *global search* [Ney 90].

2.1 Signal Analysis

The signal analysis provides a compact parameterization of the speech waveform which is considered to be appropriate for recognizing speech. Common signal processing techniques used in automatic speech recognition are based on *Mel frequency cepstral coefficients* (MFCC) [Davis & Mermelstein 80] or *perceptual linear prediction* (PLP) [Hermansky 89]. In this thesis, the MFCC signal analysis front-end of the RWTH speech recognizer is used. A schematic overview of the feature extraction process is depicted in Figure 2.2. For a thorough description see [Welling 99].

Starting point of the signal analysis is the digitized speech signal, which is emphasized to compensate for the characteristics of the glottis mute and the high pass filtering caused by the lips [Wakita 73]. Since speech is a highly non-stationary signal, the emphasized waveform is truncated into short, overlapping segments by multiplying the signal with a sliding window. The resulting segments are almost stationary and can be further processed by means of a short term spectral analysis based on a *Discrete Fourier Transform* (DFT). To alleviate the distortion of the short term spectrum caused by windowing the signal, the window function is required to

satisfy certain spectral properties. In this thesis, a *Hamming* window is used whose geometry is specified through a window length of 25ms and a frame shift of 10ms [Hamming 89, p. 118]. Every 10ms, the Fourier power spectrum is computed on the windowed signal using a *Fast Fourier Transform* (FFT). The frequency axis of the power spectrum is warped with respect to the *Mel* frequency scale which accounts for the varying frequency resolution of the human ear. A bank of overlapping, on the Mel scale equally spaced, triangular frequency filters is applied to smooth the power spectrum. The logarithm of the filter bank outputs is taken to reduce the dynamic range of the signal and to model the non-linear relationship between the intensity of a sound and its percepted volume. Due to the overlap between adjacent filters, the outputs are highly correlated. The filter outputs can, however, easily be decorrelated and transformed into the cepstral space through a *discrete cosine transform* since the respective correlation matrix has almost *Toeplitz* structure [Davis & Mermelstein 80]. The acoustic features thus obtained are referred to as *Mel frequency cepstral coefficients* (MFCCs). Higher order cepstral coefficients are usually discarded since they only reflect the fine structure of the harmonics which makes them inappropriate for recognizing speech. The lower order coefficients form a rough approximation of the short-term spectral envelope and are thus more suitable for recognizing a spoken utterance. Every 10 milliseconds, the lower order coefficients are computed and combined into an acoustic observation vector. Depending on the quality and bandwidth of the recorded speech signal, acoustic observation vectors typically comprise between 12 (telephone-line recorded data) and 16 (close-talk microphone) MFCCs. Observation vector are optionally augmented with dynamic

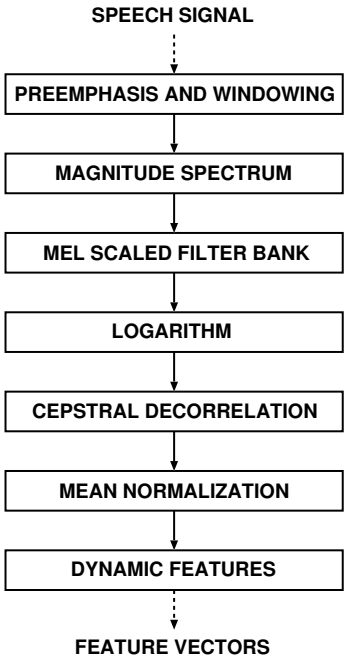


Figure 2.2: Mel frequency cepstral coefficient feature extraction.

features by computing, e.g., the first and second order temporal derivatives of the cepstral coefficients. In addition, adjacent vectors may be concatenated and mapped into a different feature space using linear or affine transformations. Commonly used linear transforms include the *Linear Discriminant Analysis* (LDA) [Haeb-Umbach & Ney 92a, Saon & Padmanabhan$^+$ 00], the *Maximum Likelihood Linear Transform* (MLLT) [Gopinath 98], and the *Heteroscedastic Linear Discriminant Analysis* (HLDA) [Kumar & Andreou 98a].

2.2 Acoustic Model

The acoustic model $p(x_1^T|w_1^N)$ provides a stochastic description for the realization of a sequence of acoustic observation vectors x_1^T given a word sequence w_1^N. Due to data sparsety, the model for individual words as well as the model for entire sentences is obtained by concatenating the acoustic models of basic sub-word units according to a pronunciation lexicon. Sub-word units smaller than words enable a speech recognizer to allow for recognizing words that do not occur in the training data. Thus, the recognition system can ensure that enough instances of each sub-word unit have been observed in training to allow for a reliable estimation of the underlying model parameters.

The type of sub-word units employed in a speech recognizer depends on the amount of available training data and the desired model complexity: while recognition systems designed for small vocabulary sizes ($\leqslant$ 100 words) typically apply whole word models, systems developed for the recognition of large vocabularies ($\geqslant$ 5000 words) often employ smaller sub-word units which may be composed of syllables, phonemes, or phonemes in context. Context-dependent phonemes are also referred to as *n-phones*. Commonly used sub-word units employed in large vocabulary speech recognition systems are n-phones in the context of one or two adjacent phonemes, so-called *triphones* or *quinphones*. Context-dependent phoneme models allow for capturing the varying articulation that a phoneme is subject to when it is realized in different surrounding phonetic contexts (*coarticulation*).

Typically, the constituent phones for various acoustic realizations of the same word are produced with different duration and varying spectral configuration, even if the utterances are produced by the same speaker. Each phone will therefore aggregate an a-priori unknown number of acoustic observations. The temporal distortion of different pronunciations as well as the spectral variation in the acoustic signal can be described via a *Hidden Markov Model* (HMM). A HMM is a stochastic finite state automaton that models the variation in the acoustic signal via a two-stage stochastic process [Baker 75, Rabiner 89]. The automaton is defined through a set of states with transitions connecting the states. The probability $p(x_1^T|w_1^N)$ is extended by an unobservable (*hidden*) variable s representing the states:

$$p(x_1|w_1^N) = \sum_{s_1^T} p(x_1^T, s_1^T|w_1^N). \tag{2.4}$$

The sum in Eq. (2.4) runs over all possible state sequences s_1^T that are consistent with the word sequence w_1^N. Each *path* through the automaton defines an alignment between the observation vectors x_1^T and the state sequence s_1^T. Using the decomposition rule, Eq. (2.4) can be written as:

$$p(x_1^T|w_1^N) = \sum_{s_1^T} \prod_{t=1}^{T} p(x_t|x_1^{t-1}, s_1^t; w_1^N) \cdot p(s_t|x_1^{t-1}, s_1^{t-1}; w_1^N). \tag{2.5}$$

Assuming that $p(x_t|x_1^{t-1}, s_1^t; w_1^N)$ and $p(s_t|x_1^{t-1}, s_1^{t-1}; w_1^N)$ do not depend on previous observations but on the current and the immediate predecessor state only (*first-order Markov* assumption [Rabiner & Juang 93, pp. 321–348]), the acoustic model simplifies to:

$$p(x_1^T|w_1^N) = \sum_{s_1^T} \prod_{t=1}^{T} p(x_t|s_{t-1}, s_t; w_1^N) \cdot p(s_t|s_{t-1}; w_1^N). \tag{2.6}$$

In addition, $p(x_t|s_{t-1}, s_t; w_1^N)$ is assumed to be independent of s_{t-1}:

$$p(x_1^T|w_1^N) = \sum_{s_1^T} \prod_{t=1}^{T} p(x_t|s_t; w_1^N) \cdot p(s_t|s_{t-1}; w_1^N). \tag{2.7}$$

Hence, $p(x_1^T|w_1^N)$ decomposes into a product of two stochastic models: (1) the acoustic *emission probability* $p(x_t|s_t; w_1^N)$ which denotes the probability to observe an acoustic vector x_t at time frame t in state s_t; and (2) the *transition probability* $p(s_t|s_{t-1}; w_1^N)$ for a transition from a predecessor state s_{t-1} into the current state s_t.

Often, the sum in Eq. (2.7) is replaced with the maximum. This approximation is called *Viterbi* or *maximum* approximation [Viterbi 67, Ney 90]:

$$p(x_1^T|w_1^N) \cong \max_{s_1^T} \prod_{t=1}^{T} p(x_t|s_{t-1}, s_t; w_1^N) \cdot p(s_t|s_{t-1}; w_1^N). \tag{2.8}$$

Eq. (2.8) can efficiently be evaluated using dynamic programming. For details, see [Baum 72, Rabiner & Juang 86].

2.2.1 HMM Topologies and Across-Word Models

Commonly used HMM topologies for n-phones are Markov chains consisting of either three or six states. While 3-state Markov chains typically apply a linear transition model that restricts the set of state changes to arcs leading to the same state (*loop transition*) and the immediate successor state (*forward transition*), 6-state Markov chains often provide an additional transition to the next state but one (*skip transition*). Except for whole-word models, HMM topologies used in this thesis are 6-state Markov chains in which each two successive states share a common emission distribution [Bakis 76]. Figure 2.3 depicts a partial view of a sequence of 6-state HMMs unfolded along the time axis for the first three phonemes *s*, *eh*, and *v* of the word "seven" within their triphone contexts.

Generalized n-Phone Models

If context-dependent phones are used as basic sub-word units, the number of states to be modeled will raise exponentially with the context length. Many n-phones will therefore aggregate too few or no observations at all during training to allow for a reliable estimation of the underlying model parameters. To alleviate this problem, states can be tied together, yielding so-called *generalized n-phone models* [Young 92]. Decision-tree based state clustering as employed in this thesis is a widely-used technique that allows for assigning HMMs to n-phones without the need of explicit back-off models. Details of the clustering algorithm can be found in [Beulen & Ortmanns$^+$ 99].

Across-Word Models

Surrounding phonemes do not only affect the pronunciation of a phone within a word but also at word boundaries. These coarticulatory effects across word boundaries are captured by *across-word n-phones*. Across-word n-phones account for the ending and beginning phonemes of two consecutive words as left and right phonetic context. The basic across word model implementation in the RWTH system is described in [Sixtus 03]. Modifications to this implementation used in discriminative training are discussed in Section 5.1.4.

2.2.2 Emission Probabilities

State emission probabilities can be modeled via *discrete probability distributions* [Jelinek 76], *semi-continuous probability distributions* [Huang & Jack 89], or *continuous probability distributions* [Levinson & Rabiner$^+$ 83]. Commonly used models for continuous probability distributions are mixture distributions composed of a weighted sum of *Gaussian* or *Laplacian* probability density

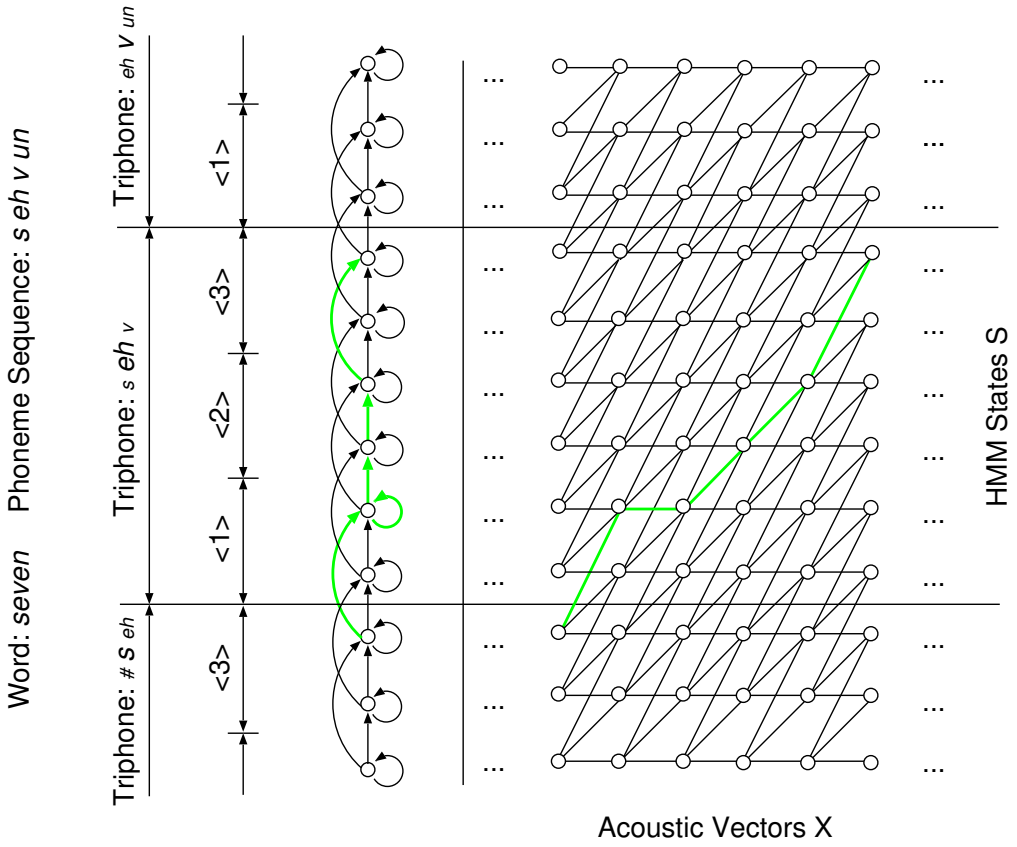


Figure 2.3: 6-state *Hidden Markov Model* (HMM) in Bakis topology unfolded along the time axis for the triphone ${}_{s}eh_{v}$ in the word "seven". The HMM segments are denoted by <1>, <2>, and <3>.

functions. In this thesis, emission probabilities are modeled as *Gaussian Mixture Models* (GMMs):

$$p(x|s) = \sum_{l=1}^{L_s} c_{sl} \cdot \mathcal{N}(x|\mu_{sl}, \mathbf{\Sigma}_k). \tag{2.9}$$

Here, L_s denotes the number of mixture components assigned with state s; c_{sl} denotes a mixture weight; and $\mathcal{N}(x|\mu_{sl}, \mathbf{\Sigma}_k)$ is a multivariate normal distribution with a D-dimensional mean vector $\mu_{sl} \in \mathbb{R}^D$ and a symmetric, positive definite matrix $\mathbf{\Sigma}_k \in \mathbb{R}^{D \times D}$ referred to as *covariance matrix*:

$$\mathcal{N}(x|\mu_{sl}, \mathbf{\Sigma}_k) = \frac{1}{\sqrt{\det(2\pi\mathbf{\Sigma}_k)}} \cdot \exp\Big\{ -\frac{1}{2} (x - \mu_{sl})^\top \mathbf{\Sigma}_k^{-1} (x - \mu_{sl}) \Big\}. \tag{2.10}$$

All mixture weights are positive and subject to a sum-to-one constraint, that is, $\sum_{l=1}^{L_s} c_{sl} = 1$. While mean vectors are specific for each density, covariance matrices may be tied across several mixture components or states. The underlying tying scheme is designated by a set of *equivalence classes* $K \subseteq S \times L$ where S denotes the total number of states and $L \coloneqq \max_{s \in S}\{L_s\}$. Each $k \in \mathcal{K}$ then comprises the set of indices of those densities that share a common covariance matrix. The set of all HMM parameters (including transition probabilities) is denoted by:

$$\theta = \{\theta_s, [p(s|\sigma)] \ : \ \sigma, s = 1, \ldots, S\}, \tag{2.11}$$

In Eq. (2.11), $\theta_s = \{c_{sl}, \theta_{sl} \ : \ l = 1, \ldots, L_s\}$ comprises the set of acoustic model parameters of densities associated with state s; $[p(s|\sigma)]$ denotes the state transition matrix; and $\theta_{sl} = \{\mu_{sl}, \mathbf{\Sigma}_k\}$ denotes the mean and variance parameter of the Gaussian mixture component l in state s. In the following sections, probability distributions depending on a parameter set θ are denoted by p_θ.

2.3 Language Model

The language model $p(w_1^N)$ provides a prior probability for the word sequence $w_1^N = w_1, \ldots, w_N$. Thus, it inherently aims at capturing the syntax, semantics, and pragmatics of a language. Since language models are independent of acoustic observations, their parameters can be estimated from large text collections as, for instance, newspapers, journal articles, or web content. Due to a theoretically infinite number of possible word sequences, language models require suitable model assumptions to make the estimation problem practicable. For large vocabulary speech recognition, *m-gram* language models have become widely accepted [Bahl & Jelinek$^+$ 83]. An m-gram language model is based on the assumption that a sequence of words follows an $(m-1)$-th order Markov process, that is, the probability of a word w_n is supposed to depend only on its $m-1$ predecessor words:

$$p(w_1^N) = \prod_{n=1}^{N} p(w_n|w_1^{n-1}) \tag{2.12}$$

$$\underset{\text{model assumption}}{=} \prod_{n=1}^{N} p(w_n|w_{n-m+1}^{n-1}). \tag{2.13}$$

The word sequence $h_n \coloneqq w_{n-m+1}^{n-1}$ is referred to as *history* of length $m-1$ of the word w_n with the definition $h_n \coloneqq w_1^{n-1}$ if $n < m$, and $h_n \coloneqq \varnothing$ if $n-1 < n-m+1$. The compactness of a language model can be measured by its *perplexity* (PP):

$$PP = \left[\prod_{n=1}^{N} p(w_n|w_{n-m+1}^{n-1}) \right]^{-1/N}. \tag{2.14}$$

The perplexity is an inverse geometric mean and can be interpreted as the average number of choices to continue a word at position n given its $m-1$ predecessor words.

2.4 Global Search

Given a sequence of acoustic observations x_1^T, the objective of the global search is to find that word sequence which maximizes the a-posteriori probability:

$$\left[w_1^N\right]_{\text{opt}} = \underset{w_1^N}{\text{argmax}} \left\{p(w_1^N|x_1^T)\right\} \tag{2.15}$$

$$= \underset{w_1^N}{\text{argmax}} \left\{p(w_1^N) \cdot p(x_1^T|w_1^N)\right\}. \tag{2.16}$$

In principle, the decoder has to align the sequence of acoustic observations x_1^T with all possible state sequences s_1^T that are consistent with a word sequence w_1^N. Using m-gram language models and an acoustic model based on HMMs, the optimization problem is given by:

$$\left[w_1^N\right]_{\text{opt}} = \underset{w_1^N}{\text{argmax}} \left\{ \left[\prod_{n=1}^{N} p(w_n|w_{n-m+1}^{n-1}) \right] \cdot \left[\sum_{s_1^T} \prod_{t=1}^{T} p(x_t|s_t; w_1^N) \cdot p(s_t|s_{t-1}; w_1^N) \right] \right\} \tag{2.17}$$

The complexity of the search process can be reduced by approximating the sum over all paths with the *Viterbi* or maximum approximation:

$$\left[w_1^N\right]_{\text{opt}} \cong \underset{w_1^N}{\text{argmax}} \left\{ \left[\prod_{n=1}^{N} p(w_n|w_{n-m+1}^{n-1}) \right] \cdot \left[\max_{s_1^T} \prod_{t=1}^{T} p(x_t|s_t; w_1^N) \cdot p(s_t|s_{t-1}; w_1^N) \right] \right\} \tag{2.18}$$

The objective of the Viterbi approximation is not to yield a good estimation of the sum over all state sequences; instead, it is introduced under the assumption that the word sequence induced by the most probable state sequence is identical with the exact optimum of Eq. (2.17). In addition, the Viterbi approximation allows for structuring the search space more efficiently by assigning unique time boundaries to each word hypothesis, which is essential for tracing back the optimal decision. An efficient implementation of the Viterbi search based on *dynamic programming* is described in [Ney & Aubert 96].

The organization of the global search may follow a *depth-first* or a *breadth-first* search strategy. The depth-first search strategy is employed by the *A*-search* or *stack-decoding* algorithm. In the breadth-first search strategy, which is used in the *Viterbi*-search, all state hypotheses are expanded time-synchronously [Vintsyuk 71, Baker 75, Sakoe 79, Ney 84]. Since the probabilities of all active state hypotheses are computed at each time frame, likelihoods of competing state hypotheses are comparable. To make the search process more efficient, less likely hypotheses are removed from the search space (*pruning*). This results in a significant reduction in the amount of active hypotheses. Pruning is essential in the Viterbi-search as it keeps the number of active hypotheses small. Otherwise the search space would grow exponentially with the duration of an utterance. An exact evaluation of Eq. (2.18) is therefore only feasible for small vocabulary sizes and short sentences.

The most important pruning strategy used in the Viterbi-search is the so-called *beam-search*. At each time frame, beam-search expands only those hypotheses for which the corresponding likelihood does not fall below a certain threshold defined by the currently best scored partial sentence hypothesis [Lowerre 76, Ney & Mergel$^+$ 87, Ortmanns & Ney 95]. Beam-search is not guaranteed to find the optimal solution because the best path might already be pruned at an early stage in the search process. In practice, however, search errors are often negligible if the pruning parameters are adjusted properly.

2.5 Acoustic Model Training

Let $r = 1, \ldots, R$ enumerate the utterances of a training corpus $(\mathcal{X}, \mathcal{W}) = \{(X_r, W_r)_{r=1,\ldots,R}\}$, each consisting of a sequence of acoustic observation vectors $X_r = x_{r1}, \ldots, x_{rT_r}$ together with the corresponding spoken word sequence $W_r = w_{r1}, \ldots, w_{rN_r}$. A standard training based on the *Maximum Likelihood* (ML) criterion aims at finding a parameter set θ which maximizes the probability of the acoustic training data given the spoken word sequence:

$$\mathcal{F}_{\text{ML}}^*(\theta) = \prod_{r=1}^{R} p_\theta(X_r|W_r) \stackrel{!}{=} \max. \tag{2.19}$$

Usually, the objective function Eq. (2.19) is denoted in a logarithmic form:

$$\mathcal{F}_{\text{ML}}(\theta) = \frac{1}{R} \sum_{r=1}^{R} \log p_\theta(X_r|W_r). \tag{2.20}$$

If the acoustic model $p_\theta(X_r|W_r)$ is based on HMMs with Gaussian mixture distributions, the objective function $\mathcal{F}_{\text{ML}}(\theta)$ can be maximized using the *Expectation Maximization* (EM) algorithm [Dempster & Laird$^+$ 77]. The EM algorithm is a consistent and easy to handle iterative training

procedure that allows for finding ML estimates of parameters in probabilistic models which depend on unobservable (hidden) variables. For Gaussian HMMs, the EM algorithm leads to update equations that are equivalent to the re-estimation formulae first derived in [Baum & Petrie$^+$ 70].

2.5.1 Maximum Likelihood Training of Acoustic Models

The derivative of $\mathcal{F}_{\mathrm{ML}}$ with respect to the parameters of the acoustic model yields[2]:

$$\frac{\partial \mathcal{F}_{\mathrm{ML}}(\theta)}{\partial \theta_{sl}} = \frac{1}{R} \sum_{r=1}^{R} \sum_{t=1}^{T} p_\theta(s_t{=}s|X_r, W_r) \cdot \frac{\partial \log p(x_{rt}|\theta_s)}{\partial \theta_{sl}}. \tag{2.21}$$

Here, $p_\theta(s_t{=}s|X_r, W_r)$ denotes the *word-conditioned state posterior probability* that, given a sequence of acoustic observations X_r, the time alignment paths for the word sequence W_r lead through state s at time frame t. Word-conditioned state posterior probabilities are also referred to as *conditional forward-backward* (FB) probabilities. They can be interpreted as weights reflecting the certainty by which the acoustic observation vector x_{rt} is emitted by state s. To simplify the notation, word-conditioned state posterior probabilities are denoted by:

$$\gamma_{rt}(s|W) \coloneqq p_\theta(s_t{=}s|X, W). \tag{2.22}$$

Maximum Approximation for Mixture Densities

Due to the exponential decay of Gaussian probability density functions, the likelihood of a mixture model is typically dominated by a single density (cf. Figure 2.4). Hence, the approximation error will be small if the sum over all densities is replaced with its maximum. Let $l_\theta(x, s)$ denote the index of the mixture component that dominates the sum over all densities given an observation x:

$$l_\theta(x, s) \coloneqq \operatorname*{argmax}_{l \in \{1,\ldots,L_s\}} \left\{ c_{sl} \cdot p(x|\theta_{sl}) \right\}. \tag{2.23}$$

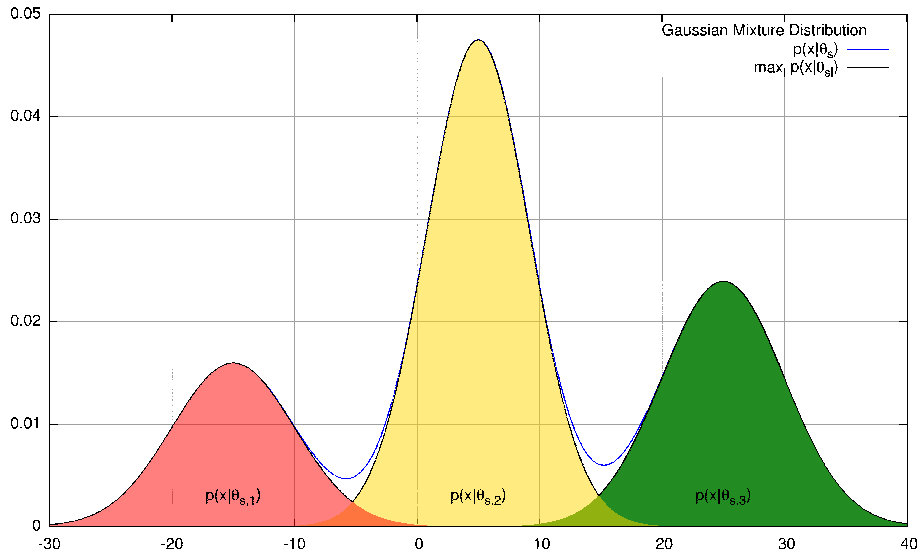


Figure 2.4: Maximum approximation for Gaussian mixture densities.

[2] The detailed calculations are shown in Section B.1.1.

Then, the derivative of the logarithm of the emission probabilities with respect to a density specific parameter θ_{sl} yields under the maximum approximation:

$$\frac{\partial \log p(x|\theta_s)}{\partial \theta_{sl}} = \frac{c_{sl} \cdot p(x|\theta_{sl})}{\sum_{\tilde{l}} c_{s\tilde{l}} \cdot p(x|\theta_{s\tilde{l}})} \cdot \frac{\partial \log p(x|\theta_{sl})}{\partial \theta_{sl}} \tag{2.24}$$

$$\overset{\text{max. approx.}}{\cong} \delta_{l,l_\theta(x,s)} \cdot \frac{\partial \log \left[c_{sl} \cdot p(x|\theta_{sl})\right]}{\partial \theta_{sl}}. \tag{2.25}$$

If not stated otherwise, the maximum approximation for mixture densities is used in all experiments reported in this thesis.

2.5.2 Conditional Forward-Backward Probabilities

The definition of the state posterior probabilities is extended by integrating the relative mixture weights:

$$\gamma_{rt}(s, l|W) \coloneqq \gamma_{rt}(s|W) \cdot \frac{c_{sl} \cdot p(x_{rt}|\theta_{sl})}{\sum_{\tilde{l}} c_{s\tilde{l}} \cdot p(x_{rt}|\theta_{s\tilde{l}})}. \tag{2.26}$$

Thus, Eq. (2.21) can be written as:

$$\frac{\partial \mathcal{F}_{\mathrm{ML}}(\theta)}{\partial \theta_{sl}} = \frac{1}{R} \sum_{r=1}^{R} \sum_{t=1}^{T_r} \gamma_{rt}(s, l|W_r) \cdot \frac{\partial}{\partial \theta_{sl}} \log p(x_{rt}|\theta_{sl}). \tag{2.27}$$

Similarly, the derivative of $\mathcal{F}_{\mathrm{ML}}(\theta)$ with respect to a state transition probability yields:

$$\frac{\partial \mathcal{F}_{\mathrm{ML}}(\theta)}{\partial p(s|\sigma)} = \frac{1}{p(s|\sigma)} \cdot \frac{1}{R} \sum_{r=1}^{R} \sum_{t=1}^{T_r} \gamma_{rt}(\sigma, s|W_r). \tag{2.28}$$

The quantity $\gamma_{rt}(\sigma, s|W) \coloneqq p_\theta(s_{t-1}{=}\sigma, s_t{=}s|X_r, W)$ denotes the probability that for a sequence of acoustic observations X_r a time alignment path for the word sequence W leads through state σ at time frame $t-1$ and through state s at time frame t. The conditional FB probabilities $\gamma_{rt}(s|W)$ can be computed based on $\gamma_{rt}(\sigma, s|W)$ by carrying out the sum over all predecessor states σ:

$$\gamma_{rt}(s|W) = \sum_{\sigma} \gamma_{rt}(\sigma, s|W). \tag{2.29}$$

Under a *Bakis* model, the predecessor states σ are constrained to states which connect state s via a *loop*, a *forward*, or a *skip* transition. An efficient computation scheme for estimating the FB probabilities is based on the decomposition of $\gamma_{rt}(\sigma, s|W)$ into the *forward* and the *backward* paths:

$$\gamma_{rt}(\sigma, s|W) = \frac{p_\theta(X_r, s_{t-1} = \sigma, s_t = s|W)}{p_\theta(X_r|W)} \tag{2.30}$$

$$= \underbrace{\left[\sum_{[s_1^{t-2}, s_{t-1}=\sigma]|W} p_\theta(x_{r,1}^{\;\;t-1}, s_1^{t-1})\right]}_{\substack{a_{r,t-1}(\sigma;W) \\ (\textit{forward}\text{ probability})}} \cdot \frac{p(s|\sigma)}{p_\theta(X_r|W)} \cdot \underbrace{\left[\sum_{[s_t=s, s_{t+1}^{T_r}]|W} p_\theta(x_{rt}^{\;\;T_r}, s_t^{T_r})\right]}_{\substack{b_{rt}(s;W) \\ (\textit{backward}\text{ probability})}}$$

The *forward* probability, $a_{r,t-1}(\sigma; W)$, is defined as the sum over the joint probabilities $p_\theta(x_{r,1}^{t-1}, s_1^{t-1})$ of all partial paths that enter state σ at time frame $t-1$. Analogously, the *backward* probability, $b_{rt}(s; W)$, is defined as the sum over the joint probabilities $p_\theta(x_{rt}^{T_r}, s_t^{T_r})$ of all partial paths that follow state s at time frame t (cf. Figure 2.5). Since a direct evaluation of Eq. (2.30) is inefficient because the number of required multiplications and summations would raise exponentially with the duration of the utterance, both quantities are computed by means of dynamic programming. The dynamic programming uses the recursive form of either quantity which, in case of the forward probability, is given by:

$$a_{rt}(s; W) = p_\theta(x_{rt}|s, W) \cdot \left(\sum_\sigma a_{r,t-1}(\sigma; W) \cdot p(s|\sigma) \right). \tag{2.31}$$

Here, the sum runs over all states σ that precede state s. Correspondingly, the recursion formula for the *backward* probability is given by:

$$b_{rt}(s; W) = \left(\sum_\sigma b_{r,t+1}(\sigma; W) \cdot p_\theta(x_{r,t+1}|\sigma) \cdot p(\sigma|s) \right), \tag{2.32}$$

with the sum running over all state hypotheses σ that succeed state s. The forward probabilities can be determined via a time-synchronous computation of the elements of the forward matrix $[a_{rt}(s; W)]$. Similarly, the backward matrix $[b_{rt}(s; W)]$ is obtained by computing its elements in descending order over the time axis. With the following identities for the normalization factor:

$$p_\theta(X_r|W) \equiv a_{r,T_r}(s_{T_r}; W) \equiv b_{r,0}(s_0; W) \tag{2.33}$$

and the starting values $a_{r,0}(s_0; W) = b_{r,T_r}(s_{T_r}; W) = 1$ (where s_0 denotes the initial state of the word automaton for the the word sequence W, and s_{T_r} denotes its final state), the FB probabilities suffice the following identities:

$$\gamma_{rt}(s|W) = a_{rt}(s; W) \cdot b_{rt}(s; W) \cdot \frac{1}{p_\theta(X_r|W)}, \tag{2.34}$$

$$\gamma_{rt}(\sigma, s|W) = a_{r,t-1}(\sigma; W) \cdot p(x_{rt}|\theta_s) \cdot p(s|\sigma) \cdot b_{rt}(s; W) \cdot \frac{1}{p_\theta(X_r|W)}. \tag{2.35}$$

Numerical Aspects

In practice, both the forward and the backward pass require the summation over likelihoods that are represented in a log-domain. If the likelihood terms have different orders of magnitudes, the summation requires a numerically stable implementation which can be based on the *Kingsbury-Rayner* formula [Kingsbury & Rayner 71]. This formula suggests the following computation scheme for the sum over two probabilities p_1 and p_2:

$$-\log(p_1 + p_2) = -\log\max\{p_1, p_2\} \; - \; \log\left(1 + e^{\log\min\{p_1,p_2\} - \log\max\{p_1,p_2\}}\right). \tag{2.36}$$

Viterbi Approximation

Under the *Viterbi* approximation, only the most probable time alignment path is taken into account to estimate $\gamma_{rt}(s|W)$ (cf. Figure 2.5). Due to the normalization constraint for probability distributions, the FB probabilities of state hypotheses along the Viterbi path are equal to 1 while state hypotheses off the Viterbi path are assumed to have a probability of 0:

$$\gamma_{rt}(s|W) \overset{\text{Viterbi}}{\cong} \delta_{s,s_t(X_r,W)}. \tag{2.37}$$

Here, the symbol δ denotes the *Kronecker* function[3], and $s_t(X_r, W)$ is the index of the state hypothesis at time frame t for the *best* time alignment path given the word sequence W and the sequence of acoustic observation vectors X_r.

[3] The *Kronecker* function $\delta_{i,j} = 1$ if $i = j$ and 0 otherwise.

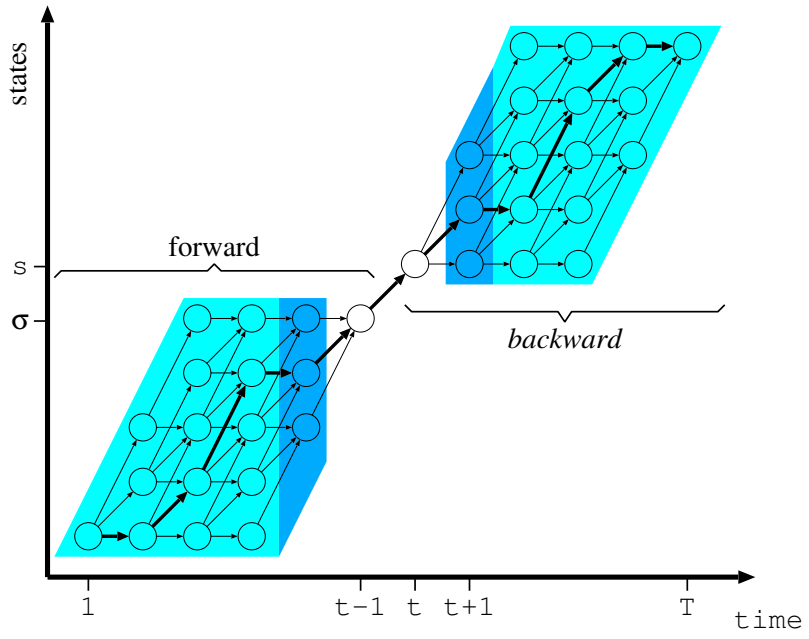


Figure 2.5: Decomposition of $\gamma_{rt}(\sigma, s|W)$ into the forward and the backward probabilities. The highlighted (bold printed) arrows indicate the best alignment path according to the *Viterbi* approximation.

2.5.3 Baum-Welch Training

The ML objective function $\mathcal{F}_{\mathrm{ML}}(\theta)$ cannot be optimized directly via the EM algorithm. Instead, the EM algorithm maximizes an auxiliary function $\mathcal{Q}$ built from the conditional a-posteriori probabilities $p_\theta(s_1^T|X_r, W_r)$:

$$\mathcal{Q}(\bar{\theta}|\theta) = \frac{1}{R}\sum_{r=1}^{R}\sum_{[s_1^{T_r}]|W_r} p_\theta(s_1^{T_r}|X_r)\cdot\log p_{\bar{\theta}}(X_r, s_1^{T_r}) \tag{2.38}$$

$$= \frac{1}{R}\sum_{s}\sum_{r=1}^{R}\sum_{t=1}^{T_r}\Big[\gamma_{rt}(s|W_r)\cdot\log p(x_{rt}|\bar{\theta}_{s_t}) + \sum_{\sigma}\gamma_{rt}(\sigma, s|W_r)\cdot\log p(s|\sigma)\Big] \tag{2.39}$$

The auxiliary function $\mathcal{Q}$ is maximized with respect to the new parameter set $\bar{\theta}$ where the estimation of $p_\theta(s_1^{T_r}|X_r, W_r)$ is based on the parameter set θ obtained from the previous iteration. It can be shown that $Q(\bar{\theta}|\theta)$ is a growth transformation for $\mathcal{F}_{\mathrm{ML}}(\bar{\theta})$, that is, any parameter set $\bar{\theta}$ increasing the auxiliary function also implies an increase in the objective function [Baum & Sell 68]:

$$\mathcal{Q}(\bar{\theta}|\theta) \geqslant \mathcal{Q}(\theta|\theta) \;\Rightarrow\; \mathcal{F}_{\mathrm{ML}}(\bar{\theta}) \geqslant \mathcal{F}_{\mathrm{ML}}(\theta). \tag{2.40}$$

The derivative of the auxiliary function with respect to the new parameter set $\bar{\theta}$ results in a closed form expression for the ML update equations which depend only on the former parameter set θ. The update equations can also be derived from a special case of the *extended unifying approach* for a class of discriminative training criteria presented in Chapter 4.

2.6 The Discriminative Approach

In [Nádas 83], it is shown that the ML estimator leads to an optimal parameter set if the following three assumptions are satisfied:

- The training samples arise from the assumed distribution family.
- The distribution family is well-behaved.
- The number of training samples is sufficiently large.

Provided that these assumptions hold, there would be no reason to estimate the model parameters with a training criterion other than the ML estimator. In practice, however, these requirements are usually not met, and one has to deal with the problem that the ML criterion aims at optimizing the parameters of an incorrect model using a suboptimal estimator. This raises the question whether there are alternative training methods that will lead to improved classification error rates if the parameters are to be estimated for a given but *wrong* family of model distributions. This is in fact the case, and an instance of such an estimator is given by the class of *discriminative training criteria*.

In contrast to ML based training methods, which aim at optimizing each class region individually, discriminative criteria also take class extraneous data into account and thus focus on enhancing class separability. Therefore, discriminative methods often result in lower error rates, in particular when the *true* distribution does not belong to the family of *model* distributions. However, a substantial gain in terms of recognition performance can only be expected if there is an alternative parameter set which differs from the ML estimate, yet is able to compensate for a mismatch between the model distribution and the stochastic process that generated the training samples. Provided that this mismatch is not incidental but representative and generalizable to unseen observations, the discriminatively trained parameter set will outperform the ML estimate. This effect is illustrated in the following example for a simple two-class recognition problem.

2.6.1 Discriminative Training for a Simple Two-Class Recognition Problem

Suppose we are given a two-class recognition problem in which the *true* distribution of either class is generated from a Gaussian *probability density function* (pdf) with a class-independent, *full* covariance matrix $\mathbf{\Sigma}$. From each class, a total number of 1,000 training samples is randomly (i.i.d.) drawn using the true distribution as the underlying generating stochastic process. The prior probability for each class shall be 0.5, resulting in the following *true* distributions for the class-posterior probabilities:

$$p(k|x) = \frac{p(k) \cdot p(x|k)}{\sum_{\tilde{k}} p(\tilde{k}) \cdot p(x|\tilde{k})}, \qquad k \in \{1, 2\} \tag{2.41}$$

with $p(k) = 0.5$ for $k \in \{1, 2\}$ and $p(x|k) = \mathcal{N}\big(x|\mu_k, \mathbf{\Sigma}\big)$ where

$$\mu_1 = (-0.5,\, 0.5)^\top, \quad \mu_2 = (0.1,\, 0.1)^\top, \quad \text{and} \quad \mathbf{\Sigma} = \begin{pmatrix} 0.9 & 0.8 \\ 0.8 & 0.9 \end{pmatrix}. \tag{2.42}$$

In the first scenario (Figure 2.6a), the model distribution is chosen from the same family of distributions to which the true probability distribution belongs, that is, the model distribution is a normal distribution with a *class-independent, full* covariance matrix. Because the *true* distribution can be expressed by the model distribution for a certain choice of the model parameters, the ML

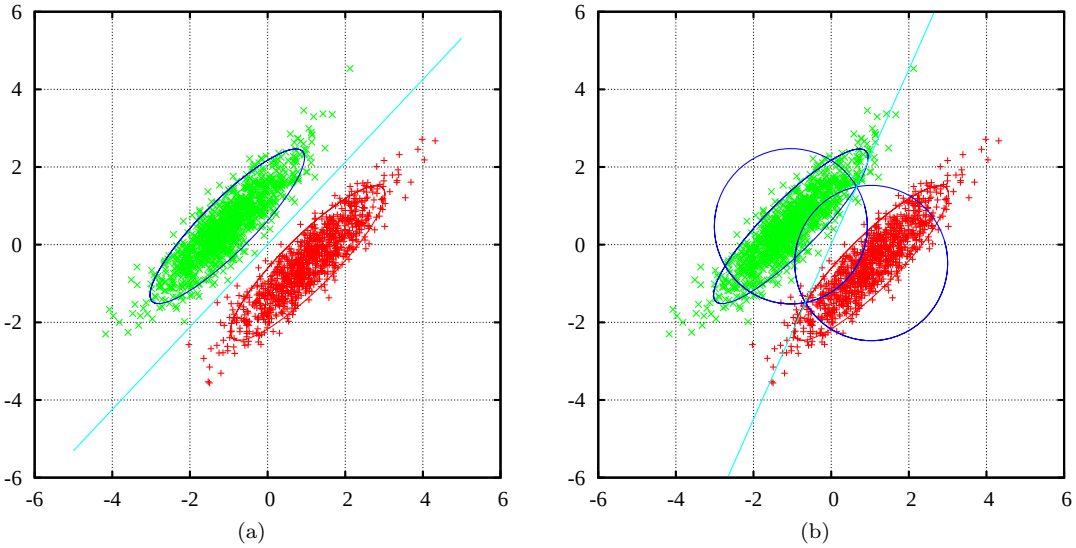


Figure 2.6: Maximum Likelihood estimates for a two-class problem with a class-independent *full* covariance matrix (a) and a class independent *diagonal* covariance matrix (b). Both classes have equal prior probability. The number of training samples is 1000 for each class.

estimator is able to find the optimal solution, and the resulting model leads to a perfect class separation of the training samples[4].

In the second scenario (Figure 2.6b), the family of model distributions is again assumed to be a normal distribution with a tied covariance matrix. This time, however, $\mathbf{\Sigma}$ is subject to a diagonal modeling constraint. This causes a mismatch between the model distribution and the true distribution, and hence the ML estimator finds only a suboptimal parameter set which causes some classification errors on the training data. In our example, 20 samples are assigned to the wrong class which corresponds to a 2.0% classification error rate.

A better solution is provided by a discriminative learning method known as the *Maximum Mutual Information* (MMI) criterion. The MMI criterion is defined as the sum over the logarithms of the class posterior probabilities of all training samples r:

$$\mathcal{F}_{\mathrm{MMI}}(\theta) = \frac{1}{R}\sum_{r=1}^{R}\log p_\theta(k_r|x_r) \tag{2.43}$$

$$= \frac{1}{R}\sum_{r=1}^{R}\log\frac{p(k_r)\cdot p_\theta(x_r|k_r)}{\sum_c p(c)\cdot p_\theta(x_r|c)}. \tag{2.44}$$

From Eq. (2.44) it becomes apparent that an optimization of the MMI objective function aims at maximizing the class conditional probabilities $p_\theta(x_r|k_r)$ over all training samples r ($r = 1,\ldots,R$) while simultaneously the sum over the class conditional probabilities of all competing classes $p_\theta(x_r|c)$ weighted with their respective prior probabilities $p(c)$ is minimized.

As shown in Figure 2.7a, the MMI criterion finds an alternative parameter set for which the decision boundary again provides an optimal separation of the training data according to their

[4] In order to keep this example simple, we do not introduce an additional held-out set but evaluate the performance of the estimated model parameters on the training data.

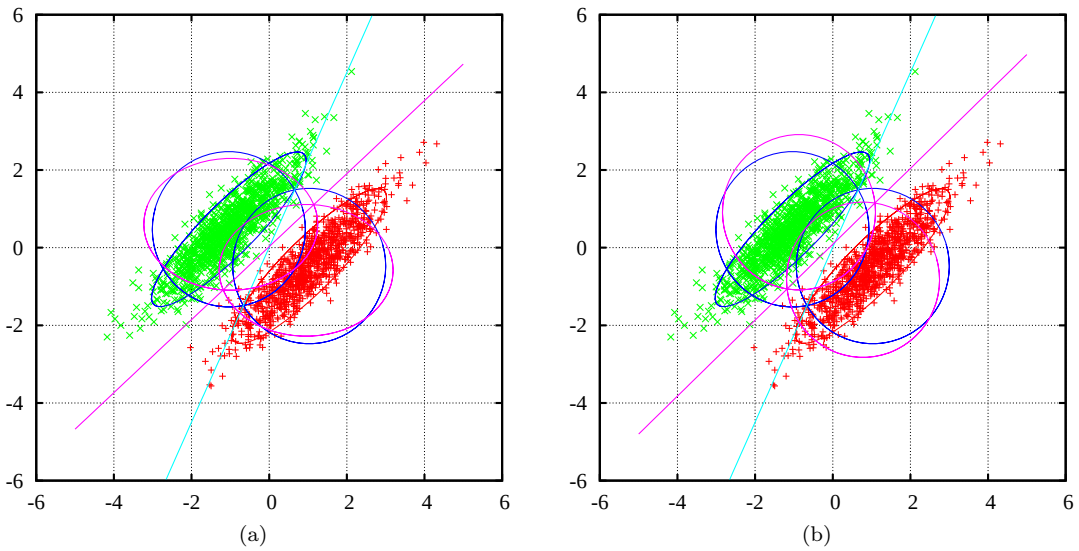


Figure 2.7: Maximum Mutual Information estimates for a two-class problem with a class-independent *diagonal* covariance matrix (a). In (b) the ML estimation of the diagonal covariance matrix is used. Both classes have equal prior probability. The total number of training samples is 1000 for each class.

class labels. Compared to the ML estimate, the major difference in the MMI estimate turns out to be the covariance matrix whose diagonal modeling constraint was the reason for the model mismatch.

An alternative solution can be found if the update rules for the MMI criterion are constrained to re-estimations of the mean vectors while the diagonal covariance matrix is kept fixed to its ML estimate. As shown in Figure 2.7b, the MMI estimate now moves the class centroids further away from their empirical means, and again the resulting decision boundary provides an optimal separation of the training data[5].

Besides the MMI criterion, several other discriminative training criteria have been proposed for automatic speech recognition. An overview of the state-of-the-art is given in Chapter 3.

2.7 Summary

This chapter gave a brief introduction to automatic speech recognition and described the models used to capture the stochastic relation between a speech signal and the spoken word sequence. Starting from Bayes' decision rule, the objective of finding the word sequence which maximizes the a-posteriori probability induced the basic architecture of an automatic speech recognizer. The training of the acoustic model under the Maximum Likelihood criterion was described in detail. The discriminative approach was motivated by means of a simple two class recognition problem which illustrated how the Maximum Mutual Information criterion will improve the decision boundary compared to a Maximum Likelihood estimator if the true distribution cannot be expressed by the model distribution.

[5]Experiments on fixing the variance parameters in a large scale discriminative training for automatic speech recognition are described in Section 6.3.7.

Chapter 3

State-of-the-Art and Scientific Goals

This chapter presents an overview of important findings and results on various aspects of discriminative training for automatic speech recognition. Based on the state-of-the-art as outlined in Sections 3.1–3.5, the scientific goals for this thesis are defined in Section 3.6.

3.1 Discriminative Training Criteria

Besides the *Maximum Mutual Information* (MMI) criterion, several other discriminative training criteria were proposed and investigated for automatic speech recognition. This section presents a selection of commonly used discriminative training criteria.

3.1.1 Maximum Mutual Information Criterion

The first comparison between a ML training and the MMI criterion is reported in [Bahl & Brown$^+$ 86] for a speaker-dependent isolated word recognition system with a dictionary comprising 2,000 words. Compared to the ML trained system, the discriminative approach improved the error rate by 18% relative. In [Merialdo 88], the MMI criterion was successfully applied to optimize a speaker-dependent recognition system for continuously spoken phonemes. The objective function was maximized using a gradient based method. Particularly the approximation of the gradient used for optimizing the mixture density weights was of special importance for a long time as it leads to faster convergence rates than the update rules derived from the *Extended Baum* (EB) algorithm [Gopalakrishnan & Kanevsky$^+$ 91]. In [Chow 90], the MMI criterion was used to train the model parameters of the HMM based *BYBLOS* system on the *DARPA Resource Management* speech corpus (991 words). Competing hypotheses were encoded as 10-best lists. Similar to [Merialdo 88], parameters were optimized with a gradient based method. In addition, *code book exponents* were trained in order to weigh the dimensions of the feature space. The relative improvement of 11% in terms of word error rate was, however, accompanied by a degradation of 3.6% relative in terms of sentence error rate. In [Normandin & Morgera 91], the EB algorithm for the optimization of discrete probability distributions was applied to continuous density functions. Experiments conducted on the *TI-digits* corpus for the recognition of continuously spoken connected digit strings showed a reduction of the sentence error rate by almost 50% relative. Investigations on the same corpus using the MMI criterion for the training of different word models are reported in [Cardin & Normandin$^+$ 93]. Here, a relative improvement between 5% and 10% could be achieved.

The first use of a lattice-based discriminative training for *large vocabulary continuous speech recognition* (LVCSR) is reported in [Valtchev & Odell$^+$ 96]. Other than most previous studies, the competing model was estimated on word lattices rather than N-best lists. To reduce the running time in the second and subsequent training iterations, the search space was constrained to those sentence hypotheses that were encoded in the initial word lattices. Experiments conducted on the *Wall Street Journal* WSJ0 and WSJ1+ corpora showed a word error rate improvement between 5% and 10% relative. In [Schlüter & Müller$^+$ 99], it was experimentally shown that lattice-based

discriminative training for large vocabulary tasks is more effective if low-order language models (unigram or zerogram LMs) are used in training. Higher-order language models as, for example, a trigram language model, deteriorated the quality of the acoustic model and could even result in a performance drop compared to the ML trained baseline system. The first successful application of the MMI criterion to a large scale speech recognition system is reported in [Woodland & Povey 00]. Besides an investigation of various techniques that aim at improving the generalization of MMI trained acoustic models, the authors report several experimental results on different subsets of the NIST March 2000 HUB-5 corpus, demonstrating that the MMI criterion is able to outperform a ML trained system even for large amounts of training data of up to 265 hours of speech. Speaker adaptation based on the *Maximum Likelihood Linear Regression* (MLLR) method could be shown to achieve similar absolute improvements on top of the MMI trained system as on top of the ML trained system.

3.1.2 Frame Discrimination Criterion

A word-lattice based implementation of the MMI criterion requires an initial search pass in which lattices are produced that serve as input for collecting sufficient statistics for all subsequent training iterations. Since the process of generating lattices is computationally expensive and time consuming, the *Frame Discrimination* (FD) criterion [Kapadia 98, Povey & Woodland 99] was proposed as an alternative to a word-lattice based MMI training. The principle of the FD approach is to increase the confusion (entropy) among Gaussian mixture models by discriminating the state hypotheses for the correct transcription against the set of *all* Gaussian densities in the HMM. Compared to lattice-based MMI training, the competing model in the FD approach is less constrained which may result in improved generalization. The FD criterion could be shown to give similar results as word lattice-based MMI training on medium and large vocabulary tasks [Povey & Woodland 00]. However, due to improved optimization procedures for lattice-based MMI training in combination with increased computational power, investigations on the frame discrimination approach were discontinued.

3.1.3 Conditional Maximum Likelihood Criterion

In [Valtchev 95], it was shown that the *Conditional Maximum Likelihood* (CML) criterion as proposed in [Nádas & Nahamoo$^+$ 88] is equivalent to the MMI criterion if the language model is kept fixed. In fact, most studies on the MMI criterion for automatic speech recognition consider the language model as a knowledge source that does not change in the course of the training process. Hence in practice, the equivalence between the CML and the MMI criterion holds for the majority of all systems described in literature. Investigations on the CML criterion are reported in [Gunawardana 01, Gunawardana & Byrne 01].

3.1.4 Minimum Classification Error Criterion

In [Juang & Katagiri 92], an alternative class of discriminative training criteria represented by the *Minimum Classification Error* (MCE) principle was introduced. The MCE criterion aims at minimizing a task specific error criterion on training data. Since in general the error rate is a discontinuous function of the parameters of a classifier, the error measure needs to be smoothed appropriately to allow for optimizing the objective function with a gradient-based method. The error metric optimized under the MCE criterion depends on the particular task. Error metrics proposed include the string (or sentence) error rate, the phoneme error rate, and the state error rate [McDermott & Katagiri 97].

The MCE criterion was successfully applied to the *TI-digits* recognition task [Chou & Juang$^+$ 92] and to the *DARPA Naval Resource Management* corpus as reported in [Chou & Lee$^+$ 93,

Chou & Lee$^+$ 94]. In [Biem & Katagiri 93, Paliwal & Bacchiani$^+$ 95], the MCE criterion was used for feature extraction. *Bauer* applied the MCE criterion to improve the accuracy of an isolated digit recognizer [Bauer 97, Bauer 98]. Compared to the best ML trained system, the MCE criterion was able to further decrease the word error rate by 26% relative. In [McDermott & Katagiri 97], the MCE criterion was used for the acoustic model training of a continuous phoneme recognition task. For the investigated setting, phoneme-level based MCE training turned out to be more effective than string-level based MCE training. In [Saul & Rahim 00], an MCE-based *factor analysis* framework for variance modeling in automatic speech recognition was proposed. Factor analysis denotes a technique that aims at mapping systematic variations of the data into lower dimensional subspaces which allows a classifier to compactly represent covariance matrices for high dimensional data. The covariance matrices are represented by a small number of parameters that model the most significant correlations. These parameters can be chosen such as to minimize the number of classification errors using the MCE criterion. Compared to a ML trained system, the MCE based factor analysis was able to further reduce the error rate on an isolated name recognition task (2,426 town names) from 12.7% to 11.7%.

The first application of the MCE criterion to a large scale training set of more than 100 hours of speech is reported in [McDermott & Hazen 04] for the JUPITER weather information system which uses a recognition vocabulary of almost 2,000 words. The MCE criterion achieved a 20% relative reduction in terms of word error rate compared to the ML-trained baseline. In [McDermott & Katagiri 05], weighted finite state transducers are used to represent correct and competing string candidates. Preliminary results are reported for large vocabulary tasks comprising 30,000 words.

3.1.5 Minimum Word Error and Minimum Phone Error Criterion

The *Minimum Word Error* (MWE) criterion and the *Minimum Phone Error* (MPE) criterion aim at minimizing the expected word or phoneme error rate on training data. Originally, the MWE criterion was proposed in [Na & Jeon$^+$ 95] as *Overall Risk* criterion. First applications to large vocabulary speech recognition tasks are reported in [Kaiser & Horvat$^+$ 00] and [Kaiser & Horvat$^+$ 02]. Since the calculation of the overall risk is based on the *Levenshtein* distance between the correct transcription and a competing hypothesis, the function minimization was performed on N-best lists. In [Povey & Woodland 02], a lattice-based implementation of the MWE and MPE criterion was proposed which uses a local approximation of the Levenshtein distance that can be calculated on word and phoneme lattices. Both the MWE and MPE criterion are reported to achieve substantial performance gains on top of the MMI criterion on the *Switchboard/Call Home* corpus [Povey 04].

3.1.6 Comparison of Discriminative Training Criteria

While most investigations on discriminative training criteria report performance gains compared to a ML-trained system, only few studies present joint comparisons in which various discriminative criteria are compared with each other. In [Reichl & Ruske 95], performance gains for the MMI and MCE criterion were measured on a speaker independent phoneme recognition task, showing that the MCE criterion is able to outperform the MMI criterion. In [Schlüter & Macherey 98], a unifying approach for a class of discriminative training criteria is proposed which includes the Gini, the MMI, and the MCE criterion. The unifying approach allows for optimizing several criteria within a single framework, thus making error rates obtained with different criteria directly comparable. A thorough investigation of the unifying approach for small vocabulary tasks can be found in [Macherey 98]. Results for large vocabulary tasks are reported in [Schlüter 00], [Schlüter & Macherey$^+$ 01], and [Macherey & Haferkamp$^+$ 05].

3.2 Parameter Optimization

In contrast to a ML-based training, optimization procedures for discriminative criteria require the choice of an explicit step size. Hence, discriminative training methods are less convenient to handle than training procedures based on, e.g., the EM algorithm. While early studies on discriminative training typically applied stochastic *gradient descent* (GD) methods, more recent investigations almost exclusively use the *Extended Baum* (EB) algorithm.

3.2.1 Extended Baum Algorithm and Gradient Descent

In [Gopalakrishnan & Kanevsky$^+$ 89] and [Gopalakrishnan & Kanevsky$^+$ 91], an extension of the *Baum-Welch* algorithm to the optimization of rational objective functions is proposed. This extension, called the *Extended Baum* (EB) algorithm, allows for optimizing the model parameters of discrete HMMs under the MMI criterion. In [Normandin 91], the EB algorithm for rational objective functions was extended and successfully applied to the parameter estimation problem of continuous *probability density functions* (PDFs) in a discriminative framework. Since for continuous PDFs, the order of magnitude of the iteration constants for which convergence is guaranteed is impractically large, *Normandin* proposed a heuristic to adjust the step size such that learning rates become viable. In [Kanevsky 95], the EB algorithm for rational objective functions was formally extended to real-valued functions which gives the application of the EB algorithm to continuous HMMs a mathematically sound justification. In [Schlüter & Macherey$^+$ 97], it was shown that for a proper choice of the step size, the GD-based optimization of Gaussian mixture models can lead to update equations that are almost identical with the re-estimation equations derived in the EB algorithm.

3.2.2 Convergence Issues

In [Gunawardana & Byrne 01], a simple and elegant proof for the EB algorithm is presented which is based on a general form of the *Jensen* inequality. Although the claim for the existence of a finite iteration constant does not hold, the paper provides new insights into the discriminative optimization procedure. Two mathematically correct proofs for the existence of finite iteration constants for the EB algorithm applied to real-valued objective functions are presented in [Kanevsky 03, Kanevsky 04, Kanevsky 05] and [Axelrod & Goel$^+$ 04]. Both proofs do not, however, provide tight lower bounds on setting the iteration constants. Hence, choosing the optimal step size in discriminative training still remains an unsolved problem.

3.2.3 Parameter Tying

The majority of discriminatively trained systems in automatic speech recognition either uses density specific variances or employs a single globally pooled variance. Compared to a globally pooled variance, density specific variances typically result in faster convergence rates on the training data because the choice of the step sizes is subject to fewer constraints and can thus be adjusted more effectively. However, a potential drawback is the possibility that faster convergence rates may reduce the ability of the trained model to generalize to unseen data. An investigation on different parameter tying schemes for discriminative training and their effect on the convergence speed is reported in [Macherey & Schlüter$^+$ 04].

3.2.4 Enhancing Generalization

In [Povey & Woodland 99], the FD approach was investigated for large vocabulary tasks and compared with the performance gains obtained by using the MMI criterion. Since under the FD

criterion, the competing model is less constraint than in lattice-based MMI training, FD may result in improved generalization. Experiments performed on the *DARPA Resource Management* (RM) speech corpus (991 words) and the *North American Business* (NAB) News task using a 65k word recognition vocabulary showed a relative reduction of 8.3% in terms of word error rate for the FD criterion over a ML-trained system on the RM corpus, and a 2% relative reduction for the FD criterion over the ML-trained baseline system on the NAB corpus. Compared to the MMI criterion, the FD criterion could be shown to give similar improvements on the NAB corpus.

In [Povey & Woodland 02], a technique called *I-smoothing* was proposed, which is reported to enhance the generalization of discriminatively trained parameters. I-smoothing is closely related to the H-criterion, which uses a fixed interpolation between the ML objective function (H=0) and the MMI objective function (H=1). In contrast to the H-criterion, I-smoothing increases the number of data points assigned to a Gaussian by a constant term while keeping the average data values and average squared data values the same. On the *Switchboard/Call Home* corpus, MMI in combination with I-smoothing improved the word error rate by 0.4% absolute over MMI alone, which achieved a word error rate of 41.8% on top of the ML-trained baseline which obtained a word error rate of 46.6%. I-smoothing in combination with the MPE criterion was able to further reduce the word error rate by 0.6% absolute.

3.2.5 Levels of Approximation for Discriminative Training

Most implementations for discriminative training use the Viterbi approximation to generate competing hypotheses for word-lattices or N-best lists. The Viterbi approximation is not, however, mandatory for collecting sufficient statistics in discriminative training. In [Macherey 98], a discriminative *Baum-Welch* training was developed and implemented which, compared to a conventional discriminative training under the Viterbi approximation, could achieve small but consistent gains of up to 4% relative in terms of word error rate on the SIETILL corpus for the recognition of continuously spoken connected German digit strings.

3.2.6 Asymptotic Behavior of Discriminative Training Criteria

The asymptotic behavior of model distributions in discriminative training has been investigated for only a small number of discriminative training criteria. In [Schlüter & Ney 00], it is shown that the MCE criterion provides an upper bound to the true Bayes error rate which is independent of the corresponding model distribution. Moreover, it is demonstrated that, in the limiting case of an infinite amount of training data, a model-free optimization of the MCE criterion leads to a closed form solution. In [Ney 03], it is shown that any model distribution optimized under the MMI criterion approximates the true Bayes error rate in the asymptotic case that an infinite amount of training data is used.

3.3 Discriminative Acoustic Modeling

While most studies on discriminative training aim at optimizing the parameters of the acoustic model, discriminative methods were also successfully applied to acoustic modeling. Examples are *discriminative model complexity control*, criteria for *discriminative mixture density splitting*, and *discriminative linear feature transformations*.

3.3.1 Discriminative Model Complexity Control

In [Liu & Gales 04], the optimal dimensionality for a *Heteroscedastic Linear Discriminant Analysis* (HLDA) is determined using a criterion based on marginalizing an MMI growth function. The same

approach is also used for model compression. Experiments conducted on a subset of the HUB-5 corpus showed a small reduction in terms of word error rate of 0.2% absolute compared to the ML trained baseline system which achieved a WER of 29.8%.

3.3.2 Approaches to Mixture Density Splitting

In [Normandin 95], a novel approach to Gaussian mixture component splitting based on the MMI criterion was proposed. The basic idea is to increase the acoustic resolution only in those distributions where a discrimination problem is identified. Although the best ML trained system could not be outperformed, the new splitting approach required significantly fewer mixture components than the baseline approach. In [Macherey 98, Schlüter & Macherey$^+$ 99], a combined MMI/ML approach for mixture density splitting was developed and evaluated on a small vocabulary task. Similar to [Normandin 95], a model evaluation measure based on the MMI criterion was used to identify mixture components that are to be split up. However, in contrast to [Normandin 95], the increased parameter set was trained until convergence using the ML criterion which prevented over-fitting and lead to improved generalization to unseen data. Experiments conducted on the SIETILL corpus for the recognition of telephone line recorded German digit strings showed that the combined splitting approach performed better than discriminative training with conventional splitting and as good as discriminative training after the new splitting approach.

3.3.3 Discriminative Linear Transformations

Discriminative methods are not limited to the estimation of Gaussian mixture models but can also be applied to train linear feature transformations. In [Macherey 98, Schlüter 00], a linear transformation called *Linear MMI Analysis* (LMA) is proposed as an extension of the *Linear Discriminant Analysis* (LDA). In contrast to the LDA transformation, the statistics for the *between class scatter* matrix under the LMA approach are collected based on the competing model of the MMI criterion. Compared to a standard LDA transformation, the LMA improved the word error rate between 8% and 14% relative on the SIETILL corpus for the recognition of telephone line recorded connected German digit strings. In [McDonough & Waibel 02], a linear transformation used for speaker adaptation is estimated in a discriminative framework. Both the transformation and the Gaussian mixture models are alternately trained. In [Tsakalidis & Doumpiotis$^+$ 02, Tsakalidis & Doumpiotis$^+$ 05], discriminative linear transforms are employed for feature normalization and speaker adaptive training. Compared to a ML-based training procedure, both the feature normalization and the speaker adaptive training improved the word error rate by approximately 0.8% absolute on the SWITCHBOARD-1 and SWITCHBOARD-2 data sets.

3.4 Direct Models for Automatic Speech Recognition

Traditionally, statistical models for automatic speech recognition are based on generative models such as HMMs. A framework for a new statistical model was presented in [Likhododev & Gao 02] and investigated on a phoneme recognition task. Other than traditional HMMs, the posterior probabilities of state sequences given the sequence of acoustic observations are computed directly using a *Maximum Entropy Markov Model* (MEMM) [McCallum & Freitag$^+$ 00]. In contrast to HMMs, observations are not conditioned on states but states are conditioned on observations or sequences of observations. Thus, the model allows for combining many different types of features. First experiments conducted on a phoneme recognition task improved the frame recognition accuracy from 62.4% (HMM) to 65.6% (MEMM). Word error rates obtained with the direct model by rescoring N-best hypotheses are reported in [Kuo & Gao 03]. For a special setting that excludes

the incorporation of a language model, the MEMM based system achieved a WER of 28.6% while the HMM based system obtained a WER of 32.7%.

3.5 Discriminative Model Combination

A framework for discriminative model combination was proposed in [Beyerlein 98]. The approach employs a log-linear combination of various acoustic models and language models. In contrast to a Maximum Entropy approach, the coefficients of the log-linear combination were optimized on training data using discriminative methods. Experiments conducted on the male portion of the WSJ-5k corpus reduced the word error rate of 3.5% for a baseline system built with across-word triphone models and a 4-gram language model by 0.3% absolute using a combination of within-word and across-word triphone models plus a bi-, a tri-, and a 4-gram language model.

3.6 Scientific Goals

Discriminative training criteria were shown to have the potential to outperform even the best ML trained systems for automatic speech recognition. However, despite a large number of studies on discriminative training, several questions and problems are still open and have not yet been addressed in literature. This section lists some of these questions and defines, based hereupon, the scientific goals for this thesis.

3.6.1 Extended Unifying Approach for a Class of Discriminative Training Criteria

Although some publications jointly compare a small number of discriminative training criteria for automatic speech recognition, there is no general framework which, in addition to the MMI and MCE criterion, also covers training criteria more recently proposed as, for example, the MWE and the MPE criterion. Hence, performance gains achieved with a particular discriminative training criterion are usually not comparable with other criteria since the underlying implementations are too different. In this thesis, a novel unifying approach is proposed that, in addition to the MMI and MCE criterion, also includes the MPE and the MWE criterion. The proposed framework is an extension of the unifying approach presented in [Schlüter & Macherey 98]. It is shown that a small set of parameterized re-estimation equations is sufficient to perform the function maximization for any criterion included in the extended unifying approach. Experimental results are presented for a large number of discriminative training criteria.

3.6.2 Word-Lattice Based Minimum Classification Error Training

While the MCE criterion could be shown to outperform MMI-trained systems on small vocabulary tasks, MCE-based training has not yet been successfully applied to large vocabulary speech recognition tasks. Moreover, all studies investigating the MCE criterion in automatic speech recognition use N-best lists to represent the set of competing hypotheses. One reason for this is that N-best lists provide an easy means to exclude the spoken word sequence from the set of hypotheses considered for discrimination. However, even for very large N, the number of sentence hypotheses encoded in an N-best list is outnumbered by several orders of magnitude by the number of candidates represented in a word lattice. Hence, word-lattice based MCE training can be expected to be more effective than discriminating against a list of the top N candidates.

In this thesis, the first successful lattice-based implementation of the MCE criterion is presented for small and large vocabulary speech recognition tasks. The optimization is performed within the extended unifying approach for a class of discriminative training criteria. Performance gains

obtained with the MCE criterion are compared with other criteria that include the MMI, the MWE, and the MPE criterion.

3.6.3 Efficient Construction of Markov Networks and Generation of High-Quality Word Lattices for Discriminative Training

In this thesis, an improved search algorithm based on word-conditioned Markov networks is developed. Markov networks generalize the concept of lexical prefix trees and can be considered as a hybrid approach between the compact but rigid structure of lexical prefix trees and the more flexible but often resource-intensive finite state transducers. Similar to transducers, Markov networks can be optimized on a state level. In addition, they provide an easy means to compile special acoustic models as, for instance, across word models, into the network without the need of changing the implementation of the underlying search algorithm. However, unlike finite state transducers, Markov networks are less memory consumptive and pertain the separation of the acoustic and the language model recombination. The latter becomes important if the language model it too large to fit into the memory of a single machine and thus needs to be distributed over several language model servers (cf. [Brants & Popat$^+$ 07]). Even for large vocabulary sizes of 65k words and more, Markov networks can be compiled very efficiently in terms of both running time and memory. In this thesis, results are reported for Markov networks comprising vocabulary sizes of more than 80k words.

3.6.4 Tying Schemes for Covariance Matrices in Discriminative Training

Discriminative objective functions are usually optimized within the *Extended Baum* (EB) algorithm. However, since for continuous distributions no method is known up to now that guarantees fast and stable convergence, the parameter estimation problem of Gaussian mixture models typically depends on setting the iteration constants according to a heuristic which ensures that the variance re-estimations are positive definite. In case of density specific variances, this leads to a system of quadratic inequalities. However, if tied variances are used, the inequalities become more complicated, and often the resulting constants turn out to be too large to be appropriate for discriminative training. In this thesis, two alternative approaches of setting the iteration constants are proposed and investigated. Both methods are independent of the variance tying scheme and result in faster convergence rates.

3.6.5 Discriminative Training with Full Covariance Matrices

If full covariance matrices are used, it is not obvious how the iteration constants should be set in the EB algorithm. In [Povey 04, p. 140], this problem was tackled by first setting all off-diagonals to zero and then by doubling the resulting iteration constants until the update equations for the full covariance matrices lead to solutions that are positive definite. In this thesis, an analytically closed form expression is derived that is based on the solution of *quadratic eigenvalue problems*. It is shown that under a diagonal modeling constraint, the commonly used heuristic to choose the iteration constants is included as a special case.

3.6.6 Efficient Training Procedures for Distributed Computing on Clusters with Inhomogeneous Architectures

Since discriminative training is a time intensive process, a client-server model for distributed computing on clusters with inhomogeneous architectures is developed. The client-server model is based on the TCP/IP protocol and allows for concurrently collecting sufficient statistics that are needed to re-estimate parameters in discriminative training.

3.6.7 Comparison of Maximum Entropy and Discriminative Training for Automatic Speech Recognition

In [Likhododev & Gao 02], a novel approach to automatic speech recognition was proposed that aims at modeling posterior probabilities of sequences of acoustic observations directly. Although this direct posterior model is conceptually a discriminative approach, error rates obtained with the new model have only been compared with a ML trained system. In this thesis, an alternative approach to a direct model is proposed that is bootstrapped with a conventional Gaussian HMM. The training procedure for the new approach employs the *Generalized Iterative Scaling* algorithm [Darroch & Ratcliff 72] and is compared analytically and experimentally with the *Extended Baum* algorithm.

3.6.8 Upper Bound on Complexity of Error Surfaces

This thesis presents a novel algorithm to efficiently compute and represent the exact and unsmoothed error surface over all sentence hypotheses that are encoded in a word lattice if all parameter settings of a log-linear model are considered that lie along an arbitrary line in the parameter space. While the number of sentence hypotheses encoded in a word lattice is exponential in the lattice size, the complexity of the error surface is shown to be always linearly bounded in the number of lattice arcs. This bound is independent of the underlying error metric.

3.6.9 Discriminative Linear Transformations

Linear transformations as, for instance, the *Linear Discriminant Analysis* (LDA), the *Maximum Likelihood Linear Transform* (MLLT), and the *Heteroscedastic Linear Discriminant Analysis* (HLDA) are important components in many state-of-the-art systems for automatic speech recognition. So far, these transformations have been trained under the ML criterion. In this thesis, a novel framework is presented that allows for estimating linear transformations in a discriminative manner. First experimental results are presented on a large vocabulary speech recognition task.

3.7 Summary

This chapter provided an overview of the state-of-the-art in discriminative training. Major achievements in discriminative training criteria, parameter optimization methods, discriminative acoustic modeling, and discriminative model combination were covered based on a selection of papers and articles that advanced the field in automatic speech recognition. Based on the state-of-the-art, the scientific goals for this thesis were defined.

Chapter 4

Discriminative Training

This chapter presents an *extended unifying approach* for a class of discriminative training criteria. Besides the MMI and the MCE criterion, the framework also captures objective functions more recently proposed, including the MWE and the MPE criterion. In Section 4.1, properties of some of the objective functions contained in the extended unifying approach are discussed. In Section 4.2, the re-estimation equations for optimizing the MMI objective function within the *Extended Baum* algorithm are derived and the heuristics for setting the iteration constants are generalized to the case that *full* covariance matrices with arbitrary tying schemes are used. Aspects of convergence control are discussed in Section 4.3. In Section 4.4, a new set of parameterized re-estimation equations is derived that generalizes the optimization procedure to all criteria contained in the extended unifying approach. The chapter concludes with some final remarks on growth transformations for discriminative objective functions in Section 4.5.

4.1 A General View of Discriminative Training Criteria

In [Schlüter & Macherey 98], a unifying approach for a class of discriminative training criteria was presented that allows for optimizing several objective functions – among them the *Maximum Mutual Information* criterion and the *Minimum Classification Error* criterion – within a single framework. In this section, the approach is extended such that it also captures other criteria more recently proposed. These criteria include the *Jeffreys*' criterion [Macherey & Haferkamp$^+$ 05], the *Chernoff Affinity* [Chernoff 52], the *Diversity Index* [Macherey & Haferkamp$^+$ 05], the *Minimum Word Error* criterion, and the *Minimum Phone Error* criterion [Povey & Woodland 02].

Let $(\mathcal{X}, \mathcal{W}) \coloneqq (X_r, W_r)_{r=1,\ldots,R}$ enumerate the training utterances of a speech corpus. Each training utterance shall consist of a sequence $X_r = x_{r1}, \ldots, x_{rT_r}$ of acoustic observation vectors together with the corresponding spoken word sequence $W_r = w_{r1}, \ldots, w_{rN_r}$. Furthermore, let $p_\theta(W|X_r)$ denote the class posterior probability of the word sequence W given the sequence of acoustic observations X_r, and $p_\theta(X_r|W)$ shall denote the emission probability of observing X_r given the word sequence W. In the following, the joint probability $p_\theta(X_r, W)$ is also referred to as *sentence probability* of the word sequence W. If not stated otherwise, the language model probabilities, $p(W)$, are supposed to be constant and therefore not subject to the parameter optimization problem. Finally, let $\mathcal{M}_r$ denote a set of word sequences which are considered for discrimination in utterance r. A class of discriminative training criteria can then be defined by:

$$\mathcal{F}\big(\mathcal{X}, \mathcal{W} \,|\, \theta; f, \kappa, \mathsf{G}, \{\mathcal{M}_r\}\big) = \frac{1}{R} \sum_{r=1}^{R} f\left(\log \left[\frac{\sum_{W} p^\kappa(W) \cdot p_\theta^\kappa(X_r|W) \cdot \mathsf{G}(W, W_r)}{\sum_{W \in \mathcal{M}_r} p^\kappa(W) \cdot p_\theta^\kappa(X_r|W)} \right]^{1/\kappa} \right) \tag{4.1}$$

Here, f denotes a smoothing function, $\kappa \in \mathbb{R}_{>0}$ is a weighting exponent used to re-weight sentence probabilities, and $\mathsf{G} : \mathcal{W} \times \mathcal{W} \to \mathbb{R}^+$ is a gain function that allows for rating sentence hypotheses W based on the spoken word string W_r. Typically, G reflects an error metric such as the *Levenshtein*

distance or the sentence error. The fraction inside the brackets is called the *local discriminative criterion* $\mathcal{F}_r(X_r, W_r \,|\, \theta, \kappa, \mathsf{G}, \mathcal{M})$:

$$\mathcal{F}_r\big(X_r, W_r \,|\, \theta; \kappa, \mathsf{G}, \mathcal{M}_r\big) = \frac{\sum\limits_{W} p^{\kappa}(W) \cdot p_\theta^{\kappa}(X_r|W) \cdot \mathsf{G}(W, W_r)}{\sum\limits_{W \in \mathcal{M}_r} p^{\kappa}(W) \cdot p_\theta^{\kappa}(X_r|W)}. \tag{4.2}$$

4.1.1 Extended Unifying Approach

The choice of alternative word sequences contained in the set $\mathcal{M}_r$ together with the smoothing function f, the weighting exponent κ, and the gain function G determine the particular criterion in use. Criteria contained in the extended unifying approach include the *Maximum Mutual Information* (MMI) criterion; its maximum approximation the *Corrective Training* (CT) criterion; the *Minimum Classification Error* (MCE) criterion together with its maximum approximation the *Falsifying Training* (FT); the *Diversity Index*; the *Jeffreys'* criterion; the *Chernoff Affinity*; the *Minimum Word Error* (MWE) criterion; and the *Minimum Phone Error* (MPE) criterion. In particular, the *Maximum Likelihood* (ML) criterion is contained for the special case that no competing utterances are considered[1]. Table 4.1 lists some of the criteria included in the extended unifying approach together with the choice of the criterion specific parameters f, κ, and G. Note that criteria which apply $\delta(W, W_r)$ as gain function consider only the spoken word sequence W_r in the numerator.

Maximum Likelihood (ML) Criterion

Although not a discriminative criterion in a strict sense, the ML objective function is contained as a special case in the extended unifying approach. The ML estimator is consistent which means that for any increasing and representative set of training samples the estimation of the parameters converges toward the true model parameters. However, for automatic speech recognition, the model assumptions are typically not correct, and therefore the ML estimator will return the *true* parameters of a *wrong* model in the limiting case of an infinite amount of training data. In contrast to discriminative training criteria, which concentrate on enhancing class separability by taking class extraneous data into account, the ML estimator optimizes each class region individually.

Maximum Mutual Information (MMI) and Corrective Training (CT) Criterion

Choosing $\kappa = 1$, $f(z) = z$, and $\mathsf{G} = \delta(W, W_r)$ yields the MMI criterion which is defined as the sum over the logarithms of the class posterior probabilities of the spoken word sequences W_r for each training utterance r given the corresponding acoustic observations X_r:

$$\mathcal{F}_{\mathrm{MMI}}(\theta) = \frac{1}{R} \sum_{r=1}^{R} \log p_\theta(W_r|X_r) \tag{4.3}$$

$$= \frac{1}{R} \sum_{r=1}^{R} \log \frac{p(W_r) \cdot p_\theta(X_r|W_r)}{\sum\limits_{W \in \mathcal{M}_r} p(W) \cdot p_\theta(X_r|W)}. \tag{4.4}$$

Ideally, the set $\mathcal{M}_r$ contains all possible word sequences. In practice, $\mathcal{M}_r$ is obtained through a recognition pass in which the set of alternative sentence hypotheses is limited to the candidates encoded in an N-best list or a word graph. Because sentence probabilities are not smoothed, the MMI criterion can tend to concentrate on outliers in the training data for which the corresponding class posterior probabilities are numerically close to zero. These outliers may have a strong impact

[1] The denominator in Eq. (4.1) must then be replaced by a positive constant $\neq 0$ which can be chosen arbitrarily; otherwise the criterion would not be defined.

Table 4.1: A class of discriminative training criteria covered by the extended unifying approach.

criterion	smoothing function $f(z)$	alternative word sequences $\mathcal{M}_r$	exponent κ	gain function $G(W, W_r)$
ML	z	$\varnothing$	–	$\delta(W, W_r)$
MMI	z	all	1	
CT		best	∞	
MCE	$-\frac{1}{1+e^{2\rho z}}$	all without W_r	*free*	
FT		best $\neq W_r$	∞	
Diversity Index	$\frac{1}{\rho}(e^{\rho z}-1)$	all	*free*	
Jeffreys	$\log\frac{e^z}{1-e^z}$	all	1	
Chernoff	$\frac{e^{\rho z}}{[1-e^z]^{1-\rho}}$	all	1	
MWE/MPE	e^z	all	1	$A(W, W_r)$

on both the optimization procedure and the parameter updates since already a small increase in $p_\theta(W_r|X_r)$ may result in a significant increase of the objective function. The MMI criterion is equivalent to the *Shannon Entropy* which, in case of given class priors, is also known as *Equivocation* [Shannon 48].

An approximation to the MMI criterion is given by the *Corrective Training* (CT) criterion. Here, the sum over *all* competing word sequences in the denominator is replaced with the *best recognized* sentence hypothesis. Formally, the CT criterion results from the MMI objective function in the limiting case of infinite κ:

$$\mathcal{F}_{\mathrm{CT}}(\theta) = \lim_{\kappa\to\infty} \frac{1}{R}\sum_{r=1}^{R} \log\left(\frac{p^\kappa(W_r)\cdot p_\theta^\kappa(X_r|W_r)}{\sum_{W\in\mathcal{M}_r} p^\kappa(W)\cdot p_\theta^\kappa(X_r|W)}\right)^{1/\kappa} \tag{4.5}$$

$$= \frac{1}{R}\log\frac{p(W_r)\cdot p_\theta(X_r|W_r)}{p(W_{\mathrm{opt}})\cdot p_\theta(X_r|W_{\mathrm{opt}})}. \tag{4.6}$$

An infinite κ has the same effect as the *Chebyshev* norm (l_∞-norm), which causes the sum in the denominator of Eq. (4.5) to be approximated by its maximum summand. Thus, the set $\mathcal{M}_r$ is reduced to a singleton, and only those training utterances can contribute to the CT criterion that are incorrectly recognized.

For the CT criterion, both parameters, κ and M_r, are mutually dependent and cannot be chosen independent of each other. Hence, it is equivalent whether $\mathcal{M}_r$ contains the best recognized sentence hypothesis with arbitrary choices of κ or whether κ is set to ∞ with $\mathcal{M}_r$ containing all possible sentence hypotheses. The CT criterion was first proposed in [Bahl & Brown$^+$ 88] and investigated in [Lee & Mahajan 89, Normandin & Morgera 91]. A comprehensive investigation on small vocabulary tasks can be found in [Macherey 98].

Diversity Index

The *Diversity Index of degree* ρ measures the divergence of a probability distribution from the uniform distribution. While a diversity index closer to the maximum at 0 means a larger divergence from the uniform distribution, smaller values indicate that all classes tend to be nearly

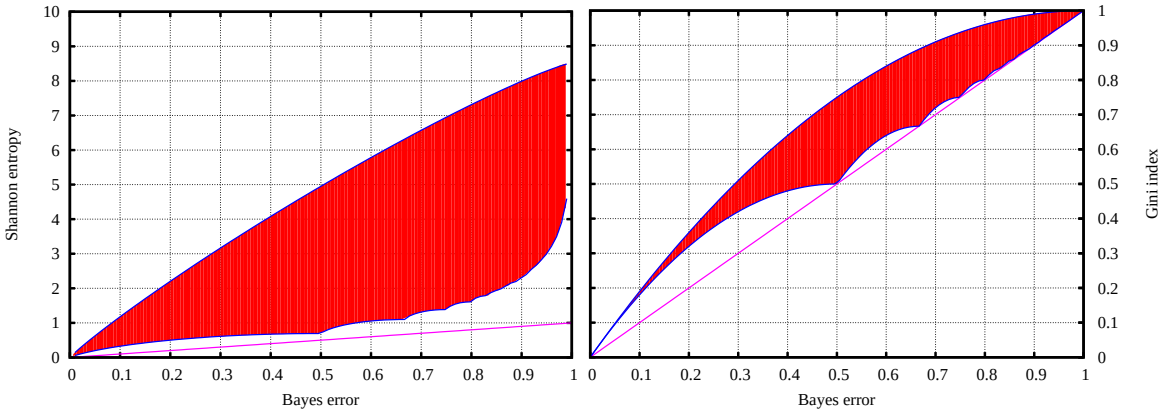


Figure 4.1: *Shannon Entropy* (left) and *Gini Index* (right) as function of the Bayes error rate for $C = 5000$ classes.

equally likely. The *Diversity Index* was introduced in [Patil & Taillie 82]. It applies the weighting function $f_\rho(z) = \frac{1}{\rho}(e^{\rho z} - 1)$ which results in the following expression for the discriminative training criterion:

$$\mathcal{F}_{\text{Diversity},\rho}(\theta) = \frac{1}{R}\sum_{r=1}^{R}\frac{1}{\rho}\Big[p_\theta^\rho(W_r|X_r) - 1\Big]. \tag{4.7}$$

Two well-known diversity indices, the *Shannon Entropy* (which is equivalent to the MMI criterion for constant class priors) and the *Gini Index*, are special cases of the *Diversity Index*. The *Shannon Entropy* results from the continuous limit as ρ approaches 0 while the *Gini Index* follows from setting $\rho = 1$:

$$\textit{Shannon Entropy}: \qquad \lim_{\rho\to 0}\mathcal{F}_{\text{Diversity},\rho}(\theta) \cong \frac{1}{R}\sum_{r=1}^{R}\log p_\theta(W_r|X_r). \tag{4.8}$$

$$\textit{Gini Index}: \qquad \mathcal{F}_{\text{Diversity},\rho=1}(\theta) \cong \frac{1}{R}\sum_{r=1}^{R}\big[p_\theta(W_r|X_r) - 1\big]. \tag{4.9}$$

Both criteria provide upper bounds to the Bayes classification error rate. The tightness of each bound results from sampling the *admissible region* using a *Monte Carlo* method. Given a function f, the admissible region A_f is defined as:

$$A_f \coloneqq \big\{(e,f) \in \mathbb{R}^2 \;\big|\; e \coloneqq 1 - \max_c\{p_c\} \;\wedge\; f \coloneqq f(\{p_c\})\big\}. \tag{4.10}$$

Figure 4.1 depicts the shape of the region A_f for the *Shannon Entropy* (left) and the *Gini Index* (right). For both criteria, the contour of the region is the convex hull of a piecewise defined function of the Bayes error e with the total number of classes C and the smoothing function f as its parameters [Schlüter & Keysers$^+$ 06]:

$$\text{upper bound:} \qquad u_{f,C}(e) = -(1-e)\cdot f(1-e) \;-\; e\cdot f\left(\tfrac{e}{C-1}\right), \qquad C \in \mathbb{N}_{>1} \tag{4.11}$$

$$\text{lower bound:} \qquad l_{f,C}(e) = \big\{\, g_{f,c}(1-e) \;:\; e \in \big[\tfrac{c-2}{c-1},\; \tfrac{c-1}{c}\big), \qquad 1 < c \leqslant C \tag{4.12}$$

with

$$g_{f,c}(x) = -x\cdot(c-1)\cdot f(x) \;-\; \big(1 - x\cdot(c-1)\big)\cdot f\big(1 - x\cdot(c-1)\big). \tag{4.13}$$

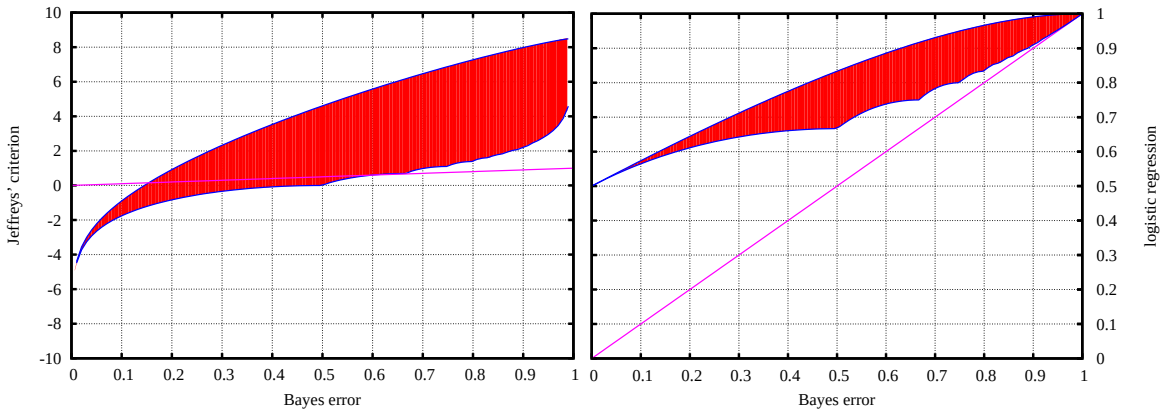


Figure 4.2: *Jeffreys' criterion* (left) and *logistic regression* (right) as function of the Bayes error rate for $C = 5000$ classes.

Since for given class priors the MMI criterion is equal to the *Shannon Entropy*, the *Diversity Index* can be used as an approximation to the MMI criterion. Depending on how ρ is chosen, the sensitivity of the MMI criterion toward outliers in the training data can be controlled.

Jeffreys' Criterion

The *Jeffreys*' criterion, which is also known as *Jeffreys' divergence*, is closely related to the *Kullback-Leibler distance* [Kullback & Leibler 51] and was first proposed in [Jeffreys 46]:

$$\mathcal{F}_{\text{Jeffreys}}(\theta) = -\frac{1}{R}\sum_{r=1}^{R}\log\frac{p_\theta(W_r|X_r)}{1-p_\theta(W_r|X_r)}. \tag{4.14}$$

The smoothing function $f(z) = \log\frac{e^z}{1-e^z}$ is not lower-bounded which means that an increase in the objective function will be large if the parameters are estimated such that no training utterance has a vanishing small posterior probability. Due to the exponential decay near the discontinuity point $z = 0$, the Jeffreys' criterion is more sensitive toward outliers in the training data than any other criterion discussed in this section. Figure 4.2 (left) depicts the admissible region for the Jeffreys' criterion as a function of the Bayes error rate.

Chernoff Affinity

The *Chernoff Affinity* was suggested in [Chernoff 52] as a generalization of *Battacharyya*'s measure of affinity [Bhattacharyya 46]. It employs the smoothing function $f_\rho(z) = -[e^z/(1-e^z)]^{\rho-1}$ with parameter $\rho \in (0,1)$ which leads to the following training criterion:

$$\mathcal{F}_{\text{Chernoff},\rho}(\theta) = -\frac{1}{R}\sum_{r=1}^{R}\left[\frac{p_\theta(W_r|X_r)}{1-p_\theta(W_r|X_r)}\right]^{\rho-1}. \tag{4.15}$$

For the special choice $\rho = 1/2$ the Chernoff affinity is equal to Bhattacharyya's measure of affinity, and the smoothing function becomes identical to the *Matushita Error* [Matushita 56]. The Matushita Error has been suggested as a distance measure for pattern recognition although it does not occur naturally as the limit of any standard discrimination rule. Figure 4.3 shows the admissible region of the Chernoff affinity for the special case $\rho = 1/2$.

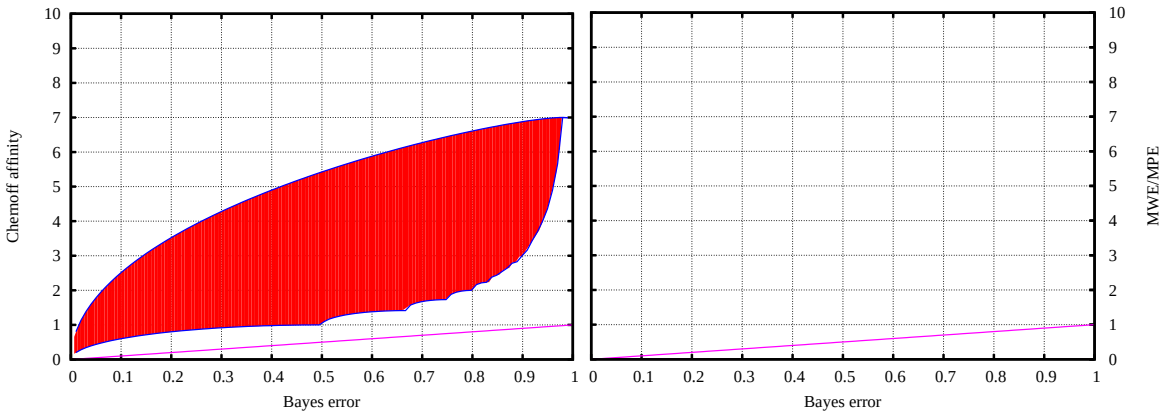


Figure 4.3: *Chernoff affinity* for $\rho = 1/2$ (left) and MWE/MPE criterion (right) as function of the Bayes error rate for $C = 50$ classes.

4.1.2 Smoothed Error Minimizing Training Criteria

Error minimizing training criteria such as the MCE, the MWE, and the MPE criterion aim at minimizing the expectation of an error related loss function on the training data. Let L denote any such loss function. Then the objective is to determine a parameter set $\bar{\theta}$ that minimizes the total costs due to classification errors:

$$\bar{\theta} = \operatorname*{argmin}_{\theta} \left\{ \frac{1}{R} \sum_{r=1}^{R} \mathsf{L}\big(W_r, \hat{W}(X_r, \theta)\big) \right\} \tag{4.16}$$

$$= \operatorname*{argmin}_{\theta} \left\{ \frac{1}{R} \sum_{r=1}^{R} \sum_{W} \mathsf{L}(W_r, W) \cdot \delta\big(\hat{W}(X_r, \theta), W\big) \right\} \tag{4.17}$$

with

$$\hat{W}(X_r, \theta) \coloneqq \operatorname*{argmax}_{W} p_\theta(W|X_r). \tag{4.18}$$

The optimization problem in Eq. (4.17) is conceptually difficult to handle which has the following reasons:

- The objective function includes an "argmin" operation which prevents the computation of a gradient.
- The objective function has many local optima: an optimization algorithm must handle this.
- The loss function L itself is typically a non-continuous step function and therefore not differentiable.

A remedy to make this class of error minimizing training criteria amenable to gradient based optimization methods is to replace Eq. (4.17) with the following expression:

$$\bar{\theta} = \operatorname*{argmin}_{\theta} \left\{ \frac{1}{R} \sum_{r=1}^{R} \sum_{W} \mathsf{L}(W_r, W) \cdot \frac{p^\kappa(W_r) \cdot p_\theta^\kappa(X_r|W_r)}{\sum_{V} p^\kappa(V) \cdot p_\theta^\kappa(X_r|V)} \right\}. \tag{4.19}$$

Eq. (4.19) is essentially a smoothed error count with a parameter κ to adjust the smoothness. Discriminative criteria like the MCE, the MWE, and the MPE criterion differ only with respect to the choice of the loss function L. While the MCE criterion typically applies a smoothed sentence error loss function, both the MWE and the MPE criterion are based on approximations of the word or phoneme error rate. The particular choice of the loss function L would not, however, affect the difference between the model-based classification error and the Bayes classification error if an infinite amount of training data was used. Since all criteria contained in the extended unifying approach are to be maximized, the loss function L is replaced with a corresponding gain function G (cf. Table 4.1).

Minimum Classification Error (MCE) and Falsifying Training (FT) Criterion

The MCE criterion aims at minimizing the expectation of a smoothed sentence error on training data. According to Bayes' decision rule, the probability of making a classification error in utterance r is given by:

$$p_B(e_r) = 1 - p(W_r|X_r) \tag{4.20}$$

$$= \frac{\sum\limits_{W \neq W_r} p(W) \cdot p(X|W)}{\sum\limits_{W} p(W) \cdot p(X|W)} \tag{4.21}$$

$$= \frac{1}{1 + \dfrac{p(W_r) \cdot p(X|W_r)}{\sum\limits_{W \neq W_r} p(W) \cdot p(X|W)}} \tag{4.22}$$

Smoothing the local error probability $p_B(e_r)$ with a sigmoid function f_ρ and carrying out the sum over all training utterances yields the MCE criterion:

$$\mathcal{F}_{\mathrm{MCE}}(\theta;\kappa) = -\frac{1}{R}\sum_{r=1}^{R} f_\rho\left(\log \frac{p^\kappa(W_r) \cdot p_\theta^\kappa(X_r|W_r)}{\sum\limits_{W \neq W_r} p^\kappa(W) \cdot p_\theta^\kappa(X_r|W)}\right). \tag{4.23}$$

The smoothing function f_ρ must be monotone increasing and needs to satisfy $f_\rho(\frac{1}{2} - z) = 1 - f_\rho(\frac{1}{2} + z)$. These requirements are met by the *logistic function*:

$$f_\rho(z) = \frac{1}{1 + e^{2\rho z}}. \tag{4.24}$$

The logistic function weighs the contribution of each training utterance on a sentence level. Depending on how the slope parameter ρ is chosen, the derivative of f_ρ gives lower weight to training utterances for which $|\log F_r(\theta)| \gg 0$. As a consequence, training sentences that are close to a decision boundary and thus likely to be affected by even small changes in the parameter set will influence the criterion the most. Conversely, reliably recognizable utterances as well as outliers in the training data are down-weighted and affect the criterion only marginally. This is also the case if incorrect transcriptions are used or if the speech signal was recorded under aggravating conditions as, for instance, in the presence of intensified background noise.

Similar to the CT criterion, the *Falsifying Training*[2] (FT) derives from the MCE criterion in the limiting case of infinite κ:

$$\mathcal{F}_{\mathrm{FT}}(\theta) = \lim_{\kappa \to \infty} \mathcal{F}_{\mathrm{MCE}}(\theta;\kappa). \tag{4.25}$$

[2] The term *to falsify* means "to disprove a hypothesis through empirical observation". It is the analog of the *Corrective Training* under the MCE criterion and was first proposed in [Macherey 98, p. 20].

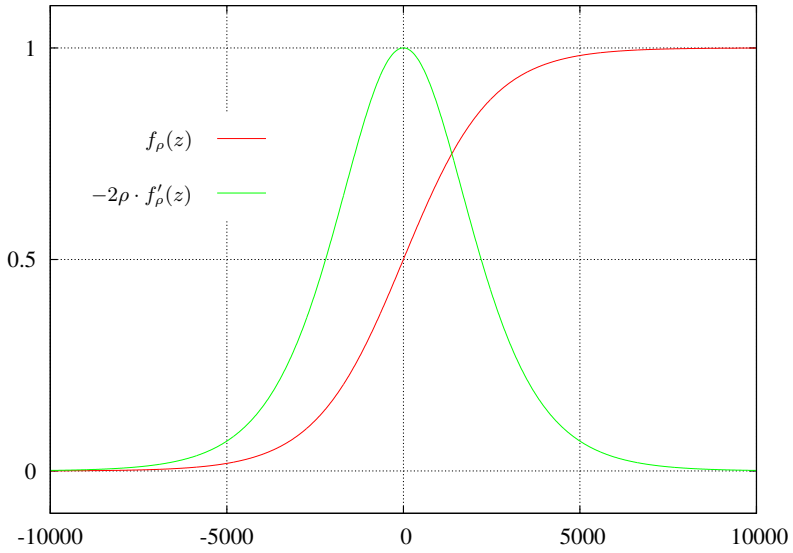


Figure 4.4: Course of the sigmoid function $f_\rho(z) = 1/(1 + e^{2\rho z})$ and its derivative $f'_\rho(z)$ for $\rho = 4 \cdot 10^{-4}$.

As before, an infinite κ has the same effect as the maximum approximation, that is, the set $\mathcal{M}_r$ becomes a singleton containing only the best recognized word sequence which does *not* match the spoken word sequence. As a consequence, either the choice of the weighting exponent κ or the representation of additional hypotheses other than the best incorrect word sequence is redundant. A thorough investigation on the FT criterion can be found in [Macherey 98].

Minimum Word Error (MWE) and Minimum Phone Error (MPE) Criterion

The objective of the *Minimum Word Error* (MWE) criterion as well as its closely related *Minimum Phone Error* (MPE) criterion is to minimize the expectation of an approximation to the word or phoneme accuracy on training data. Originally, the MWE criterion was proposed as *Overall Risk Criterion Estimation* in [Na & Jeon$^+$ 95]. An investigation on large vocabulary speech recognition tasks can be found in [Kaiser & Horvat$^+$ 00]. After an efficient lattice-based training scheme was found and successfully implemented in [Povey & Woodland 02], the criterion has received increasing interest. Both criteria compute the average transcription accuracy over all sentence hypotheses considered for discrimination:

$$\mathcal{F}_{\text{MWE/MPE}}(\theta) = \frac{1}{R} \sum_{r=1}^{R} \sum_{W \in \mathcal{M}_r} p_\theta^\kappa(W|X_r) \cdot A(W, W_r). \tag{4.26}$$

Here, $p_\theta^\kappa(W|X_r)$ is defined as the posterior probability of the sentence hypothesis W scaled by a factor κ in the log-space:

$$p_\theta^\kappa(W|X_r) \coloneqq \frac{p^\kappa(W) \cdot p_\theta^\kappa(X_r|W)}{\sum_{V \in \mathcal{M}_r} p^\kappa(V) \cdot p_\theta^\kappa(X_r|V)}. \tag{4.27}$$

Due to [Kaiser & Horvat$^+$ 00, p. 93], the function $A(W, W_r)$ is defined as the number of words in the reference transcription W_r reduced by the number of word errors in the hypothesis W. For

the MPE criterion, the accuracy is calculated analogously, however, on a phone-level rather than on a word-level. Both the MWE and the MPE criterion measure the risk of misclassification on training data and are motivated by Bayes' decision rule for minimum expected loss.

Minimum Squared Error (MSE) Criterion

The *Minimum Squared Error* (MSE) criterion is defined as the sum over the squared errors of a discriminant function g_θ which depends on a parameter θ and the ideal output values for a sequence of labeled training patterns $(x_1, k_1), \ldots, (x_n, k_n), \ldots, (x_N, k_N)$:

$$\mathcal{F}_N(g, \theta) = \frac{1}{N} \sum_{n=1}^{N} \sum_{k=1}^{K} \Big[g_\theta(x_n, k) \; - \; \delta(k, k_n) \Big]^2 . \tag{4.28}$$

The discriminant function g_θ takes two arguments: the input pattern x and the index of the hypothesized class k. The ideal output value is set to 1 for the correct class and 0 otherwise.

The MSE criterion is a special case of the *Minkovsky loss* and is often used to optimize the parameters for neural networks as, for instance, the *multi-layer perceptron* [Bishop 95]. Although the MSE criterion is formally not captured by the extended unifying approach, it can easily be integrated into the discriminative framework by replacing the sum over all classes in Eq. (4.28) with the sum over all sentence hypotheses. Since sentence posterior probabilities cannot be factorized appropriately on word lattices, the MSE criterion is defined on a word rather than on a sentence level:

$$\mathcal{F}_{\mathrm{MSE}}(\theta) = -\frac{1}{R} \sum_{r=1}^{R} \sum_{W_a w W_e \in \mathcal{M}_r} \Big[p_\theta(w | x_r{}_{t_a}^{t_e}) \; - \; \delta(w, w_r) \Big]^2 . \tag{4.29}$$

A refinement of the MSE criterion can be obtained by replacing the *Kronecker* function with a smoothed accuracy measure $A(w, w_r)$ that accounts for the relative temporal overlap between the time boundaries of the reference word w_r and a hypothesis w:

$$A(w, w_r) = \delta(w, w_r) \cdot \varrho(w, w_r) \tag{4.30}$$

with

$$\varrho(w, w_r) = \max \left\{ 0, \; \frac{\min\{t_e(w_r), \, t_e(w)\} - \max\{t_b(w_r), \, t_b(w)\} + 1}{t_e(w_r) - t_b(w_r) + 1} \right\}, \tag{4.31}$$

and $t_b(\cdot)$ and $t_e(\cdot)$ denoting the begin and end time, respectively. Similar measures of accuracy are used in the MWE and the MPE criterion (cf. Section 5.2).

4.2 Parameter Optimization

This section presents the re-estimation equations for the MMI objective function with respect to the parameters of the acoustic model. Based on the formal differentiation, a set of linear operators referred to as *discriminative averages* is introduced that encapsulate the alignments of the observation sequences of all training utterances with the states of the respective Markov automata for the spoken word sequence and the competing word sequences.

4.2.1 Formal Differentiation

The derivative of the MMI objective function with respect to the parameters of the acoustic model $\{c_{sl}, \theta_{sl}\}$ is given by the following equation:

$$\frac{\partial \mathcal{F}_{\mathrm{MMI}}(\theta)}{\partial \{c_{sl}, \theta_{sl}\}} = \frac{1}{R} \sum_{r=1}^{R} \sum_{t=1}^{T_r} \sum_{W \in \mathcal{M}_r} \left[\delta_{W,W_r} - p_\theta(W|X_r)\right] \cdot \gamma_{rt}(s|W) \cdot \\ \cdot \frac{c_{sl} \cdot p(x|\theta_{sl})}{\sum_{\tilde{l}=1}^{L_s} c_{s\tilde{l}} \cdot p(x|\theta_{s\tilde{l}})} \cdot \frac{\partial \log\left[c_{sl} \cdot p(x_{rt}|\theta_{sl})\right]}{\partial \{c_{sl}, \theta_{sl}\}}. \tag{4.32}$$

To simplify the notation, we introduce the *generalized forward-backward probabilities* $\gamma_{rt}(s)$ which are defined as the sum over the conditional FB probabilities $\gamma_{rt}(s|W)$ for all sentence hypotheses W considered for discrimination:

$$\gamma_{rt}(s) \coloneqq \sum_{W \in \mathcal{M}_r} p_\theta(W|X_r) \cdot \gamma_{rt}(s|W) = p_\theta(s|X_r), \tag{4.33}$$

$$\gamma_{rt}(s,l) \coloneqq \sum_{W \in \mathcal{M}_r} p_\theta(W|X_r) \cdot \gamma_{rt}(sl|W) = p_\theta(sl|X_r). \tag{4.34}$$

Other than $\gamma_{rt}(s|W)$, which denotes the probability that for a given sequence of acoustic observation vectors X_r a time alignment path for the word sequence W leads through state s at time frame t (cf. Sec. 2.5.1), $\gamma_{rt}(s)$ is defined as the probability that state s is hypothesized *independent* of any word sequence at time frame t given the sequence of acoustic observation vectors X_r. The independence of word sequences is due to the sum over all sentence hypotheses contained in the set $\mathcal{M}_r$ which is equivalent to computing the marginal distribution $\sum_W p(s_t{=}s, W|X_r) = p(s_t{=}s|X_r)$. An efficient computation scheme to estimate $\gamma_{rt}(s)$ is described in Chapter 5. With the definition of the generalized FB probabilities, Eq. (4.32) can be re-written as:

$$\frac{\partial \mathcal{F}_{\mathrm{MMI}}(\theta)}{\partial \{c_{sl}, \theta_{sl}\}} = \frac{1}{R} \sum_{r=1}^{R} \sum_{t=1}^{T_r} \left[\gamma_{rt}(sl|W_r) - \gamma_{rt}(s,l)\right] \cdot \frac{\partial \log\left[c_{sl} \cdot p(x_{rt}|\theta_{sl})\right]}{\partial \{c_{sl}, \theta_{sl}\}}. \tag{4.35}$$

Eq. (4.35) is composed of the difference between two averages which are based on the conditional and the generalized FB probabilities. Because these averages occur very frequently in discriminative training, they are defined as functionals over a polynomial g that takes a sequence of acoustic observation vectors $\mathcal{X}$ as its argument:

$$\Gamma_{sl}^{\mathsf{num}}(g\{\mathcal{X}\}) \coloneqq \frac{1}{R} \sum_{r=1}^{R} \sum_{t=1}^{T_r} \gamma_{rt}(s,l|W_r) \cdot g(x_{rt}), \tag{4.36}$$

$$\Gamma_{sl}^{\mathsf{den}}(g\{\mathcal{X}\}) \coloneqq \frac{1}{R} \sum_{r=1}^{R} \sum_{t=1}^{T_r} \gamma_{rt}(s,l) \cdot g(x_{rt}). \tag{4.37}$$

The superscripts num and den indicate whether the respective functional refers to the numerator or the denominator statistics of the underlying discriminative criterion. Both functionals are linear operators and preserve convexity if the polynomial g is convex in $\mathcal{X}$. The difference between $\Gamma_{sl}^{\mathsf{num}}(g\{\mathcal{X}\})$ and $\Gamma_{sl}^{\mathsf{den}}(g\{\mathcal{X}\})$ is referred to as *discriminative average*:

$$\Gamma_{sl}(g\{\mathcal{X}\}) \coloneqq \Gamma_{sl}^{\mathsf{num}}(g\{\mathcal{X}\}) - \Gamma_{sl}^{\mathsf{den}}(g\{\mathcal{X}\}), \tag{4.38}$$

Discriminative averages are also linear in g. In general, they do not, however, preserve convexity.

Carrying out the sum over all mixture components $l = 1, \ldots, L_s$ and states $s = 1, \ldots, S$ yields the marginal discriminative averages:

$$\Gamma_s^{\mathsf{num}}(g\{\mathcal{X}\}) \coloneqq \sum_{l=1}^{L_s} \Gamma_{sl}^{\mathsf{num}}(g\{\mathcal{X}\}), \qquad \Gamma_s^{\mathsf{den}}(g\{\mathcal{X}\}) \coloneqq \sum_{l=1}^{L_s} \Gamma_{sl}^{\mathsf{den}}(g\{\mathcal{X}\}), \tag{4.39}$$

$$\Gamma^{\mathsf{num}}(g\{\mathcal{X}\}) \coloneqq \sum_{s=1}^{S} \Gamma_s^{\mathsf{num}}(g\{\mathcal{X}\}), \qquad \Gamma^{\mathsf{den}}(g\{\mathcal{X}\}) \coloneqq \sum_{s=1}^{S} \Gamma_s^{\mathsf{den}}(g\{\mathcal{X}\}), \tag{4.40}$$

$$\Gamma_s(g\{\mathcal{X}\}) \coloneqq \sum_{l=1}^{L_s} \Gamma_{sl}(g\{\mathcal{X}\}), \qquad \Gamma(g\{\mathcal{X}\}) \coloneqq \sum_{s=1}^{S} \Gamma_s(g\{\mathcal{X}\}). \tag{4.41}$$

The derivative of the MMI criterion with respect to a state transition probability $p(s|\sigma)$ is obtained similarly:

$$\frac{\partial \mathcal{F}_{\mathrm{MMI}}(\theta)}{\partial p(s|\sigma)} = \frac{1}{R} \sum_{r=1}^{R} \sum_{t=1}^{T_r} \Big[\gamma_{rt}(s, \sigma|W_r) - \gamma_{rt}(s, \sigma) \Big] \cdot \frac{1}{p(s|\sigma)}, \tag{4.42}$$

where

$$\gamma_{rt}(\sigma, s) \coloneqq \sum_{W \in \mathcal{M}_r} p_\theta(W|X_r) \cdot \gamma_{rt}(\sigma, s|W). \tag{4.43}$$

As before, the derivative is split into the portion for the correct and the competing model which induces the following set of discriminative averages for state transitions:

$$\mathcal{T}_{\sigma s}^{\mathsf{num}} \coloneqq \frac{1}{R} \sum_{r=1}^{R} \sum_{t=1}^{T_r} \gamma_{rt}(\sigma, s|W), \tag{4.44}$$

$$\mathcal{T}_{\sigma s}^{\mathsf{den}} \coloneqq \frac{1}{R} \sum_{r=1}^{R} \sum_{t=1}^{T_r} \gamma_{rt}(\sigma, s), \tag{4.45}$$

$$\mathcal{T}_{\sigma s} \coloneqq \mathcal{T}_{\sigma s}^{\mathsf{num}} - \mathcal{T}_{\sigma s}^{\mathsf{den}}. \tag{4.46}$$

The gradient of the MMI criterion with respect to the parameters $\{c_{sl}, \theta_{sl}\}$ and $p(s|\sigma)$ is thus given by:

$$\frac{\partial \mathcal{F}_{\mathrm{MMI}}(\theta)}{\partial \{c_{sl}, \theta_{sl}\}} = \Gamma_{sl} \left(\frac{\partial \log \left[c_{sl} \cdot p(x_{rt}|\theta_{sl}) \right]}{\partial \{c_{sl}, \theta_{sl}\}} \right), \tag{4.47}$$

$$\frac{\partial \mathcal{F}_{\mathrm{MMI}}(\theta)}{\partial\, p(s|\sigma)} = \frac{\mathcal{T}_{\sigma s}}{p(s|\sigma)}. \tag{4.48}$$

4.2.2 Extended Baum Algorithm

An optimization procedure as convenient as the *Baum-Welch* (BW) algorithm [Baum & Petrie$^+$ 70] is not available for discriminative training, which is a harder optimization problem than Maximum-Likelihood based training procedures. In a discriminative framework, the parameter optimization is usually based on

- the *Extended Baum* (EB) algorithm or
- a *Gradient Descent* (GD) method.

The EB algorithm is due to [Gopalakrishnan & Kanevsky$^+$ 91] and generalizes the *Baum-Eagon* inequality for homogeneous polynomials[3] [Baum & Eagon 67] to rational functions[4]. An example of such a function is the MMI objective function for discrete-probability HMMs. Provided that the parameters are subject to a sum-to-one constraint, the Baum-Eagon inequality states that for any homogeneous polynomial $P(\theta)$ in variables $\theta = \{\theta_{ij}\}$ with solely positive coefficients, the transformation:

$$\bar{\theta}_{ij} = \frac{\theta_{ij} \cdot \frac{\partial P(\theta)}{\partial \theta_{ij}}}{\sum_{\tilde{\jmath}=1}^{J_i} \theta_{i\tilde{\jmath}} \cdot \frac{\partial P(\theta)}{\partial \theta_{i\tilde{\jmath}}}} \tag{4.49}$$

is a growth transformation on the manifold $\mathcal{M}_\theta = \{\forall i, j\colon \theta_{ij} \geqslant 0 \ \wedge \ \forall i : \sum_{j=1}^{J_i} \theta_{ij} = 1\}$, that is, for each $\theta \in \mathcal{M}_\theta$ we have $\bar{\theta} \in \mathcal{M}_\theta$, and it follows $P(\theta) \leqslant P(\bar{\theta})$.

In the EB algorithm, the objective is to maximize a *rational* function $P(\theta) = Q(\theta)/R(\theta)$ with Q and R being homogeneous polynomials in θ. However, instead of maximizing P directly, it suffices to maximize a polynomial $G(\bar{\theta}|\theta)$ defined by:

$$G(\bar{\theta}|\theta) = Q(\bar{\theta}) - P(\theta) \cdot R(\bar{\theta}) + \max_i\{D_i\}, \tag{4.50}$$

where $\{D_i\}$ is a set of positive, real constants. The convergence property then follows from the observation that any increase in G also implies an increase in P (a proof of this proposition is given in Section B.2.1):

$$G(\bar{\theta}|\theta) \geqslant G(\theta|\theta) \ \Rightarrow \ P(\bar{\theta}) \geqslant P(\theta). \tag{4.51}$$

The parameters $\{D_i\}$ must be constant in $\{\theta_{ij}\}$ and are to be chosen such that the resulting coefficients are nonnegative when added to $Q(\bar{\theta}) - P(\theta) \cdot R(\bar{\theta})$. In this case, $G(\bar{\theta}|\theta)$ satisfies the prerequisites of the Baum-Eagon inequality, and the new parameter set $\theta_{ij}^{(k+1)}$ can be determined through the following update rule:

$$\theta_{ij}^{(k+1)} = \frac{\theta_{ij}^{(k)} \cdot \left(\frac{\partial G(\theta|\theta^{(k)})}{\partial \theta_{ij}}\Big|_{\theta=\theta^{(k)}} + D_i \right)}{\sum_{\tilde{\jmath}=1}^{J_i} \theta_{i\tilde{\jmath}}^{(k)} \cdot \left(\frac{\partial G(\theta|\theta^{(k)})}{\partial \theta_{i\tilde{\jmath}}}\Big|_{\theta=\theta^{(k)}} + D_i \right)}. \tag{4.52}$$

The constants $\{D_i\}$ are also referred to as *iteration constants*. They control the convergence rate of the training process and can be interpreted as a reciprocal step size: the larger the choice of the iteration constants, the more alike the old and new parameters will be. Thus, setting the iteration constants too large a value results in low convergence rates, and many training iterations may be necessary to approach the local optimum. Conversely, if the iteration constants are set too small a value, convergence may be unstable and the objective function can oscillate in the course of the training process. A proper setting of the iteration constants is therefore essential in order to provide a stable and fast convergence rate. The values for $\{D_i\}$ which guarantee an increase in the objective function are, however, usually too large to be of any practical use. A guaranteed convergence is therefore often abandoned in favor of a faster but potentially unstable training itinerary.

[3] For a homogeneous polynomial, all monomes are of the same degree.
[4] A rational function is a ratio of two polynomials.

Originally the EB algorithm was designed for the optimization of discrete probability distributions in rational objective functions [Gopalakrishnan & Kanevsky$^+$ 89]. In [Normandin & Morgera 91, Normandin 91], the algorithm was extended and applied to continuous probability density functions. The idea behind this extension is based on a discretization of the underlying probability density function: by repeatedly refining the discretization level, the quantization error becomes infinitesimal and the rational approximation finally turns into the original continuous density function. The limiting case, however, causes the iteration constants to be unbounded, and hence the step size may become infinitesimal.

Normandin's derivation does not, however, provide a tight bound to the iteration constants, and the conclusion frequently drawn that the iteration constants have to be infinite in order to ensure convergence, is not stringent. This was shown, e.g., in [Kanevsky 95], where the application of the EB algorithm to general functions is strictly justified. In [Kanevsky 03, Kanevsky 04, Kanevsky 05] and [Axelrod & Goel$^+$ 04], it is explicitly shown that, for continuous objective functions, convergence can be guaranteed with finite iteration constants. All proofs show, however, only the existence of an upper bound to the iteration constants but do not provide tight bounds.

In [Schlüter & Macherey$^+$ 97], it is shown that the EB algorithm has a close relationship with gradient based optimization methods. Hence, for a special choice of the iteration constants, the proposed gradient method leads to re-estimation equations that are nearly identical with the update rules derived in the EB algorithm.

4.2.3 Auxiliary Function for Continuous HMMs

The MMI criterion for continuous HMMs is optimized according to the following auxiliary function:

$$\begin{aligned}\mathcal{S}(\bar{\theta}|\theta) = \sum_{k\in K}\Bigg\{ & \frac{1}{R}\sum_{r=1}^{R}\sum_{t=1}^{T_r}\sum_{\sigma=1}^{S}\sum_{(s,\cdot)\in k}\Big[\gamma_{rt}(\sigma,s|W_r)-\gamma_{rt}(\sigma,s)\Big]\cdot\log p_{\bar{\theta}}(x_{rt},s|\sigma) \\ & + D_k\cdot\sum_{(s,l)\in k} c_{sl}\cdot\int p(x|\theta_{sl})\cdot\log p(x|\bar{\theta}_{sl})\,dx \\ & + C_s\sum_{l=1}^{L_s} c_{sl}\cdot\log\bar{c}_{sl}+\sum_{\sigma=1}^{S}E_\sigma\cdot p(s|\sigma)\cdot\log\bar{p}(s|\sigma)\Bigg\}.\end{aligned} \tag{4.53}$$

A formal derivation of this auxiliary function is shown in Section B.2.1. The derivation is motivated by the *Conditional Maximum Likelihood* (CML) algorithm proposed in [Gunawardana 01]. The CML algorithm extends Normandin's approach for HMMs with Gaussian emission distributions to HMMs with arbitrary continuous emission distribution, yet without the need for a discrete approximation to the probability density functions. Note that for constant language model probabilities, the MMI criterion is equivalent to the CML criterion [Nádas 83].

As introduced in Section 2.2, the tying scheme for covariance matrices is defined via a set of equivalence classes $k \in K$ where each k comprises the set of indices of those Gaussians that share a common covariance matrix. The mixture density weights c_{sl} can be integrated into D_k resulting in density specific iteration constants:

$$D_{sl} \coloneqq D_k \cdot c_{sl}. \tag{4.54}$$

From a formal point of view, the iteration constants do not depend on the equivalence class k anymore. Nevertheless, it can be useful to take a common factor D_k into account when choosing the step size for parameters that belong to the same mixture distribution or across which the same covariance matrix is shared. This is demonstrated in Section 6.3.8, where different methods on choosing the step size are investigated for different variance tying schemes. In particular, it is shown that, even if density specific variances are used, it is often more effective to employ mixture

specific iteration constants that are scaled with a density specific weight than searching for the least admissible constant which guarantees positive definite variances and ensures that denominators do not become singular.

Other than the mean and variance parameters, which belong to a continuous probability space, the mixture weights c_{sl} as well as the transition probabilities $\{p(s|\sigma)\}$ can be estimated according to the EB algorithm for rational objective functions because both are instances of discrete random variables. Conceptually, the new parameters $\bar{c}_{sl}$ can therefore always be estimated such that the objective function does not decrease.

4.2.4 Re-estimation Equations

Let $\Omega_{sl}(g\{x\})$ denote the expectation value of a function $g(x)$ with a Gaussian $p(x|\theta_{sl}) = \mathcal{N}(x|\theta_{sl})$ as the probability measure:

$$\Omega_{sl}\big(g\{x\}\big) = \int p(x|\theta_{sl}) \cdot g(x)\, dx. \tag{4.55}$$

Then the derivatives of the auxiliary function with respect to the parameters of the acoustic model yield:

$$\frac{\partial \mathcal{S}(\bar{\theta}|\theta)}{\partial \bar{\theta}_{sl}} = \Gamma_{sl}\left(\frac{\partial \log p(x|\bar{\theta}_{sl})}{\partial \bar{\theta}_{sl}}\right) \; + \; D_{sl}\Omega_{sl}\left(\frac{\partial \log p(x|\bar{\theta}_{sl})}{\partial \bar{\theta}_{sl}}\right), \tag{4.56}$$

$$\frac{\partial \mathcal{S}(\bar{\theta}|\theta)}{\partial \bar{c}_{sl}} = \Gamma_{sl}\left(\frac{1}{\bar{c}_{sl}}\right) \; + \; C_s \cdot \frac{c_{sl}}{\bar{c}_{sl}}, \tag{4.57}$$

$$\frac{\partial \mathcal{S}(\bar{\theta}|\theta)}{\partial \bar{p}(s|\sigma)} = \frac{\mathcal{T}_{\sigma s}}{\bar{p}(s|\sigma)} \; + \; E_\sigma \cdot \frac{p(s|\sigma)}{\bar{p}(s|\sigma)}. \tag{4.58}$$

Since the discriminative averages are linear operators, the re-estimation equations can easily be derived using the following partial derivatives of a Gaussian probability density function:

$$\frac{\partial \log p(x|\theta_{sl})}{\partial \mu_{sl}} = \mathbf{\Sigma}_{sl}^{-1}(x - \mu_{sl}), \tag{4.59}$$

$$\frac{\partial \log p(x|\theta_{sl})}{\partial \mathbf{\Sigma}_{sl}^{-1}} = \frac{1}{2}\Big(\mathbf{\Sigma}_{sl} - (x - \mu_{sl})(x - \mu_{sl})^T\Big). \tag{4.60}$$

Together with the following identities:

$$\Omega_{sl}(1) = 1, \qquad \Omega_{sl}(x) = \mu_{sl}, \qquad \Omega_{sl}(x \cdot x^T) = \mathbf{\Sigma}_{sl} + \mu_{sl} \cdot \mu_{sl}^T. \tag{4.61}$$

the update rules for the acoustic model parameters are given by the following expressions:

Re-estimation Equations for Means:

$$\frac{\partial \mathcal{S}(\bar{\theta}|\theta)}{\partial \bar{\mu}_{sl}} \stackrel{!}{=} 0 \quad \Leftrightarrow \quad \bar{\mu}_{sl} = \frac{\Gamma_{sl}(x) + D_{sl} \cdot \mu_{sl}}{\Gamma_{sl}(1) + D_{sl}} \tag{4.62}$$

Re-estimation Equations for Covariance Matrices:

$$\frac{\partial \mathcal{S}(\bar{\theta}|\theta)}{\partial \bar{\mathbf{\Sigma}}_k^{-1}} \stackrel{!}{=} 0 \quad \Leftrightarrow$$

$$\bar{\mathbf{\Sigma}}_k = \frac{\sum\limits_{(s,l)\in k} \Big[\Gamma_{sl}(x \cdot x^\top) + D_{sl} \cdot \big[\mathbf{\Sigma}_k + \mu_{sl} \cdot \mu_{sl}^\top\big] - \big[\Gamma_{sl}(1) + D_{sl}\big] \cdot \bar{\mu}_{sl} \cdot \bar{\mu}_{sl}^\top \Big]}{\sum\limits_{(s,l)\in k} \big[\Gamma_{sl}(1) + D_{sl}\big]} \tag{4.63}$$

Re-estimation Equations for Mixture Weights and Transition Probabilities:

The sum-to-one constraint for mixture weights and transition probabilities is taken into account by incorporating a set of Lagrangian multipliers $\lambda = \{\lambda_s\}$ and $\eta = \{\eta_\sigma\}$ into the auxiliary function:

$$\mathscr{S}\big(\bar{\theta}|\theta,\lambda,\eta\big) = \mathcal{S}(\theta,\bar{\theta}) \; - \; \sum_{s=1}^{S} \lambda_s \cdot \Big(\sum_{l=1}^{L_s} \bar{c}_{sl} - 1 \Big) \; - \; \sum_{\sigma=1}^{S} \eta_\sigma \cdot \Big(\sum_{s=1}^{S} \bar{p}(s|\sigma) - 1 \Big). \tag{4.64}$$

This yields the following update equations:

$$\frac{\partial \mathscr{S}\big(\bar{\theta}|\theta,\lambda,\eta\big)}{\partial \bar{c}_{sl}} = \frac{\partial \mathscr{S}\big(\bar{\theta}|\theta,\lambda,\eta\big)}{\partial \lambda_s} \stackrel{!}{=} 0 \quad \Leftrightarrow \qquad \bar{c}_{sl} = \frac{\Gamma_{sl}(1) + C_s \cdot c_{cl}}{\Gamma_s(1) + C_s}, \tag{4.65}$$

$$\frac{\partial \mathscr{S}\big(\bar{\theta}|\theta,\lambda,\eta\big)}{\partial \bar{p}(s|\sigma)} = \frac{\partial \mathscr{S}\big(\bar{\theta}|\theta,\lambda,\eta\big)}{\partial \eta_s} \stackrel{!}{=} 0 \quad \Leftrightarrow \qquad \bar{p}(s|\sigma) = \frac{\mathcal{T}_{\sigma s} + E_\sigma \cdot p(s|\sigma)}{\mathcal{T}_\sigma + E_\sigma}, \tag{4.66}$$

with the marginal counts $\mathcal{T}_\sigma \coloneqq \sum_{s=1}^{S} \mathcal{T}_{\sigma s}$. Since the update rule for mixture weights is known to be unstable for small values of $\Gamma_{sl}(1)$, the update scheme proposed in [Povey & Woodland 00] is used instead:

$$c_{sl}^{(i+1)} = \frac{\Gamma_{sl}^{\mathsf{num}}(1) + u_{sl} \cdot c_{sl}^{(i)}}{\sum\limits_{\tilde{l}} \Gamma_{s\tilde{l}}^{\mathsf{num}}(1) + u_{s\tilde{l}} \cdot c_{s\tilde{l}}^{(i)}}, \qquad u_{sl} \coloneqq \big[\max_{\tilde{l}} \Gamma_{s\tilde{l}}^{\mathsf{den}}(1)/c_{s\tilde{l}}^{(0)} \big] - \Gamma_{sl}^{\mathsf{den}}(1)/c_{sl}^{(0)}. \tag{4.67}$$

Eq. (4.67) results in a faster increase of the objective function than the update rule according to the EB algorithm. Since it is free of any smoothing constants, the mixture weight updates are not affected by the choice of the iteration constants.

4.3 Convergence Control

Setting the iteration constants is a key issue in discriminative training. If the step size is set too small a value, convergence will be slow. If it is set too large a value, it might not increase the objective function. While for discrete HMMs, setting the iteration constants is a solved problem [Gopalakrishnan & Kanevsky$^+$ 89], a similar solution which guarantees an increase in the objective function for practicable choices of the step size is not known for continuous HMMs. A useful lower bound on setting D_{sl} was found to be the value which ensures that all variances in the update equations are positive [Normandin 91]. In case of *full* covariance matrices, this leads to a system of higher-order inequalities [Macherey & Schlüter$^+$ 04]:

$\forall x \in \mathbb{R}^D \; \forall k \in K :$

$$x^\top \cdot \bar{\mathbf{\Sigma}}_k \cdot x > 0 \tag{4.68}$$

$$\Leftrightarrow \; x^\top \cdot \sum_{(s,l)\in k} \Big(\big[\Gamma_{sl}(x \cdot x^\top) + D_{sl} \cdot (\Sigma_k + \mu_{sl}\mu_{sl}^\top)\big] \cdot \big[\Gamma_{sl}(1) + D_{sl}\big] - \big[\Gamma_{sl}(x) + D_{sl} \cdot \mu_{sl}\big]^2 \Big) \cdot x > 0 \quad \wedge \quad \forall (s,l) \in k : \; D_{sl} > -\Gamma_{sl}(1) \tag{4.69}$$

$$\Leftrightarrow \; x^\top \cdot \sum_{(s,l)\in k} \Big(D_{sl}^2 \cdot \mathbf{A}_k \; + \; D_{sl} \cdot \mathbf{B}_{sl} \; + \; \mathbf{C}_{sl} \Big) \cdot x > 0 \quad \wedge \quad \forall (s,l) \in k : \; D_{sl} > -\Gamma_{sl}(1) \tag{4.70}$$

$$\Leftarrow \; \forall (s,l) \in k : \; \Big(D_{sl}^2 \cdot \mathbf{A}_k \; + \; D_{sl} \cdot \mathbf{B}_{sl} \; + \; \mathbf{C}_{sl} \Big) \cdot x > 0 \quad \wedge \quad D_{sl} > \max\{-\Gamma_{sl}(1),\, 0\} \tag{4.71}$$

with

$$\mathbf{A}_k = \Sigma_k, \tag{4.72}$$

$$\mathbf{B}_{sl} = \Gamma_{sl}(x \cdot x^\top) + \Gamma_{sl}(1) \cdot \big(\Sigma_k + \mu_{sl}\mu_{sl}^\top\big) - 2\Gamma_{sl}(x) \cdot \mu_{sl}^\top, \tag{4.73}$$

$$\mathbf{C}_{sl} = \Gamma_{sl}(1) \cdot \Gamma_{sl}(x \cdot x^\top) - \Gamma_{sl}(x) \cdot \Gamma_{sl}(x^\top). \tag{4.74}$$

In practice, the iteration constants are often set to twice the value necessary to ensure that all variances are positive. Although this heuristic proved to be effective in many settings [Woodland & Povey 02, Macherey & Schlüter$^+$ 04], it works best in combination with density specific variances. The reason for this is that only in case of density specific variances, the lower bounds on the iteration constants are given as the roots of a system of quadratic inequalities. If, however, tight variances are used, the inequalities become more complicated and the magnitude of the iteration constants that ensure positive variances increase with the number of densities that share a common covariance matrix.

In the following section, we first address the case of density-specific variances under a full and a diagonal modeling constraint. Two novel approaches on setting the iteration constants are suggested in Section 4.3.3. The first approach refines the heuristic of using twice the value to ensure positive variances by incorporating further statistics into the auxiliary function. The second approach is based on a decomposition of the re-estimaton formulae for covariance matrices into a sum of positive definite matrices. Both methods are compared and discussed with approaches proposed in [Woodland & Povey 02] and [Schlüter & Macherey$^+$ 01]. Experimental results are shown reported Section 6.3.8.

4.3.1 Density-Specific Variances: Full and Diagonal Case

If *density-specific* full covariance matrices are used, there is a one-to-one correspondence between the equivalence classes k representing the tying scheme and the pairs (s,l). Thus, Eq. (4.71) has the form of a *quadratic eigenvalue problem* (QEP), and the largest positive real eigenvalue D_{sl} solving Eq. (4.71) forms a lower bound on the iteration constant. The QEP can be turned into a linear non-symmetric eigenvalue problem by introducing an additional (unknown) eigenvector y and solving the resulting $2D \times 2D$ eigensystem [Press & Teukolsky$^+$ 02, p. 467]:

$$\begin{pmatrix} \mathbf{0} & \mathbf{1} \\ -\mathbf{A}_{sl}^{-1} \cdot \mathbf{C}_{sl} & -\mathbf{A}_{sl}^{-1} \cdot \mathbf{B}_{sl} \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix} = D_{sl}^{\min} \cdot \begin{pmatrix} x \\ y \end{pmatrix} \tag{4.75}$$

Due to the asymmetry, the computation of the eigenvalues is numerically less stable than it would be in the symmetric case. The matrix on the left-hand side of Eq. (4.75) should therefore be balanced before it is reduced into Hessenberg form. Balancing the matrix makes it less sensitive toward rounding errors, and the actual eigenvalue problem can thus be solved with a higher precision.

If the variances are estimated under a *diagonal* modeling constraint, the QEP decomposes into a system of quadratic inequalities (one inequality for each dimension d), and the largest positive root is a lower bound on the sought iteration constant:

$$D_{sl}^{\min} = \max_d \left\{ \left(-b_d + \sqrt{b_d^2 - 4a_d c_d} \right) / 2a_d \right\} \tag{4.76}$$

with

$$a_d = \operatorname{diag}(\mathbf{A}_{sl})_d, \qquad b_d = \operatorname{diag}(\mathbf{B}_{sl})_d, \qquad c_d = \operatorname{diag}(\mathbf{C}_{sl})_d.$$

Alternative Methods on Setting the Iteration Constants

In [Woodland & Povey 02], the iteration constants D_{sl} are set on a per-Gaussian level to the maximum of (1) twice the value necessary to ensure positive variance updates for all dimensions of the Gaussian $k \equiv (s, l)$ and (2) a further constant E multiplied with the denominator occupancy $\Gamma_{sl}^{\text{den}}(1)$. The constant E is empirically chosen from the set $\{1,\, 2,\, \max_{sl}\{D_{sl}^{\min}\} / \min_{sl}\{\Gamma_{sl}^{\text{den}}(1)\}\}$. The final iteration constants are thus given by:

$$D_{sl} = \max\left\{ h \cdot D_{sl}^{\min},\; E \cdot \Gamma_{sl}^{\text{den}}(1) \right\} + \epsilon, \quad h = 2.0. \tag{4.77}$$

Although this method works well for density-specific variances, it has the disadvantage that solving the root finding problem will become more complicated if tied variances are used because the denominators in Eq. (4.58) prevent a decomposition into an analytically easy to handle expression. Thus, by expanding the fractions, the resulting polynomial is of degree $|k|+1$, and even for moderate tying schemes (e.g. $5 \leqslant |k|$), the root finding problem has to be solved numerically. A putative remedy would be to abandon the goal of finding the smallest value ensuring positive variances, and to consider the fractions in Eq. (4.58) to be independent quadratic inequalities instead. However, though the maximum over all roots would yield a valid choice for D_k, this method bears the problem that the magnitude of D_k will be controlled by densities with nearly equally occupied numerator and denominator counts, that is, for which $\Gamma_{sl}^{\text{num}}\big(g\{x\}\big) \approx \Gamma_{sl}^{\text{den}}\big(g\{x\}\big)$ holds[5]. Almost identically occupied numerator and denominator statistics lead to very small gradients in the update equations and thus result in very large iteration constants which reduce the convergence speed.

In [Schlüter & Macherey$^+$ 01], the discriminative approach was investigated for Gaussian mixture models with either a single, globally pooled variance or state specific variances. In case of state specific variances, the iteration constants were determined under the additional constraint that not only the variances have to be positive, but also the denominators in *all* re-estimation equations, including the update rules for means. This lead to the following set of inequalities:

$$\overline{\sigma}_s^2 \geqslant \alpha > 0, \qquad \Gamma_{sl}(1) + D_s \geqslant \frac{1}{\beta_s} > 0. \tag{4.78}$$

Here, α is a positive constant that provides a lower limit for the variances. Usually, α can be set to 1 since variances are typically larger by several orders of magnitudes. The lower limit to the denominator, β_s, is determined according to the following heuristic formula:

$$\frac{1}{\beta_s} = 1 + \big(|\Gamma_{s\eta_s}(1)| - 1\big) \cdot \frac{\big|\Gamma_{s\eta_s}(1)\big|}{\Gamma_{s\eta_s}^{\max}}, \tag{4.79}$$

[5] Equally occupied numerator and denominator counts may originate from densities that are almost never misclassified.

with

$$\eta_s = \operatorname*{argmax}_l \big\{|\Gamma_{sl}(1)|\big\}, \tag{4.80}$$

$$\Gamma^{\max}_{s\eta_s} = \max\big\{\Gamma^{\text{num}}_{s\eta_s}(1), \Gamma^{\text{den}}_{s\eta_s}(1)\big\}. \tag{4.81}$$

The idea behind this formula is to choose $1/\beta_s$ according to the magnitude of $\Gamma_{s\eta_s}(1)$ as far as the ratio $|\Gamma_{s\eta_s}(1)|/\Gamma^{\max}_{s\eta_s}$ is not too small. Otherwise, if the ratio is small, the contributions of $\Gamma^{\text{num}}_{s\eta_s}(1)$ and $\Gamma^{\text{den}}_{s\eta_s}(1)$ nearly cancel, and β_s approaches a fixed limit. Based on these quantities, the minimal iteration constants are given by:

$$D^{\min}_s = \max_d \left(\sum_{l=1}^{L_s} -\Gamma_{sl}(x_d^2) + 2\Gamma_{sl}(x_d)\mu_{sld} - \Gamma_{sl}(1)\mu_{sld}^2 + \beta_s\big[\Gamma_{sl}(x_d) - \Gamma_{sl}(1)\mu_{sld}\big]^2 + \alpha\Gamma_{sl}(1) \right) \Big/ \sum_{l=1}^{L_s} c_{sl}\big[\sigma_{sd}^2 - \alpha\big] \tag{4.82}$$

The final iteration constants are then chosen as follows:

$$D_s = h \cdot \max\left\{ D^{\min}_s,\ \max_l \frac{1}{c_{sl}}\left[\frac{1}{\beta_s} - \Gamma_{sl}(1)\right]\right\}. \tag{4.83}$$

This method turns out to be very effective in combination with a globally pooled variance. However, as shown in [Macherey & Schlüter$^+$ 04], the iteration constants tend to be too large if density specific variances are used.

4.3.2 Setting the Iteration Constants for Tied Variances

A general method for setting the iteration constants with arbitrary tying schemes should meet the following properties:

1. The iteration constants should not depend on additional parameters.
2. The method should not introduce further constraints that go beyond the approved heuristic of positive definite covariance matrices.
3. In case of using density specific variances, the magnitude of the iteration constants should be in the same range as the roots of the respective system of quadratic inequalities (cf. Eq. (4.76)).
4. Increasing the number of Gaussians within an equivalence class should only cause a small increase in the iteration constants.

The heuristics proposed in [Woodland & Povey 02] as well as in [Schlüter & Macherey$^+$ 01] meet these requirements only partially. The reason for this becomes apparent when re-inspecting the update rules and analyzing their effect on the iteration constants. According to Eq. (4.62), D_{sl} can be considered as a smoothing parameter which interpolates between a previous estimation of the mean μ_{sl} and the discriminative statistics $\Gamma_{sl}(x)$. However, while μ_{sl} is a normalized quantity, the discriminative statistics $\Gamma_{sl}(x)$ are not. Since $\Gamma_{sl}(x)$ is composed of the difference between two unnormalized quantities, namely $\Gamma^{\text{num}}_{sl}(x)$ and $\Gamma^{\text{den}}_{sl}(x)$, D_{sl} has to scale μ_{sl} such that $D_{sl} \cdot \mu_{sl}$ and $\big\{\Gamma^{\text{num}}_{sl}(x), \Gamma^{\text{den}}_{sl}(x)\big\}$ have the same magnitude. This scaling should, however, be separated from the actual iteration constant as it depends on the number of observations aligned with the density (s, l) rather than on discrimination. Therefore, D_{sl} is proposed to be split into two terms: a term Δ_k that depends on the equivalence class k and a density specific term $\Lambda_{sl}(1)$ which accounts for the

different magnitudes of μ_{sl} and $\{\Gamma_{sl}^{\text{num}}(x), \Gamma_{sl}^{\text{den}}(x)\}$. With the assumption that both quantities, $\Gamma_{sl}^{\text{nums}}(x)$ and $\Gamma_{sl}^{\text{num}}(x)$, should be proportional to μ_{sl}, the new parameter $\Lambda_{sl}(1)$ is set to the Maximum-Likelihood estimates of the state occupancy probabilities obtained from the preceding training iteration. By replacing the iteration constants in Eq. (4.53) according to

$$D_{sl} \rightarrow \Delta_k \cdot \Lambda_{sl}(1), \tag{4.84}$$

we obtain the following expression for the auxiliary function:

$$\begin{aligned}\mathcal{S}_K(\bar{\theta}|\theta) = \sum_{k\in K}\sum_{(s,l)\in k}\Bigg(&\frac{1}{R}\sum_{r=1}^{R}\sum_{t=1}^{T_r}\sum_{\sigma=1}^{S}\Big[\gamma_{rt}(\sigma,s|W_r)-\gamma_{rt}(\sigma,s)\Big]\cdot\log p_{\bar{\theta}}(x_{rt},s|\sigma)\\ &+\Delta_k\cdot\Lambda_{sl}(1)\int p(x|\theta_{sl})\cdot\log p(x|\bar{\theta}_{sl})\;dx\\ &+C_s\sum_{l=1}^{L_s}c_{sl}\cdot\log\bar{c}_{sl}+\sum_{\sigma=1}^{S}E_\sigma\cdot p(s|\sigma)\cdot\log\bar{p}(s|\sigma)\Bigg).\end{aligned} \tag{4.85}$$

The statistics defined in Eqs. (4.72–4.74) have to be replaced accordingly:

$$\widehat{\mathbf{A}}_{sl} = \Lambda_{sl}^2(1)\cdot\mathbf{A}_k, \quad \widehat{\mathbf{B}}_{sl} = \Lambda_{sl}(1)\cdot\mathbf{B}_{sl}. \tag{4.86}$$

Thus, the final iteration constants are given by:

$$\Delta_k = \max\Big\{h\cdot\Delta_k^{\min},\ \max_{(s,l)\in k}\big\{-\Gamma_{sl}(1)/\Lambda_{sl}(1)\big\}\Big\}+\epsilon, \tag{4.87}$$

where $\Delta_k^{\min}$ corresponds to the roots (or eigenvalues) of the respective system of quadratic inequalities. Note that the mixture weights are not affected by the new auxiliary function, that is, they can still be estimated according to the update rule in Eq. (4.67).

4.3.3 Decomposition into a Sum of Positive Definite Matrices

An alternative approach to set the iteration constants results from decomposing the update equations for the covariance matrices into the following sum:

$$\begin{aligned}\bar{\boldsymbol{\Sigma}}_k &= \Bigg(\sum_{(s,l)\in k}\overbrace{\Big[\Gamma_{sl}(x\cdot x^\top)+D_{sl}\cdot\big[\Sigma_k+\mu_{sl}\cdot\mu_{sl}^\top\big]-\big[\Gamma_{sl}(1)+D_{sl}\big]\cdot\bar{\mu}_{sl}\cdot\bar{\mu}_{sl}^\top\Big]}^{=:\mathbf{G}_{sl}}\Bigg)\\ &\qquad\Big/\sum_{(s,l)\in k}\underbrace{\big[\Gamma_{sl}(1)+D_{sl}\big]}_{=:h_{sl}}\end{aligned} \tag{4.88}$$

$$= \Bigg(\sum_{(s,l)\in k}\mathbf{G}_{sl}\Bigg)\Big/\Bigg(\sum_{(s,l)\in k}h_{sl}\Bigg). \tag{4.89}$$

If the iteration constants are chosen such that (1) all matrices $\mathbf{G}_{sl}$ are positive definite and (2) all denominators h_{sl} are larger than zero, then $\bar{\boldsymbol{\Sigma}}_k$ is positive definite by construction:

$$\forall(s,l)\in k:\ \det\mathbf{G}_{sl}>0\ \wedge\ h_{sl}>0 \quad\Longrightarrow\quad \det\bar{\boldsymbol{\Sigma}}_k>0. \tag{4.90}$$

As shown in Section 6.3.4, this method provides very fast convergence speeds for tied covariance matrices. At the same time, it is independent of the underlying tying scheme.

4.4 Optimization in the Extended Unifying Approach

So far, the parameter optimization problem was solely discussed for the MMI criterion. In this section, the function maximization is generalized to *all* criteria included in the extended unifying approach. Other than for the MMI criterion, both the smoothing function f as well as the expectation value of the gain function G have now to be taken into account. The following derivation is based on the assumption that the gain function $G(W, W_r)$ is decomposable into a sum of local costs. This constraint allows us to efficiently estimate expected costs over word or phoneme hypotheses on lattices (cf. Chapter 5).

4.4.1 Formal Differentiation

With the definition of the local discriminative criterion $\mathcal{F}_r(\theta)$ in Eq. (4.2), the extended unifying approach can be written as:

$$\mathcal{F} = \frac{1}{R} \sum_{r=1}^{R} f\big(\log \mathcal{F}_r\big). \tag{4.91}$$

In the following, acoustic probabilities are considered with respect to phoneme hypotheses $h := \{q, \tau, t\}$ where q denotes a phoneme with start time τ and end time t. Each word sequence with given phoneme boundary times can thus be represented by a corresponding sequence of phoneme hypotheses h_1^M:

$$p(x_1^T|w_1^N) = p(x_1^T|h_1^M). \tag{4.92}$$

Similarly, the language model probabilities are redistributed over phoneme sequences:

$$p(w_1^N) = p(h_1^M). \tag{4.93}$$

Both the acoustic and the language model probabilities can then be factorized as follows:

$$p(x_1^T|h_1^M) = \prod_{m=1}^{M} p(x_{t_{m-1}}^{t_m}|h_m), \tag{4.94}$$

$$p(h_1^M) = \prod_{m=1}^{M} p(h_m|h_{m-1}), \tag{4.95}$$

using a unigram or bigram language model which is mapped onto the phoneme hypotheses whenever a word boundary is hypothesized:

$$p(h_m|h_{m-1}) = \begin{cases} 1 & \text{if } w(h_m) = w(h_{m-1}) \\ p(w|v) & \text{if } w(h_m) = w \text{ and } w(h_{m-1}) = v. \end{cases} \tag{4.96}$$

Note that in the first case, h_m and h_{m-1} are part of the same word hypothesis w while in the second case, h_m and h_{m-1} belong to different word hypotheses, even if w is equal to v.

Assuming that the gain function $G(W, W_r)$ can be decomposed into a sum of local costs:

$$G(W, W_r) = \sum_{h \in W} \ell(h, W_r), \tag{4.97}$$

the local MPE and MWE criterion (as well as any other criterion with a local cost function defined on a word or phoneme level) can be re-written as follows:

$$\mathcal{F}_r(\theta) = \frac{\sum_W p^\kappa(W) \cdot p_\theta^\kappa(X_r|W) \cdot \mathcal{G}(W, W_r)}{\sum_{W \in \mathcal{M}_r} p^\kappa(W) \cdot p_\theta^\kappa(X_r|W)} \tag{4.98}$$

$$= \frac{\sum_{h_1^M} p^\kappa(h_1^M) \cdot p_\theta^\kappa(X_r|h_1^M) \cdot \sum_{m=1}^{M} \ell(h_m, W_r)}{\sum_{h_1^M \in \mathcal{M}_r} p^\kappa(h_1^M) \cdot p_\theta^\kappa(X_r|h_1^M)}. \tag{4.99}$$

In principal, the above stated construction is not limited to phoneme or word sequences, but can be applied to sentence hypotheses as well. In this case, the sequence h_1^M becomes identical with the sentence hypothesis W, that is, $M = 1$ and $h_1 = W$. As a consequence, sentence-based training criteria such as the MMI and MCE criterion are captured by this formulation as well. Since the gain function G must account for the level of locality of a hypothesis h, sentence-based training criteria typically apply the *Kronecker* delta $\delta(W, W_r)$ as cost function whereas word and phoneme-based criteria such as the MWE and MPE criterion apply approximations of the word or phoneme accuracy (cf. Table 4.1).

Under the assumption that the phoneme boundary times are given for each hypothesis in the sequence h_1^M, the derivative of the acoustic probability with respect to θ_{sl} yields:

$$\frac{\partial p_\theta^\kappa(X_r|h_1^M)}{\partial \theta_{sl}} = \kappa \cdot p_\theta^\kappa(X_r|h_1^M) \cdot \frac{\partial}{\partial \theta_{sl}} \log p_\theta(X_r|h_1^M) \tag{4.100}$$

$$= \kappa \cdot p_\theta^\kappa(X_r|h_1^M) \cdot \sum_{m=1}^{M} \frac{\partial}{\partial \theta_{sl}} \log p_\theta(x_r{}_{\tau_m}^{t_m}|h_m). \tag{4.101}$$

Hence, the formal derivative of $\mathcal{F}$ with respect to θ_{sl} gives:

$$\frac{\partial \mathcal{F}}{\partial \theta_{sl}} = \frac{1}{R} \sum_{r=1}^{R} f'\big(\log \mathcal{F}_r\big) \cdot \frac{1}{\mathcal{F}_r} \cdot \frac{\partial \mathcal{F}_r}{\partial \theta_{sl}}, \tag{4.102}$$

where the partial derivative of the local criterion $\mathcal{F}_r$ with respect to θ_{sl} is given by:

$$\frac{\partial \mathcal{F}_r}{\partial \theta_{sl}} = \kappa \cdot \sum_h \Bigg(\frac{1}{p_\theta(h|X_r)} \cdot \sum_{h_1^M : \exists m \; h_m = h} p_\theta(h_1^M|X_r) \cdot \sum_{m=1}^{M} \ell(h_m, W_r) - \sum_{h_1^M} p_\theta(h_1^M|X_r) \cdot \sum_{m=1}^{M} \ell(h_m, |W_r) \Bigg) \cdot p_\theta(h|X_r) \cdot \cdot \sum_{t=\tau(h)}^{t(h)} p_\theta(s_t{=}s|x_r{}_{\tau(h)}^{t(h)}, h) \cdot \frac{\partial}{\partial \theta_{sl}} \log p(x_{rt}|\theta_s). \tag{4.103}$$

Here, the first term in the brackets can be interpreted as the expectation value of the local costs:

$$c_r(h) \coloneqq \frac{\sum_{h_1^M : \exists m \; h_m = h} p_\theta(h_1^M|X_r) \cdot \sum_{m=1}^{M} \ell(h_m, W_r)}{p_\theta(h|X_r)}. \tag{4.104}$$

Table 4.2: Smoothing functions and their first-order derivatives.

criterion	smoothing function $f(z)$	first order derivative $f'(z)$
ML	z	1
MMI	z	1
CT		
MCE	$-\frac{1}{1+e^{2\rho z}}$	$\frac{\rho}{2}\cdot\left[1-\tanh^2(\rho z)\right]$
FT		
Diversity Index	$\frac{1}{\rho}(e^{\rho z}-1)$	$e^{\rho z}-\frac{1}{\rho}$
Jeffreys	$\log\frac{e^z}{1-e^{\bar{z}}}$	$\frac{1}{1-e^{\bar{z}}}$
Chernoff	$\frac{e^{\rho z}}{\left[1-e^{z}\right]^{1-\rho}}$	$\left[\rho+(1-\rho)\cdot\frac{e^z}{1-e^{\bar{z}}}\right]\cdot f(z)$
MWE/MPE	e^z	e^z

The second term in the brackets is the local discriminative criterion itself:

$$\mathcal{F}_r(\theta)=\sum_{h_1^M} p_\theta(h_1^M|X_r)\cdot\sum_{m=1}^{M}\ell(h_m,W_r)\equiv\bar{c}_r. \tag{4.105}$$

Hence, with the following definition:

$$\eta_r(h;\theta)\coloneqq\frac{c_r(h)}{\mathcal{F}_r(\theta)}, \tag{4.106}$$

the partial derivative of the discriminative criterion $\mathcal{F}$ with respect to the parameters of the acoustic model θ yields:

$$\frac{\partial\mathcal{F}_r}{\partial\theta_{sl}}=\kappa\cdot\sum_h\Big(c_r(h)\;-\;\mathcal{F}_r(\theta)\Big)\cdot p_\theta(h|X_r)\cdot\sum_{t=\tau(h)}^{t(h)}p_\theta(s_t{=}s|x_{r\,\tau(h)}^{\;\;t(h)},h)\cdot\frac{\partial}{\partial\theta_{sl}}\log p(x_{rt}|\theta_s). \tag{4.107}$$

The discriminative averages need to be redefined in a similar way. To accomplish this, we rewrite the numerator and the denominator statistics by incorporating an additional binary variable $b\in\{0,1\}$ that indicates whether the gain function takes only the spoken word sequence ($b=0$) or *all* possible sentence hypotheses ($b=1$) into account.

$$\begin{aligned}\Gamma_{sl,b}^{\mathsf{num}}\big(g(\mathcal{X})\big)\coloneqq\frac{\kappa}{R}\sum_{r=1}^{R}f'\big(\log\mathcal{F}_r(\theta)\big)\cdot\sum_h\max\big\{0,\;\eta_r(h;\theta)-\delta_{b,1}\big\}\cdot\\ \cdot p_\theta(h|X_r)\cdot\sum_{t=\tau(h)}^{t(h)}\gamma_{rt}(s,l|h)\cdot g(x_{rt})\end{aligned} \tag{4.108}$$

$$\begin{aligned}\Gamma_{sl,b}^{\mathsf{den}}\big(g(\mathcal{X})\big)\coloneqq\frac{\kappa}{R}\sum_{r=1}^{R}f'\big(\log\mathcal{F}_r(\theta)\big)\cdot\sum_h\max\big\{0,\;1-\delta_{b,1}\cdot\eta_r(h;\theta)\big\}\cdot\\ \cdot p_\theta(h|X_r)\cdot\sum_{t=\tau(h)}^{t(h)}\gamma_{rt}(s,l|h)\cdot g(x_{rt})\end{aligned} \tag{4.109}$$

Due to the following identity:

$$\max\left\{0,\, \eta_r(h;\theta) - \delta_{b,1}\right\} \;-\; \max\left\{0,\, 1 - \delta_{b,1}\cdot\eta_r(h;\theta)\right\} \;=\; \eta_r(h;\theta) - 1, \tag{4.110}$$

the variable b cancels in the difference between the numerator and the denominator statistics, and the discriminative averages can thus be written as:

$$\Gamma_{sl}\big(g(\mathcal{X})\big) \coloneqq \frac{\kappa}{R}\sum_{r=1}^{R} f'\big(\log\mathcal{F}_r(\theta)\big)\cdot\sum_{h}\big[\eta_r(h;\theta)-1\big]\cdot\cdot p_\theta(h|X_r)\cdot\sum_{t=\tau(h)}^{t(h)}\gamma_{rt}(s,l|h)\cdot g(x_{rt}). \tag{4.111}$$

To finish the optimization in the extended unifying approach, it remains to redefine the auxiliary function which is now given by:

$$\begin{aligned}\mathscr{S}_K(\bar\theta|\theta) = \sum_{k\in K}\Bigg(&\frac{\kappa}{R}\sum_{r=1}^{R} f'\big(\log\mathcal{F}_r(\theta)\big)\cdot\sum_{h}\big[\eta_r(h;\theta)-1\big]\cdot\\ &\cdot p_\theta(h|X_r)\cdot\sum_{t=\tau(h)}^{t(h)}\sum_{\sigma=1}^{S}\sum_{(s,\cdot)\in k}\gamma_{rt}(\sigma,s|h)\cdot\log p_{\bar\theta}(x_{rt},s|l)\\ &+ D_k\cdot\sum_{(s,l)\in k} c_{sl}\cdot\int p(x|\theta_{sl})\cdot\log p(x|\bar\theta_{sl})\,dx\\ &+ C_s\sum_{l=1}^{L_s} c_{sl}\cdot\log\bar c_{sl} + \sum_{\sigma=1}^{S} E_\sigma\cdot p(s|\sigma)\cdot\log\bar p(s|\sigma)\Bigg).\end{aligned} \tag{4.112}$$

Note that the functional form of the re-estimation equations for the parameters of the acoustic model does not change in the extended unifying approach. Therefore, the update-rules defined in Eqs. (4.62–4.67) remain valid. The same holds for the choice of the iteration constants as discussed in Section 4.3.

4.5 Growth Transformations and Auxiliary Functions

For many optimization problems, a direct maximization of a given objective function is often not possible since non-linear dependencies on the function parameters prevent the derivation of easy-to-use update equations from its gradient. In many cases, the objective function can then be replaced with an *auxiliary function* which is easier to optimize and whose maximization implies an increase in the original objective function. This section derives some general rules for constructing auxiliary functions that can be used to optimize discriminative objective functions. These rules motivated the auxiliary function used for optimizing the MMI objective function with tied covariance matrices in the *extended unifying approach* (cf. Section 4.4).

Suppose we are given an objective function $\mathcal{F}(\theta)$ that depends on a parameter set θ. We further assume that the gradient of $\mathcal{F}$ with respect to θ exists and can be decomposed into a product of two functions f and g:

$$\frac{\partial\mathcal{F}(\theta)}{\partial\theta} = f(\theta)\cdot\frac{\partial}{\partial\theta}\,g(\theta), \tag{4.113}$$

where $\partial g(\theta)/\partial\theta$ is supposed to have an analytically easy to handle dependency on θ as, for instance, linear or quadratic in θ, while all other dependencies on θ that are difficult to separate shall be

subsumed by $f(\theta)$. A new function $\mathcal{G}$ depending on two parameter sets $\bar{\theta}$ and θ can then be defined as follows:

$$\mathcal{G}(\bar{\theta}|\theta) = f(\theta) \cdot g(\bar{\theta}). \tag{4.114}$$

An important property of $\mathcal{G}$ is that its gradient with respect to $\bar{\theta}$ is equal to the gradient of $\mathcal{F}$ at $\bar{\theta} = \theta$:

$$\left.\frac{\partial \mathcal{G}(\bar{\theta}|\theta)}{\partial \bar{\theta}}\right|_{\bar{\theta}=\theta} = \left.\frac{\partial \mathcal{F}(\bar{\theta})}{\partial \bar{\theta}}\right|_{\bar{\theta}=\theta}. \tag{4.115}$$

If $\mathcal{G}(\bar{\theta}|\theta)$ is convex in $\bar{\theta}$ for fixed θ, then Eq. (4.114) can directly be used as an auxiliary function to maximize $\mathcal{F}(\theta)$. This holds for the $\mathcal{Q}$ function in the EM algorithm (cf. Eq. (2.38)). Functions which do not have this property can often be made convex by adding a smoothing function h scaled with a constant D (*regularization*):

$$\mathcal{G}(\bar{\theta}|\theta) = f(\theta) \cdot g(\bar{\theta}). \; + \; D \cdot h(\bar{\theta}|\theta). \tag{4.116}$$

Here, h and D are to be chosen such that the following constraints are satisfied:

- The resulting function $\mathcal{G}(\bar{\theta}|\theta)$ is (locally) convex in $\bar{\theta}$ for fixed θ:

$$f(\theta) \cdot \frac{\partial}{\partial^2 \bar{\theta}}\, g(\bar{\theta}) \; + \; D \cdot \left.\frac{\partial h(\bar{\theta}|\theta)}{\partial^2 \bar{\theta}}\right|_{\bar{\theta}=\theta} \geqslant 0 \tag{4.117}$$

- The gradient of h with respect to $\bar{\theta}$ vanishes at $\bar{\theta} = \theta$, i.e.:

$$\left.\frac{\partial h(\bar{\theta}|\theta)}{\partial \bar{\theta}}\right|_{\bar{\theta}=\theta} = 0. \tag{4.118}$$

If both conditions hold, Eq. (4.116) is called an auxiliary function for $\mathcal{F}(\theta)$. For the special case that the parameters to be optimized are subject to a sum-to-one constraint, the restrictions given by Eq. (4.118) are not required and the objective function can directly be maximized using, e.g., the EB algorithm.

4.6 Summary

In this chapter, an extended unifying approach for a class of discriminative training criteria was introduced that includes the Maximum Mutual Information criterion, the Minimum Classification Error criterion, the Minimum Word Error criterion, the Minimum Phone Error criterion, the Diversity Index, the Jeffreys criterion, and the Chernoff criterion. Several criteria were discussed in detail, and refined upper bounds to the Bayes classification error rate were presented by sampling the admissible region using Monte Carlo experiments. A mathematical framework was presented that allows for handling the parameter optimization problem in the case that Gaussian mixture densities with arbitrary tying schemes and full covariance matrices are used. Re-estimation equations and iteration constants were derived for the MMI criterion and then generalized to all criteria covered by the extended unifying approach. The problem of choosing the iteration constants for the case that full covariance matrices are used was formulated as the problem of finding the eigenvalues of a non-symmetric quadratic eigenvalue problem. Two novel methods on setting the iteration constants were suggested, one of which taking additional Maximum Likelihood statistics into account to even out the different ranges of normalized (means) and unnormalized (discriminative averages) quantities in the re-estimation equations. A discussion on growth transformations and a canonical way to construct auxiliary functions for discriminative objective functions concluded this chapter.

Chapter 5

Estimating Discriminative Statistics

In discriminative training, each training utterance W_r is discriminated against a set $\mathcal{M}_r$ of competing hypotheses. Although ideally this set should contain all possible word sequences, in practice, $\mathcal{M}_r$ is usually restricted to sentence alternatives explored in a previous recognition pass. This approximation can be justified by the fact that for given model parameters only a small fraction of the probability mass accounts for sentences that were pruned or have not been hypothesized during recognition.

Alternative sentence hypotheses can be represented via N-best lists or word lattices. Due to their simple structure, N-best lists provide an easy means to estimate posterior probabilities or to determine the expected loss under an error metric such as the *Levenshtein* distance which is difficult to compute on word lattices. However, adjacent candidates in N-best lists often differ only by a few words or phonemes which makes the representation highly redundant, and even for very large N only a small fraction of the hypotheses generated during search can be represented. Other than N-best lists, word lattices capture a significantly larger amount of sentence hypotheses which exceeds the number of candidates encoded in an N-best list by several orders of magnitude. Since word lattices represent a larger fraction of the search space, they are preferably used in discriminative training.

This chapter gives a detailed description of the word-lattice based estimation of discriminative statistics. Conceptually, the word lattices used in this thesis are generated from a modified search algorithm which extends the concept of word-conditioned tree copies to *networks of HMMs* [Macherey & Ney 02]. HMM networks are *finite state machines* (FSMs) which can be statically compiled and optimized on the acoustic model level. Thus they provide an easy and flexible means to incorporate special acoustic models as, for example, across-word models and complex Markov topologies into the search network without the need of specializing the implementation of the search or changing the hypotheses management.

While FSMs compile the language model as a special automaton into the network, HMM networks as used in this thesis keep the acoustic model strictly separated from the language model. Except for the language model look-ahead pruning (cf. Section 5.1.5), the word hypotheses are scored with their language model probabilities if and only if a word end is hypothesized. This has two advantages: first, the memory usage is significantly smaller compared to a language model that is represented as a finite state machine and compiled into the acoustic network by composing two FSMs; second, the language model can be implemented as an external service which returns language model probabilities for requested m-grams[1].

Depending on the structure of the HMM network, the search space can be unconstrained as required by the denominator model or it may be restricted to the set of sentence hypotheses that are alignments of different pronunciations of the spoken word sequence as needed by the numerator model.

[1] To keep the network traffic low, LM requests can be batched and sent to the LM server once every 10ms. Thus, the LM recombination remains fast enough to be applicable to online applications where it is essential that response times are short. If the size of the model is too large to fit into the memory of a single machine, the language model service can be distributed over multiple machines [Brants & Popat$^+$ 07].

The remainder of this chapter is organized as follows: Section 5.1 describes the construction of the numerator and the denominator lattices. Section 5.2 presents a forward-backward (FB) algorithm for estimating word probabilities on word lattices. Section 5.3 illustrates how word and phoneme accuracies can be estimated on lattices using a modified FB algorithm. Section 5.4 presents a novel algorithm that allows for optimizing the MCE objective function on word lattices. Techniques to accelerate the training process presented in Section 5.5 conclude this chapter.

5.1 Modeling Correct and Competing Classes

In lattice-based discriminative training it is important that the numerator and the denominator lattices are built consistently, that is, an implementation must ensure that corresponding hypotheses which occur in either lattice within the same temporal and linguistic context are equal with respect to their local scores and segmentation points. If this were not the case, the discriminative averages would impose a further direction onto the gradient of the objective function even when the criterion is already in a local optimum. To make the construction of the numerator and the denominator lattices consistent, the lexical prefix tree used in the global search is replaced with an utterance-specific HMM network. This has the advantage that a single implementation of the search algorithm suffices to compute the forced alignment of the spoken word sequence ("numerator" model) or to perform an unconstrained search ("denominator model").

5.1.1 HMM Networks for the Correct Model

The construction of numerator lattices is based on HMM networks that are specific for each training utterance. Given a word sequence W_r in training, the HMM network for the numerator model needs to encode *all* combinations of pronunciation variants as provided by the dictionary. Similar to the denominator lattices, numerator lattices may contain alternative alignments of the same word sequence. This is an important difference compared to the implementation used e.g. in [Schlüter 00] where numerator lattices were constrained to contain only the best alignment path of the spoken word sequence in a single pronunciation variant. The reason why numerator networks require this additional level of complexity is that denominator lattices may contain multiple alignments for various pronunciations of the spoken word sequence. Thus, if the corresponding paths were missing in the numerator lattice, statistics drawn from alternative alignments would be discriminated against the numerator model even though they represent the correct class.

To avoid this problem, all pronunciation variants of the spoken word sequence are compiled into the HMM network for the numerator model. At word boundaries, coarticulated as well as not-coarticulated phoneme models are compiled into the network to account for across word and within word contexts. Word end hypotheses are inserted into the lattice *before* they are recombined, which does not only increase the number of alternative paths that otherwise would be discarded due to the word-pair approximation [Ney & Aubert 94], but also allows for representing the same word simultaneously in a coarticulated as well as in a non-coarticulated context.

HMM networks for the numerator model are optimized on the acoustic model level. To accomplish this, state sequences with identical model indices are combined provided that the merged states do not introduce new paths into the network. Figure 5.1 depicts an example of an HMM network for the word sequence "*he is*". For simplicity, the pronunciation variants of the word "*he*" are assumed to be "*h e*" and "*h i*", respectively. For the word "*is*", the constituent phoneme sequences are assumed to be "*i s*" and "*ih s*". Word end nodes in the HMM network are labeled with their language model histories together with the sentence position at which the respective m-gram occurs. The information about the sentence position is important to prevent that instances of the same m-gram will be recombined if they refer to different sentence positions in the spoken word sequence.

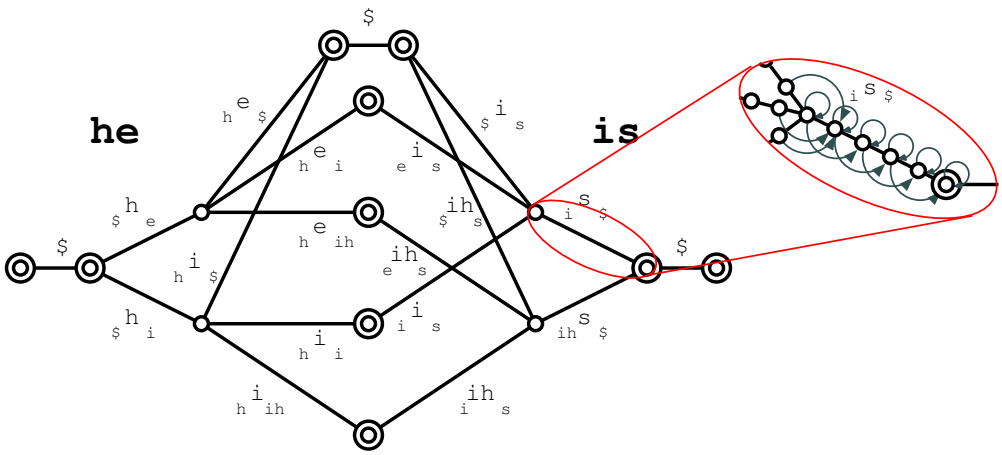


Figure 5.1: Hidden Markov model network for the numerator model to constrain the search space onto the word sequence "he is". For simplicity, the pronunciation variants of the word "*he*" are assumed to be "*h e*" and "*h i*", respectively. For the word "*is*", the constituent phoneme sequences are assumed to be "*i s*" and "*ih s*". Double circles in the network indicate word boundaries.

5.1.2 HMM Networks for the Competing Model

Sentence hypotheses for the competing model are encoded as word lattices that are produced during an unconstrained recognition. The underlying HMM network is based on a lexical prefix tree whose structure is extended with multi-word phrases plus additional arcs and nodes that account for across-word phoneme contexts at word boundaries. During construction, the network is optimized by merging adjacent nodes with identical model indices. The resulting network is minimal with respect to the number of arcs and transitions, and thus induces a very small search space. If across word models are used, the HMM network may have multiple source nodes. The number of additional source nodes (fan-in arcs) corresponds with the number of central phonemes that occur in the last phoneme generation of words contained in the pronunciation lexicon. Similarly, the number of final states (fan-out arcs) at word ends is equal to the number of different central phonemes in the first phoneme generation. Figure 5.2 depicts the structure of an HMM network for a three-word vocabulary. The pause model is denoted by '$'. To reduce the memory usage during search, the HMM network is embedded into a simplified data structure which contains for each node only the model index of the Gaussian mixture distribution and a list of reachable successor nodes. If a node represents a word boundary, it stores in addition the respective word index. To reduce the number of cache misses during search, nodes in the simplified data structure are linearly arranged in memory by ordering them according to a breadth-first-search traversal of the HMM network.

5.1.3 Sizes of HMM Networks for Different Speech Corpora

Table 5.1 shows sizes of HMM networks in terms of number of arcs and nodes for different speech corpora[2] using either whole word models (SieTill) or within word and across word models (Wsj-5k, Nab-20k, Nab-65k, Hub-4). Note that in case of across-word models, the respective HMM network contains the complete set of fan-in and fan-out arcs.

[2] A thorough description of the speech corpora is given in Section 6.2.

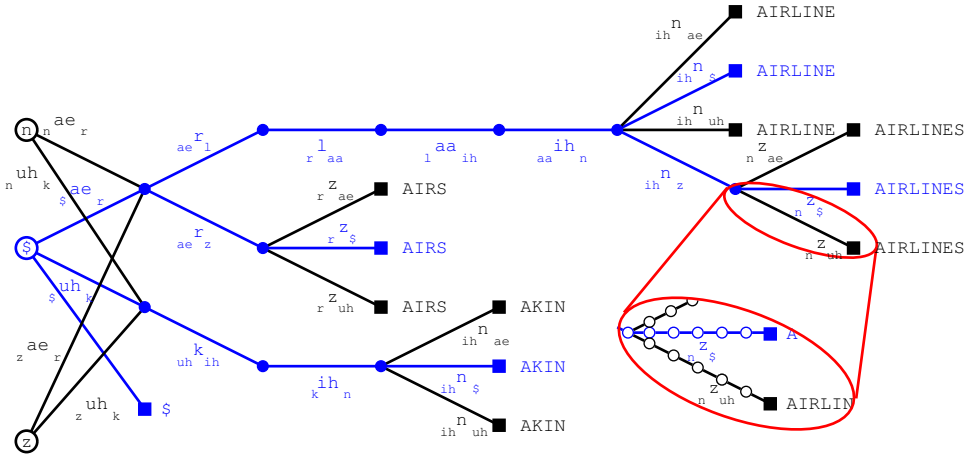


Figure 5.2: Structure of an HMM network for the competing model built from a lexical prefix tree (blue). The network has been extended with additional arcs (black) in order to account for across-word phoneme models at word boundaries. The source node tagged with the '$' symbol induces the lexical prefix tree for words with not-coarticulated left across-word contexts. All other source nodes induce prefix trees for word hypotheses with coarticulated left across-word contexts.

Table 5.1: Sizes of HMM networks for different speech corpora.

corpus	set	voc. size	# lemmata (end nodes)	word boundary model	# links	# states	# LM-LA nodes
SieTill	train	11	12	whole word	429	430	–
	test	11	12	whole word	429	430	–
Wsj-5k	train	10 135	10 770	within word	181 178	181 179	18 498
	test	5 009	5 656	within word	97 332	97 333	9 710
Nab-20k	train	15 015	16 990	within word	287 914	287 915	29 009
			(730 528)	across word	2 585 506	2 502 990	28 058
	test	19 980	22 412	within word	358 864	358 863	38 253
			(3 008 668)	across word	9 122 081	9 011 095	113 981
Nab-65k	train	65 431	70 751	within word	1 068 335	1 068 336	119 525
			(3 113 001)	across word	9 408 189	9 291 058	115 062
	test	64 738	69 970	within word	1 059 624	1 059 625	118 250
			(3 008 668)	across word	9 122 081	9 011 095	113 981
Hub-4	train	69 621	85 246	within word	1 270 745	1 269 003	144 128
			(3 113 001)	across word	9 408 189	9 291 058	115 062
	test	66 843	81 932	within word	1 234 909	1 233 167	138 858
			(3 008 668)	across word	9 122 081	9 011 095	113 981

5.1.4 Organization of the Search

The basic search strategy for decoding the spoken word sequence is based on the one-pass algorithm described in [Vintsyuk 71, Bridle & Brown+ 82, Ney 84, Ney & Aubert 96]. The objective of the search is to find that word sequence which maximizes the a-posteriori probability :

$$[w_1^N]_{\text{opt}} = \underset{w_1^N, N}{\text{argmax}} \left\{ p(w_1^N | x_1^T) \right\} \tag{5.1}$$

$$= \underset{w_1^N, N}{\text{argmax}} \left\{ p(w_1^N) \cdot \sum_{[s_1^T] | w_1^N} p(x_1^T, s_1^T \mid w_1^N) \right\} \tag{5.2}$$

$$\cong \underset{w_1^N, N}{\text{argmax}} \left\{ p(w_1^N) \cdot \max_{[s_1^T] | w_1^N} \left\{ p(x_1^T, s_1^T | w_1^N) \right\} \right\} \tag{5.3}$$

Taking across-word contexts into account, the HMM state sequence induced by a word hypothesis does not only depend on the word itself but also on its left and right across-word context. These contexts are either given by the neighboring words for a coarticulated word transition or set to '$' in case that a not-coarticulated word transition is hypothesized. The additional dependency is accounted for by introducing a corresponding sequence of across-word contexts $[l, r]_1^N$ for each word sequence w_1^N [Sixtus 03, p. 38]:

$$p(x_1^T | w_1^N) \cong \max_{s_1^T} \left\{ p(x_1^T, s_1^T \mid w_1^N) \right\} \tag{5.4}$$

$$= \max_{s_1^T} \left\{ \sum_{[l,r]_1^N} p(x_1^T, s_1^T, [l, r]_1^N \mid w_1^N) \right\} \tag{5.5}$$

$$\cong \max_{s_1^T, [l,r]_1^N} \left\{ p([l, r]_1^N \mid w_1^N) \cdot p(x_1^T, s_1^T \mid [l, w, r]_1^N) \right\} \tag{5.6}$$

Usually, the probability distribution $p([l, r]_1^N \mid w_1^N)$ is not trained but approximated as follows:

$$p([l, r]_1^N \mid w_1^N) = \begin{cases} 1 & \text{if } [l, r]_1^N \text{ and } w_1^N \text{ are consistent,} \\ 0 & \text{otherwise.} \end{cases} \tag{5.7}$$

The sequence $[l, r]_1^N$ is called *consistent* if it contains for each word pair in w_1^N either a coarticulated or a not-coarticulated word transition[3]. To simplify the notation, the term $p([l, r]_1^N \mid w_1^N)$ and the optimization over $[l, r]_1^N$ are omitted in the following. The emission probability $p(x_1^T, s_1^T \mid [l, w, r]_1^N)$ can be further decomposed into the contributions of the individual word hypotheses:

$$\max_{s_1^T} p(x_1^T, s_1^T \mid [l, w, r]_1^N) \tag{5.8}$$

$$\cong \max_{s_1^T} \prod_{t=1}^{T} \left\{ p(x_t \mid s_t, [l, w, r]_1^N) \cdot p(s_t \mid s_{t-1}, [l, w, r]_1^N) \right\} \tag{5.9}$$

$$= \max_{t_1^N} \prod_{n=1}^{N} \underbrace{\left\{ \max_{s_{t_{n-1}+1}^{t_n}} \prod_{t=t_{n-1}+1}^{t_n} \left\{ p(x_t \mid s_t, [l, w, r]_n) \cdot p(s_t \mid s_{t-1}, [l, w, r]_n) \right\} \right\}}_{=: h([l, w, r]_n;\, t_{n-1}, t_n)} \tag{5.10}$$

$$= \max_{t_1^N} \prod_{n=1}^{N} h([l, w, r]_n;\, t_{n-1}, t_n). \tag{5.11}$$

[3] The search network is assumed to be structured such that inconsistent across word contexts cannot be hypothesized.

Here, $t_0 = 0$ and $t_N = T$. The quantity $h([l, w, r]_n; t_{n-1}, t_n)$ denotes the probability that the word hypothesis w_n in the across word context $[l, r]_n$ emits the sequence of acoustic observations $x_{t_{n-1}+1}, ..., x_{t_n}$ It is needed to define the local acoustic scores of arc hypotheses in word lattices (cf. Section 5.1.6).

Eq. (5.3) is evaluated by performing a time-synchronous search on the HMM network. Since each source node of the underlying HMM network induces a lexical prefix tree, the concept of word conditioned tree search remains applicable. Assuming a trigram language model, the search can be described through the following two quantities [Ortmanns & Ney$^+$ 97]:

$Q_{uv}(t, s)$ $\coloneqq$ Probability of the best partial path which ends at time frame t in state s in a copy of the lexical prefix tree induced by the HMM network for the language model history (u, v).

$B_{uv}(t, s)$ $\coloneqq$ Time index at which the best partial path ending at time frame t in state s has entered the source node in a copy of the lexical prefix tree induced by the HMM network for the language model history (u, v).

This leads to the following recursion equations for the dynamic programming:

- Recursion within an HMM network (recombination on HMM-state level):

$$Q_{uv}(t, s) = \max_{\sigma} \big\{p(x_t, s \,|\, \sigma) \cdot Q_{uv}(t-1, \sigma)\big\}, \tag{5.12}$$

$$\sigma_{uv}^{\max}(t, s) = \operatorname*{argmax}_{\sigma'} \big\{p(x_t, s|\sigma') \cdot Q_{uv}(t-1, \sigma')\big\}, \tag{5.13}$$

$$B_{uv}(t, s) = B_{uv}(t-1, \sigma_{uv}^{\max}(t, s)). \tag{5.14}$$

- Recursion at word transitions (recombination on word level):

$$Q_{vw}(t, s_0(r)) = \max_{u} \big\{p(w|u, v) \cdot Q_{uv}(t, S_{(w,r)})\big\}, \tag{5.15}$$

$$B_{vw}(t, s_0(r)) = t. \tag{5.16}$$

For each hypothesized right across-word context r, a final state $S_{(w,r)}$ is provided which has a unique corresponding initial state $s_0(r)$ ("*source node*"). Each initial state induces a new successor tree which is conditioned on its left across-word context r and a language model history (v, w).

5.1.5 Language Model Look-Ahead Pruning with HMM Networks

Since each source node in an HMM network induces a lexical prefix tree (cf. Figure 5.2), nodes in the network can uniquely be mapped onto the nodes of a *language model look ahead tree* derived from a regular lexical prefix tree. The language model look-ahead is a pruning technique that alleviates some of the overhead caused by the tree organization of the lexicon [Steinbiss & Tran$^+$ 94, Odell & Valtchev$^+$ 94, Alleva & Huang$^+$ 96, Ortmanns & Ney$^+$ 96a]. For each node in the induced lexical prefix tree, the language model probability of the most likely word reachable from this position is added to the acoustic probability of the corresponding state hypotheses, thus making the language model pruning more effective. To obtain a tree structure from an HMM network, the following sequence of transformation steps is performed: first, all source nodes in the HMM network are collapsed with the source node from which not-coarticulated across-word contexts are started. Similarly, leaf nodes representing coarticulated word ends are merged with the corresponding leaf nodes of not-coarticulated word ends (cf. Figure 5.3). The HMM network is then further processed by means of two graph transformations. The first transformation replaces a diamond with a 3-state chain and maps the index of the erased node onto the index of the middle node (cf. Figure 5.4, left). The second graph operation removes the middle node in a

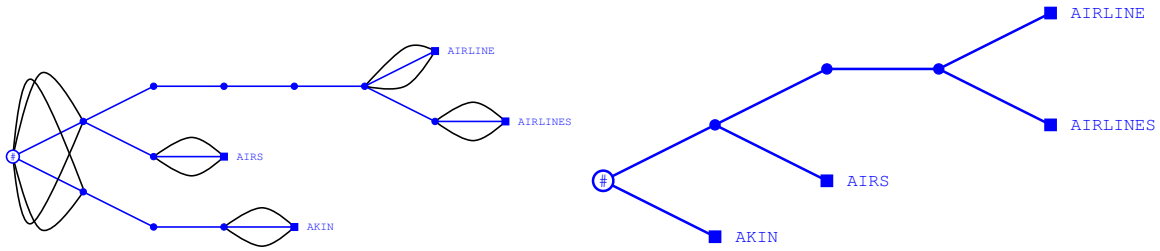


Figure 5.3: Mapping the states of a lexical prefix graph onto the nodes of a language model look ahead tree. In a first step, all graph nodes with an indegree of zero are mapped onto the root node representing the non-coarticulated across-word context. Similarly, each node with an outdegree equal to zero is collapsed with a graph node representing the same word end. Afterwards, all substructures in the graph which are either chains consisting of more than two states or diamonds are replaced with subgraphs as depicted in Figure 5.4. The replacements are carried out until the resulting structure is free of any 3-state chain and diamond.

3-state chain and maps its index onto the index of the bottom node (cf. Figure 5.4, right). Both transformations are repeatedly applied until the resulting graph contains neither a 3-state chain nor a diamond. The graph thus obtained is a tree and allows for factorizing the language model probabilities according to the language model look-ahead pruning. Optionally, higher levels of the look ahead tree can be cut off to make the structure more compact [Ortmanns & Ney$^+$ 96b]. The sequence of node indices onto which deleted states are mapped define a mapping $\mathcal{I}$ which assigns nodes in the original HMM network to nodes in the language model look-ahead tree. To construct this map, $\mathcal{I}$ is initialized with the identity and then refined with each graph transformation by including the next update rule as illustrated in Figure 5.4. The resulting function is an injective mapping of node indices of an HMM network onto the node indices of the language model look ahead tree.

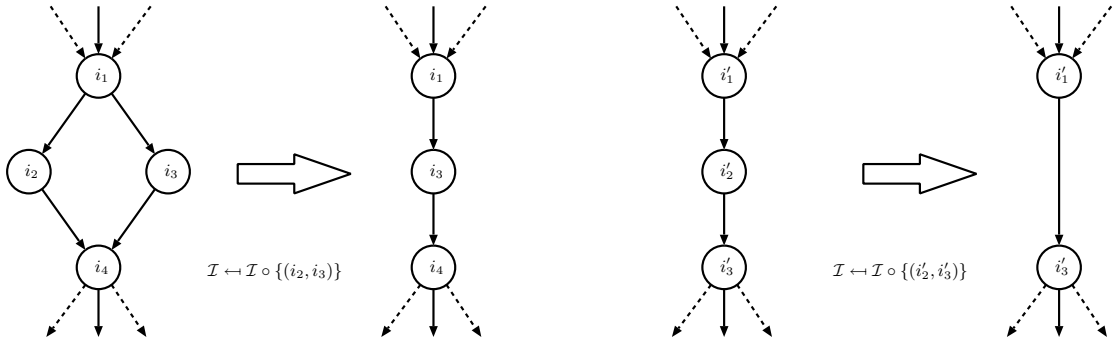


Figure 5.4: Graph transformations to convert a diamond (left) and a 3-state chain (right).

5.1.6 Construction of Word Lattices

A *word lattice* is a weighted, directed, acyclic graph that consists of a set of nodes together with a set of edges linking the nodes. Two special nodes are the source node which has no incoming edges and the sink node which has no outgoing edges. Edges represent word hypotheses and have associated scores or costs. Nodes mark time frames within the acoustic speech signal. Each path through the graph defines a sentence hypothesis. Unlike unconditioned word graphs as used, e.g., in [Sixtus 03, Schlüter 00], lattice nodes defined in this thesis are conditioned on their language model history. This has the advantage that the implementation of the FB algorithm for estimating word posterior probabilities simplifies compared to [Sixtus 03] since language model scores can directly be stored at word graph edges and do not have to be re-computed when performing the FB algorithm.

Assuming that an m-gram language model is used, the following quantity is introduced to construct word graphs using dynamic programming:

$$H(v_2^m, r(v_m), t) \;\coloneqq\; \text{Probability of the best partial sentence hypothesis that ends with the word hypotheses } v_2^m \text{ in the right across-word context } r(v_m) \text{ at time frame } t.$$

The dynamic programming is then based on the following recursion:

$$h([l,w,r];\,\tau,t) \coloneqq \max_{s_{\tau+1}^t} \left\{ p(x_{\tau+1}^t, s_{\tau+1}^t \,|\, [l,w,r]) \;:\; s_{\tau+1} = s_0(l) \wedge s_t = S_{(w,r)} \right\} \tag{5.17}$$

$$\begin{aligned} H(v_2^m, r(v_m); t) \coloneqq \max_{v_1} \Big\{ & p(v_m | v_1^{m-1}) \cdot \max_{\tau} \big\{ H(v_1^{m-1}, r(v_{m-1}); \tau) \cdot \\ & h([l(v_m), v_m, r(v_m)]; \tau, t) \;:\; r(v_{m-1}) = l(v_m) \big\} \Big\} \end{aligned} \tag{5.18}$$

As defined in Eq. (5.10), $h([l,w,r];\,\tau,t)$ denotes the probability that the word hypothesis w in the across word context $[l,r]$ emits the sequence of acoustic observations $x_{\tau+1},\ldots,x_t$. The contribution of a word hypothesis w_n to the joint distribution $p(x_1^t, w_1^n)$ can then be decomposed as follows:

$$\underbrace{x_1, ..., x_\tau}_{H(w_1^{n-1}, r(w_{n-1}); \tau)} \quad \underbrace{x_{\tau+1}, ..., x_t}_{h([l(w_n),\, w_n,\, r(w_n)]; \tau, t)} \quad \underbrace{x_{t+1}, ..., x_T}_{\cdots}. \tag{5.19}$$

Initial experiments showed that word lattices should meet certain requirements in order to be useful in discriminative training. In [Schlüter 00], an optimal word graph was considered to be a lattice which contains each word sequence at most once. If the same word sequence was represented multiple times (differing only with respect to its Viterbi segmentation points) the estimation of the word posterior probabilities could be distorted because a larger amount of the probability mass would distribute over recurring hypotheses than over less frequently represented words. The estimation of the word posterior probabilities might thus be less reliable, and the discriminative training could lead to reduced performance gains.

This problem is, however, primarily caused by word lattices where nodes are not conditioned on their language model context. Although unconditioned word lattices are typically more compact, they also tend to concentrate the probability mass on multiply represented hypotheses that, despite the use of the word pair approximation, may emerge from the presence of redundant silence edges or alternative alignments of the same word sequence [Wessel 02, p. 59]. However, while redundant silence edges may easily be removed through a graph post-processing, redundant sentence hypotheses are difficult to eliminate since in general a constituent word hypothesis does not belong to a single sentence hypothesis but may be part of several other sentence hypotheses, too. Therefore, excluding a word hypothesis from a path requires a reorganization of the whole lattice, which, in general, cannot be done efficiently. If lattice nodes are instead conditioned on their language model history, redundant arcs do not occur and the problem does not arise.

Formal Structure of Word Lattices

A word lattice $G = (\mathcal{V}, \mathcal{E})$ is a directed, acyclic, weighted graph that consists of a set of nodes $\mathcal{V}$ and a set of edges $\mathcal{E}$ connecting the nodes. Each node $n \in \mathcal{V}$ is a triple:

$$n = (v_1^{m-1}, \varphi, t) \in \mathcal{V} \subseteq \mathcal{W}^* \times \Phi \times \mathbb{N}, \tag{5.20}$$

where v_1^{m-1} refers to an $(m-1)$-gram language model history, φ denotes an across-word phoneme context, and t marks a time index. An edge $q \in \mathcal{E} \subseteq [\Phi \times \mathcal{W} \times \Phi] \times \mathbb{N}^2 \times \mathcal{V}^2 \times \mathbb{R}^3 \times \{0,1\}$ in the word graph is defined as an 11-tuple:

$$q \coloneqq \big([l, v_m, r], \tau, t, n_b, n_e, p(v_m|v_1^{m-1}), h_q, H_q, b\big). \tag{5.21}$$

Here, $[l, v_m, r]$ denotes a word hypotheses v_m within the left across-word context l and the right across-word context r; τ and t denote the time boundaries in which $[l, v_m, r]$ is hypothesized; n_b and n_e designate the begin node and the end node linked by q; $p(v_m|v_1^{m-1})$, h_q, and H_q refer to the language model probability $p(v_m|v_1^{m-1})$, the acoustic score $h_q \coloneqq h([l, v_m, r]; \tau, t)$, and the probability of the best partial sentence hypothesis $H_q \coloneqq H(v_2^m, r, t)$ ending in arc q; and b indicates whether the word hypothesis associated with arc q matches the spoken word for this time interval ($b = 1$) or not ($b = 0$). If no information on the spoken word sequence is available, b defaults to 0. The 11-tuple is implemented as a structure comprising the following components:

$$\begin{aligned}
\texttt{xwRight} &:= r(v_m) \\
\texttt{xwLeft} &:= l(v_m) \\
\texttt{wrdIdx} &:= v_m \\
\texttt{begTim} &:= \tau + 1 \\
\texttt{endTim} &:= t \\
\texttt{fwdNod} &:= \Phi(v_1^{m-1}, l(v_m), \tau + 1) \\
\texttt{bwdNod} &:= \Psi(v_2^m, r(v_m), t) \\
\texttt{lanSco} &:= -\log p(v_m|v_1^{m-1}) \\
\texttt{acuSco} &:= -\log h([l(v_m), v_m, r(v_m)]; \tau, t) \\
\texttt{scoMin} &:= -\log H(v_2^m, r(v_m); t) \\
\texttt{correct} &:= \delta(w_r, v_m)
\end{aligned}$$

Here, Φ and Ψ map the begin and end nodes of the word lattice onto unique node indices. The lattice structure is used to encode numerator as well as denominator lattices. Figure 5.5 illustrates a simplified example of a numerator lattice containing the spoken word sequence together with a corresponding denominator lattice. To ensure that the denominator lattice contains all sentence hypotheses represented by the numerator lattice, the numerator lattice is merged into the denominator lattice as depicted in Figure 5.6 before it is used in training. Numerator and denominator lattices may contain alternative alignments of the same word sequence.

5.2 Forward-Backward Probabilities on Word Lattices

The FB algorithm on word lattices allows for estimating the generalized FB probabilities, $\gamma_{rt}(s)$, based on word hypotheses encoded in a lattice. Taking a weighting exponent $\kappa \in (0, \infty)$ into account, the generalized FB probabilities are given by:

$$\gamma_{rt}(s) = \sum_{W \in \mathcal{M}_r} \frac{p_\theta^\kappa(W, X_r)}{\sum_{V \in \mathcal{M}_r} p_\theta^\kappa(V, X_r)} \cdot \gamma_{rt}(s|W) \tag{5.22}$$

$$\underset{(\kappa=1)}{=} \sum_{W \in \mathcal{M}_r} p_\theta(W|X_r) \cdot \gamma_{rt}(s|W). \tag{5.23}$$

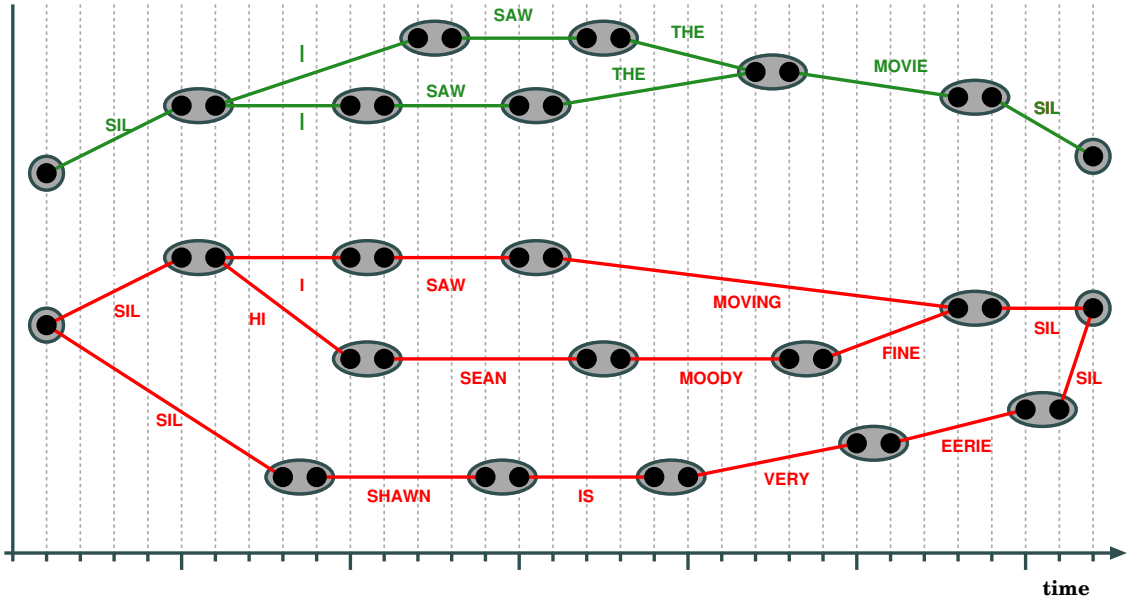


Figure 5.5: Word lattices for the numerator (green) and the denominator model (red) before being merged. The numerator lattice is constrained through an HMM network to only contain alignments of the spoken word string, whereas the denominator lattice is obtained from an unconstrained search.

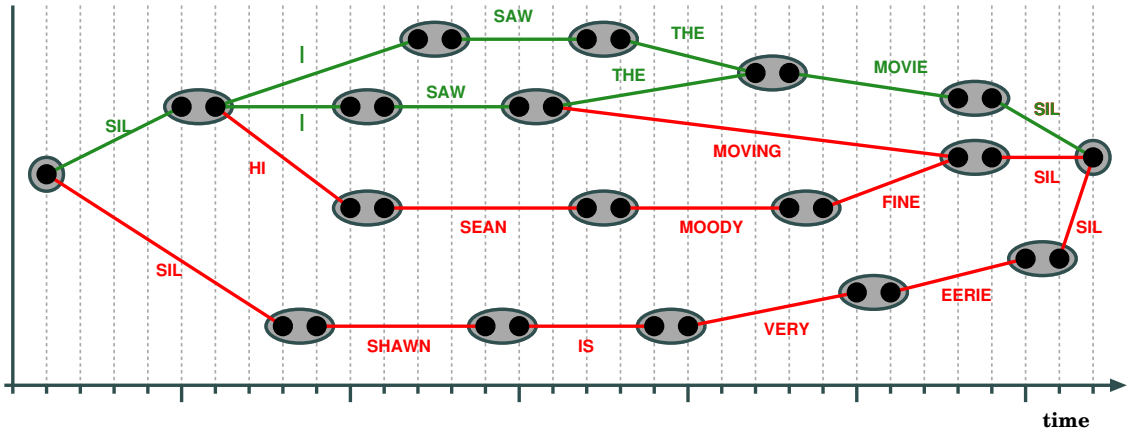


Figure 5.6: Denominator lattice after being merged with the numerator lattice. Words marked red are tagged as wrong, while words marked green are tagged as correct.

Due to the Viterbi approximation, each sentence hypothesis represented in a word lattice has unique segmentation points. Hence, the time alignment can be recomputed for each word hypothesis without taking surrounding word hypotheses into account. Let $Q = q_1, ..., q_{n-1}, q, q_{n+1}, ..., q_N$ denote a path in a word lattice with the corresponding word sequence $W_f = w(q_1), ..., w(q_{n-1})$, $w = w(q)$, and $W_b = w(q_{n+1}), ..., w(q_N)$, that is, W_f is defined as the sequence of word labels of a partial path starting at the source node and leading to arc q, and W_b is defined as the sequence of

word labels along a partial path starting after arc q and ending in the sink node of the lattice. Then the sum over all word sequences in Eq. 5.22 can be decomposed into a sum over partial sentence hypotheses $W = W_f \circ w \circ W_b$. Note that the lattice structure guarantees the consistency of all word sequences $W_f \circ w \circ W_b$, that is, the right across-word context for W_f matches the left across-word context of w, and the right across-word context of w matches the left across-word context for W_b. The same holds for the language model context. To simplify the notation, across-word contexts are omitted in the following. With $p_\theta([w,\tau,t]|X_r)$ denoting the probability that w is hypothesized within the time interval $[\tau+1,t]$ given the sequence of acoustic observations X_r:

$$p_\theta([w,\tau,t]|X_r) \coloneqq \sum_{W_f,W_b} \frac{p_\theta^\kappa(W_f,w,W_b,X_r)}{\sum\limits_{V\in\mathcal{M}_r} p_\theta^\kappa(V,X_r)}, \tag{5.24}$$

the generalized FB probabilities are computed as:

$$\gamma_{rt'}(s) = \sum_{[w,\tau,t]:\tau<t'\leqslant t} p_\theta([w,\tau,t]|X_r)\cdot\gamma_{r\tau}(s_{\tau}{=}s|w). \tag{5.25}$$

Here, the sum runs over all word hypotheses w which span time frame t'. In case that $\kappa = 1$, the word probability $p_\theta([w,\tau,t]|X_r)$ corresponds to the posterior probability of hypothesizing w within the time boundaries $[\tau+1,\ t]$, independent of any surrounding word sequences W_f and W_b.

Because lattices may contain different paths going through a certain arc, two equivalence relations are defined to decompose the computation of the word probabilities into the portion for the forward probability and the portion for the backward probability. The equivalence class representing a set of forward paths is denoted by $[q]_{\sim_f}$ and comprises the set of all partial paths that start at the source node q_1 and end in arc q:

$$[q]_{\sim_f} \coloneqq \{q_1,...,q_n\ :\ q_n = q\}. \tag{5.26}$$

Similarly, the equivalence class $[q]_{\sim_b}$ is defined as the set of all partial paths which start at arc q and end in the sink node q_N:

$$[q]_{\sim_b} \coloneqq \{q_n,...,q_N\ :\ q_n = q\}. \tag{5.27}$$

The forward and the backward probability of an arc q can then be denoted as follows:

$$\alpha\big([q]_{\sim_f}\big) = \sum_{q_1,\ldots,q_n\in[q]_{\sim_f}} \prod_{i=1}^{n} h_{q_i}^\kappa \cdot p^\kappa(v_{i_m}|v_{i_1}^{i_{m-1}}) \qquad \text{(forward probability)} \tag{5.28}$$

$$\beta\big([q]_{\sim_b}\big) = \sum_{q_n,\ldots,q_N\in[q]_{\sim_b}} \prod_{i=n}^{N} h_{q_i}^\kappa \cdot p^\kappa(v_{i_m}|v_{i_1}^{i_{m-1}}) \qquad \text{(backward probability)} \tag{5.29}$$

The recursive form of the forward probability is given by:

$$\alpha\big([q]_{\sim_f}\big) = \sum_{q_1,\ldots,q_n\in[q]_{\sim_f}} \prod_{i=1}^{n} h_{q_i}^\kappa \cdot p^\kappa(v_{i_m}|v_{i_1}^{i_{m-1}}) \tag{5.30}$$

$$= h_q^\kappa \cdot p^\kappa(v_m|v_1^{m-1}) \cdot \underbrace{\sum_{q_1,\ldots,q_n\in[q]_{\sim_f}} \prod_{i=1}^{n-1} h_{q_i}^\kappa \cdot p^\kappa(v_{i_m}|v_{i_1}^{i_{m-1}})}_{=\sum\limits_{q':\,q_1,\ldots,q',q_n\in[q]_{\sim_f}} \alpha([q']_{\sim_f})} \tag{5.31}$$

$$= h_q^\kappa \cdot p^\kappa(v_m|v_1^{m-1}) \cdot \sum_{q':\,q_1,\ldots,q',q_n\in[q]_{\sim_f}} \alpha\big([q']_{\sim_f}\big). \tag{5.32}$$

Analogously, the recursive form of the backward probability is given by:

$$\beta\big([q]_{\sim_b}\big) = \sum_{q_n,\ldots,q_N \in [q]_{\sim_b}} \prod_{i=n}^{N} h_{q_i}^{\kappa} \cdot p^{\kappa}(v_{i_m}|v_{i_1}^{i_{m-1}}) \tag{5.33}$$

$$= h_q^{\kappa} \cdot p^{\kappa}(v_m|v_1^{m-1}) \cdot \underbrace{\sum_{q,q',\ldots,q_N \in [q]_{\sim_b}} \prod_{i=n+1}^{N} h_{q_i}^{\kappa} \cdot p^{\kappa}(v_{i_m}|v_{i_1}^{i_{m-1}})}_{= \sum\limits_{q' \,:\, q,q',\ldots,q_N \in [q]_{\sim_b}} \beta([q']_{\sim_b})} \tag{5.34}$$

$$= h_q^{\kappa} \cdot p^{\kappa}(v_m|v_1^{m-1}) \cdot \sum_{q' \,:\, q,q',\ldots,q_N \in [q]_{\sim_b}} \beta\big([q']_{\sim_b}\big). \tag{5.35}$$

Both functions allow for estimating the forward and the backward probabilities on word lattices by means of dynamic programming. Because word lattice arcs are conditioned on their language model context, the FB algorithm can be formulated in a very compact and elegant way. For this purpose, let $G = (\mathcal{V}, \mathcal{E})$ denote a word graph with begin nodes $n_b = (v_1^{m-1}, l, \tau)$, $n_b' = (u_1^{m-1}, l', \tau')$ and end nodes $n_e = (v_2^m, r, t)$, $n_e' = (u_2^m, r', t')$ for $\{n_b, n_b', n_e, n_e'\} \subseteq \mathcal{V}$. Moreover, let

$$q = ([l, v_m, r], \tau, t, n_b, n_e, p(v_m|v_1^{m-1}), h_q, H_q, b) \in \mathcal{E}, \tag{5.36}$$

$$q' = ([l', u_m, r'], \tau', t', n_b', n_e', p(u_m|u_1^{m-1}), h_{q'}, H_{q'}, b') \in \mathcal{E} \tag{5.37}$$

denote two arcs in the word graph whose acoustic scores are given by $h_q = h([l, v_m, r]; \tau, t)$ and $h_{q'} = h([l', u_m, r']; \tau', t')$, respectively. The probabilities of the best partial sentence hypotheses leading to either of these arcs shall be given by $H_q = H(v_2^m, r, t)$ and $H_{q'} = H(u_2^m, r', t')$. We now define a *predecessor function* $\mathsf{head}(q)$, which comprises the set of all arcs preceding arc q, and a *successor function* $\mathsf{tail}(q)$ which contains the set of all arcs succeeding arc q:

$$\mathsf{head}(q) = \{q' \in \mathcal{V} \,:\, r' = l \wedge t' = \tau \wedge u_2^m = v_1^{m-1}\} \tag{5.38}$$

$$\mathsf{tail}(q) = \{q' \in \mathcal{V} \,:\, l'' = r \wedge \tau'' = t \wedge u_1^{m-1} = v_2^m\} \tag{5.39}$$

Word probabilities can then be estimated according to Algorithm 1. If $\kappa = 1$, then each arc weight γ_q corresponds to a word posterior probability. The generalized FB probabilities are obtained by carrying out the sum over the conditional FB probabilities over all arcs that span time frame t weighted with their respective word probabilities:

$$\gamma_{rt'}(s) = \sum_{q \,:\, \tau < t' \leqslant t} \gamma_{rt'}(s|[l, v_m, r]) \cdot \gamma_q. \tag{5.40}$$

Note that the FB algorithm can also be applied to phoneme or state graphs.

Algorithm 1 Forward-Backward Probabilities on Word Lattices

Forward Pass:

$$\alpha_{\varnothing} \coloneqq 1$$
$$\alpha_q \coloneqq \sum_{r \in \mathsf{head}(q)} p^{\kappa}(X|q) \cdot p^{\kappa}(q|r) \cdot \alpha_r$$

Backward Pass:

$$\beta_{\varnothing} \coloneqq 1$$
$$\beta_q \coloneqq \sum_{r \in \mathsf{tail}(q)} p^{\kappa}(X|r) \cdot p^{\kappa}(r|q) \cdot \beta_r$$

Normalization:

$$\chi \coloneqq \sum_{q \in \mathcal{E}:\mathsf{tail}(q)=\varnothing} \alpha_q$$
$$\gamma_q \coloneqq \frac{\alpha_q \cdot \beta_q}{\chi}$$

5.2.1 Scaling Log Likelihoods

Both the log-probabilities of the acoustic model and the language model are typically scaled with a constant κ. The choice of this weighting exponent controls the extent to which alternative sentence hypotheses affect the estimation of word probabilities. As κ becomes large, the probability mass concentrates preferably on the first best hypothesis; as κ becomes smaller, less likely hypotheses get a larger relative weight and are increasingly taken into account. If κ is set to 0, all paths in the lattice become equally likely, and the posterior distribution will be uniform. In practice, κ is typically set to a value close to the inverse of the language model scaling factor[4]. Scaling the log likelihoods often leads to improved generalization to unseen test data because a larger amount of sentence hypotheses obtain a non-vanishing weight which makes the estimation of the discriminative averages more reliable. Similar to Section 2.5.2, the *Kingsbury-Rayner* formula is used to compute sums over probabilities that are represented in the log-domain [Kingsbury & Rayner 71].

5.2.2 Improved Forward-Backward Pruning

An effective means to prune a lattice is the FB pruning as suggested in [Sixtus & Ortmanns 99]. This pruning method is refined as follows to guarantee that the most probable sentence hypotheses are pertained. The first step in the modified FB pruning applies Algorithm 1 with the sum being replaced by the maximum operation. Each arc in a lattice is thus assigned to the score of the best path (or sentence hypothesis) that runs through that arc. In a second step, all arc hypotheses are sorted in ascending order with respect to the scores computed in step 1. The lattice is then pruned by keeping a prefix of the sorted array up to the desired graph density while all remaining arcs (except for ties) are discarded. In contrast to [Sixtus & Ortmanns 99], the improved FB pruning has the following properties:

[4] Language model probabilities and word insertion penalties are both weighted with the language model scaling factor.

- The suggested method requires only a single parameter which is the desired graph density. This graph density can either be specified in absolute terms or be set relative with respect to the input lattice size. The specification of a threshold that makes implicit assumptions on the distribution of the posterior probabilities is not required.
- Per construction, the pruned graph never contains an open ended path.
- It is guaranteed that any N-best list computed from the pruned graph contains exactly the same hypotheses with the same costs as if it were computed from the unpruned graph.

5.2.3 Unigram Lattice Rescoring

In [Schlüter & Müller$^+$ 99], it was found that a discriminative training based on the MMI criterion works best if the numerator and the denominator lattices are rescored with a unigram language model, irrespective of the language model used for testing. The reason for this is that a higher-order language model serves as a filter which puts higher weight onto linguistically more probable sentence hypotheses, and thus reduces the weight of alternative sentence hypotheses. If, however, a lower-order language model is used, competing hypotheses become more likely which improves the ability of a discriminatively trained system to generalize to unseen data.

If not stated otherwise, the numerator and the denominator lattices are constructed using either a bigram or a trigram language model. The lattices are then rescored with a unigram language model before they are used in training. The rescoring is done without changing the structure of the lattice, that is, after rescoring, the language model nodes still reflect the order of the language model used during recognition. An interesting follow-up question is whether the language model used in decoding (i.e., before the unigram rescoring is applied) has an effect on the test set performance. This question is investigated in Section 6.3.11.

5.2.4 Discriminative Training Procedure

Algorithm 2 illustrates the major steps in a lattice-based discriminative training. Both the numerator and the denominator lattices are only generated in the first iteration and stored for later use in a lattice repository. From the second iteration onwards, lattices are read from the repository and rescored with the acoustic model obtained from the previous iteration. At the beginning of each iteration, accumulators for collecting and aggregating discriminative statistics are initialized. For each training utterance r, an HMM network for the spoken word sequence W_r is constructed and passed to the search procedure. The HMM network constrains the search space onto the pronunciation variants of the spoken word sequence. The output of the search is a word lattice for the numerator model. The arcs of the numerator lattice are then rescored with a unigram language model and weighted with the FB probabilities as determined based on Algorithm 1. Optionally, the word lattice is pruned using the refined FB pruning as described in Section 5.2.2. The acoustic observation vectors are aligned with the word automata for all arcs in the numerator lattice, and the sufficient statistics are aggregated in the accumulators for the correct model.

The statistics for the competing model are determined similarly: first, the denominator lattice is constructed via an unconstrained search and merged with the numerator lattice. From that on, all subsequent steps are identical with the processing of the numerator lattices, that is, a unigram rescoring is performed followed by the computation of the FB probabilities. The denominator lattice is optionally pruned before the acoustic observations are aligned with the word automata for the arcs in the denominator lattice. The resulting conditional FB probabilities are weighted with their arc posterior weights, and the sufficient statistics are aggregated in the accumulators for the competing model. At the end of each training pass, the iteration constants are determined and the parameters of the new acoustic model are re-estimated.

Algorithm 2 Discriminative Training Procedure

LOOP OVER PASSES: $i = 1, ..., I$
- **INITIALIZE RUNNING SUMS FOR ALL MIXTURES AND DENSITIES**
- **FOR EACH SENTENCE** $r = 1, ..., R$ **OF TRAINING DATA DO**
  - **IF** $(i$ == $1)$ **THEN**
    - **CREATE HMM NETWORK** $\mathcal{M}_r^{\text{num}}$ **OF THE SENTENCE** W_r
    - **SEARCH**$(\mathcal{M}_r^{\text{num}}, X_r) \rightarrow G(\mathcal{V}_r^{\text{num}}, \mathcal{E}_r^{\text{num}})$ **(unigram lattice rescoring)**
    - **WRITE**$(G(\mathcal{V}_r^{\text{num}}, \mathcal{E}_r^{\text{num}})$
    - **CREATE HMM NETWORK** $\mathcal{M}^{\text{den}}$ **FOR UNCONSTRAINED RECOGNITION**
    - **SEARCH**$(\mathcal{M}^{\text{den}}, X_r) \rightarrow G(\mathcal{V}_r^{\text{den}}, \mathcal{E}_r^{\text{den}})$ **(unigram lattice rescoring)**
    - **MERGE LATTICES:** $\mathcal{V}_r^{\text{den}} \leftarrow (\mathcal{V}_r^{\text{num}} \cup \mathcal{V}_r^{\text{den}}($, $\mathcal{E}_r^{\text{den}} \leftarrow (\mathcal{E}_r^{\text{num}} \cup \mathcal{E}_r^{\text{den}})$
    - **WRITE**$(G(\mathcal{V}_r^{\text{den}}, \mathcal{E}_r^{\text{den}})$
  - **ELSE**
    - **READ**$(G(\mathcal{V}_r^{\text{num}}, \mathcal{E}_r^{\text{num}})$ **(acoustic rescoring)**
    - **READ**$(G(\mathcal{V}_r^{\text{den}}, \mathcal{E}_r^{\text{den}})$ **(acoustic rescoring)**
  - **COMPUTE FB PROBABILITIES ON** $G(\mathcal{V}_r^{\text{num}}, \mathcal{E}_r^{\text{num}})$
  - **FOR EACH ARC** $q \in \mathcal{E}_r^{\text{num}}$ **DO**
    - **FOR EACH TIME FRAME** $t' = \tau + 1, ..., t$ **DO**
      - **ACCUMULATE COUNTS OF THE NUMERATOR MODEL:**
        $(q, t') \rightarrow (s, l)$
        squares: $\Gamma_{sl}^{\text{num}}(\{x^2\})$ += $\gamma_q \cdot x_{rt'} \cdot x_{rt'}^\top$
        observations: $\Gamma_{sl}^{\text{num}}(\{x\})$ += $\gamma_q \cdot x_{rt'}$
        weights: $\Gamma_{sl}^{\text{num}}(1)$ += γ_q
  - **COMPUTE FB PROBABILITIES ON** $G(\mathcal{V}_r^{\text{den}}, \mathcal{E}_r^{\text{den}})$
  - **FOR EACH ARC** $q \in \mathcal{E}_r^{\text{den}}$ **DO**
    - **FOR EACH TIME FRAME** $t' = \tau + 1, ..., t$ **DO**
      - **ACCUMULATE COUNTS OF THE DENOMINATOR MODEL:**
        $(q, t') \rightarrow (s, l)$
        squares: $\Gamma_{sl}^{\text{den}}(\{x^2\})$ += $\gamma_q \cdot x_{rt'} \cdot x_{rt'}^\top$
        observations: $\Gamma_{sl}^{\text{den}}(\{x\})$ += $\gamma_q \cdot x_{rt'}$
        weights: $\Gamma_{sl}^{\text{den}}(1)$ += γ_q
  - **DETERMINE ITERATION CONSTANTS** $\{D_{sl}\}$
  - **REESTIMATE PARAMETERS** $\theta^{(i)} \leftarrow \text{argmax}_\theta \mathcal{S}_K(\theta|\theta^{(i-1)})$

5.3 Forward-Backward Accuracies on Word Lattices

The FB algorithm provides an efficient means to estimate arc posterior probabilities on word graphs. These arc posterior probabilities can be used to compute the expectation value of a function f over a word sequence w_1^N if f decomposes over the constituent words of the sentence hypothesis: $f(w_1^N) = f(w_1) \circ ... \circ f(w_N)$. Such a decomposition is not, however, always possible, and a prominent example for which this is known to be difficult is the *Minimum Bayes Risk* (MBR) decision rule in automatic speech recognition. The MBR decision rule aims at finding that sentence hypothesis which minimizes the expected loss under the word error rate:

$$\hat{W} = \underset{W}{\text{argmin}} \sum_{V \in \mathcal{M}_r} p(V|X_r) \cdot \mathcal{L}(V, W) \tag{5.41}$$

$$= \underset{W}{\text{argmin}} \sum_{V \in \mathcal{M}_r} p(V, X) \cdot \mathcal{L}(V, W). \tag{5.42}$$

Due to the intrinsic complexity of the Levenshtein distance, the computation of the expected word error rate cannot be factorized efficiently on lattices such that the principle of dynamic programming would be applicable. As a remedy, one can either segment the lattice [Kumar & Byrne 02, Goel & Kumar$^+$ 04] or replace the Levenshtein distance with a local approximation.

Closely related to the idea of Minimum Bayes Risk decoding is the *Minimum Word Error* (MWE) criterion which aims at minimizing the expected word error rate on training data. Since the word error loss function is based on the Levenshtein distance, the exact MWE criterion (and therefore also the exact *Minimum Phone Error* (MPE) criterion) requires in the worst-case a loop over all sentence hypotheses encoded in the lattice. Although this has an exponential time complexity, in practice this may be less problematic as was shown in [Heigold & Macherey$^+$ 05] where the exact MWE criterion is used to discriminatively train the parameters of a large scale speech recognition system. An alternative to the exact lattice-based MWE and MPE criterion is to either use N-best lists [Kaiser & Horvat$^+$ 00] or to replace the Levenshtein distance with a local approximation that allows for decomposing the loss function over the arcs of a lattice. The following accuracy measure proposed in [Povey & Woodland 02] provides such a local approximation of the Levenshtein distance:

$$\mathsf{A}(w, w_r) = \begin{cases} -1 + 2\varrho(w, w_r) & \text{if } w = w_r \\ -1 + \varrho(w, w_r) & \text{if } w \neq w_r \end{cases} \tag{5.43}$$

with

$$\varrho(w, w_r) \coloneqq \max\left\{0,\ \frac{\min\left\{t_e(w_r), t_e(w)\right\} - \max\left\{t_b(w_r), t_b(w)\right\}}{t_e(w_r) - t_b(w_r) + 1}\right\}. \tag{5.44}$$

Eq. 5.43 captures the relative temporal overlap between a reference word w_r and a hypothesis w. If the reference and the hypothesis have the same word identity, the sign of $\mathsf{A}(w, w_r)$ will be positive; otherwise, the sign is negative. Figure 5.7 illustrates how the temporal overlap between a single reference and a hypothesis is computed. The approximated word accuracy is a local function and can easily be extended to word lattices where arcs in the numerator lattice serve as references while arcs in the denominator lattice are the hypotheses. Since numerator lattices may contain multiple pronunciation variants with different alignments of the spoken word sequence, the raw accuracy is computed by maximizing over all reference words contained in the numerator lattice:

$$\mathsf{A}(w) \coloneqq \max_{w_r} \mathsf{A}(w, w_r). \tag{5.45}$$

The raw word accuracy does not take across-word contexts into account. Instead, it is defined on the word identity itself. Hence, even for different across-word contexts, both the reference and the hypothesis may still yield the maximum possible accuracy of 1.0. The same holds for the MPE criterion where a pair of n-phones is considered to match if both n-phones have the same

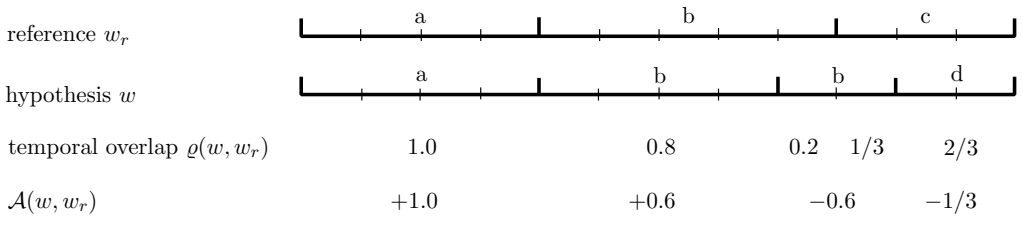


Figure 5.7: Relative temporal overlap between a single reference w_r and a hypothesis w together with a measure of accuracy $\mathsf{A}(w, w_r)$ for the *Minimum Word Error* criterion.

Algorithm 3 Forward-Backward Accuracies on Word Lattices

Forward Pass:

$$\bar{\alpha}_{\varnothing} \coloneqq 0$$
$$\bar{\alpha}_q \coloneqq \sum_{r \in \mathsf{head}(q)} p^{\kappa}(X|q) \cdot p^{\kappa}(q|r) \cdot \frac{\alpha_r}{\alpha_q} \cdot \Big[\bar{\alpha}_r + A(q)\Big]$$

Backward Pass:

$$\bar{\beta}_{\varnothing} \coloneqq 0$$
$$\bar{\beta}_q \coloneqq \sum_{r \in \mathsf{tail}(q)} p^{\kappa}(X|r) \cdot p^{\kappa}(r|q) \cdot \frac{\beta_r}{\beta_q} \cdot \Big[\bar{\beta}_r + A(r)\Big]$$

Normalization:

$$c(q) \coloneqq \bar{\alpha}_q + \bar{\beta}_q$$
$$\bar{\chi} \coloneqq \frac{\sum_{q \in \mathcal{E}:\mathsf{tail}(q)=\varnothing} \bar{\alpha}_q \cdot \alpha_q}{\sum_{q \in \mathcal{E}:\mathsf{tail}(q)=\varnothing} \alpha_q}$$
$$\bar{\gamma}_q \coloneqq \gamma_q \cdot \big[c(q) - \bar{\chi}\big]$$

central phoneme; left and right phonetic contexts are ignored. Figure 5.8 and Figure 5.9 illustrate the estimation of the raw word accuracy on a word lattice. Note that the average accuracy over all sentences encoded in the lattice is subtracted from each arc accuracy. Hence, the sum over the accuracies of all arcs crossing a certain time frame is always equal to zero. The sign of each accuracy indicates the "level of correctness" of that arc relative to the average sentence hypothesis. Arcs along a path with primarily positive FB accuracies are therefore "more correct" than arcs that constitute the average sentence hypothesis. Correspondingly, arcs with a negative accuracy belong to paths with a higher expected error than the average sentence hypothesis.

Arc accuracies can be computed using a modified FB algorithm. The computation requires the FB probabilities γ_q which are estimated based on Algorithm 1. The raw word (or phoneme) accuracies are then determined for each arc according to Eq. 5.45. The combination of the FB probabilities with the raw accuracies according to Algorithm 3 yields the forward-backward accuracies.

5.3.1 Higher Order Statistics on Word Lattices

Other useful quantities can be estimated based on Algorithm 3 by redefining the raw word accuracy. Hence, if $A(q) \coloneqq 1$, Algorithm 3 returns the expected sentence length. If $A(q) \coloneqq 0$, Algorithm 3 yields the usual FB probabilities. To estimate the expectation values of word frequencies, the raw word accuracy can be replaced with a vector indicating the word index of the arc it refers to. Let $\mathcal{W}(G)$ denote the vocabulary constrained to words that occur in a word lattice G. Then, with $A(q) \coloneqq (\delta(w_q, v_1), ..., \delta(w_q, v_n))^{\top}$ and v_i denoting the ith entry in the vocabulary ($1 \leqslant i \leqslant n$), Algorithm 3 allows for estimating expected word frequencies which are important in many information retrieval tasks [Macherey & Viechtbauer$^+$ 02, Macherey & Viechtbauer$^+$ 03].

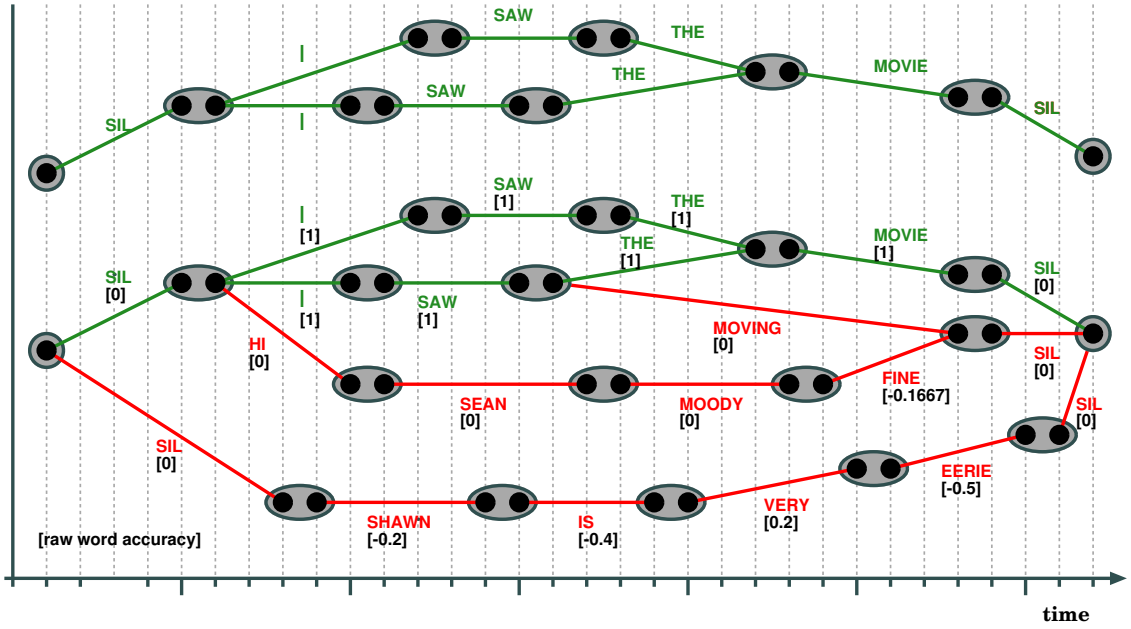


Figure 5.8: Raw word accuracy computed on a word lattice

5.4 Word-Lattice Based MCE Training

The MCE criterion aims at minimizing a smoothed sentence error on training data. Thus, together with the MWE and the MPE criterion, the MCE criterion belongs to the class of error minimizing training criteria. Although the MCE criterion could be shown to give consistently better results on small vocabulary tasks compared to the MMI criterion [Macherey 98, Schlüter & Macherey 98, Schlüter & Macherey$^+$ 01], there are only few publications that investigate the MCE criterion on large vocabulary tasks [McDermott & Hazen 04, McDermott & Katagiri 05]. One reason is that the MCE criterion requires the exclusion of the correct class from the set of all competing classes. For automatic speech recognition this means that the spoken word sequence has to be removed from the set of all word sequences considered for discrimination, which can be difficult to accomplish if the set of competing word sequences is encoded as a word lattice. Since for discriminative training it is essential to use lattices that may contain multiple alignments and pronunciation variants of the spoken utterance, arcs cannot uniquely be assigned to either the correct or a competing sentence hypothesis without changing the structure of the lattice. One possible remedy would be to use N-best lists which was examined in [Paliwal & Bacchiani$^+$ 95]. Another alternative is the use of a *Finite State Machine* (FSM) library where a transducer built for each utterance encodes the set of competing word sequences considered for discrimination. The spoken word sequence can then be excluded using standard FSM operations. In general, this procedure is however less efficient than directly using a word lattice since the FSM operations involve the determinization of the input lattice which may have an exponential space complexity.

This section presents a novel algorithm that allows for optimizing the acoustic model parameters of a large scale speech recognition system under the MCE criterion using word lattices. The statistics for both the correct and the competing model are collected solely from word graphs without the use of N-best lists. Particularly for long utterances, the number of sentence alternatives thus taken into account increases by several orders of magnitude compared to N-best lists.

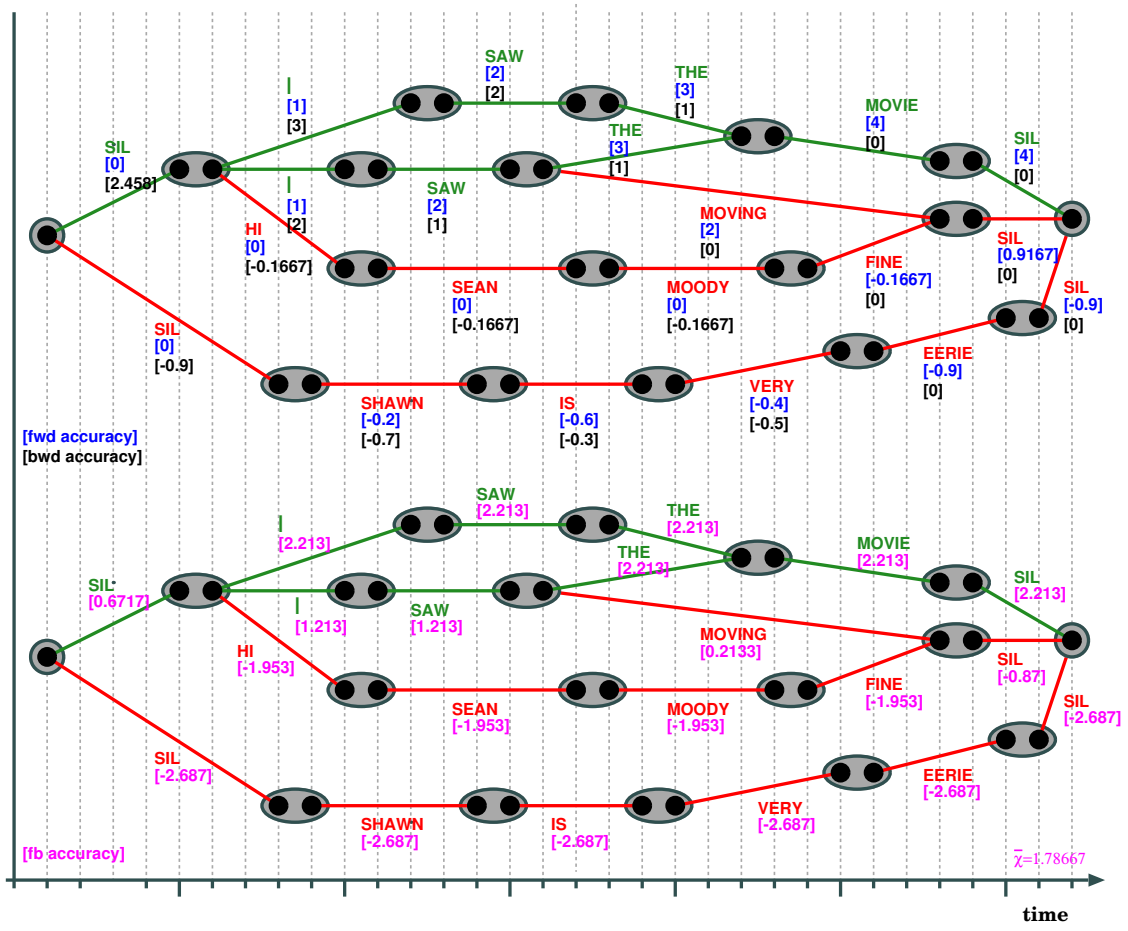


Figure 5.9: Forward-backward accuracies estimated on a word lattice.

5.4.1 Excluding the Correct Word Sequence from a Lattice

In lattice-based MCE training, the set of competing hypotheses comprises all word sequences represented in a word graph, except the spoken word sequence W_r. Therefore, estimating the word probabilities under the MCE criterion similar to lattice-based MMI training requires excluding the spoken word sequence from the word graph. However, since particular words of the spoken word sequence may be part of other sentence hypotheses, too, this could be difficult to accomplish since the affected arcs cannot be removed without changing the structure of the lattice. The algorithm suggested in this section follows a different approach. First, the sum over *all* word sequences in the word graph (represented by $\mathcal{M}_r$) including the spoken word sequence is computed. In a second step, the probability of the spoken word sequence is then subtracted from the word probabilities computed in the previous step. The probability of hypothesizing a word w within the time frames $[t_b, t_e]$ under the MCE criterion can thus be written as [Macherey 98, Schlüter & Macherey$^+$ 01,

Macherey & Haferkamp+ 05]:

$$q_{[t_b,\,t_e]}(w|X_r) = \frac{\sum\limits_{\substack{\{W\in\mathcal{M}_r|W\neq W_r \\ \wedge\, w_{[t_b,\,t_e]}\in W\}}} p_\lambda^\alpha(X_r,W)}{\sum\limits_{\{V\in\mathcal{M}_r|V\neq W_r\}} p_\lambda^\alpha(X_r,V)} \tag{5.46}$$

$$= \frac{\overbrace{\sum\limits_{\substack{\{W\in\mathcal{M}_r| \\ w_{[t_b,\,t_e]}\in W\}}} p_\lambda^\alpha(X_r,W)}^{(*)} - \overbrace{\sum\limits_{\substack{\{W\in\mathcal{M}_r|W=W_r \\ \wedge\, w_{[t_b,\,t_e]}\in W\}}} p_\lambda^\alpha(X_r,W)}^{(**)}}{\sum\limits_{V\in\mathcal{M}_r} p_\lambda^\alpha(X_r,V) - \sum\limits_{\{V\in\mathcal{M}_r|V=W_r\}} p_\lambda^\alpha(X_r,V)} \tag{5.47}$$

In detail, the procedure to efficiently compute arc weights for a lattice-based MCE training comprises the following steps:

1. Label *all* alignments of the spoken word sequence in the denominator lattice. The corresponding sub lattice is equivalent to the numerator lattice.

2. Compute the arc posteriors in the numerator and the denominator lattice using the forward-backward algorithm (similar to lattice-based MMI training).

3. Subtract the posteriors of labeled arcs in the denominator lattice by the corresponding numerator arc posteriors.

Note that the denominator arc posteriors are always greater than or equal to their corresponding numerator arc posteriors. Hence, Eq. (5.47) is guaranteed to be nonnegative.

5.4.2 Numerical Aspects

Eq. (5.47) needs to be evaluated for those training utterance only, for which $(*)$ is greater than $(**)$:

$$\sum_{\substack{\{W\in\mathcal{M}_r| \\ w_{[t_b,\,t_e]}\in W\}}} p_\lambda^\alpha(X_r,W) - \sum_{\substack{\{W\in\mathcal{M}_r|W=W_r \\ \wedge\, w_{[t_b,\,t_e]}\in W\}}} p_\lambda^\alpha(X_r,W) \;\geqslant\; \epsilon \tag{5.48}$$

The minimum amount by which both probabilities must differ to require an evaluation can be determined by finding the least floating-point value that is larger than the negative log-probability of $(*)$. This value can be determined by decomposing $(*)$ into a normalized fraction (mantissa) M and an exponent E (usually raised to a power of $B = 2$) using the ISO/IEC conform function `frexp`: $(*) = M \times B^E$. The normalized fraction lies within the range $[-1, 1]$ and has to be increased by the smallest representable floating-point number ε_0 that, when being added to 1.0, yields a floating-point number larger than 1.0. The value of ε_0 is a machine-dependent constant provided by any ANSI conform C and C++ compiler. The increased fraction is then combined with the original exponent using the ISO/IEC function `ldexp`: $(M + \varepsilon_0) \times B^E$. A similar operation can be applied in order to identify lattice arcs whose contribution to the aggregated sufficient statistics can be neglected if $\gamma_{q,\mathrm{MCE}}^{\mathsf{den}} + \varepsilon \leqslant \gamma_{q,\mathrm{MCE}}^{\mathsf{num}}$.

5.5 Methods for Accelerating the Training Procedure

Discriminative training procedures require significantly more CPU time than ML based training methods. Typically, the predominant share of the total running time accounts for the generation and rescoring of word lattices in order to collect sufficient statistics for re-estimating the parameters of the acoustic model. To reduce the time spent in each training iteration, three methods were developed that aim at accelerating different aspects of the discriminative training procedure:

- Lattice generation: the search is performed with a dynamically adjusted pruning threshold.
- Lattice rescoring: use of a constrained search space that is restricted to candidates encoded in a lattice within their Viterbi segmentation points.
- Parallelization: a client-server model for distributed computing

The first two methods take advantage of the fact that the spoken word sequence is known during training. This allows for focusing the search more effectively by using tighter beam widths which leads to a reduced number of distance calculations during training.

5.5.1 Dynamically Adjusted Pruning Thresholds

This method is based on an utterance specific choice of the pruning thresholds. Each recognition pass is initially started with a very tight acoustic pruning threshold that is sufficient to prevent search errors for approximately 90% of all training utterances. For the remaining 10% of the training utterances, the pruning threshold is gradually increased until search errors can be excluded. The criterion to decide whether a search error occurred compares the sentence probability of the best recognized sentence hypotheses $p_\theta(W_{\text{opt}}, X_r)$ with the spoken word sequence $p_\theta(W_r, X_r)$, where the latter is obtained through a forced alignment. If the likelihood of the best recognized word sequence (unconstrained search) is less than the likelihood obtained from the forced alignment of the spoken word sequence (constrained search), a search error occurred since the spoken word sequence provides a better explanation for the sequence of acoustic observations under the current model than the recognized word sequence. In this case the recognition is repeated with a larger beam size. If, instead, the likelihood of the best recognized sentence hypothesis is larger than or equal to the sentence probability of the spoken word sequence, the search result is assumed to be free of recognition errors, and W_{opt} is supposed to be the best competing word sequence among all possible sentence hypotheses. The method of dynamically adjusted pruning thresholds is primarily applied in the first pass in order to generate initial word lattices on large training corpora. Subsequent training phases typically collect their sufficient statistics from re-scoring the lattice repositories produced in the first iteration.

5.5.2 Constrained Recognition

To further reduce the computational costs for estimating the competing model once initial word lattices are available, the search space is restricted to those sentence hypotheses that are encoded in a word lattice. This can be accomplished by either (1) rescoring the word hypotheses in a word lattice *within* their Viterbi segmentation points or (2) by converting the word lattice into a corresponding HMM network and carrying out the search on the hypotheses space thus induced. Note that both methods cannot produce word sequences other than those contained in the initial word lattice. Method (2) may, however, hypothesize new word boundary times i.e., new Viterbi segmentation points. Thus, it allows the training procedure to relax some of the constraints caused by re-using word lattices produced in the initial iteration.

5.5.3 A Client-Server Model for Distributed Computing

In each iteration, the training algorithm proceeds over all training utterances, generates or re-scores word lattices for the numerator and the denominator model, and aggregates sufficient statistics in order to re-estimate the acoustic model parameters at the end of the training pass. Since each utterance can be processed independent of other training utterances, the training procedure can be parallelized. To accomplish this, a client-server model based on the TCP/IP protocol for distributed computing was developed to distribute the computational load over multiple machines. At the beginning of each training iteration, a master job spawns off a set of worker jobs of which each is assigned to a certain portion of the training corpus. Once started, a worker job either generates or re-scores a set of word lattices, collects sufficient statistics, and sends the result back to the master. The master job aggregates all incoming statistics and starts the re-estimation process once all worker jobs have successfully finished their tasks. After re-estimating the acoustic model, the master job starts the next training iteration by sending out requests to the worker jobs (broadcast) to re-score the lattice repository using the updated parameter set. Depending on the scheduling policy which controls how jobs are assigned to machines, the master job usually monitors the worker jobs but does not generate or re-score any lattices by itself. This helps to reduce the resources allocated by the master job which can therefore run with a higher priority. Running with a higher priority also prevents the master job from getting evicted which is important since the master has to stay alive during the entire training process. Other than the master job, worker jobs may terminate themselves once they have successfully sent their sufficient statistics to the master and received a confirmation in return. Since worker jobs are terminated at the end of an iteration, they need to allocate less CPU time which, considering their larger memory requirements, can be useful to schedule them earlier and to run them with a higher priority. The overhead caused by additional start-up and shut-down times are usually negligible compared to the time spent for generating and re-scoring lattices.

The information exchange between the worker and the master jobs is primarily performed over the network. Only the lattice repository, the acoustic model, and – in case that new lattices need to be generated – the language model are loaded from disk. If lattices are read from a repository, the language model does not need to be reloaded again since the lattice arcs are already scored with the respective language model probabilities. This helps to further reduce the memory requirements and thus to increase the number of jobs that get scheduled on a highly utilized data center. To alleviate delays and time-outs due to an increased network traffic when multiple worker jobs try to connect to the master simultaneously, incoming requests are immediately processed by the master. To accomplish this, the procedure that receives incoming messages and sets up connections to the worker jobs runs in a separate thread. Once the connection to the worker has been established, the master enqueues the descriptor that defines the connection. The worker changes its state into "idle" and waits until the master job sends a message to signal that it is ready to receive the sufficient statistics. If the connection to the master job cannot be established because the network may be temporarily down or the maximum number of descriptors that the operating system can handle are depleted, the worker job tries to reconnect to the master once a time-out of 30 seconds has elapsed. To reduce the network traffic, data exchanged between worker and master jobs are batched into packages of 4KByte before they are sent to the recipient. Jobs that are waiting for a message can optionally sent a request to enforce an immediate transmission even before the final package size of 4KByte has been reached (*flush*). Since the information sent over the network is usually not encrypted, data packages are tagged with a unique key that identifies the recipient job and prevents that the communication between jobs of other users in the cluster may accidentally interfere with the data flow between the master and the worker jobs. The keys to identify packages are generated by the master job and passed to the worker jobs as an additional argument on the command line when a worker is setup. Therefore, the communication protocol does not have to support a safe exchange of keys.

5.6 Summary

This chapter described the estimation and collection of sufficient statistics in lattice-based discriminative training. To model the correct and the competing classes, a novel framework based on Hidden Markov Model networks was introduced. HMM networks can be considered as a hybrid approach between finite state transducers and a search implementation based on word-dependent tree copies. Complex acoustic models like across word models, pronunciation variants, and multi-word phrases can easily be compiled into HMM networks without the need of changing the implementation of the search. Network sizes for various speech corpora and recognition tasks were studied, showing that HMM networks provide a very compact representation even if large vocabulary sizes of more than 80k words are used. A set of elementary graph transformations was defined that allows for mapping the nodes and arcs of an HMM network onto the nodes of a language model look-ahead tree, thus making the concept of language model look-ahead pruning applicable to the new search architecture. The construction of word lattices was formally defined and two forward-backward algorithms for computing posterior probabilities and word accuracies were discussed. A refined version of a forward-backward pruning algorithm for lattices was presented that improves over the algorithm described in [Sixtus & Ortmanns 99]. The new FB pruning has the advantage that it never produces dead paths. Moreover, the algorithm guarantees that the N-best candidates computed from the pruned lattice are always identical with the N-best candidates contained in the unpruned lattice. The new FB algorithm does not require a pruning threshold; instead, it suffices to specify the desired graph density which makes this pruning technique invariant with respect to the range of acoustic scores and log-posterior probabilities. Section 5.4 described details of the first successful implementation of a lattice-based Minimum Classification Error training for large vocabulary sizes. Methods to accelerate the training procedure presented in Section 5.5 concluded this chapter.

Chapter 6

Experimental Results

This chapter reports on experiments conducted on various speech recognition tasks with different levels of difficulty. Key aspects of the investigation are (1) the dependency between model complexity and training performance; (2) a comparison of different techniques to choose the step size in the update equations; (3) the effect of using Gaussian mixture densities with different variance tying schemes; and (4) a comparison of discriminative training criteria contained in the extended unifying approach. Results are presented on the SieTill corpus, the Wall Street Journal recognition tasks (Wsj-5k, Nab-20k and Nab-65k), and the Hub-4 corpus. Section 6.1 defines the evaluation criteria and the method used to estimate confidence intervals for significance tests. Section 6.2 describes the speech corpora and presents the ML-trained baseline systems which serve as starting point for the discriminative training. Section 6.3 presents experimental results on the four areas listed above. The chapter ends with a summary in Section 6.4.

6.1 Evaluation Metrics and Significance Tests

System performance is mainly reported in terms of *word error rate* (WER) which is defined as the minimum number of basic edit operations (*insertions, deletions, and substitutions*) that are needed to transform the spoken word sequence into the recognized word sequence [Levenshtein 65]. Besides the WER, results are also reported for the *sentence* (or *string*) *error rate* (SER) which reflects the amount of misrecognized utterances; the *phoneme error rate* (PER) which is defined as the minimum number of edit operations needed to transform the sequence of spoken phonemes into the sequence of recognized phonemes; and the *graph error rate* (GER) which is defined as the minimum over the WERs of all sentence hypotheses represented in a word graph. Word graph densities are quantified as follows:

- the average number of *arcs per spoken word* (avg. APS);
- the average number of *arcs per recognized word* (avg. APR); and
- the average number of *arcs per time frame* (avg. APT).

6.1.1 Confidence Intervals and Significance Tests

Confidence intervals are estimated based on the *bootstrap re-sampling normal approximation* method [Efron 79, Noreen 89]. The bootstrap method aims at yielding a better estimation of the true error rate of a classifier by generating a large number of virtual test sets over which the individual error rates are averaged. The virtual test sets are obtained through sampling the original test set with replacement (so-called *Monte Carlo simulations*). The averaged error rate is considered to be more reliable than the error rate measured on the original test set. If not stated otherwise, the number of virtual test sets is set to $100,001$ with each virtual set comprising 10,000 utterances that are chosen randomly with replacement from the original evaluation set.

Pairwise system comparisons are based on the *paired bootstrap* method [Bisani & Ney 04, Koehn 04]. Virtual test sets in the this method are generated through a consistent selection of candidates with replacement from each system. The confidence intervals are estimated over the differences between the word error rates of corresponding virtual test sets. Improvements are considered to be statistically significant if the left boundary of the resulting confidence interval is larger than zero. Confidence intervals are computed for the 70%, 90%, 95%, and the 99% quantile.

6.2 Speech Corpora and Baseline Systems

Experiments were conducted on five speech corpora with different levels of difficulty, including elementary digit string recognition (SieTill), recognition of read speech (Wall Street Journal tasks), and transcriptions of recorded broadcasts (Hub-4). The following subsections briefly describe each recognition task and summarize some corpus statistics. Speech recognition systems trained under the ML criterion define the starting point for the discriminative training runs.

6.2.1 SieTill

Experiments for continuously spoken connected digit string recognition were performed on the SieTill corpus which is a collection of telephone-line recorded German digit strings [Eisele & Haeb-Umbach$^+$ 96]. The recognition vocabulary comprises the ten German digits plus the pronunciation variant *zwo* for *zwei* (two). All utterances were spoken by adult speakers. Table 6.1 summarizes some corpus statistics.

Table 6.1: Corpus statistics for the SieTill corpus for the recognition of telephone line recorded continuously spoken connected German digit strings.

	SieTill					
	training			test		
	male	female	all	male	female	all
amount of acoustic data [h]	5:58	5:20	11:18	6:03	5:20	11:23
silence portion [%]	54.7	53.7	54.2	55.5	53.7	54.7
# speakers	191	171	362	188	168	356
# sentences	6 886	6 150	13 036	6 938	6 176	13 114
# words	22 631	20 226	42 857	22 881	20 205	43 086
# lexicon words	11			11		
# pronunciations	11			11		
unknown words [%]	0.0			0.0		
perplexity (zerogram LM)	11.0			11.0		

The SieTill recognition system is based on gender-dependent whole-word HMMs with continuous Gaussian emission distributions of up to 32 densities per state. Each gender is modeled with 214 distinct states plus one state for silence. Observation vectors comprise 12 MFCCs with first derivatives and the second derivative of the first component, resulting in a $12 + 12 + 1 = 25$ dimensional feature vector. Each three adjacent input frames are concatenated to a $3 \times 25 = 75$ dimensional super vector and projected into a 25 dimensional subspace via a *Linear Discriminant Analysis* (LDA). Recognition systems were built for Gaussian mixture models with either a globally-pooled variance vector or density-specific variances. In case of density-specific variances, the feature space is further transformed with a *Maximum Likelihood Linear Transform* (MLLT) which is applied on top of the LDA matrix (cf. Chapter 9). All baseline recognition systems were

Table 6.2: Word error rates (WER) and sentence error rates (SER) measured on the SieTill corpus for a ML trained baseline system using gender-dependent whole-word models consisting of 214 distinct states per gender plus one state for silence. Results are presented for both single and mixture densities using either a globally pooled variance vector (var=1) or density specific variances (var=29).

			SieTill					
			train			test		
var	dns/mix	error	male	female	m+f	male	female	m+f
1	1	WER[%]	3.43	2.71	3.09	4.01	4.24	4.12
		SER[%]	9.12	7.40	8.31	10.25	11.06	10.63
	32	WER[%]	0.92	0.72	0.83	1.92	2.25	2.07
		SER[%]	2.61	2.18	2.41	5.10	6.10	5.57
29	1	WER[%]	3.63	3.01	3.34	4.28	4.08	4.18
		SER[%]	9.92	8.47	9.24	11.06	10.69	10.88
	32	WER[%]	0.78	0.38	0.59	1.57	1.50	1.54
		SER[%]	2.28	1.16	1.75	4.19	4.19	4.19

trained under the ML criterion in the Viterbi approximation. Word error rates thus achieved range between 4.12% (pooled variance) and 4.18% (density-specific variances) for single densities, and between 2.07% (pooled variance) and 1.54% (density-specific variances) using mixture densities. Table 6.2 lists the baseline results. Table 6.3 shows the word error rates for the different settings together with their confidence interval estimations based on the bootstrap re-sampling normal approximation method for the 70%, the 90%, the 95%, and the 99% interval. All ML trained systems were used to generate high-density word lattices for both the correct ("numerator") and the competing ("denominator") model. The resulting word graphs were pruned using the improved version of the *forward-backward* pruning described in Section 5.2.2.

Table 6.3: Confidence intervals estimated on the SieTill corpus using the bootstrap re-sampling normal approximation method [Efron 79].

				SieTill					
				baseline			confidence intervals			
set	var	dns/mix	gender	del- ins- sub	WER[%]	70%	90%	95%	99%
test	1	1	male	175-125- 617	4.01	±0.13	±0.21	±0.25	±0.33
			female	187-137- 533	4.24	±0.14	±0.22	±0.26	±0.34
			m+f	362-262-1150	4.12	±0.14	±0.22	±0.26	±0.34
		32	male	112- 96- 232	1.92	±0.09	±0.15	±0.18	±0.23
			female	81-127- 246	2.25	±0.10	±0.16	±0.19	±0.24
			m+f	193-223- 478	2.07	±0.10	±0.15	±0.18	±0.24
	29	1	male	251-329- 399	4.28	±0.14	±0.22	±0.26	±0.34
			female	215-249- 360	4.08	±0.13	±0.21	±0.25	±0.33
			m+f	466-578- 759	4.18	±0.14	±0.22	±0.26	±0.34
		32	male	84- 99- 177	1.57	±0.08	±0.13	±0.16	±0.21
			female	58- 88- 158	1.50	±0.08	±0.13	±0.15	±0.20
			m+f	142-187- 335	1.54	±0.08	±0.13	±0.15	±0.20

6.2.2 North American Business Corpora

The NORTH AMERICAN BUSINESS (NAB) corpora were compiled in 1991 as part of the *DARPA Spoken Language Program* to support research on large-vocabulary *Continuous Speech Recognition* (DARPA-CSR). The first two DARPA-CSR corpora consist of read speech with texts drawn from a machine-readable collection of WALL STREET JOURNAL (WSJ) news texts and are thus often known as WSJ0 and WSJ1. Later sections of these corpus sets contain read texts from additional sources of NORTH AMERICAN BUSINESS news and other domains. The texts were selected to fall within either a 5,000-word or a 20,000-word subset of the WSJ text corpus. Some spontaneous dictation is included in addition to the read speech. The dictation portion was recorded from journalists who dictated hypothetical news articles. To emphasize the size of the recognition vocabulary, the corpora are referred to as WSJ-5k, NAB-20k, and NAB-65k, respectively.

Wall Street Journal 5k (WSJ-5k)

The WALL STREET JOURNAL 5k (WSJ-5k) speech corpus is the first of two corpora collected for the DARPA-CSR initiative. The texts selected for the 5,000-word subset have a closed recognition vocabulary. Table 6.4 summarizes some corpus statistics.

Table 6.4: Corpus statistics for the WALL STREET JOURNAL 5K (WSJ-5k) speech corpus for the recognition of read speech of texts drawn from a machine-readable collection of *Wall Street Journal* news texts. The read texts were selected to fall within a 5,000-word subset of the WSJ text corpus.

	WSJ-5k					
	training		development		evaluation	
	male	female	male	female	male	female
amount of acoustic data [h]	7:26	7:51	0:27	0:19	0:21	0:19
silence portion [%]	19.3	19.3	21.8	22.0	25.6	27.6
# speakers	42	42	6	4	4	4
# sentences	3 586	3 654	247	163	163	167
# words	64 905	66 071	4 067	2 717	2 653	2 700
# phonemes	277 883	281 879	16 780	11 575	11 345	11 210
# lexicon words	10 133		5 007		5 007	
# pronunciations	10 771		5 656		5 656	
unknown words [%]	0.0		0.0		0.0	
perplexity (trigram LM)	293.8		57.8		52.8	

The WSJ-5k recognition system uses 2000 decision-tree based, gender-independent within-word triphone states plus one state for the silence model. The states are modeled as Gaussian mixture distributions with a total of 149k densities. All densities share a single, diagonal variance vector. The observation vectors comprise 16 MFCCs together with their first derivatives and the second derivative of the energy, resulting in a $16 + 16 + 1 = 33$ dimensional feature vector. Each five adjacent input frames are concatenated to a $5 \times 33 = 165$ dimensional super vector and projected into a 33 dimensional subspace via a *Linear Discriminant Analysis* (LDA). The baseline recognizer was trained under the ML criterion in the Viterbi approximation and achieves a WER of 4.19% on the combined development and evaluation set[1] as shown in Table 6.5. Table 6.6 shows the

[1] Since the official WSJ0 corpus does not provide a development set, the 410 sentences were extracted from 10 new

respective confidence intervals.

Table 6.5: Baseline recognition results produced on the WSJ-5k corpus using ML-trained Gaussian HMMs with a single, globally pooled, diagonal covariance matrix.

	WSJ-5k			
	train	dev	eval	dev+eval
avg. #arcs per spoken word	334.77	171.89	187.66	178.92
avg. #arcs per best rec. word	241.23	128.05	136.50	131.82
avg. #arcs per time frame	80.42	72.33	68.63	70.61
GER [%]	0.19	0.25	0.07	0.17
PER [%]	1.98	5.53	4.89	5.25
WER [%]	6.86	4.55	3.74	4.19
SER [%]	67.71	40.49	34.85	37.97

Table 6.6: Confidence intervals estimated on the WSJ-5k corpus using the bootstrap re-sampling normal approximation method [Efron 79].

	WSJ-5k					
	baseline		confidence intervals			
set	del-ins-sub	WER[%]	70%	90%	95%	99%
dev	55- 24-230	4.55	±0.07	±0.11	±0.14	±0.18
eval	24- 25-151	3.74	±0.07	±0.11	±0.13	±0.17
dev+eval	79- 49-381	4.19	±0.04	±0.06	±0.07	±0.09

Wall Street Journal 20k and 64k (NAB-20k, NAB-64k)

The November'94 NORTH AMERICAN BUSINESS (NAB) training corpus consists of the 84 speakers of the WSJ0 corpus (see above) plus 200 additional speakers of the WSJ1 corpus, resulting in a total of 81 hours of speech. The WSJ1 corpus comprises 37 474 training utterance (over 81 hours of speech), 4000 of which are the result of spontaneous dictations spoken by journalists with varying degrees of experience in dictation. Two recognition vocabularies are defined, comprising 20k and 65k words, respectively. Table 6.7 summarizes some corpus statistics.

Recognition results were produced on the NAB *Nov. '94 Hub-1* development and evaluation set. Both the 20k and the 65k recognition system use 7000 decision-tree based, gender-independent across-word triphone states plus one state for silence. The states are modeled as Gaussian mixture distributions with a total of 412k densities and a single, globally pooled, diagonal covariance matrix. The observation vectors consist of 16 MFCCs without temporal derivatives. Each 7 adjacent input frames are concatenated to a $7 \times 16 = 112$ dimensional super vector and projected into a 32 dimensional subspace via an LDA transformation. The ML trained speech recognition system achieves a WER of 11.41% using the 20k recognition vocabulary, and 9.22% using the 65k recognition vocabulary on the combined development and evaluation corpus as shown in Table 6.9.

As a contrastive result, a recognition system for the same task was built that uses 3000 decision-tree based, gender-independent *within-word* triphone states plus one state for silence. Other than

speakers of the *North American Business* task.

the across-word system, the observation vectors in the within-word system comprise 16 MFCCs together with their first derivatives and the second derivative of the energy. Each 3 adjacent input frames are concatenated to a $3 \times 33 = 99$ dimensional super vector and projected into a 33 dimensional subspace via a LDA. Table 6.8 shows the word error rates for the within-word system. Confidence interval estimations are shown in Table 6.10 for both the across word system and the within word system.

Table 6.7: Corpus statistics for the NORTH AMERICAN BUSINESS (NAB) corpus for the recognition of read speech of texts drawn from a machine-readable collection of *Wall Street Journal* (WSJ) news texts. The corpus, which includes the 84 speakers of the WSJ0 corpus, is augmented by 200 additional speakers of the WSJ1 corpus. Recognition tasks are defined for a 20k and a 65k words comprising vocabulary referred to as NAB-20k and NAB-65k, respectively.

	NAB-20k / NAB-65k					
	training		development		evaluation	
	male	female	male	female	male	female
amount of acoustic data [*h*]	39:43	41:40	0:24	0:24	0:27	0:26
silence portion [%]	25.8	26.1	19.6	17.3	18.5	17.7
# speakers	142	142	10	10	10	10
# sentences	18 704	18 770	155	155	156	160
# words	319 629	322 445	3 671	3 716	4 135	4 058
# phonemes	1 334 464	1 344 335	15 038	15 316	16 426	16 190
# lexicon words	15 013		19 978 / 64 735		19 978 / 64 735	
# pronunciations	16 990		22 412 / 69 970		22 412 / 69 970	
unknown words [%]	0.0		2.7 / 0.5		2.6 / 0.9	
perplexity (trigram LM)	181.9		124.5 / 145.9		136.8 / 144.2	

Table 6.8: Baseline recognition results achieved on the NAB-20k and the NAB-65k speech corpus using 3000 decision-tree based, gender-independent within-word triphone states trained under the ML criterion.

	NAB within-word systems							
	NAB-20k				NAB-65k			
	train	dev	eval	dev+eval	train	dev	eval	dev+eval
avg. APS	168.34	333.10	341.81	337.50	200.68	329.44	361.48	345.61
avg. APR	123.08	249.42	265.72	257.65	146.31	252.86	281.86	267.50
avg. APT	75.82	170.14	179.91	175.24	90.93	174.21	189.12	182.00
GER [%]	0.31	4.43	3.89	4.15	0.35	2.06	1.86	1.95
PER [%]	2.54	9.40	8.99	9.18	2.57	6.27	6.21	6.23
WER [%]	6.05	12.93	12.75	12.84	6.26	10.53	10.47	10.50
SER [%]	53.12	79.68	79.75	79.71	53.77	74.84	75.63	75.24

Table 6.9: Baseline recognition results produced on the NAB-20k and the NAB-65k speech corpus using 7000 decision-tree based, gender-independent across-word triphone states trained under the ML criterion.

	NAB across-word systems							
	NAB-20k				NAB-65k			
	train	dev	eval	dev+eval	train	dev	eval	dev+eval
avg. APS	81.38	122.20	128.94	125.60	63.84	95.87	100.16	98.03
avg. APR	59.46	94.02	102.38	98.24	46.81	74.66	79.92	77.32
avg. APT	42.99	77.93	75.68	76.76	34.28	62.48	60.40	61.40
GER [%]	0.28	4.63	4.17	4.39	0.29	2.26	2.29	2.28
PER [%]	1.85	8.65	8.47	8.55	1.88	5.52	5.70	5.61
WER [%]	4.89	11.45	11.38	11.41	5.01	9.07	9.35	9.22
SER [%]	45.88	73.78	76.27	75.08	46.27	66.77	71.20	69.01

Table 6.10: Confidence intervals estimated on the NAB-20k and the NAB-65k speech corpus using the bootstrap re-sampling normal approximation method [Efron 79].

			NAB confidence intervals					
			baseline		confidence intervals			
corpus		set	del- ins- sub	WER[%]	70%	90%	95%	99%
NAB-20k	WW	dev	113-171- 671	12.93	±0.16	±0.25	±0.29	±0.39
		eval	158-184- 703	12.75	±0.13	±0.20	±0.24	±0.32
		dev+eval	271-355-1374	12.84	±0.07	±0.11	±0.14	±0.18
	XW	dev	102-160- 584	11.45	±0.14	±0.23	±0.27	±0.36
		eval	124-173- 635	11.38	±0.12	±0.19	±0.23	±0.30
		dev+eval	226-333-1219	11.41	±0.07	±0.11	±0.13	±0.17
NAB-65k	WW	dev	128-104- 546	10.53	±0.15	±0.23	±0.28	±0.36
		eval	163-112- 583	10.47	±0.12	±0.19	±0.22	±0.29
		dev+eval	291-216-1129	10.50	±0.07	±0.11	±0.13	±0.17
	XW	dev	115- 93- 462	9.07	±0.13	±0.21	±0.25	±0.33
		eval	126-108- 532	9.35	±0.11	±0.17	±0.20	±0.27
		dev+eval	241-201- 994	9.27	±0.06	±0.10	±0.12	±0.15

6.2.3 Broadcast News Transcription Task (HUB-4)

The 1996 BROADCAST NEWS SPEECH CORPUS consists of American English transcribed television and radio broadcasts from ABC, CNN, and CSPAN television networks as well as NPR and PRI radio networks with corresponding transcripts. The recordings cover six different conditions ranging from clean to heavily degraded speech [Garofolo & Fiscus$^+$ 97]. The primary motivation for this collection was to provide training data for the DARPA HUB-4 Project on continuous speech recognition in the broadcast domain. Table 6.11 contains some corpus statistics.

The speech recognition system built for the HUB-4 task uses 4000 decision-tree based, gender-dependent across-word triphone states plus one state for silence. For each gender, two systems were trained that differ with respect to the variance tying scheme. The first system uses a single, globally pooled diagonal covariance matrix and the second system employs density-specific diagonal covariance matrices. Generalized triphone states are modeled with Gaussian mixture

Table 6.11: Corpus statistics for the HUB-4 data collection for the recognition of speech in the broadcast domain. The collection covers six different conditions ranging from clean to heavily degraded speech.

	HUB-4			
	training		test	
	male	female	male	female
amount of acoustic data [h]	63:29	32:50	1:48	1:06
silence portion [%]	14.8	14.5	16.0	10.5
# speakers	275	274	*N.N.*	*N.N.*
# sentences	17 148	8 914	396	261
# words	698 609	350 917	19 497	12 196
# phonemes	2 651 125	1 362 770	75 803	47 721
# lexicon words	66 269			
# pronunciations	81 931			
unknown words [%]	0.0	0.0	1.0	1.2
perplexity (trigram LM)	225.4	240.8	221.3	207.8

distributions. The system trained on the male speaker portion uses either 358k densities with a single, globally pooled variance vector, or 241k densities with a total of 62k diagonal covariance matrices. Similarly, the system trained on the female speaker portion uses either 246k densities with a globally pooled diagonal variance, or 148k densities tied over approximately 33.8k variances. The observation vectors used in either system consist of 16 MFCCs. Each 9 consecutive input frames are concatenated to a $9 \times 16 = 144$ dimensional super vector and projected into a 45 dimensional subspace using a LDA transformation. In case of the multi-variance systems, the 45 dimensional feature space is further transformed via an MLLT which is applied on top of the LDA matrix.

Table 6.12: HUB-4 baseline recognition results produced with a system using 4000 decision-tree based, gender-dependent across-word triphone states plus one state for silence. States are modeled as Gaussian mixture densities that share a single, globally tied diagonal covariance matrix. Other than the lattices generated for the male speaker portion, the word lattices for the female speaker portion were built using an integrated unigram language model (cf. Section 6.3.11). As a consequence, the average graph densities are substantially smaller compared to trigram-decoded lattices.

	HUB-4, pooled variance					
	train			test		
	male	female	m+f	male	female	m+f
avg. #arcs per spoken word	221.35	27.07	115.61	146.53	52.76	109.28
avg. #arcs per best rec. word	135.18	17.45	71.23	96.54	35.40	72.25
avg. #arcs per time frame	24.63	10.86	19.93	45.52	23.71	37.21
GER [%]	6.06	2.73	4.95	10.70	10.34	10.56
PER [%]	10.86	10.67	10.80	15.72	14.32	15.18
WER [%]	15.13	18.77	16.35	23.39	20.93	22.44
SER [%]	83.76	89.45	85.71	92.68	88.12	90.87

All systems were trained under the ML criterion in the Viterbi approximation. Tables 6.12 and 6.13 show the baseline results together with the average word graph densities for the numerator and the denominator lattices. Table 6.14 shows the baseline error rates for the multi-variance system separated by gender and recording conditions. Confidence interval estimations are shown in Table 6.15.

Table 6.13: HUB-4 baseline recognition results produced with a system using 4000 decision-tree based, gender-dependent across-word triphone states plus one state for silence. States are modeled as Gaussian mixture densities with density specific, diagonal covariance matrices.

	HUB-4, density-specific variances					
	train			test		
	male	female	m+f	male	female	m+f
avg. #arcs per spoken word	221.35	35.68	121.50	377.54	52.76	248.52
avg. #arcs per best rec. word	133.29	24.65	75.33	246.56	35.40	162.54
avg. #arcs per time frame	24.63	14.99	21.34	109.43	23.71	76.78
GER [%]	6.06	2.08	4.73	9.71	10.34	9.96
PER [%]	10.96	12.53	11.50	15.55	14.37	15.09
WER [%]	15.36	22.17	17.64	22.82	20.65	21.98
SER [%]	83.63	91.17	86.21	92.68	89.27	91.32

Table 6.14: HUB-4 baseline results. Both systems employ across-word models with density-specific, diagonal variances estimated in an LDA-MLLT transformed feature space. Word error rates are shown separately for each gender and broken down by recording conditions. The recording conditions range form clean speech (f0) to heavily degraded speech (fx).

		HUB-4							
gender	dns/mix	f0	f1	f2	f3	f4	f5	fx	all
male	1.00	18.44	30.25	37.82	36.42	29.71	34.18	41.55	28.31
	1.99	17.14	27.49	36.72	32.55	28.57	32.81	39.72	26.63
	3.89	16.05	26.19	35.03	32.02	27.23	28.71	37.68	25.27
	6.92	15.08	25.21	34.01	32.78	26.46	26.95	35.70	24.33
	10.53	14.60	24.89	32.59	31.79	25.68	24.41	34.23	23.50
	13.22	13.97	24.58	32.16	31.79	25.99	22.85	33.59	23.07
	14.77	14.17	24.54	32.03	31.79	25.48	23.24	33.31	23.05
	15.56	14.03	24.54	31.49	32.40	24.65	22.65	32.89	22.81
female	1.00	14.32	28.23	38.65	30.45	29.48	27.52	39.62	24.21
	1.99	13.27	26.37	36.15	28.64	27.79	26.85	38.52	22.75
	3.62	12.50	25.55	35.69	30.45	27.33	26.17	36.89	21.99
	5.58	12.27	24.65	35.01	30.00	26.33	22.82	35.06	21.28
	7.09	12.15	24.42	34.78	28.18	26.25	24.16	34.24	21.05
	7.93	11.68	24.56	35.12	31.82	24.48	22.82	34.70	20.82
	8.32	11.55	24.53	34.89	30.45	24.87	22.82	34.52	20.74
	8.48	11.75	24.16	34.09	30.45	24.87	22.82	33.88	20.61

Table 6.15: Confidence intervals estimated on the HUB-4 corpus using the bootstrap re-sampling normal approximation method.

			HUB-4					
			baseline		confidence intervals			
set	var. model	gender	del- ins- sub	WER[%]	70%	90%	95%	99%
test	pooled	male	939- 852-2770	23.39	±0.10	±0.15	±0.18	±0.24
		female	597- 352-1636	20.93	±0.09	±0.15	±0.18	±0.23
		m+f	1536-1204-4406	22.44	±0.15	±0.23	±0.28	±0.36
	mix.-spec.	male	1039- 760-2650	22.82	±0.09	±0.15	±0.17	±0.23
		female	646- 328-1576	20.65	±0.08	±0.13	±0.16	±0.21
		m+f	1685-1088-4226	21.98	±0.13	±0.21	±0.25	±0.33

6.3 Experiments

Experiments for lattice-based discriminative training were performed on the WSJ-5k corpus, the NAB-20k corpus, the NAB-65k corpus, and the HUB-4 corpus. If not stated otherwise, discriminative training runs always start from the ML-trained baseline as described in Section 6.2.

6.3.1 Initial Settings and Defaults

To describe the experimental setups as briefly as possible, some settings and parameters are only specified if they differ from a default. Default settings are as follows:

- *Exact match alignment versus constrained search*: exact match is default.
- *Language model used in decoding*: if not stated otherwise, all language models used for decoding the training and the test data are trigram language models.
- *Language model re-scoring in training*: if not stated otherwise, the language models used to re-score training lattices are unigram language models. The unigram language model uses the same scaling factor and the same word insertion penalties as the language model used for generating the lattices.
- *Lattice generation*: all lattices are generated in a one-pass integrated trigram search. The lattices used in discriminative training are re-scored with a unigram language model. The unigram lattice re-scoring leaves the lattice structure unchanged, i.e, after re-scoring, all lattice nodes remain connected with the same adjacent nodes to which they were connected before the re-scoring.
- *Acoustic scaling factor* κ: the scaling factor κ is set to the inverse of the language model scaling factor. This yields 1.0 for SIETILL; 1/18 for WSJ-5k; 1/19 for both the NAB-20k and the NAB-65k within-word system, and 1/22 for the respective across word systems. The gender-dependent HUB-4 recognition systems use a scaling factor of 1/20.
- Choice of iteration constants: if not stated otherwise, the default method for choosing the iteration constants is the Λ method as described in Section 4.3.2.
- *Optimization Method*: If not stated otherwise, all discriminatively trained systems are optimized with the *Extended Baum* (EB) algorithm. By default, the iteration constants are scaled with a factor $h = 1.1$.

6.3.2 Exact Match Alignment vs. Constrained Search

Experiments comparing the exact match alignment with the constrained search algorithm were conducted on the WSJ-5k corpus. Table 6.16 shows the results for either keeping the time boundaries (Viterbi segmentation points) of word hypotheses represented in a lattice a constant or allowing the search algorithm to refine the time boundaries before the forward-backward probabilities are estimated. The former method is called *exact match alignment* while the latter is referred to as *constrained search*. In contrast to a full search, the HMM network for the denominator model in the constrained search is restricted to those hypotheses that are encoded in a word lattice. Both the exact match alignment and the constrained search cannot hypothesize word sequences other than those for which there is a corresponding path in the input word lattice. Figure 6.1 shows the evolution of the MMI criterion in the course of the iteration index. The corresponding word error rates measured on the combined development and evaluation set are shown in Figure 6.2 and listed in Table 6.16.

Although, Table 6.16 shows a slightly better performance for the exact match alignment over the constrained search, there is no statistically significant difference between either method. The exact match alignment is, however, substantially faster than the constrained search algorithm and allows for reducing the total training time by more than a factor of two. Hence, if not stated otherwise, the exact match alignment is used for all further experiments reported in this chapter.

Table 6.16: Comparison of the *exact match alignment* with the *constrained search algorithm* on the WSJ-5k corpus.

	WSJ-5k								
	development			evaluation			dev + eval		
	PER	WER	SER	PER	WER	SER	PER	WER	SER
ML	5.61	4.72	41.2	4.98	3.90	36.7	5.33	4.36	39.2
MMI exact match	5.29	4.26	39.0	4.81	3.64	34.9	5.08	3.99	37.2
MMI constrained search	5.38	4.33	39.8	4.87	3.72	34.6	5.15	4.06	37.4

6.3.3 Increasing the Model Complexity

The following experiments show how the performance of a discriminative training under the MMI criterion is affected by changing the model complexity. The initial model is adopted from the WSJ-5k system which is a fairly simple within-word speech recognition system. The model complexity is then gradually increased by first enlarging the recognition vocabulary from 5k to 20k words and switching to the NAB-20k corpus using different ML-trained parameter sets. Settings that are varied include the dimensionality of the LDA transformation matrix, the number of generalized triphone states, and the total number of mixture densities. The model mismatch due to the presence of out-of-vocabulary words is further reduced by switching from the 20k to the 65k recognition vocabulary. Finally, coarticulatory effects are explicitly captured by using across-word models. All experiments pursue the question as to what extent the performance of a discriminatively trained system is affected by the initial setting of the ML-trained parameter set and how the presence or absence of a certain model changes the recognition rates thus obtained.

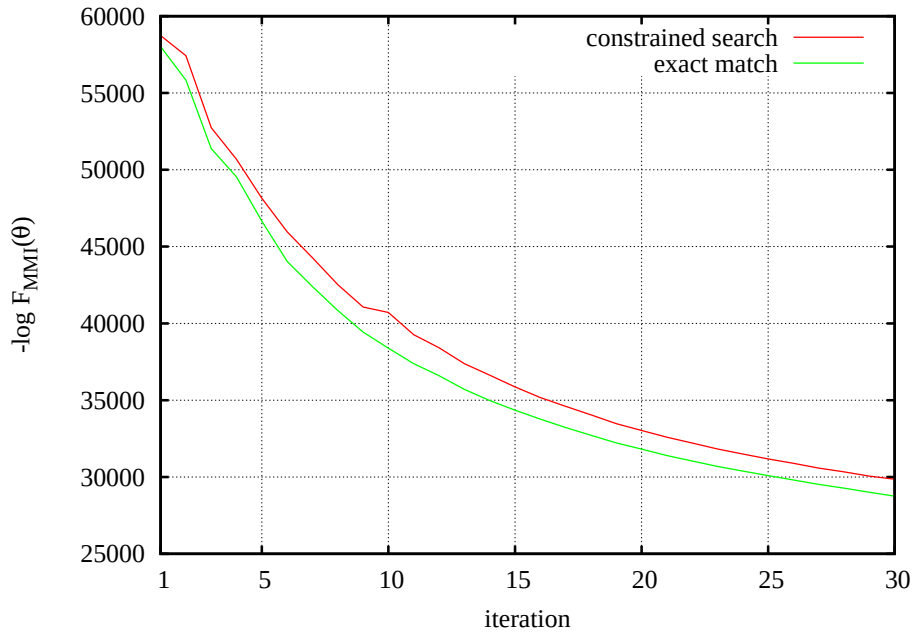


Figure 6.1: Evolution of the MMI criterion as a function of the iteration index for the *exact match alignment* and the *constrained search algorithm* on the combined development + evaluation data of the WSJ-5k corpus.

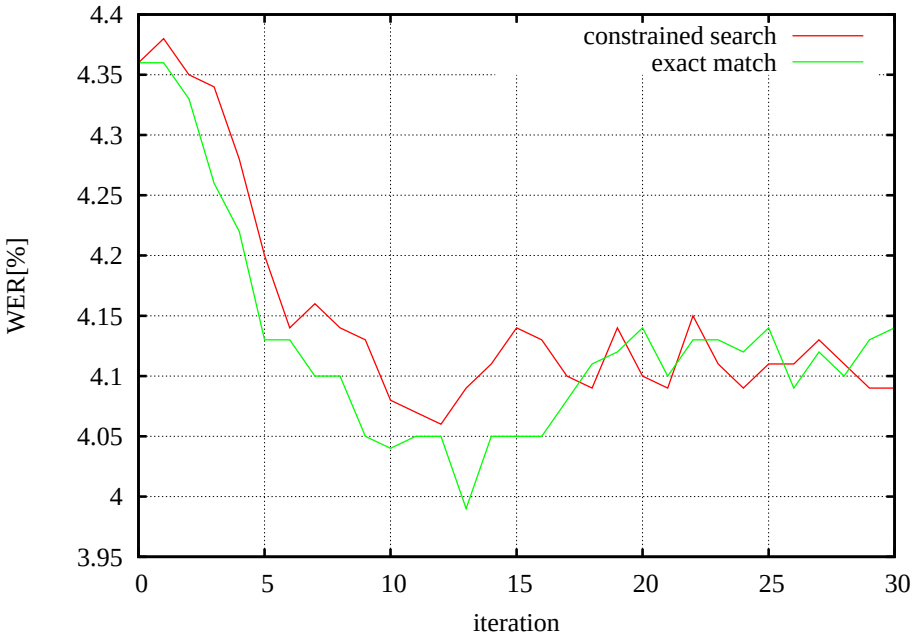


Figure 6.2: Evolution of the word error rate as a function of the iteration index for the *exact match alignment* and the *constrained search algorithm* on the combined development + evaluation data of the WSJ-5k corpus.

Increasing the Vocabulary Size

The acoustic model employed by the WSJ-5k recognition system was trained on a comparatively small amount of training data. In this Section, the effect of increasing the amount of training data as well as the vocabulary size is investigated by switching from the WSJ-5k corpus to the NAB-20k corpus. To make the recognition results of both systems comparable, the acoustic model trained for the WSJ-5k system is plugged into the NAB-20k system. Table 6.17 shows the results in terms of word error rates for the ML trained baseline systems and the discriminative system optimized under the MMI criterion. In both cases the discriminatively trained system improves the word error rate between 3.6% and 6.8% relative over the corresponding ML baseline. However, despite the significant decrease in terms of word error rate obtained on the WSJ-5k corpus by the discriminative model, it does not improve over the ML-trained acoustic model if the latter is trained on a substantially larger amount of training data as, in this example, the NAB-20k corpus.

Table 6.17: Recognition results on the NAB-20k test corpus using acoustic models trained on the WSJ-5k training data and the NAB-20k training data, respectively. Both models are optimized under the ML and the MMI criterion and are applied to the NAB-20k test corpus.

		NAB-20k								
		development			evaluation			dev + eval		
train	criterion	PER	WER	SER	PER	WER	SER	PER	WER	SER
WSJ-5k	ML	12.70	14.12	78.71	12.63	14.48	79.11	12.67	14.31	78.91
	MMI	12.45	13.69	79.03	12.35	13.90	79.11	12.40	13.80	79.07
NAB-20k	ML	9.40	12.93	79.68	8.99	12.75	79.75	9.18	12.84	79.71
	MMI	9.44	12.08	75.48	8.83	11.88	75.32	9.12	11.97	75.40

MMI Training Starting from an Improved ML Baseline

The following experiment investigates to what extent the MMI criterion is able to outperform the initial ML-trained error rate if an improved ML-trained baseline system is used as starting point. Experiments were conducted on the NAB-20k corpus. Other than the baseline setting described in Section 6.2, the improved system uses 2500 generalized within-word triphone states with a larger set of 522k Gaussian mixture densities. Another difference is the LDA transformation matrix which now maps a 99-dimensional super vector (obtained by concatenating each 3 adjacent observation vectors) into a 33-dimensional feature space. Compared to the previous system, the improved system achieves a word error rate of 12.39%, which is a modest gain of 0.2% absolute over the previous baseline system. Despite the improvement in the initial system, the MMI criterion is still able to further reduce the word error rate by almost 1% on the combined development and evaluation set. Hence, the improvements obtained with an MMI training are additive and do not seem to be affected by changes in the number of generalized n-phone states or the estimation of the LDA transformation matrix. Figure 6.3 shows the evolution of the MMI criterion in the course of the training iterations. The corresponding word error rates measured on the combined development plus evaluation set are shown in Table 6.4. We also note that the recognition performance over the first five iterations of the discriminative training run initially deteriorates the test set performance which may be due to the larger number of mixture densities used for the acoustic model. Many of these densities seem to be redundant and are not necessary to achieve a high recognition performance. As shown in Figure 6.5, the discriminative training decreases the number of mixture components in each iteration until it achieves a more appropriate model complexity of less than 490k densities.

Table 6.18: Baseline results obtained on the NAB-20k corpus with an improved, ML-trained, within-word recognition system.

	NAB-20k			
	train	dev	eval	dev + eval
avg. #arcs per spoken word	186.94	65.70	66.16	65.93
avg. #arcs per best rec. word	144.93	51.71	53.28	52.50
avg. #arcs per time frame	81.52	34.48	36.75	35.67
GER [%]	0.0	5.06	4.37	4.70
SER [%]	49.58	76.45	78.16	77.32
WER [%]	5.30	12.25	12.52	12.39
PER [%]	2.08	9.04	8.85	8.94

Table 6.19: Recognition results on the NAB-20k corpus for a discriminative training under the MMI criterion on top of an improved ML-trained within-word baseline system.

	NAB-20k								
	development			evaluation			dev + eval		
	PER	WER	SER	PER	WER	SER	PER	WER	SER
ML	9.04	12.25	76.45	8.85	12.52	78.16	8.94	12.39	77.32
MMI	8.85	11.56	74.52	8.31	11.36	76.58	8.56	11.46	75.56

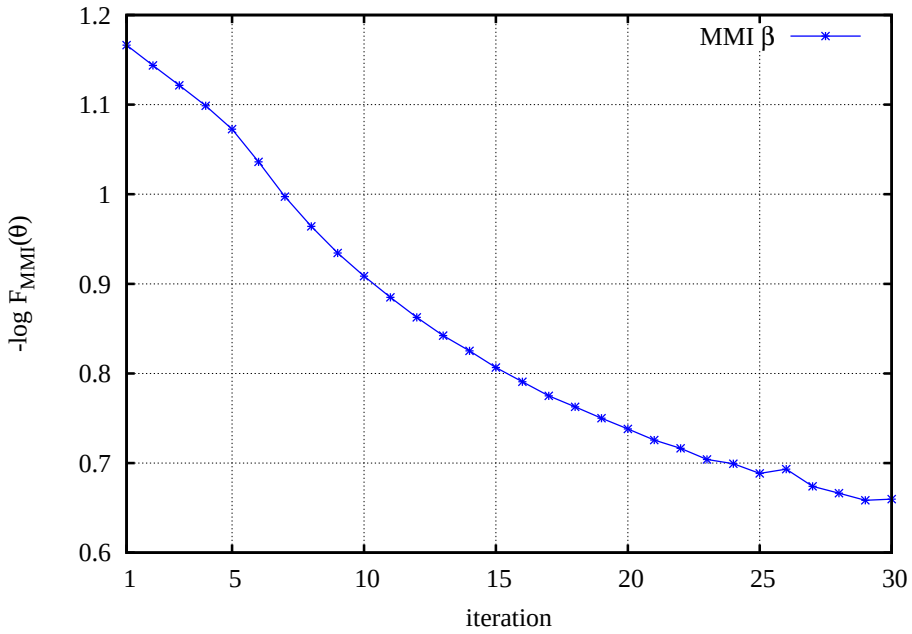


Figure 6.3: Evolution of the MMI objective function on the combined development + evaluation set in the course of the iteration index starting from an improved ML-trained baseline system for the NAB-20k corpus.

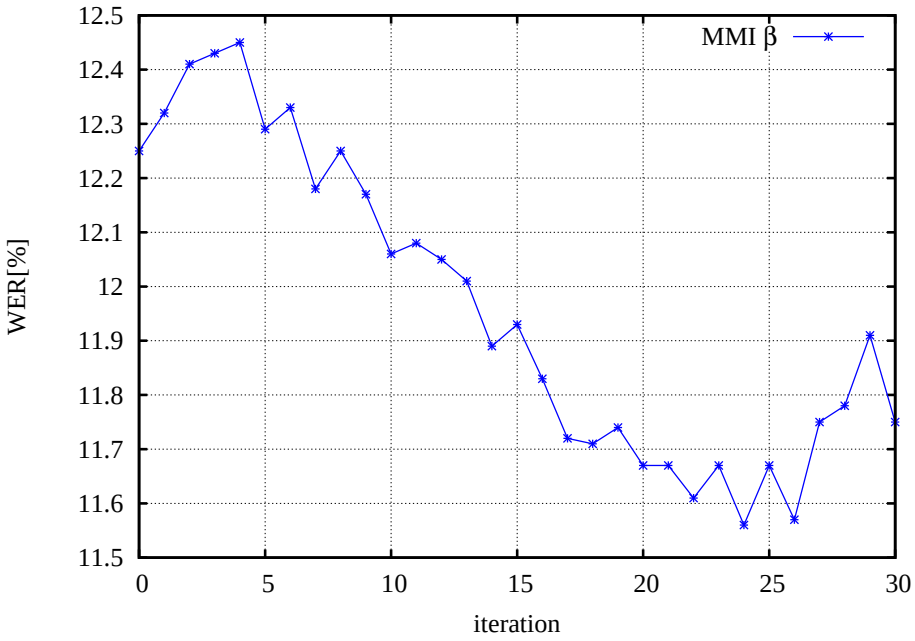


Figure 6.4: Evolution of the word error rate on the combined development + evaluation set in the course of the iteration index starting from an improved ML-trained baseline system for the NAB-20k corpus.

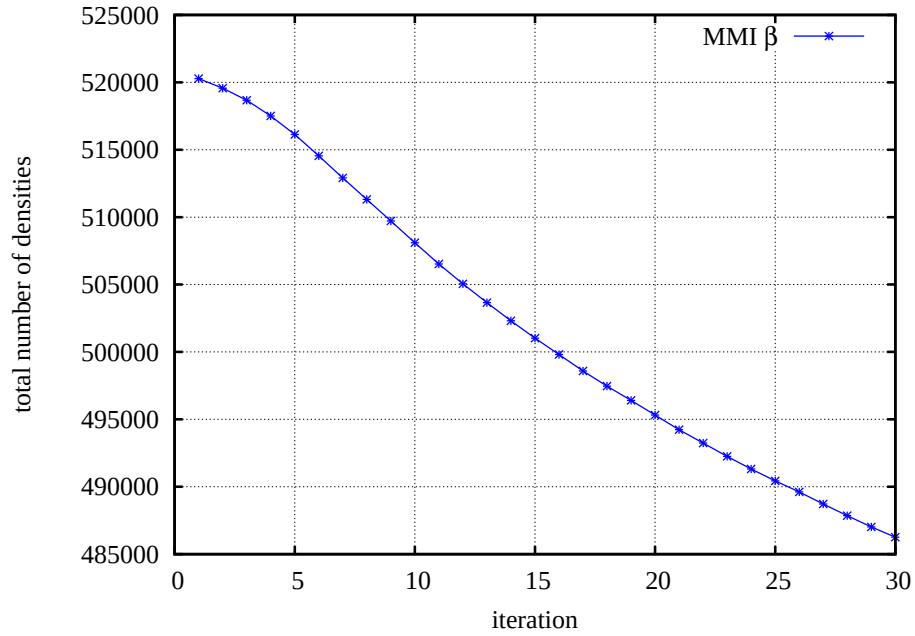


Figure 6.5: Evolution of the number of densities in the course of the iteration index for a discriminative training under the MMI criterion starting from an improved ML-trained baseline system on the NAB-20k corpus.

6.3.4 Optimization Using the Fast Method

The *fast* method proposed in Section 4.3.3 is an optimization procedure that can lead to significantly faster convergence rates compared to traditional optimization schemes as proposed in [Schlüter & Macherey$^+$ 01] and [Woodland & Povey 02]. Although the *fast* method is based on the same heuristic according to which the step size is chosen such that the variances in the update-equations are positive definite, it cannot be derived from the auxiliary function used in the Extended Baum algorithm. Figure 6.6 shows the evolution of the MMI objective function in the course of the iteration index on the NAB-20k corpus using the conventional β method as suggested in [Schlüter & Macherey$^+$ 01] and, starting from the 6th iteration on, using in addition the *fast* method. Figure 6.7 shows the corresponding word error rates. The *fast* method is able to optimize the training criterion significantly faster than the β optimization scheme. It does not, however, result in a statistically significant gain on the development set as shown in Table 6.22. Note that the experiments were conducted on a different ML-trained baseline system. The corresponding baseline is shown in Table 6.21. Table 6.20 lists the names for the different optimization schemes.

Table 6.20: Naming scheme for optimization methods used in discriminative training.

Optimization Method	Reference
E	[Woodland & Povey 02] (cf. Sec. 4.3.1, p. 45)
Λ	cf. Sec. 4.3.2, p. 46
β	[Schlüter & Macherey$^+$ 01] (cf. Sec. 4.3.1, p. 45)
fast	cf. Sec. 4.3.3, p. 47

Table 6.21: Baseline recognition results obtained with a within-word system on the NAB-20k corpus. The baseline serves as starting point for the investigation of the *fast* training method.

	NAB-20k	
	train	dev
avg. #arcs per spoken word	195.9	72.4
avg. #arcs per best rec. word	151.7	56.2
avg. #arcs per time frame	80.5	36.9
GER [%]	0.0	4.9
SER [%]	64.6	78.4
WER [%]	8.3	12.6
PER [%]	2.9	9.1

Table 6.22: Fast discriminative training with the MMI criterion on the NAB-20k corpus.

	NAB-20k development		
	PER	WER	SER
ML	9.1	12.6	78.4
MMI beta	8.7	11.6	76.1
MMI fast	8.7	11.5	75.2

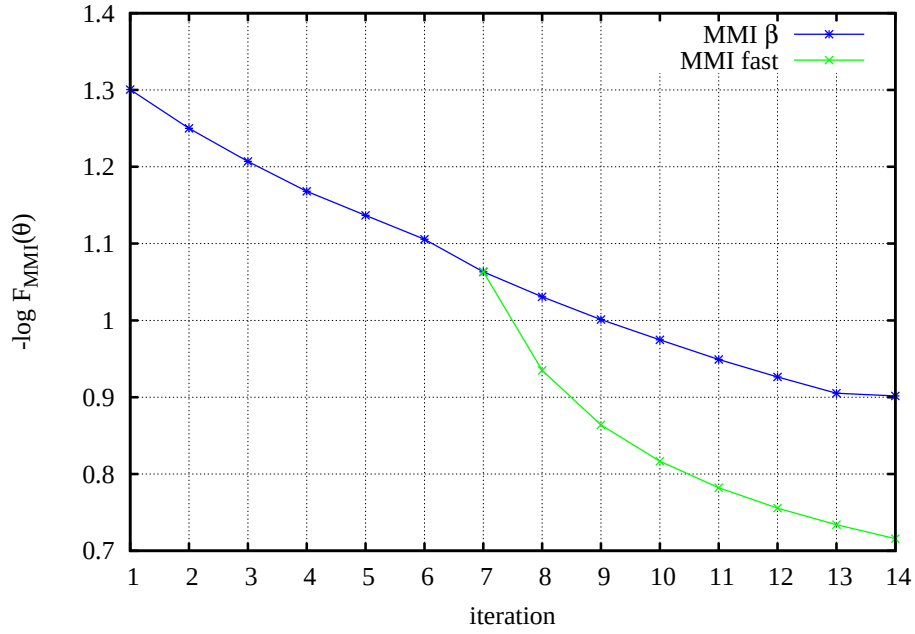


Figure 6.6: Evolution of the MMI objective function in the course of the iteration index on the NAB-20k corpus using the conventional β method and, starting from the 6th iteration on, using in addition the *fast* method.

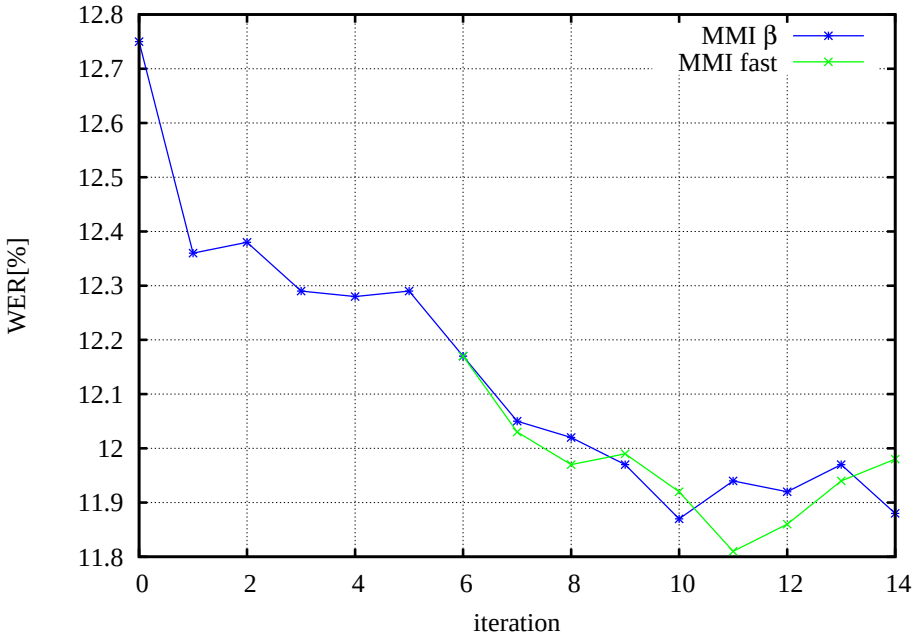


Figure 6.7: Evolution of the word error rate in the course of the iteration index on the NAB-20k corpus using the conventional β method and, starting from the 6th iteration on, using in addition the *fast* method.

Effect of Out-of-Vocabulary Words

A large share of recognition errors in automatic speech recognition systems is caused by unknown words which are not represented in the pronunciation dictionary. Typically, unknown words trigger additional errors in their neighborhood because the recognizer may hypothesize other words as surrogates to explain the acoustic observations which then expose the wrong context to the language model. This may even further reduce the likelihood of sentence hypotheses that will have a low word error rate if their acoustic model scores do not compensate for the higher language model costs.

To measure how unknown words affect the performance of a discriminatively trained system, the acoustic models obtained from the improved baseline system (cf. the previous section) were used to decode the test set of the Nab-20k corpus with an increased pronunciation dictionary comprising 65k words. Due to the enlarged recognition vocabulary the lattices for the development and the evaluation data had to be rebuilt. The resulting word graph densities and error rates are shown in Table 6.23.

Table 6.24 shows the recognition results after increasing the vocabulary size for the Nab-20k system from 20k to 65k words. Both the Nab-20k system and the Nab-65k system use the same acoustic model which was trained under the MMI criterion on the Nab-20k training data. Only the size of the recognition vocabulary as well as the language model were changed. As was the case for the Nab-20k system, the improvement of more than 1% absolute over the ML-trained baseline still carries over to the Nab-65k system. This indicates that an acoustic model trained under the MMI criterion does not necessarily achieve improved recognition accuracies for words which are in the direct neighborhood of unknown words.

Table 6.23: ML-trained baseline results on the Nab-65k recognition tasks using the improved acoustic model from Table 6.18 and an enlarged vocabulary size of 65k words.

	Nab-65k			
	train	dev	eval	dev + eval
avg. #arcs per spoken word	186.94	62.35	66.84	64.62
avg. #arcs per best rec. word	144.93	50.34	54.50	52.44
avg. #arcs per time frame	81.52	34.15	37.22	35.75
GER [%]	0.0	2.49	2.34	2.41
SER [%]	49.58	71.94	74.37	73.16
WER [%]	5.30	10.32	10.31	10.31
PER [%]	2.08	6.06	6.12	6.09

Table 6.24: Discriminative training based on the MMI criterion using the improved ML trained within-word baseline system from Table 6.23.

	Nab-65k								
	development			evaluation			dev + eval		
	PER	WER	SER	PER	WER	SER	PER	WER	SER
ML	6.06	10.32	71.94	6.12	10.31	74.37	6.09	10.31	73.16
MMI	5.80	9.27	66.45	5.57	9.14	70.89	5.68	9.20	68.69

Across-Word Modeling

Table 6.25 shows the recognition performance on the NAB-20k corpus if within word models are replaced with across word models. Compared to the ML-trained baseline system, which achieves a WER of 11.47%, the discriminative training under the MMI criterion is able to further reduce the word error rate by 0.43% absolute. Although this gain is still substantial, the MMI criterion does not achieve the 1% improvement on top of the ML-trained system that can be achieved if within-word models are used. This indicates that a discriminative training under the MMI criterion alleviates some of the effects that are caused by using simpler word boundary models. The same observation can be made when the discriminatively trained acoustic models are applied to the NAB-65k recognition task as shown in Table 6.26. Here, the improvement on the combined development and evaluation set yields 0.35% absolute in terms of word error rate. For comparison, both tables also show results for the MCE criterion, the MPE criterion, and the MWE criterion.

Table 6.25: Discriminative training using the *Maximum Mutual Information* (MMI) criterion, the *Minimum Classification Error* (MCE) criterion, the *Minimum Phone Error* (MPE) criterion, and the *Minimum Word Error* (MWE) criterion on top a *Maximum Likelihood* (ML) trained baseline across-word system for the NAB-20k corpus.

	NAB-20k								
	development			evaluation			dev + eval		
	PER	WER	SER	PER	WER	SER	PER	WER	SER
ML	8.65	11.45	73.87	8.47	11.38	76.27	8.55	11.41	75.08
MMI	8.54	11.11	72.58	8.24	11.09	74.68	8.38	11.10	73.64
MCE	8.49	11.11	74.19	8.18	10.97	75.32	8.32	11.04	74.76
MPE	8.55	11.01	72.58	8.25	11.03	73.73	8.39	11.02	73.16
MWE	8.50	11.15	72.90	8.24	11.02	74.68	8.36	11.08	73.80

Table 6.26: Discriminative training using the *Maximum Mutual Information* (MMI) criterion, the *Minimum Classification Error* (MCE) criterion, the *Minimum Phone Error* (MPE) criterion, and the *Minimum Word Error* (MWE) criterion on top a *Maximum Likelihood* (ML) trained baseline across-word system for the NAB-65k corpus.

	NAB-65k								
	development			evaluation			dev + eval		
	PER	WER	SER	PER	WER	SER	PER	WER	SER
ML	5.52	9.07	66.77	5.70	9.35	71.20	5.61	9.22	69.01
MMI	5.46	8.99	66.77	5.50	9.02	69.62	5.48	9.01	68.21
MCE	5.35	8.81	67.10	5.45	9.04	69.30	5.41	8.93	68.21
MWE	5.39	8.81	66.13	5.53	9.18	70.25	5.46	9.01	68.21
MPE	5.50	8.84	67.10	5.44	8.97	68.99	5.47	8.91	68.05

Training with Increased Vocabulary Sizes

In the previous subsection the effect of optimizing across-word models with different discriminative training criteria was investigated. This section investigates how different vocabulary sizes affect the recognition performance. As before, the investigation is conducted on the NAB-20k and the NAB-65k corpus for both a within-word system and an across-word system. The central question guiding the experiments is whether a discriminative training can benefit from an increased vocabulary size in training as this may help to improve generalization. Table 6.27 shows the recognition results when the vocabulary sizes in training and recognition are increased. The third column, "train", shows the size of the recognition vocabulary used for generating the training lattices, and the fourth column, "test", indicates the vocabulary size used to decode the test utterances. Note that only a discriminative training may benefit from an increased training vocabulary whereas the ML criterion is not able to exploit an enlarged pronunciation dictionary during the acoustic model training.

Although one could expect that the increased vocabulary size in training would be reflected in an improved recognition performance, Table 6.27 clearly shows that there is almost no difference in the word error rates. This is partially an effect of the pronunciation dictionary used in training which is already nearly closed if the 20k lexicon is used. Hence, increasing the vocabulary size does not lead to improved generalization. Even if the additional words in the pronunciation lexicon are acoustically similar to entries in the 20k word-comprising subset, there will be no additional gain since the models for the correct and the competing classes are discriminated on a sub word level for which the similarity is typically much higher than on the word level (cf. the phoneme error rates in Table 6.27 which are significantly lower than the corresponding word error rates). Since the gradient in the optimization procedure is determined by the state occupancy probabilities, its direction is primarily affected by recognition errors on a sub-phoneme level rather than on a word level. The outcome of this experiment might, however, show different results, if the language model parameters were also subject to a discriminative training procedure rather than being kept fixed.

Table 6.27: Discriminative training under the MMI criterion on the NAB-65k corpus on top of a ML-trained within-word system and a ML-trained across-system, respectively. The table shows the effect that using different vocabulary sizes in training has on recognition performance.

				NAB-20k, NAB-65k								
		vocab. size		development			evaluation			dev + eval		
model	crit.	train	test	PER	WER	SER	PER	WER	SER	PER	WER	SER
WW	ML	20k	20k	9.40	12.93	79.68	8.99	12.75	79.75	9.18	12.84	79.71
			65k	6.27	10.53	74.84	6.21	10.47	75.63	6.23	10.50	75.24
	MMI	20k	20k	9.27	12.02	75.48	8.65	11.75	75.32	8.94	11.88	75.40
			65k	6.25	10.02	70.65	5.92	9.50	70.25	6.08	9.74	70.45
		65k	20k	9.22	11.99	75.81	8.69	11.86	75.95	8.94	11.93	75.88
			65k	6.21	9.91	71.29	5.94	9.59	70.25	6.06	9.74	70.77
XW	ML	20k	20k	8.65	11.45	73.87	8.47	11.38	76.27	8.55	11.41	75.08
			65k	5.52	9.07	66.77	5.70	9.35	71.20	5.61	9.22	69.01
	MMI	20k	20k	8.54	11.11	72.58	8.24	11.09	74.68	8.38	11.10	73.64
			65k	5.46	8.99	66.77	5.50	9.02	69.62	5.48	9.01	68.21
		65k	20k	8.52	11.29	72.90	8.27	11.11	74.37	8.39	11.19	73.64
			65k	5.38	8.83	66.45	5.50	9.07	68.99	5.44	8.95	67.73

6.3.5 Model Complexity versus System Performance

The previous sections investigated how increasing the model complexity affects the performance of a discriminative training. Figure 6.8 illustrates this gain in a learning curve. While the change in terms of word error rate for many system parameters such as the dimensionality of the feature space, the number of mixture densities, and the number of generalized n-phone states is additive when switching from an ML-trained system to a discriminatively trained system, the use of across-word models results in a reduced performance gain. Hence, the absence of an explicit model that allows for capturing inter-word coarticulation can partially be compensated by a discriminative training procedure.

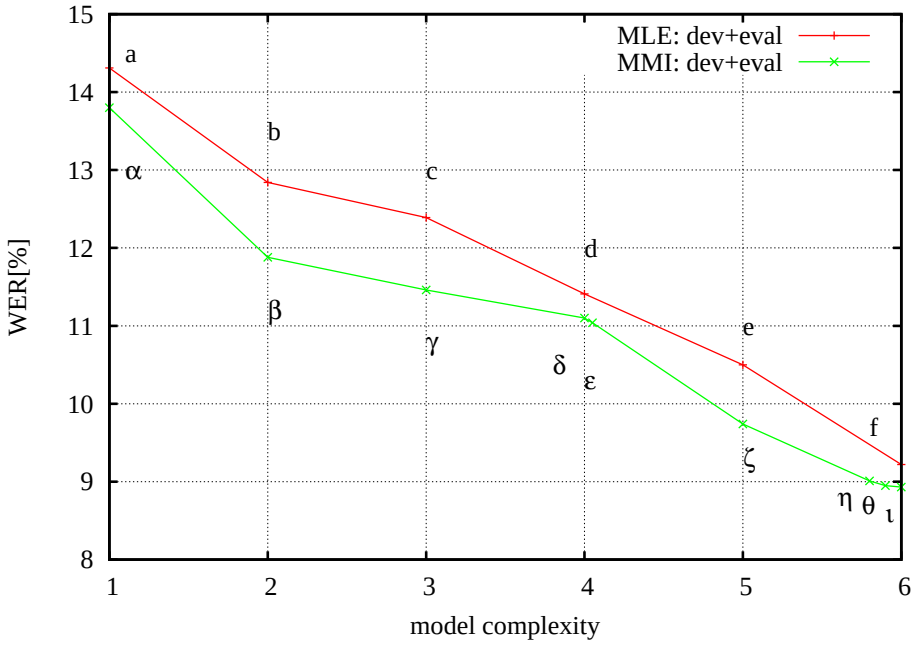


Figure 6.8: Learning curve on the Wall Street Journal corpora. Illustrated is the performance of various ML-trained systems, which differ with respect to their model complexity, and corresponding systems trained under the MMI criterion. The model complexity increases from left to right. The ML-trained baseline systems are defined as follows: (a) shows the performance in terms of *word error rate* (WER) if the acoustic model trained on the Wsj-5k corpus is applied to the Nab-20k corpus; (b) is a within-word system trained on the Nab-20k corpus; (c) is an improved version of system (b); (d) shows the transition from within-word to across-word models; (e) depicts the performance of the within-word system as defined under (b) but using the 65k recognition vocabulary; and (f) shows the performance of the across-word system using the 65k words comprising recognition vocabulary of the Nab-65k corpus. The corresponding discriminatively trained systems were optimized under the MMI criterion (α-δ,ζ-θ) or the MCE criterion (ϵ,ι), respectively. System (θ) uses the 65k words recognition vocabulary also in training.

Comparison with CU-HTK

Table 6.28 compares the performance gains achieved on the NAB-65k test corpus with results published in [Povey & Woodland 02, Povey 04] by the *University of Cambridge* (CU). The Cambridge recognition systems is based on the HTK toolkit [Young & Evermann+ 06] and uses 6399 decision-tree based, gender independent 3-state HMM across-word triphone models. States are modeled as Gaussian mixture distributions with density-specific variances and 12 densities per mixture. The observation vectors consist of 12 cepstral coefficients plus the log energy as well as the first and the second order derivatives, resulting in a 39 dimensional feature vector. Each three adjacent input frames are concatenated to a $3 \times 39 = 117$ dimensional super vector and projected into a 32 dimensional subspace via an LDA transformation. The baseline recognizer employs a trigram language model. Table 6.28 shows results obtained with the MMI criterion, the MCE criterion, the MWE criterion, and the MPE criterion.

Table 6.28: Comparison of recognition results obtained with the RWTH speech recognizer and the CU-HTK system on the NAB-65k corpus. Both systems use discriminatively trained across-word models.

	NAB-65k, WER[%]					
	RWTH			CU-HTK [Povey & Woodland 02] [Povey 04]		
criterion	dev	eval	dev + eval	dev	eval	dev + eval
ML	9.07	9.35	9.22	9.34	9.80	9.57
MMI	8.99	9.02	9.01	8.80	9.40	9.10
MMI (65k in train)	8.83	9.07	8.95			
MCE	8.81	9.04	8.93	–	–	–
MWE	8.81	9.18	9.01	–	–	–
MPE	8.84	8.97	8.91	8.70	9.29	8.99

6.3.6 Discriminative Training with Tied Covariance Matrices

Subject to the investigation in this section is the question whether a single variance vector pooled over all Gaussian densities can prevail over a mixture- or a density-specific variance model. Experiments conducted in the past at RWTH Aachen University were often inconclusive and did not result in a clear answer [Schlüter & Macherey+ 97]. In fact, many experiments even showed a performance loss when instead of a globally pooled variance model, mixture- or density-specific variances were estimated in an LDA-transformed feature space. As will be pointed out in Chapter 9, the main reason for this performance loss can be attributed to the diagonal modeling constraint which is typically imposed on the variance model if the amount of available training data is not sufficient to allow for estimating full covariance matrices. However, since the LDA transformation cannot diagonalize more than two covariance matrices simultaneously, the model assumption of diagonal variances is typically violated as soon as mixture specific variances are used. One way to alleviate this problem is to combine the LDA transformation matrix with a *Maximum Likelihood Linear Transform* (MLLT). The MLLT warps the feature space such that the loss between the full covariance model and the diagonal variance model is minimized[2]. The MLLT is a key component when using multiple variances in a speech recognition system. Multiple variances also affect the discriminative training procedure. As discussed in Chapter 4, the use of several variances may

[2] Details of the MLLT transformation are discussed in Chapter 9.

provide better constraints on how to choose the step size in the *Extended Baum* (EB) algorithm. In practice, the step size is typically chosen under the constraint that the resulting variances have to be positive (or positive definite in case that full covariance matrices are used). Therefore, the number of variance parameters in the acoustic model has a direct effect on the convergence speed and thus on the recognition performance. If only a single, globally pooled variance vector is used, the step sizes of hundreds of thousands of parameters have to be adjusted based on the small number of components in the variance vector. Typically, this increases the number of iterations required until the training criterion converges by a factor of 5 to 10. In contrast to that, the use of multiple variance vectors provides additional constraints on how to choose the iteration constants and thus allows for adjusting the step sizes more effectively. As a consequence, training times can be reduced and the number of iterations required until convergence is reached decreases substantially.

To investigate the effect of using a single, globally pooled variance vector compared to density specific variances, experiments were first conducted on the SieTill corpus for continuously spoken connected German digit strings. Table 6.29 shows the recognition results in terms of word error rates for different variance tying schemes and three methods on choosing the step size in a discriminative training under the MMI criterion. Both systems use an LDA transformation matrix. In case of density specific variances, the LDA is combined with a MLLT transformation matrix. Figure 6.9 depicts the evolution of the MMI training criterion in the course of the number of training iterations for various variance tying schemes and different methods for choosing the step size. Figure 6.10 shows the corresponding word error rates on the SieTill test corpus. The WERs obtained with this setting are the best results ever reported on this task.

Table 6.29: Recognition results for different variance tying schemes and three methods on choosing the step size for a discriminative training under the MMI criterion. Both systems use an LDA transformation matrix which, in case of the density specific variances, is combined with an additional MLLT transformation matrix.

male + female		WER[%]			
		ML	MMI		
# variance	# densities		D_{sl}	D_s	Δ_k
1+1	6.8k+6.8k	2.04	1.95	1.78	1.74
6.2k+6.2k	6.8k+6.8k	1.56	1.36	1.46	1.36

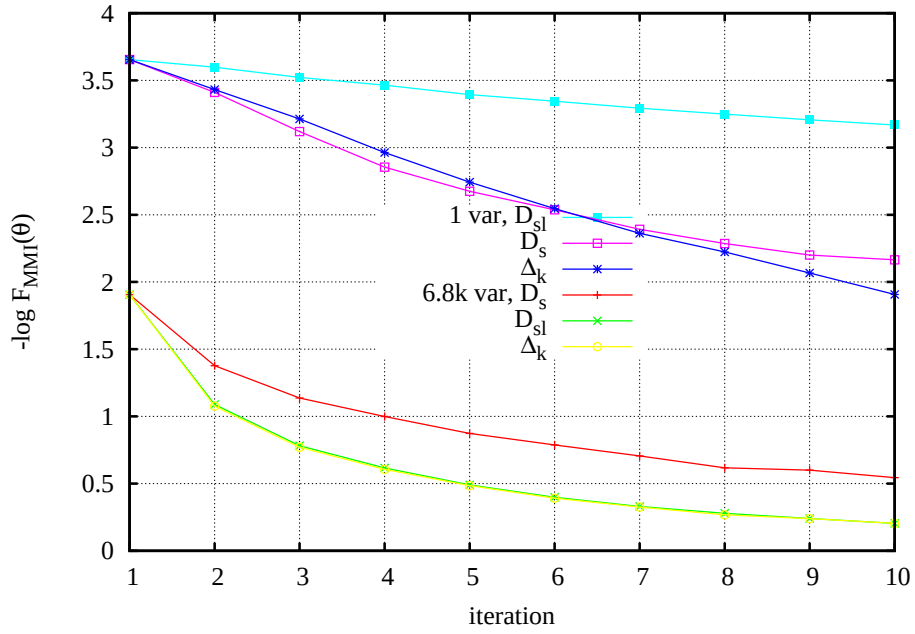


Figure 6.9: Evolution of the MMI training criterion on the male speaker portion of the SieTill corpus in the course of the number of training iterations for various variance tying schemes and different methods for choosing the step size.

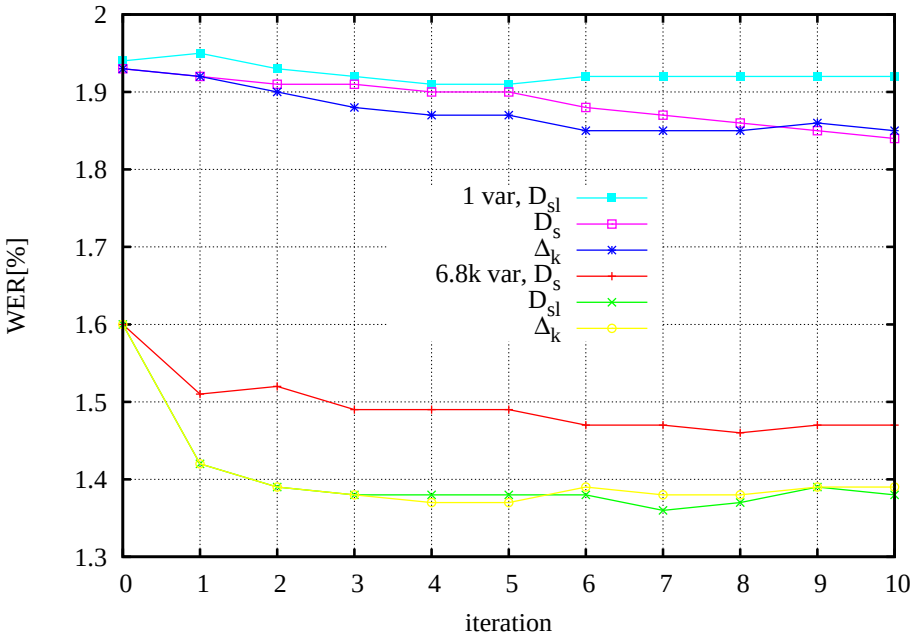


Figure 6.10: Evolution of the word error rate on the male speaker portion of the SieTill corpus in the course of the number of training iterations for various variance tying schemes and different methods for choosing the step size.

6.3.7 Large Scale Discriminative Training on Hub-4

The experiments described in this section were mainly conducted on the HUB-4 corpus. Subject to the investigation are the following aspects: the optimal word graph density that is required in lattice-based discriminative training to make the training procedure most effective; the choice of the iteration constants when different variance tying schemes are used; the question of how scaling the acoustic probabilities affects the estimation of the forward-backward probabilities; and how the variance re-estimation affects the recognition performance. Since the comparison of different tying schemes is an important aspect in this thesis, all results are reported for two recognition systems: one using a single globally pooled variance model and one using density specific variances.

Effect of Increasing the Graph Density

To study the effect of how the graph density affects the performance of a discriminative training, word lattices were pruned with the refined forward-backward pruning as described in Section 5.2.2 using different values for the desired number of arcs per timeframe. The resulting graph densities correspond to a threshold of the log-posterior probabilities ranging from 1.0 to 15.0. Table 6.30 shows the graph densities on the training set together with the WERs measured on the male speaker portion of the HUB-4 test corpus as a function of the pruning threshold τ. The Gaussian mixture models either employ a globally pooled variance vector or use density specific variances. In both settings, increasing the lattice density leads to an improved recognition performance on the test data. While for density specific variances, the optimal lattice size can already be achieved using graph densities with an average of less than 91 hypotheses per time frame, the best results for the system with a pooled variance vector was achieved using a lattice size of approximately 110 arc hypotheses per time frame. Since the initial lattices were generated using a forward-backward pruning with a corresponding threshold of 15.0, a further increase in the lattice size could not be performed without generating new lattices.

Table 6.30: Graph densities on the training set and the WERs measured on the male speaker portion of test data of the HUB-4 corpus as a function of the pruning threshold τ.

		HUB-4, male speaker portion								
		train			test					
					pooled variance			mix.-spec. variances		
criterion	τ	APS	APR	APT	PER	WER	SER	PER	WER	SER
ML	–	–	–	–	15.72	23.39	92.68	15.53	22.81	92.42
MMI	1.0	1.49	1.40	1.42	15.27	22.59	92.68	15.38	22.48	92.93
	2.0	2.12	1.98	1.86	15.25	22.55	92.42	15.38	22.46	92.68
	3.0	8.93	8.38	6.48	15.23	22.37	92.68	15.36	22.48	92.17
	4.0	20.99	19.66	14.52	15.18	22.33	92.42	15.32	22.35	92.42
	5.0	37.66	35.18	25.26	15.13	22.32	92.93	15.29	22.41	92.17
	6.0	58.75	54.74	38.73	15.11	22.24	92.93	15.32	22.36	92.68
	7.0	81.85	76.16	53.72	15.16	22.27	92.42	15.25	22.34	92.68
	8.0	103.53	96.28	68.28	15.14	22.24	92.68	15.17	22.22	92.42
	9.0	121.40	112.86	80.85	15.10	22.20	92.68	15.14	22.09	92.93
	10.0	134.63	125.15	90.73	15.10	22.21	92.93	15.08	22.03	93.18
	11.0	143.77	133.63	97.99	15.05	22.17	92.68	15.11	22.06	92.68
	12.0	149.68	139.12	103.00	15.09	22.21	92.42	15.09	22.03	92.42
	13.0	153.36	142.54	106.30	15.07	22.16	92.68	15.10	22.04	92.17
	14.0	155.62	144.64	108.46	15.03	22.15	92.68	15.10	22.12	92.17
	15.0	156.98	145.90	109.83	15.06	22.13	92.68	15.06	22.04	92.17

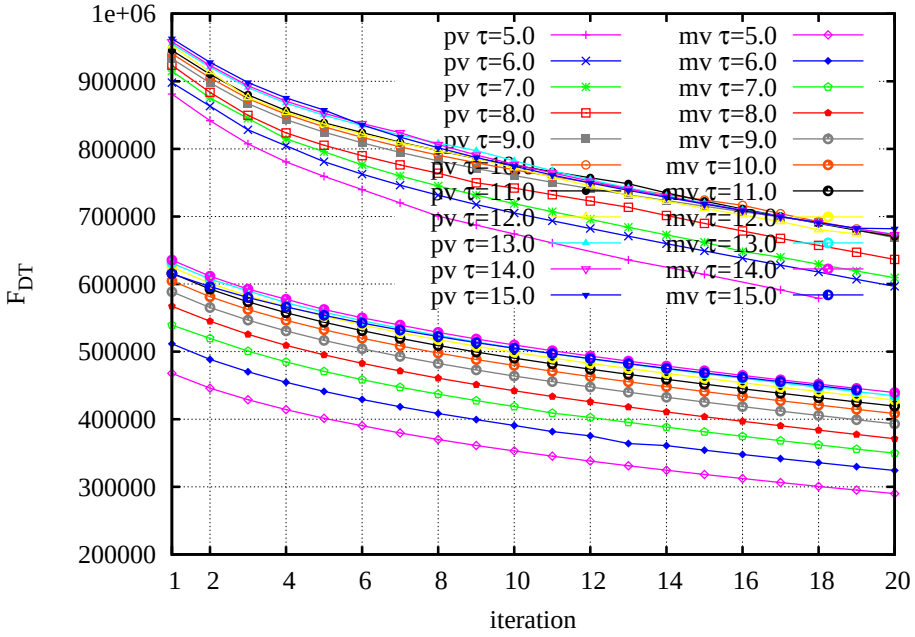


Figure 6.11: Evolution of the MMI criterion in the course of the iteration index on the HUB-4 corpus for different graph densities τ using a single, globally pooled variance vector or density specific variances, respectively.

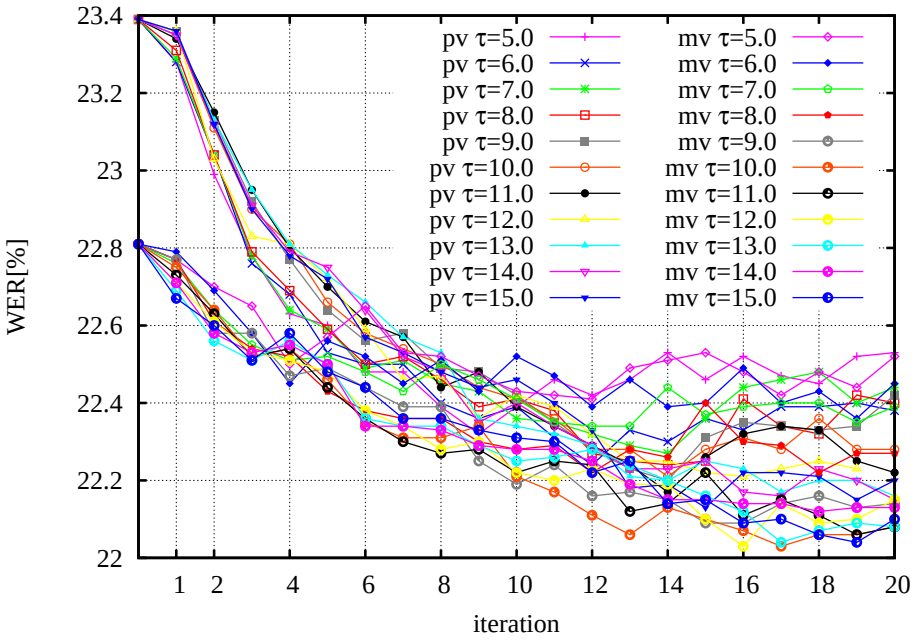


Figure 6.12: Evolution of the *word error rate* (WER) in the course of the iteration index on the HUB-4 corpus for different graph densities τ using a single, globally pooled variance vector or density specific variances, respectively.

Note that even for very small graph densities with an average number of just 1.4 hypotheses per recognized word, the discriminative training under the MMI criterion yields already substantial gains over the ML-trained baseline system. Hence, the pooled variance system is able to improve the initial word error rate by 0.8% absolute using a graph density under which the MMI criterion is almost indistinguishable from the CT criterion. If density specific variances are used, the performance gain obtained with the smallest tested lattice density is 0.33% absolute in terms of word error rate. Though this is less than the absolute improvement obtained with the single variance system, it is still significant. For the optimal lattice densities, the absolute improvement can be as large as 1.26% in terms of word error rate for the single variance system, and 0.77% absolute using density specific variances.

Figure 6.11 shows the evolution of the MMI criterion in the course of the iteration index for different graph densities using either a single, globally pooled variance vector or density specific variances. Figure 6.12 shows the corresponding word error rates on the test data in the course of the iteration index. Note that for both settings the number of training iterations necessary to reach the optimum does not seem to be correlated with the graph density.

In practice, doubling the lattice density also doubles the runtime of the training procedure. Hence, if training time is of the essence, the trade-off between performance gain and total amount of CPU time may become important. If not stated otherwise, the graph densities used in all subsequent experiments corresponds to a pruning threshold of $\tau = 7.0$.

Effect of Changing the Step Size

To study the effect that reducing the step size has on the test set performance, different values for h were investigated for the β method (cf. Section 4.3.1). For the system using a pooled variance vector, h was chosen from the interval $[1.1, ..., 2.1]$, while for the system with density specific variances, $h \in [1.1, 10.0]$. Table 6.31 shows the results on the HUB-4 corpus. In both cases, the choice of the step size neither has a significant effect on the number of training iterations that

Table 6.31: Effect of increasing the step size on the recognition performance of the HUB-4 system for a discriminative training under the MMI criterion using across-word models with a single, globally pooled variance or density-specific variances, respectively.

		HUB-4, male speaker portion							
		pooled variance				mix.-spec. variances			
criterion	h	iter.	PER[%]	WER[%]	SER[%]	iter.	PER[%]	WER[%]	SER[%]
ML	–	–	15.72	23.39	92.68	–	15.53	22.81	92.42
MMI	1.1	13	15.13	22.30	92.68	5	15.25	22.28	92.68
	1.2	13	15.12	22.29	92.68	3	15.28	22.33	92.93
	1.3	14	15.14	22.30	92.42	4	15.26	22.30	92.68
	1.4	13	15.10	22.24	92.68	5	15.25	22.29	92.68
	1.5	13	15.10	22.24	92.68	6	15.21	22.29	92.68
	1.6	13	15.10	22.21	92.42	5	15.27	22.31	92.68
	1.7	13	15.10	22.23	92.42	5	15.26	22.30	92.68
	1.8	13	15.10	22.25	92.42	5	15.26	22.31	92.68
	1.9	14	15.12	22.23	92.42	5	15.24	22.27	92.93
	2.0	14	15.10	22.23	92.42	5	15.26	22.32	92.93
	2.1	14	15.10	22.23	92.42	6	15.22	22.29	92.68
	5.0	14	15.13	22.30	92.42	6	15.24	22.28	92.93
	10.0	14	15.14	22.34	92.42	5	15.24	22.30	92.93

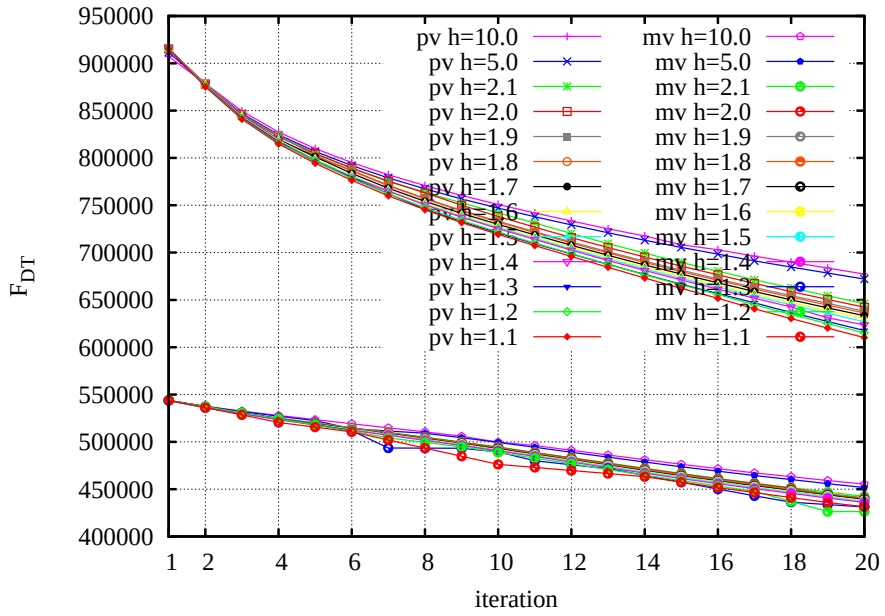


Figure 6.13: Evolution of the MMI criterion in the course of the iteration index on the HUB-4 corpus using different step sizes h.

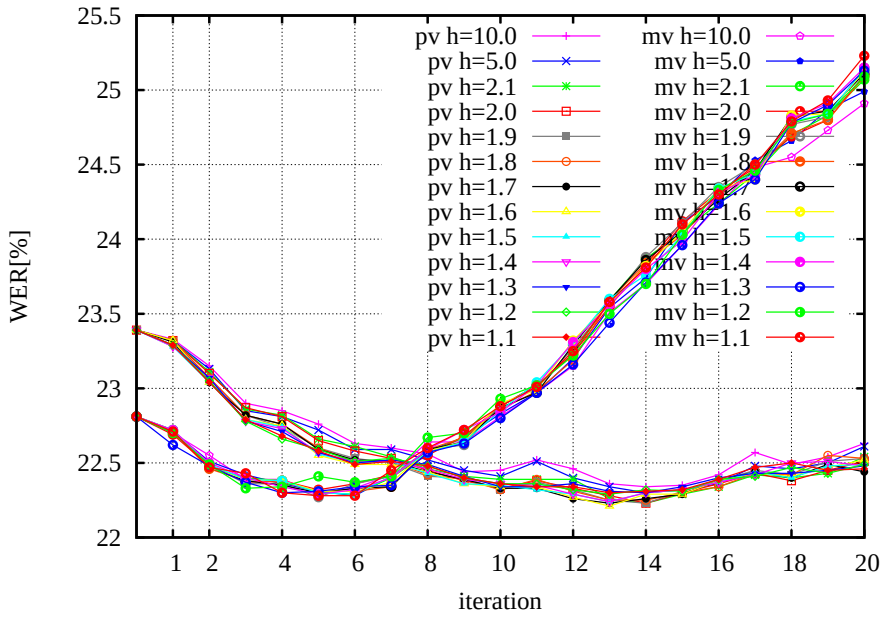


Figure 6.14: Evolution of the *word error rate* (WER) in the course of the iteration index on the HUB-4 corpus using different step sizes h.

are required to reach the best word error rate on test data, nor does the absolute value in terms of word error rate in the optimum change with respect to the chosen step size. This is also an effect

of using high-density word lattices and large amounts of training data. Both settings introduce enough variation into the competing model so that overestimating the step size becomes less likely. Figure 6.13 depicts the evolution of the MMI objective function in the course of the iteration index. Figure 6.14 shows the corresponding word error rates.

Effect of Changing the Acoustic Scaling Factor

This experiment investigates the effect of scaling the acoustic log-probabilities in training. For this purpose, several training runs were carried out using the MMI criterion with different choices of the parameter κ. Table 6.32 shows the results obtained on the HUB-4 corpus for two recognition systems, one using a single, globally pooled variance vector and the other using density specific variances. For both systems, the optimal value leading to the best test set performance was found close to the inverse of the language model scaling factor which confirms results reported in [Schlüter & Müller$^+$ 99]. However, while in case of the multi variance system the performance loss caused by underestimating the optimal value for the scaling factor does not change significantly from the performance loss when the scaling factor is overestimated, the behavior is different when a single, globally pooled variance vector is used. In this case, choosing the acoustic scaling factor too small a value leads to only marginal improvements for the MMI criterion over the ML-trained baseline. Hence, a proper choice of the scaling factor is essential in discriminative training if only a small number of variance parameters is used. Figures 6.15 depicts the evolution of the MMI objective function in the course of the iteration index on the HUB-4 corpus for both the single variance recognition system and the recognition system using density specific variances. Figure 6.16 shows the test set performance in terms of WER as a function of the iteration index for both systems.

Table 6.32: Effect of changing the acoustic scaling factor on the recognition performance of the HUB-4 system for a discriminative training under the MMI criterion using across-word models with either a single, globally pooled variance vector or density-specific variances.

		HUB-4, male speaker portion					
		pooled variance			mix.-spec. variances		
criterion	κ	PER[%]	WER[%]	SER[%]	PER[%]	WER[%]	SER[%]
ML	–	15.72	23.39	92.68	15.53	22.81	92.42
MMI	0.001	15.51	23.21	92.93	15.32	22.43	92.42
	0.005	15.52	23.14	92.68	15.29	22.31	92.42
	0.01	15.46	23.09	92.42	15.26	22.28	92.68
	0.05	15.06	22.11	92.42	15.26	22.35	92.93
	0.1	15.13	22.30	92.68	15.34	22.33	92.42
	0.2	15.21	22.36	92.42	15.37	22.44	92.42
	0.4	15.25	22.45	92.17	15.44	22.48	92.93
	0.8	15.24	22.50	91.92	15.50	22.61	92.68
	1.0	15.29	22.54	92.17	15.49	22.62	92.68

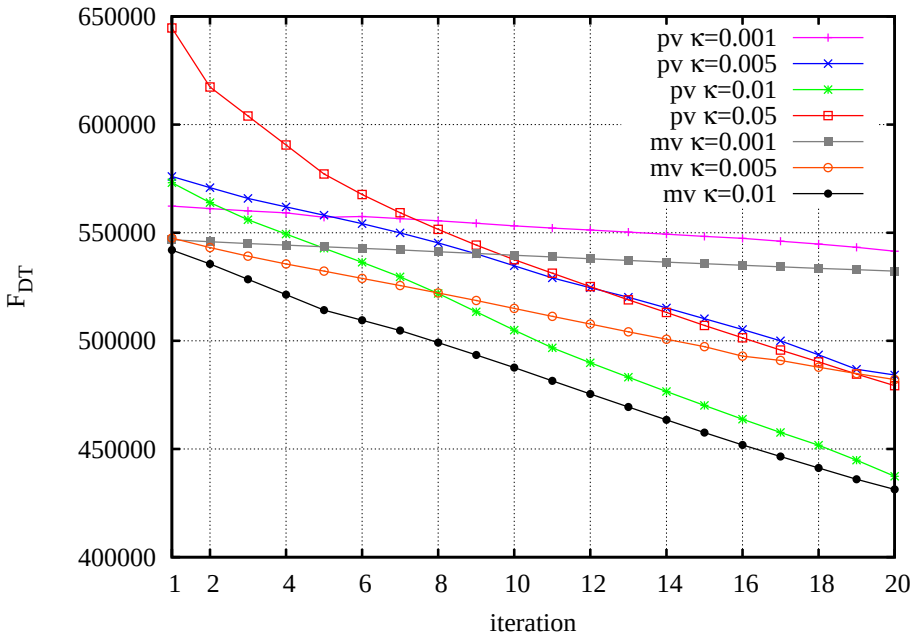


Figure 6.15: Evolution of the MMI criterion in the course of the iteration index on the HUB-4 corpus using different acoustic scaling factors κ.

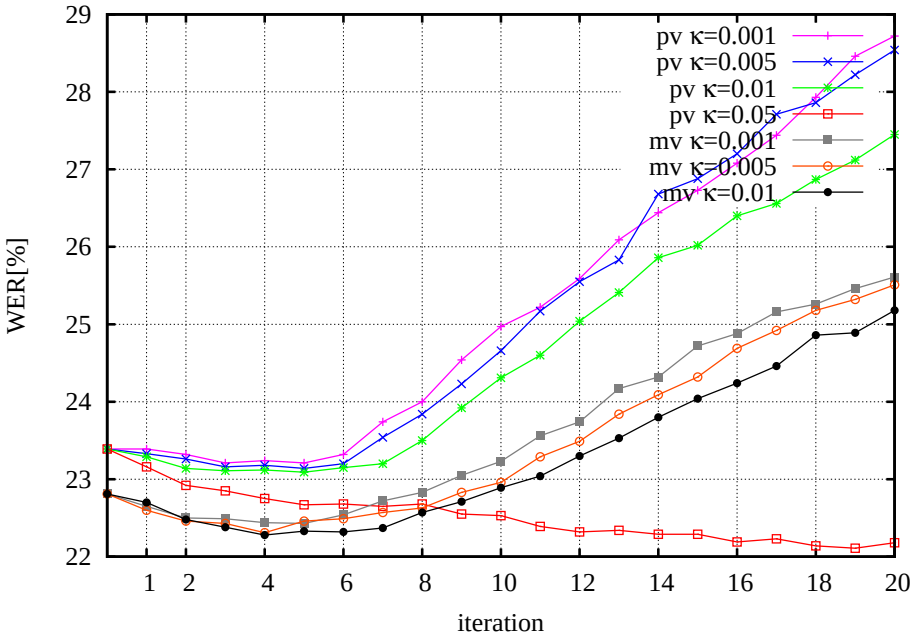


Figure 6.16: Evolution of the *word error rate* (WER) in the course of the iteration index on the HUB-4 corpus using different acoustic scaling factors κ.

Training with Fixed Variance Parameters

A major difference between Gaussian HMMs with a single, globally pooled variance vector and HMMs with density specific variances is the fact that the variance will cancel if it is shared across all mixture components. This rises the question whether it is necessary to re-estimate the variance parameter at all if a single, globally pooled variance model is used, and, if so, to what extent the test set performance will be affected. The same question can of course be investigated for systems that use multiple variances. Table 6.33 shows the recognition results obtained on the male speaker portion of the HUB-4 corpus for recognition systems trained under the MMI criterion using either a single, globally pooled variance vector or density specific variances. Error rates are reported for the case that variances are either kept fixed to their initial ML estimates or that they are updated according to the EB re-estimation equations. Interestingly, fixing the values for the variance parameters slightly improves the recognition results in both cases. Thus, compared to the setting in which the variances are updated, the MMI criterion achieves an additional gain of 0.15% in terms of word error rate for the system with as globally pooled variance vector and an additional improvement of 0.12% for the system using density specific variances. This result may be somewhat surprising since the magnitude of the variances provides an important constraint to determine the step size in discriminative training. However, as shown in the artificial example in Section 2.6.1, fixing the variance parameters in discriminative training results in a larger change of the mean parameters which will move further away from their empirical means and thus try to compensate for the reduced flexibility of the model if the variances are constrained to remain unchanged. Note that fixing the variance parameters also means that the step size does virtually not change across the training iterations. This might give some evidence that the optimal step size is not necessarily a function of the variance parameters.

Figure 6.17 shows the evolution of the MMI criterion in the course of the iteration index with and without the variance updates for the HUB-4 system using a pooled variance vector and the system using density specific variances. Figure 6.18 depicts the corresponding evolution of the WER for both settings in the course of the iteration index. To illustrate the change in the variances, Figure 6.19 depicts the logarithm of the average over the determinants of all variance vectors in the parameter set θ.

Table 6.33: Recognition results obtained on the HUB-4 test corpus (male speaker portion) with a discriminative training under the MMI criterion with and without variance re-estimation, respectively. If no variance updates are performed, the variance parameters are fixed over the whole training process to their initial ML-based estimates.

		HUB-4, male speaker portion					
		pooled variance			mix.-spec. variances		
criterion	var. update	PER[%]	WER[%]	SER[%]	PER[%]	WER[%]	SER[%]
ML	yes	15.72	23.39	92.68	15.53	22.81	92.42
MMI	yes	15.16	22.27	92.42	15.25	22.34	92.68
	no	15.08	22.12	92.42	15.21	22.22	92.68

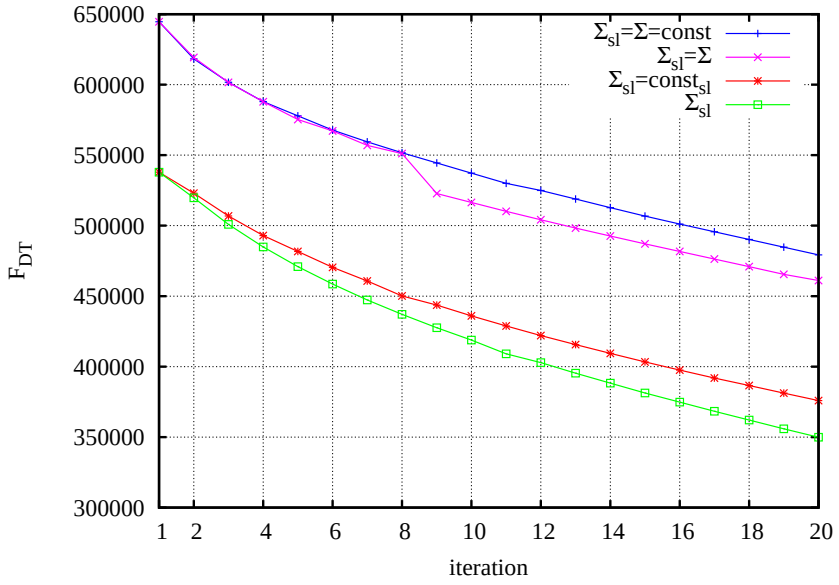


Figure 6.17: Evolution of the MMI criterion in the course of the iteration index on the HUB-4 corpus using variance parameters that are either fixed to their initial ML estimates (const) or which are re-estimated according to the MMI update rule. The function progressions are shown for both a system using a single, globally pooled variance ($\Sigma_{sl} = \Sigma$) and a system using density specific variances.

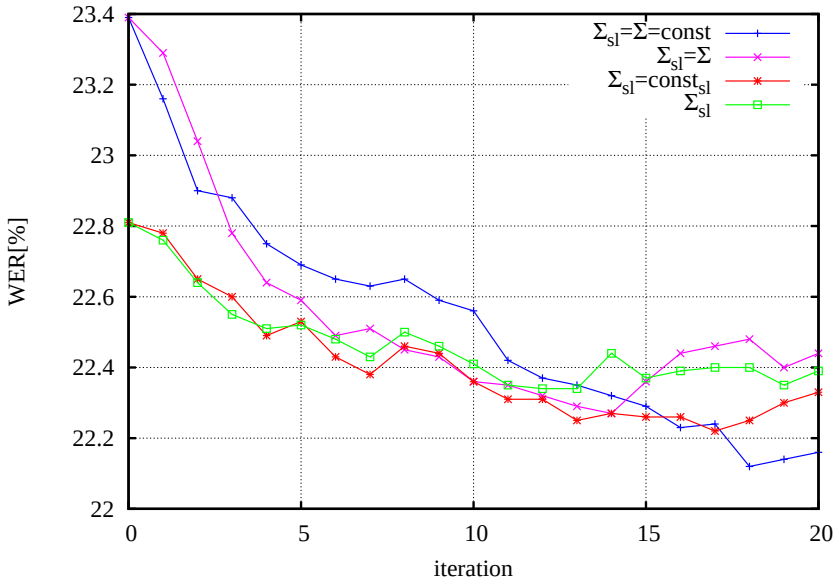


Figure 6.18: Evolution of the *word error rate* (WER) in the course of the iteration index on the HUB-4 corpus with variance parameters that are either fixed to their initial ML estimates (const) or which are re-estimated according to the MMI update rule. The progressions of the functions are shown for both a system using a single, globally pooled variance vector ($\boldsymbol{\Sigma}_{sl} = \boldsymbol{\Sigma}$) and a system using density specific variances ($\boldsymbol{\Sigma}_{sl}$).

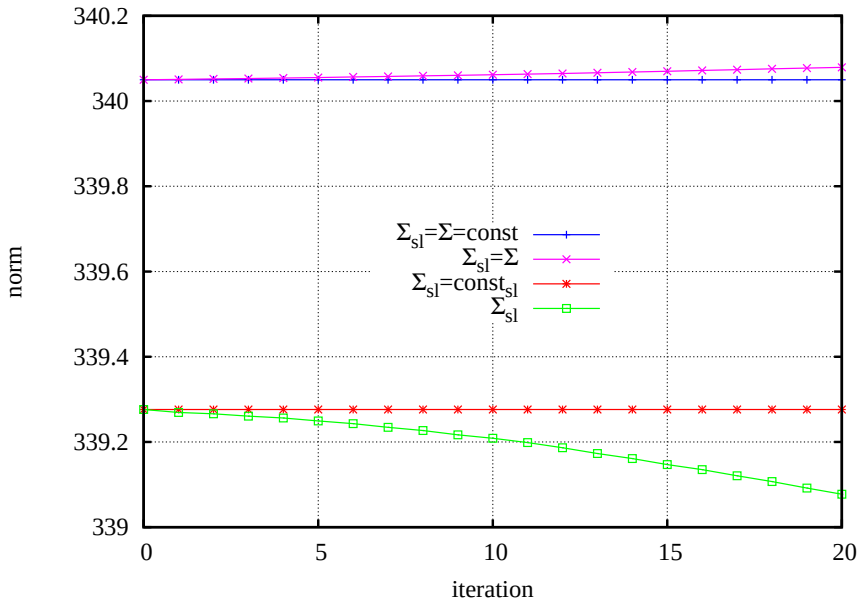


Figure 6.19: Evolution of the normalization term averaged over the determinants of all variances in the course of the iteration index on the HUB-4 corpus for an MMI training with variance parameters that are either fixed to their initial ML estimates (const) or which are re-estimated in each iteration according to the MMI update rule. The function progressions are shown for both a system using a single, globally pooled variance vector ($\boldsymbol{\Sigma}_{sl} = \Sigma$) and a system using density specific variances ($\boldsymbol{\Sigma}_{sl}$).

6.3.8 Comparison of Optimization Methods

This section presents a comprehensive investigation on different methods for choosing the step size in discriminative training. The methods being compared include the choice of the iteration constant according to the β method as proposed in [Schlüter & Macherey$^+$ 01]; an extension of the β method denoted by $D_k \cdot c_{sl}$ which is a generalization to arbitrary tying schemes (cf. Section 4.3); the Λ method (cf. Section 4.3.3); two variants of the *fast* method (cf. Section 4.3.3); and the E method as proposed in [Woodland & Povey 02]. A thorough discussion of either method is given in Section 4.3. Table 6.34 shows the recognition results on the male speaker portion of the HUB-4 corpus together with the iteration index at which the reported word error rate was achieved if the variance parameters are kept fixed. Figure 6.20 shows the evolution of the MMI criterion in the course of the iteration index. Figure 6.21 shows the corresponding word error rates obtained on the test data. Results for experiments conducted on the male speaker portion of the HUB-4 corpus including variance updates are shown in Table 6.35 and Figures 6.22 and 6.23. While the β method and its refinement, the $D_k \cdot c_{sl}$ method, achieve good results if pooled variances are used, the E method performs best if density specific variances are employed. Both *fast* methods achieve their optimum in terms of word error rate already after the second training iteration. However, they tend to overfit the training data and therefore do not result in larger improvements on the test set. Overall, the Λ method achieves the best results independent of the underlying variance tying scheme. These results are confirmed by experiments conducted on the WSJ-5k corpus as shown in Table 6.36 and illustrated in Figures 6.28 and 6.29. Hence, if not stated otherwise, the Λ method is the default optimization method for all experiments reported hereafter.

Table 6.34: Effect of different methods for setting the iteration constants in a discriminative training under the MMI criterion if the variance parameters are kept fixed to their initial ML-trained estimates. Recognition error rates are shown on the HUB-4 test corpus (male speaker portion) for two across-word systems, one using a single, globally pooled variance and the other using density specific variances. The column "iter" indicates in which iteration the optimum was reached.

		HUB-4, male speaker portion							
		pooled variance				mix.-spec. variances			
criterion	method	iter.	PER	WER	SER	iter.	PER	WER	SER
ML	–	–	15.72	23.39	92.68	–	15.53	22.81	92.42
MMI	RWTH	18	15.04	22.21	92.42	18	15.26	22.29	92.93
	$D_k \cdot c_{sl}$	18	15.08	22.12	92.42	17	15.21	22.22	92.68
	Λ	17	15.03	22.17	92.93	7	15.17	22.10	93.18
	fast	2	15.52	22.95	92.93	2	15.33	22.45	92.93
	fast+$\Gamma_{sl}(1)$	2	15.50	22.99	92.93	2	15.33	22.45	92.93
	$E = 1.0$	3	15.18	22.48	92.93	5	15.10	22.09	93.18
	$E = 2.0$	7	15.14	22.32	92.93	11	15.18	22.16	93.43
	E = halfmax	8	15.16	22.35	92.68	11	15.18	22.19	93.18

Table 6.35: Effect of different methods for setting the iteration constants in a discriminative training under the MMI criterion if the variance parameters are re-estimated at the end of each training iteration. Recognition error rates are shown on the HUB-4 test corpus (male speaker portion) for two across-word systems, one using a single, globally pooled variance and the other using density specific variances. The column "iter" indicates in which iteration the optimum was reached.

		HUB-4, male speaker portion							
		pooled variance				mix.-spec. variances			
criterion	method	iter.	PER	WER	SER	iter.	PER	WER	SER
ML	–	–	15.72	23.39	92.68	–	15.53	22.81	92.42
MMI	RWTH	17	15.08	22.19	92.68	18	15.27	22.29	92.93
	$D_k \cdot c_{sl}$	14	15.16	22.27	92.42	12	15.25	22.34	92.68
	Λ	10	15.09	22.32	92.93	5	15.07	22.04	93.18
	fast	1	15.47	23.10	92.42	1	15.27	22.25	92.68
	fast+$\Gamma_{sl}(1)$	5	15.19	22.43	92.68	5	15.11	22.13	93.18
	$E = 1.0$	6	15.23	22.35	92.68	5	15.10	22.08	92.93
	$E = 2.0$	9	15.14	22.28	92.68	10	15.14	22.15	93.18
	E = halfmax	10	15.17	22.41	92.93	4	15.18	22.27	92.68

Table 6.36: Comparison of different methods to control the convergence rate for a discriminative training under the MMI criterion on the WSJ-5k corpus.

		WSJ-5k								
		development			evaluation			dev + eval		
		PER	WER	SER	PER	WER	SER	PER	WER	SER
ML		5.61	4.72	41.2	4.98	3.90	36.7	5.33	4.36	39.2
MMI	β	5.29	4.26	39.0	4.81	3.64	34.9	5.08	3.99	37.2
	E	5.42	4.45	41.0	4.89	3.79	35.2	5.18	4.16	38.4
	Λ	5.31	4.25	38.5	4.80	3.61	34.2	5.09	3.96	36.6
	fast	5.32	4.38	38.1	4.76	3.57	32.7	5.07	4.02	35.7

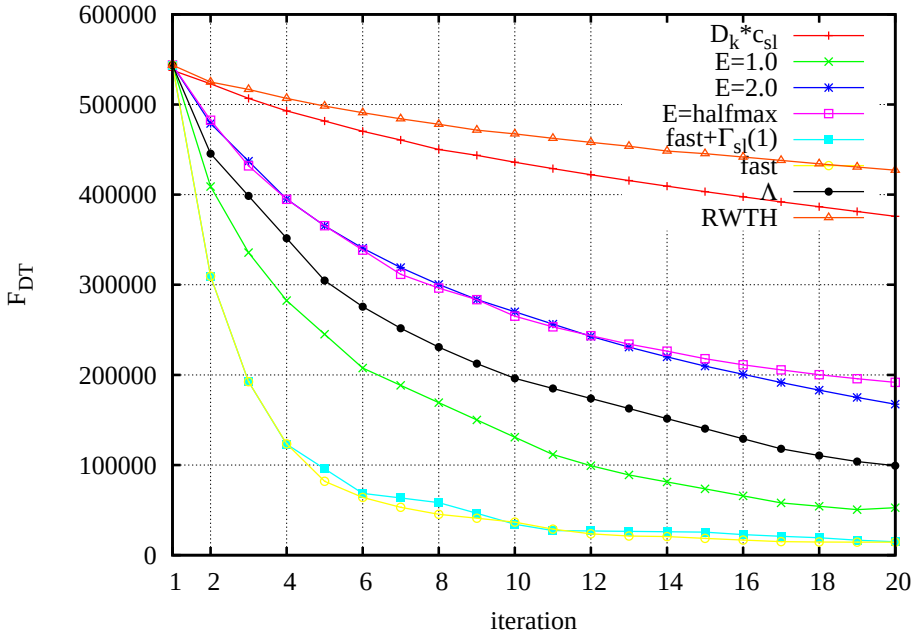


Figure 6.20: Evolution of the MMI training criterion in the course of the iteration index on the HUB-4 corpus for different methods for choosing the iteration constants. The recognition system uses density specific variances that were not re-estimated but kept fixed to their initial ML estimates over the whole training process.

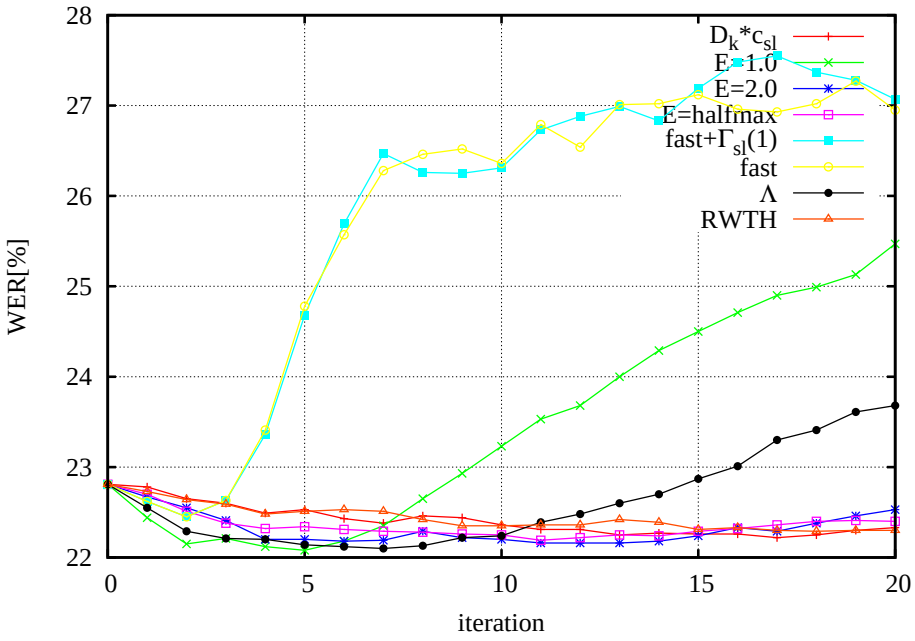


Figure 6.21: Evolution of the word error rate in the course of the iteration index on the HUB-4 corpus for different methods for choosing the iteration constants. The recognition system uses density specific variances. The variance parameters were not re-estimated but kept fixed to their initial ML estimates over the whole training process.

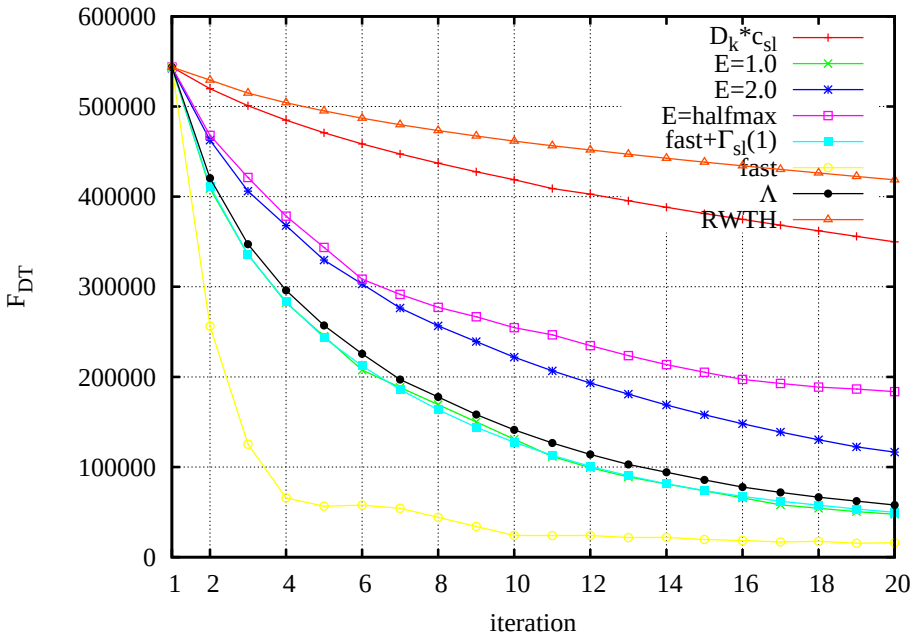


Figure 6.22: Evolution of the MMI training criterion in the course of the iteration index on the HUB-4 corpus for different methods for choosing the iteration constants. The recognition system uses density specific variances. In contrast to the setting described in Figure 6.20, all variance parameters were re-estimated.

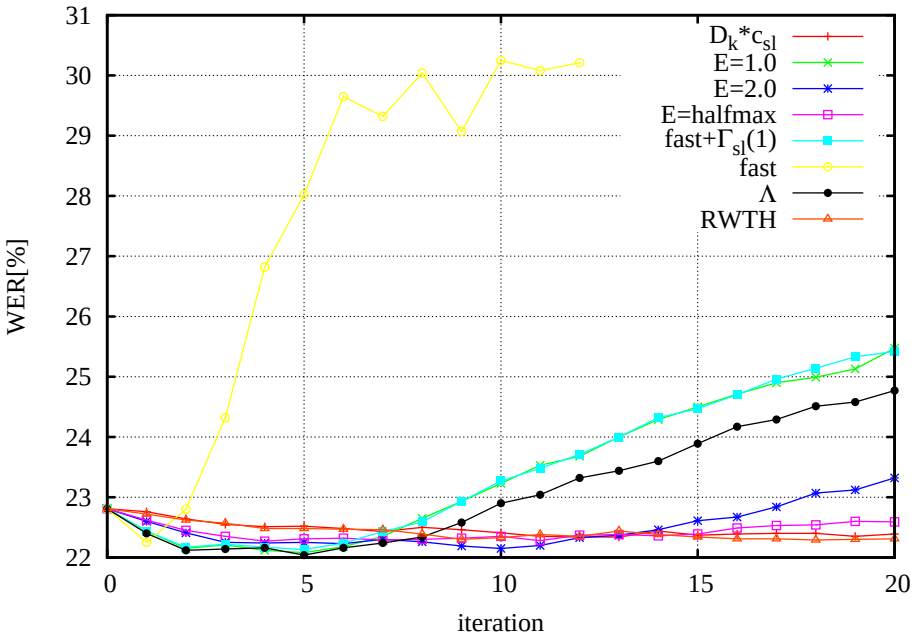


Figure 6.23: Evolution of the *word error rate* WER in the course of the iteration index on the HUB-4 corpus for different methods for choosing the iteration constants. The recognition system uses density specific variances. In contrast to the setting described in Figure 6.21, all variance parameters were re-estimated.

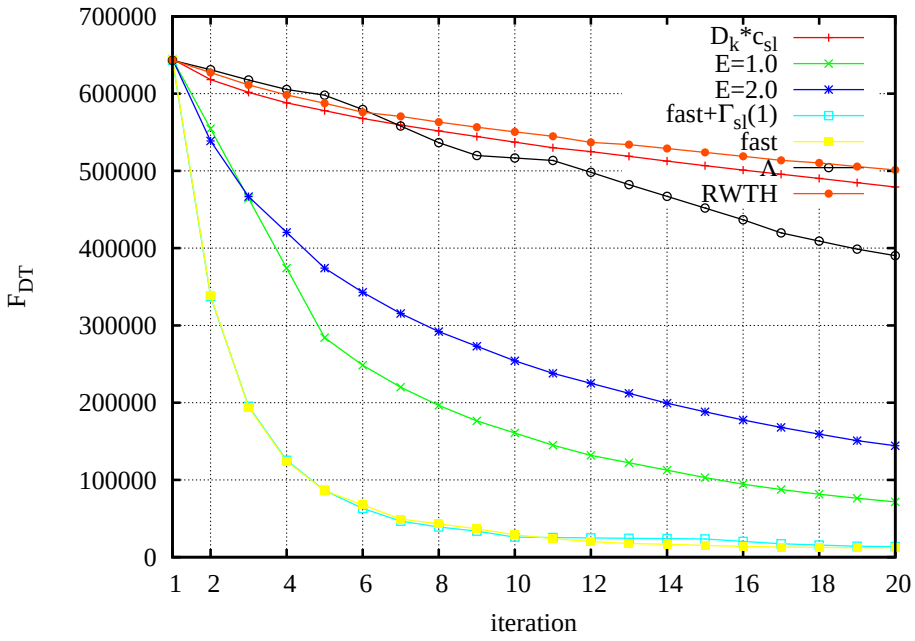


Figure 6.24: Evolution of the MMI training criterion in the course of the iteration index on the HUB-4 corpus for different methods for choosing the iteration constants. The recognition system uses a single, globally pooled variance vector. The variance parameters were not re-estimated not kept fixed to their initial ML estimates over the whole training process.

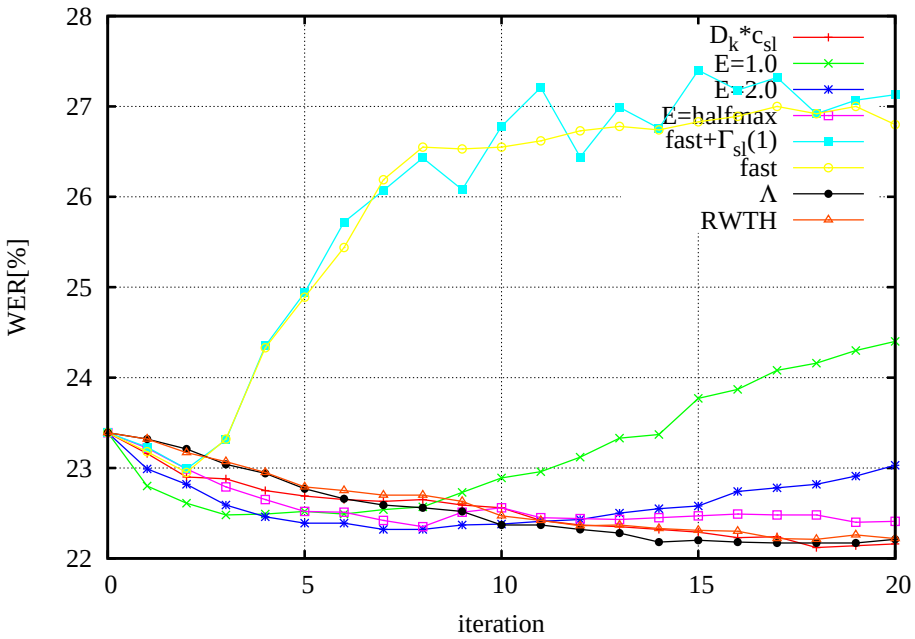


Figure 6.25: Evolution of the *word error rate* (WER) in the course of the iteration index on the HUB-4 corpus for different methods for choosing the iteration constants. The recognition system uses a single, globally pooled variance vector. The variance parameters were not re-estimated but kept fixed to their initial ML estimates over the whole training process.

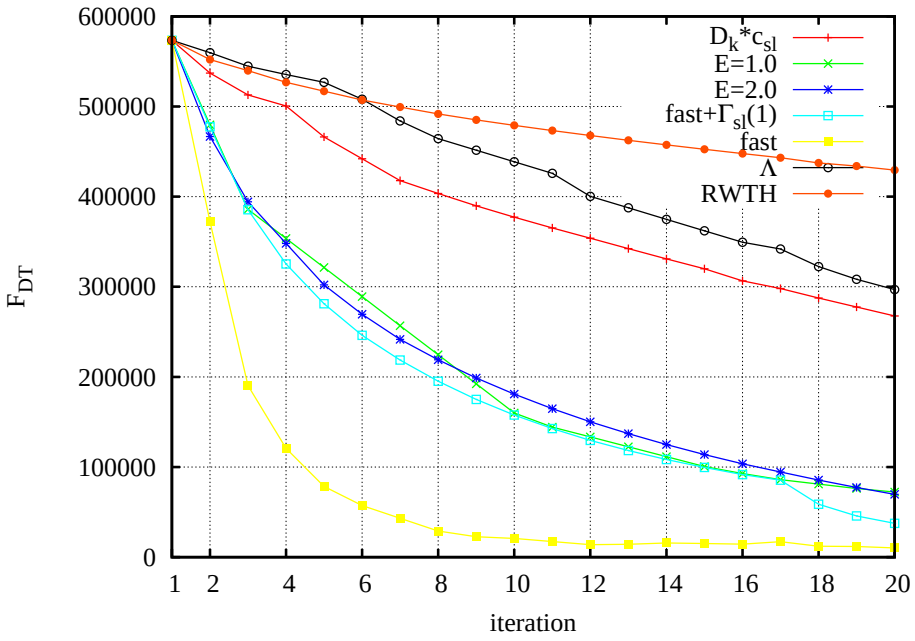


Figure 6.26: Evolution of the MMI training criterion in the course of the iteration index on the HUB-4 corpus for different methods for choosing the iteration constants. The recognition system uses a single, globally pooled variance vector. In contrast to the setting described in Figure 6.24, the variance parameters were re-estimated.

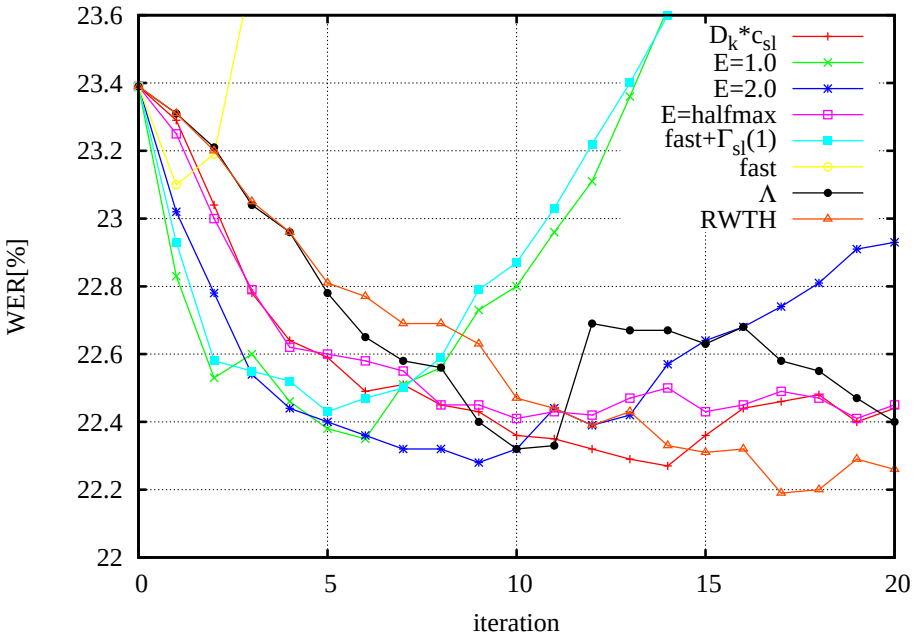


Figure 6.27: Evolution of the *word error rate* WER in the course of the iteration index on the HUB-4 corpus for different methods for choosing the iteration constants. The recognition system uses a single, globally pooled variance vector. In contrast to the setting described in Figure 6.25, the variance parameters were re-estimated.

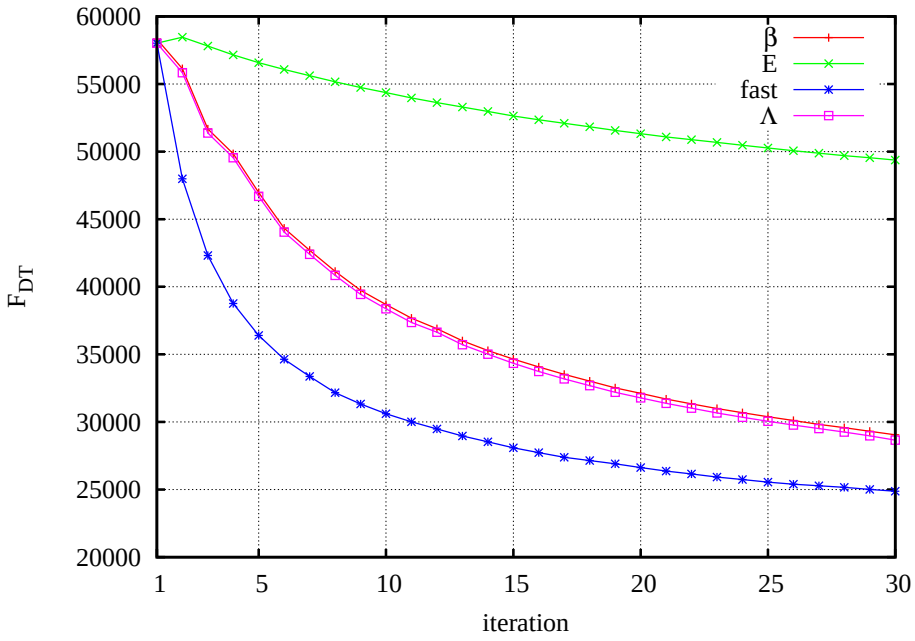


Figure 6.28: Evolution of the MMI training criterion in the course of the iteration index on the WSJ-5k corpus for different methods for choosing the iteration constants. The recognition system uses a single, globally pooled variance vector. The variance parameters were re-estimated.

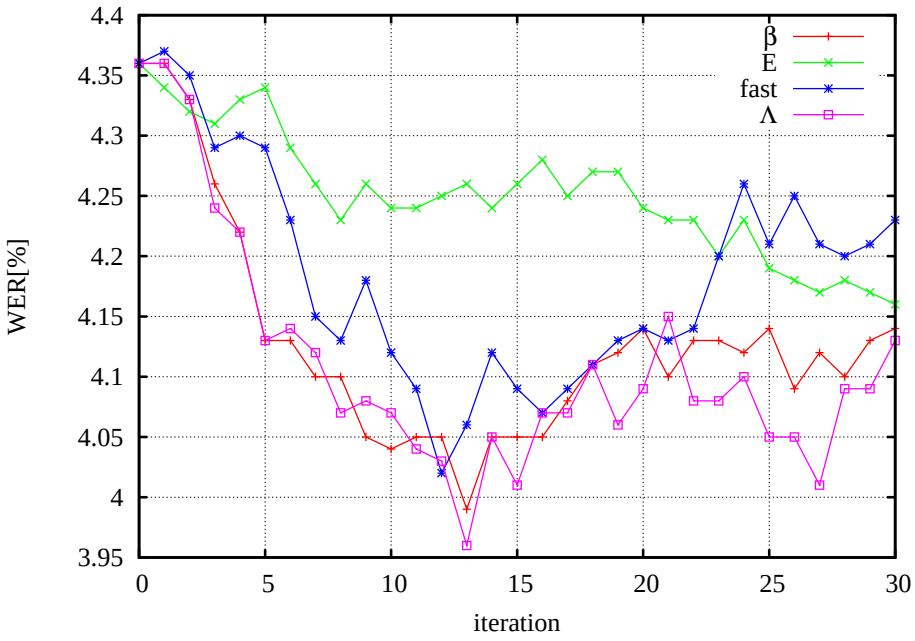


Figure 6.29: Evolution of the *word error rate* WER in the course of the iteration index on the WSJ-5k corpus for different methods for choosing the iteration constants. The recognition system uses a single, globally pooled variance vector. The variance parameters were re-estimated.

6.3.9 Lattice-based Minimum Classification Error Training

This section presents experimental results for a large-scale lattice-based *Minimum Classification Error* (MCE) training. The experiments were conducted on the WSJ-5k corpus and on the male speaker portion of the HUB-4 corpus. As discussed in Section 4.1.2, the MCE criterion approximates the smoothed sentence error on training data using the logistic regression as sigmoid function. The slope of the sigmoid function is controlled by a parameter ρ. Table 6.37 shows how different choices of ρ affect the test set performance of the WSJ-5k corpus. Here the optimum is reached for $\rho = 4 \times 10^{-5}$. On the male speaker portion of the HUB-4 corpus, the optimal value for the slope parameter turns out be within a similar range as shown in Table 6.38. The recognition system using density specific variances performs best if the slope parameter is set to 10^{-5}. For the pooled variance system, the optimal choice is 10^{-3} which yields a word error rate of 22.29%. The recognition performance for this value does not significantly differ from the choice $\rho = 10^{-5}$, which yields a slightly higher word error rate of 22.34%. In general, smaller values for the slope parameter tend to result in lower error rates. It is therefore usually safe to underestimate the value of ρ whereas overestimating typically results in only negligible performance gains compared to the ML-trained baseline system.

Table 6.37: Effect of changing the slope parameter in the sigmoid function of the *Minimum Classification Error* (MCE) criterion on the WSJ-5k corpus.

		WSJ-5k								
		dev			eval			dev+eval		
criterion	ρ	PER	WER	SER	PER	WER	SER	PER	WER	SER
ML	–	5.53	4.55	40.49	4.89	3.74	34.85	5.25	4.19	37.97
MCE	4×10^{-3}	5.16	4.05	37.07	4.73	3.53	33.33	4.97	3.82	35.41
	1×10^{-3}	5.18	4.08	37.07	4.77	3.61	33.64	5.00	3.87	35.54
	4×10^{-4}	5.20	4.07	37.80	4.74	3.61	33.33	4.99	3.86	35.81
	1×10^{-4}	5.19	4.04	38.05	4.77	3.57	33.94	5.00	3.83	36.22
	4×10^{-5}	5.17	4.02	38.54	4.72	3.47	33.33	4.97	3.78	36.22
	1×10^{-5}	5.22	4.10	38.78	4.72	3.47	33.33	5.00	3.82	36.35
	4×10^{-6}	5.18	4.05	38.78	4.75	3.53	33.64	4.99	3.82	36.49
	1×10^{-6}	5.17	4.07	37.32	4.71	3.53	32.73	4.97	3.83	35.27

Table 6.38: Effect of changing the slope parameter ρ in the sigmoid function of the *Minimum Classification Error* (MCE) criterion on the HUB-4 test corpus (male speaker portion) using an across-word system with a single, globally pooled variance vector or density-specific variances, respectively.

		HUB-4, male speaker portion					
		pooled variance			mix.-spec. variances		
criterion	ρ	PER[%]	WER[%]	SER[%]	PER[%]	WER[%]	SER[%]
ML	–	15.72	23.39	92.68	15.53	22.81	92.42
MCE	1.0	15.70	23.27	92.42	15.53	22.81	92.42
	0.1	15.51	23.03	92.68	15.38	22.67	92.17
	0.01	15.20	22.44	92.93	15.04	22.17	93.18
	10^{-3}	15.10	22.29	92.93	14.99	22.15	93.18
	10^{-4}	15.30	22.30	92.68	14.97	22.12	93.18
	10^{-5}	15.12	22.34	92.68	14.97	22.11	92.93
	10^{-6}	15.18	22.34	92.17	14.97	22.11	92.93

6.3.10 Comparison of Discriminative Training Criteria

The extended unifying approach for a class of discriminative training criteria as proposed in Chapter 4 allows for expressing different objective functions within a single framework. This makes performance gains obtained with a large number of training criteria are directly comparable. Table 6.39 shows results in terms of word error rates on the male speaker portion of the HUB-4 test corpus for several training criteria including the *Maximum Mutual Information* (MMI) criterion; the *Corrective Training* (CT) criterion; the *Minimum Classification Error* (MCE) criterion; its maximum approximation, the *Falsifying Training* (FT) criterion; the Diversity Index together with its two special instances, the *Shannon* criterion and the *Gini* criterion; the *Jeffreys* criterion; the *Chernoff* criterion; and the *Minimum Word Error* (MWE) criterion together with the closely related *Minimum Phone Error* (MPE) criterion. As before, results for all criteria are shown for two systems, one using a single, globally pooled variance vector and one using density specific variances.

Table 6.39 reveals some interesting results: (1) the MWE criterion as well as the MPE criterion are not able to outperform the MMI criterion or the MCE criterion on this task. (2) The Diversity Index and its special case, the Gini criterion, achieve results on both the single variance system and the multi-variance system that are similar to the recognition performance obtained with the MMI criterion. (3) All criteria achieve statistically significant improvements on top of the ML-trained baseline systems.

Figure 6.30 shows the evolution of the training criteria in the course of the iteration index. Figure 6.31 shows the corresponding word error rates measured on the test corpus for the pooled variance system. Analogous plots for the systems using density specific variances are shown in Figures 6.32 and 6.32.

Overall, the MCE criterion turns out to be the best performing criterion which yields the highest improvement over the ML-trained baseline system. Thus, with the single variance system, the MCE criterion is able to reduce the initial word error by 1.34%. With the system using density-specific variances, the MCE criterion achieves a reduction in terms of word error rate of 0.81% absolute. The performance gain obtained with the MCE criterion on the HUB-4 corpus does not, however, differ significantly from the performance gains obtained with criteria such as the MMI criterion, the Diversity Index or the Gini criterion.

Similar results were obtained on the WSJ-5k corpus as shown in Table 6.40. Here, the MWE criterion achieves the best result with a word error rate of 3.74% on the combined development and evaluation corpus. Gains obtained with the MCE criterion, which yields a word error rate of 3.78%, do not differ significantly from the MWE result. Figure 6.34 shows the evolution of the training criteria in the course of the iteration index. Figure 6.35 shows the corresponding word error rates measured on the test corpus.

Table 6.39: Comparison of multiple discriminative training criteria in the extended unifying approach. Error rates were measured on the HUB-4 test corpus (male speaker portion). The ML-trained baseline system employs across-word models and uses a single, globally pooled variance vector or mixture-specific variances, respectively. The criteria listed comprise the *Maximum Likelihood* (ML) criterion, the *Maximum Mutual Information* (MMI) criterion, the *Corrective Training* (CT) criterion, the *Minimum Classification Error* (MCE) criterion, the *Falsifying Training* (FT) criterion, the *Diversity Index*, the *Shannon* criterion, the *Gini* criterion, the *Jeffreys* criterion, the *Chernoff* criterion, the *Minimum Word Error* (MWE) criterion, and the *Minimum Phone Error* (MPE) criterion.

	HUB-4, male speaker portion					
	pooled variance			mix.-spec. variances		
criterion	PER[%]	WER[%]	SER[%]	PER[%]	WER[%]	SER[%]
ML	15.72	23.39	92.68	15.53	22.81	92.42
MMI	15.03	22.09	92.93	14.99	22.09	92.93
CT	15.28	22.52	92.17	15.39	22.57	92.17
MCE	15.02	22.05	92.68	14.94	22.00	93.18
FT	15.50	22.79	91.92	15.39	22.58	92.68
Diversity Index	14.97	22.06	92.93	15.02	22.14	92.93
Shannon	15.07	22.23	93.18	14.96	22.12	93.18
Gini	14.99	22.09	92.42	15.02	22.11	93.18
Jeffreys	15.10	22.15	92.42	15.15	22.30	92.42
Chernoff	15.72	23.33	92.42	15.29	22.54	91.92
MWE	15.19	22.35	91.92	15.16	22.28	92.17
MPE	15.20	22.37	92.17	15.11	22.27	92.42

Table 6.40: Comparison of multiple discriminative training criteria in the extended unifying approach. Error rates were measured on the WSJ-5k corpus. The criteria shown comprise the *Maximum Likelihood* (ML) criterion, the *Gini* criterion, the *Jeffreys* criterion, the *Maximum Mutual Information* (MMI) criterion, the *Minimum Classification Error* (MCE) criterion, the *Minimum Word Error* (MWE) criterion, and the *Minimum Phone Error* (MPE) criterion.

	WSJ-5k								
	dev			eval			dev+eval		
criterion	PER	WER	SER	PER	WER	SER	PER	WER	SER
ML	5.53	4.55	40.49	4.89	3.74	34.85	5.25	4.19	37.97
Gini	5.36	4.26	38.29	4.90	3.81	35.76	5.16	4.06	37.16
Jeffreys	5.21	4.08	38.05	4.76	3.51	33.94	5.01	3.83	36.22
MMI	5.25	4.07	38.78	4.76	3.53	33.64	5.03	3.83	36.49
MCE	5.17	4.02	38.54	4.72	3.47	33.33	4.97	3.78	36.22
MWE	5.20	3.98	37.56	4.74	3.44	31.82	5.00	3.74	35.00
MPE	5.39	4.17	38.29	4.85	3.62	33.64	5.15	3.93	36.22

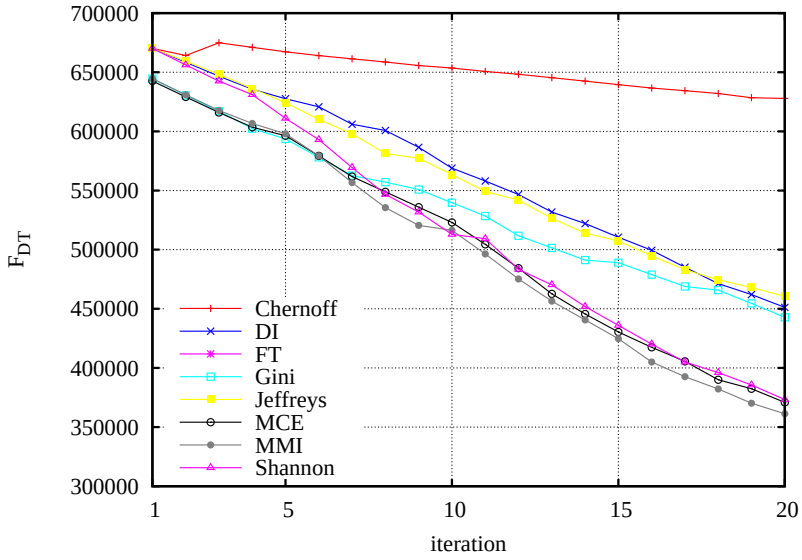


Figure 6.30: Evolution of multiple objective functions in the course of the iteration index for a class of discriminative training criteria on the HUB-4 corpus (male speaker portion) using a system with a single, globally pooled variance. Training criteria depicted comprise the *Chernoff* criterion, the *Diversity Index* (DI), the *Falsifying Training* (FT), the *Corrective Training* (CT) criterion, the *Gini* criterion, the *Jeffreys* criterion, the *Minimum Classification Error* (MCE) criterion, the *Maximum Mutual Information* (MMI) criterion, the *Minimum Phone Error* (MPE) criterion, and the *Minimum Word Error* (MWE) criterion.

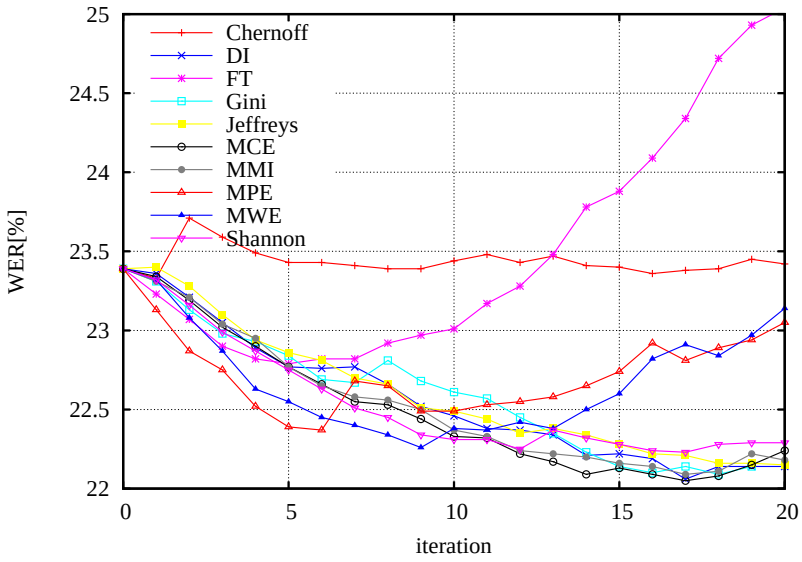


Figure 6.31: Evolution of *word error rates* (WER) in the course of the iteration index for a class of discriminative training criteria on the HUB-4 corpus (male speaker portion) using a system with a single, globally pooled variance. Training criteria depicted comprise the *Chernoff* criterion, the *Diversity Index* (DI), the *Falsifying Training* (FT), the *Corrective Training* (CT) criterion, the *Gini* criterion, the *Jeffreys* criterion, the *Minimum Classification Error* (MCE) criterion, the *Maximum Mutual Information* (MMI) criterion, the *Minimum Phone Error* (MPE) criterion, and the *Minimum Word Error* (MWE) criterion.

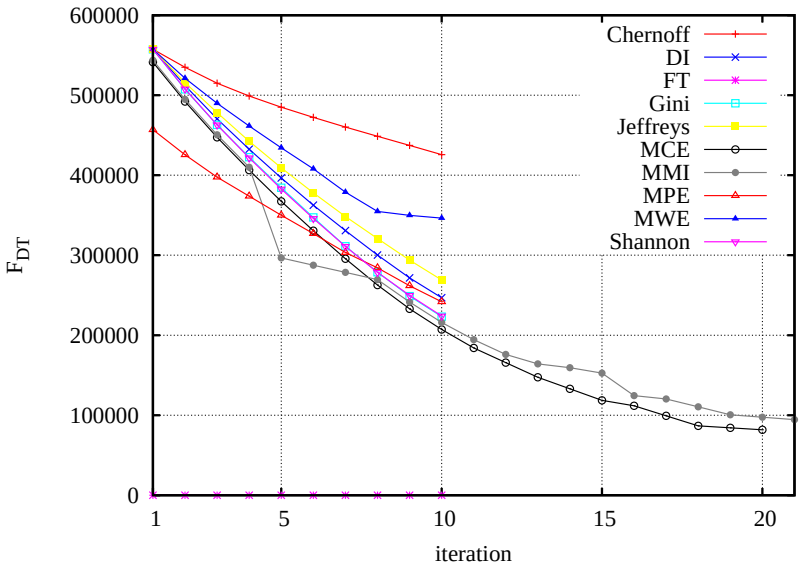


Figure 6.32: Evolution of multiple objective functions in the course of the iteration index for a class of discriminative training criteria on the HUB-4 corpus (male speaker portion) using a system with density specific variances. Training criteria depicted comprise the *Chernoff* criterion, the *Diversity Index* (DI), the *Falsifying Training* (FT), the *Corrective Training* (CT) criterion, the *Gini* criterion, the *Jeffreys* criterion, the *Minimum Classification Error* (MCE) criterion, the *Maximum Mutual Information* (MMI) criterion, the *Minimum Phone Error* (MPE) criterion, and the *Minimum Word Error* (MWE) criterion.

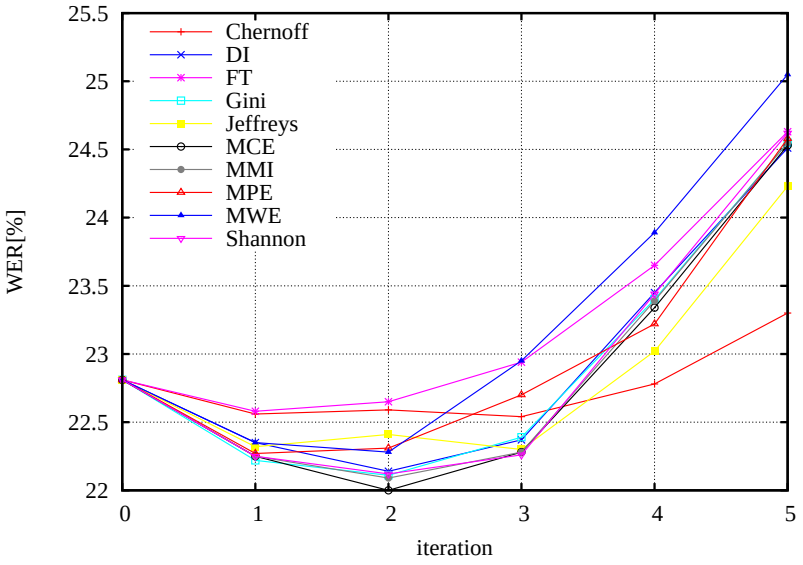


Figure 6.33: Evolution of *word error rates* (WER) in the course of the iteration index for a class of discriminative training criteria on the HUB-4 corpus (male speaker portion) using a system with density specific variance. Training criteria depicted comprise the *Chernoff* criterion, the *Diversity Index* (DI), the *Falsifying Training* (FT), the *Corrective Training* (CT) criterion, the *Gini* criterion, the *Jeffreys* criterion, the *Minimum Classification Error* (MCE) criterion, the *Maximum Mutual Information* (MMI) criterion, the *Minimum Phone Error* (MPE) criterion, and the *Minimum Word Error* (MWE) criterion.

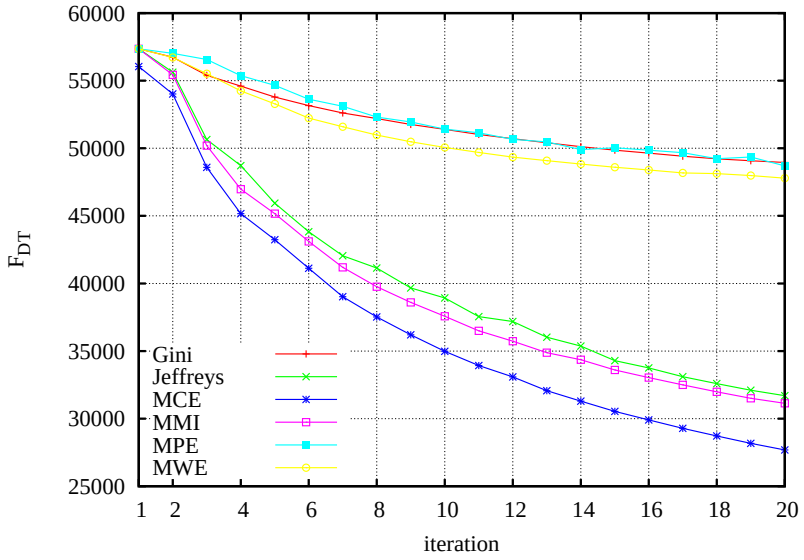


Figure 6.34: Evolution of multiple objective functions for a class of discriminative training criteria in the course of the iteration index on the WSJ-5k corpus. The system uses a single, globally pooled variance. Training criteria depicted comprise the *Gini* criterion, the *Jeffreys* criterion, the *Minimum Classification Error* (MCE) criterion, the *Maximum Mutual Information* (MMI) criterion, the *Minimum Phone Error* (MPE) criterion, and the *Minimum Word Error* (MWE) criterion.

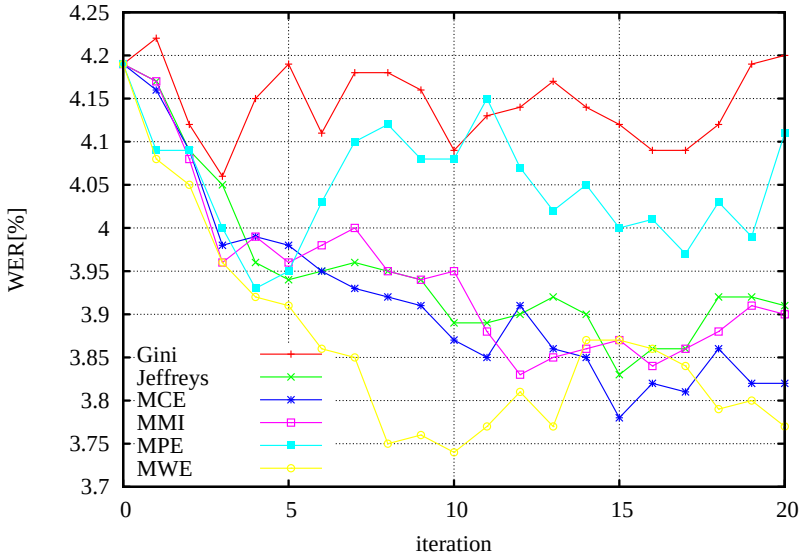


Figure 6.35: Evolution of *word error rates* (WERs) for a class of discriminative training criteria in the course of the iteration index on the WSJ-5k corpus. The system uses a single, globally pooled variance. Training criteria depicted comprise the *Gini* criterion, the *Jeffreys* criterion, the *Minimum Classification Error* (MCE) criterion, the *Maximum Mutual Information* (MMI) criterion, the *Minimum Phone Error* (MPE) criterion, and the *Minimum Word Error* (MWE) criterion.

6.3.11 Unigram Decoded Lattices in Training

In [Schlüter & Müller+ 99], it was shown that a discriminative training procedure will be more effective if the lattices used in training are re-scored with a unigram language model before the sufficient statistics are collected from the numerator and the denominator lattices. The reason for this is that competing hypotheses, which are acoustically similar to the spoken word sequence but do not fit into the linguistic context, become otherwise less probable because they get penalized by higher language model costs. As a consequence, the forward-backward probabilities of these hypotheses will be relatively small. A low-order language model like a unigram language model does not take surrounding words into account and thus operates only locally on the arc hypothesis itself. Therefore, a larger fraction of hypotheses will yield a higher weight which may improve generalization to unseen data.

An extension of this idea is to directly use a unigram language model in order to decode the training corpus and to produce initial word lattices. Since nodes in a unigram-decoded word lattice have no associated language model history, they can connect a larger variety of word hypotheses and thus result in word lattices with a larger diversity of alternative sentence hypotheses compared to trigram-decoded word lattices of similar size. Again, this may help to improve generalization to unseen data.

Table 6.41 shows the word error rates obtained with a lattice-based MMI training on the male speaker portion of the HUB-4 corpus using either trigram-, bigram-, or unigram-decoded word lattices. Compared to trigram-decoded word lattices, unigram-decoded lattices result in an additional decrease of 0.17% in terms of word error rate.

Table 6.42 shows results obtained with the MCE criterion trained on unigram-decoded word lattices. Compared to trigram-decoded word lattices (cf. Table 6.41), the unigram-decoded word lattices result in an additional 0.2% decrease in terms of word error rate for the MCE criterion which achieves a word error rate reduction of 1.00% over the ML-trained baseline system. Similar results could be observed on the female speaker portion of the HUB-4 corpus as shown in Table 6.43.

Table 6.41: Effect of using different language models to decode and generate initial lattices for a discriminative training under the MMI on the HUB-4 corpus (male speaker portions). The notation mg-1g means that the lattices were produced using an integrated m-gram search before they are rescored with a unigram language model. Results are presented for an across-word system with density-specific variances and a uni-, bi-, and trigram language model used for decoding the training corpus.

		HUB-4, male speaker portion		
		mix.-spec. variances		
criterion	n-gram	PER[%]	WER[%]	SER[%]
ML	—	15.53	22.81	92.42
MMI	3g-1g	15.31	22.08	92.68
	2g-1g	15.28	21.99	92.93
	1g-1g	15.17	21.91	92.68

6.3.12 Performance Gain with I-Smoothing

Table 6.44 shows the effect of I-smoothing, a technique developed in [Povey & Woodland 02] to improve the ability of a discriminatively trained parameter set to generalize to unseen data. I-smoothing is closely related to the H-criterion which uses a fixed interpolation between the ML objective function (H=0) and the MMI objective function (H=1). Unlike the H-criterion, I-

Table 6.42: Effect of using unigram-decoded lattices for a discriminative training under the *Maximum Mutual Information* (MMI) criterion and the *Minimum Classification Error* (MCE) criterion on the HUB-4 corpus (male speaker portion) using an across-word system with a single, globally pooled variance or density-specific variances, respectively.

	HUB-4, male speaker portion					
	pooled variance			mix.-spec. variances		
criterion	PER[%]	WER[%]	SER[%]	PER[%]	WER[%]	SER[%]
ML	15.72	23.39	92.68	15.53	22.81	92.42
MMI	15.02	22.22	92.68	15.17	21.91	92.68
MCE	15.08	22.25	92.93	15.10	21.81	92.93

Table 6.43: Effect of using unigram-decoded lattices for a discriminative training under the *Maximum Mutual Information* (MMI) criterion and the *Minimum Classification Error* (MCE) criterion on the HUB-4 corpus (female speaker portion) using an across-word system with a single, globally pooled variance or density-specific variances, respectively.

	HUB-4, female speaker portion					
	pooled variance			mix.-spec. variances		
criterion	PER[%]	WER[%]	SER[%]	PER[%]	WER[%]	SER[%]
ML	14.32	20.93	88.12	14.37	20.65	89.27
MMI	13.97	20.17	88.51	13.83	19.91	89.66
MCE	14.03	20.13	89.27	13.81	19.90	89.66

smoothing increases the number of data points assigned to a Gaussian by a constant term while keeping the average data values and average squared data values the same. Word error rates on the HUB-4 corpus (male speaker portion) using single densities with density specific variances trained under the MMI criterion are shown in Table 6.44. Here, I-smoothing can decrease the word error rate by 0.4% absolute if the optimal parameter set is chosen.

Table 6.45 and Table 6.46 show the performance of the HUB-4 system trained under the MMI criterion. Word error rates are shown for the number of mixture components per state versus the number of training iterations (Table 6.45) and recording conditions (Table 6.46).

6.3.13 Comparison with CU-HTK

Table 6.47 shows a comparison of word error rates for various discriminative training criteria between the CU-HTK system and the RWTH speech recognition system. Compared to the RWTH system, the CU-HTK baseline is more than 7% higher in terms of word error rate which gives a discriminative training procedure additional headroom for larger relative improvements. Overall, the MCE criterion gives the best performance, although there is no significant difference compared to a highly optimized system trained under the MMI criterion.

Table 6.44: Error rates obtained with I-smoothing and single densities trained under the MMI criterion on the male speaker portion of the HUB-4 corpus.

	HUB-4, male speaker portion		
	mix.-spec. variances		
I-smoothing	PER[%]	WER[%]	SER[%]
0	18.12	27.27	94.95
50	18.12	27.28	94.95
100	18.14	27.30	94.95
150	18.14	27.27	94.95
200	18.13	27.24	94.95
250	18.16	27.27	94.95
500	18.13	27.17	94.70
750	18.05	27.11	94.70
1000	17.97	27.06	95.45
2000	17.94	26.87	94.70
5000	18.07	27.03	94.95
15000	18.29	27.53	95.20

Table 6.45: Performance of a discriminative training under the MMI criterion measured on the HUB-4 corpus (male speaker portion) for an across-word system using mixture-specific variances. The MMI performance is shown for each iteration and separated by number of densities per mixture (dns/mix).

	HUB-4, male speaker portion						
	MLE	MMI (iteration)					
dns/mix	–	1	2	3	4	5	impr.
1.00	28.31	27.39	27.27	28.18	30.29	32.32	1.04
1.99	26.63	25.90	25.90	27.36	29.24	31.45	0.73
3.89	25.27	24.72	24.70	26.20	28.19	30.00	0.55
6.92	24.33	23.78	23.65	24.94	27.06	28.87	0.68
10.53	23.50	23.06	23.06	24.41	26.12	27.87	0.44
13.22	23.07	22.56	22.62	23.45	24.72	26.22	0.51
14.77	23.05	22.45	22.48	22.72	23.58	24.68	0.60
15.56	22.81	22.25	22.09	22.28	23.39	24.55	0.72

Table 6.46: Performance of a discriminative training under the MMI criterion measured on the HUB-4 corpus (male speaker portion) for an across-word system using mixture-specific variances. The MMI performance is broken down by recording conditions ranging from clean speech (f0) to heavily degraded speech (fx). Results are shown for an increasing number of densities per mixture (dns/mix).

	HUB-4, male speaker portion									
	MLE	MMI (recording condition)								
dns/mix	–	f0	f1	f2	f3	f4	f5	fx	all	impr.
1.00	28.31	17.77	28.71	36.37	34.60	29.04	30.47	41.69	27.27	1.04
1.99	26.63	16.68	26.89	35.46	32.25	28.00	31.05	38.31	25.90	0.73
3.89	25.27	15.40	26.19	34.28	31.87	27.80	26.37	35.92	24.72	0.55
6.92	24.33	14.61	25.53	33.21	30.80	25.48	24.80	34.44	23.65	0.68
10.53	23.50	14.32	24.75	31.87	31.71	24.91	24.41	32.96	23.06	0.44
13.22	23.07	13.68	24.12	31.28	31.41	25.22	22.85	32.89	22.56	0.51
14.77	23.05	13.64	24.16	31.31	31.49	24.60	22.85	32.18	22.45	0.60
15.56	22.81	13.52	24.09	30.98	31.26	23.36	21.09	31.41	22.09	0.72

Table 6.47: Comparison of recognition results obtained with the RWTH speech recognizer and the CU-HTK system as published in [Povey 04, p. 113] on the HUB-4 corpus. Both systems use discriminatively trained across-word models.

		HUB-4, WER[%]							
criterion	site	f0	f1	f2	f3	f4	f5	fx	avg.
ML	CU-HTK	11.6	26.2	38.7	–	24.6	24.8	55.4	29.6
	RWTH	13.1	24.4	31.9	32.1	24.8	22.8	33.4	22.0
MMI	CU-HTK	12.0	24.4	34.5	–	22.8	23.4	51.8	27.9
		11.1	24.4	35.6	–	22.8	22.6	52.7	27.8
	RWTH	12.6	23.2	31.3	31.5	23.7	21.0	31.9	21.1
MWE	RWTH	12.7	23.6	32.0	31.0	23.7	20.1	32.2	21.4
MPE	CU-HTK	10.6	22.9	33.7	–	21.4	22.8	48.9	26.2
	RWTH	12.8	23.6	31.5	30.8	24.0	20.3	32.4	21.4
MCE	RWTH	12.6	23.0	31.3	31.3	23.8	20.4	31.7	21.0

6.4 Conclusions

This chapter presented and discussed a large number of experimental results for discriminative training in automatic speech recognition. Besides some fundamental investigations that addressed the question of how increasing the model complexity affects the performance gains achieved with a discriminative training, several other aspects were pursued, including a discriminative training with different variance tying schemes and a comparison of methods to adjust the step size in the *Extended Baum* algorithm if multiple variances are used. The *fast*-method for choosing the step size could be shown to provide much better convergence rates although it suffers from overfitting the training data. The suggested Λ-method showed a consistently good performance on all settings independent of the underlying variance tying scheme.

Section 6.3.9 presented results for the first successful implementation of a lattice-based *Minimum Classification Error* training.

In Section 6.3.10, a large number of discriminative training criteria was investigated and compared within the extended unifying approach suggested in Chapter 4. In particular, both the MWE and the MPE criterion were not able to outperform the MMI criterion on the Hub-4 corpus for the investigated settings. Performance gains in the same order of magnitude as obtained with the MMI criterion could also be achieved with the Shannon and the Gini criterion. Both criteria are special cases of the Diversity index. The Jeffreys criterion could be shown to yield a solid performance if a globally pooled variance vector is used. Overall, the best results could be obtained with the MCE criterion which performed best on the Hub-4 corpus for the globally pooled variance system as well as on the system using density-specific variances.

As demonstrated in Section 6.3.11, both the MMI and MCE trained systems could be further improved by using unigram-decoded lattices in training. This advances the results found in [Schlüter & Müller$^+$ 99] according to which lattices used in discriminative training should be rescored with a low-order language model.

On the SieTill corpus, a word error rate of 1.36% could be achieved with an MMI training of Gaussian HMMs using density specific variances that were estimated in a combined LDA-MLLT transformed feature space. The step size was adjusted using the Λ method as described in Section 4.3.2. The reported word error rate is the lowest error rate ever reported on this corpus and it defines the new top mark for this task.

Summarizing the results from this chapter, the recommended setting for a discriminative training in automatic speech recognition employs Gaussian mixture densities with density-specific variances that are optimized in a combined LDA-MLLT transformed feature space under the MCE criterion using the Λ method for adjusting the step size. Word graphs produced in the first iteration should be generated using an integrated unigram search that is performed on HMM networks as described in Chapter 5.

Chapter 7

Maximum Entropy and Discriminative Training

Over the past years, *Maximum Entropy* (ME) based training methods were successfully applied to various tasks in natural language processing as, for instance, language modeling [Rosenfeld 96], part of speech tagging [Radnaparkhi 96], language understanding [Bender & Macherey$^+$ 03], and statistical machine translation [Och & Ney 02]. However, unlike text-based natural language processing, ME based methods have hardly been investigated for automatic speech recognition. An exemption is the work by [Likhododev & Gao 02] who apply the ME approach to estimate the parameters of a direct posterior model for the recognition phonemes. The approach generalizes the *Maximum Entropy Markov Model* (MEMM) proposed by [McCallum & Freitag$^+$ 00] such that sequential processes with complex contextual information can be trained.

One reason why ME based training methods are rarely used in acoustic modeling is due to the problem that the optimization procedure is difficult to handle if the objective function contains hidden variables. Hidden variables may lead to non-convex objective functions, and therefore, some training algorithms commonly used in a ME framework like the *Generalized Iterative Scaling* (GIS) algorithm as well as its faster version, the *Improved Iterative Scaling* (IIS), cannot be applied without major obstacles. Automatic speech recognition has, however, at least two types of events which are not observable and thus have to be described via hidden variables: (1) the alignment of a sequence of acoustic observations with the states of a Markov chain and (2) the component weights of a mixture of (typically Gaussian) densities. Moreover, since the ME framework aims at optimizing the class posterior probabilities, the objective function corresponds with the *Maximum Mutual Information* (MMI) criterion, and in ASR there is already an effective optimization method known as the *Extended Baum* (EB) algorithm that allows for an efficient training of the model parameters under the MMI criterion.

Nevertheless, as is shown in this chapter, the principal of ME can still be applied to discriminatively train the acoustic model parameters of a speech recognizer based on Gaussian densities. To accomplish this, some of the premises that come with the GIS algorithm have to be relaxed. The approach suggested in this chapter employs feature functions based on cepstral coefficients as provided by the signal analysis front-ends of most speech recognizers (cf. Section 2.1). This is different form [Likhododev & Gao 02] who use rank-based lists of Gaussian model indices as feature functions.

Experiments were conducted on the SieTill speech corpus for the recognition of telephone line recorded German connected digit strings. Since the ME framework takes competing classes into account, performance gains are not only compared with the results of a *Maximum Likelihood* (ML) trained speech recognizer but also with a discriminative training under the MMI criterion.

The remainder of this chapter is organized as follows. Section 7.1 briefly describes the ME principal and introducess the GIS algorithm. In addition, it is shown how acoustic models based on Gaussian HMMs can be embedded into the log-linear framework. In Section 7.2, the update rules for the GIS algorithm are derived and compared analytically with the re-estimation equations obtained from a "conventional" discriminative approach based on the EB algorithm. Experimental results are presented in Section 7.3. Section 7.4 puts this work into the context of more recent developments in this area. The chapter concludes with a summary in Section 7.5.

7.1 The Maximum Entropy Principle

Given a set of training samples, the ME approach seeks to choose a distribution that is consistent with *constraints* derived from the training data while making as few assumptions as possible. It can be shown that the resulting distribution is well defined and amounts to a *log-linear model* [Darroch & Ratcliff 72].

The constraints derived from the training data are typically expressed via a set of *feature functions*. Let c and c' denote two class labels and let b denote a condition imposed on an observation x. The label c' is the *predicted* class (hypothesis) for the observation x while c shall designate x's *true* class membership. A feature function $f_{b,c'}(x,c)$ is defined as a function that returns a value $\alpha_c \in \mathbb{R}_{>0}$ if and only if (1) the predicted class c' corresponds with the true class c and (2) the observation x satisfies the condition b. Otherwise, the feature value shall be zero:

$$f_{b,c'}(x,c) := \begin{cases} \alpha_c > 0 & \text{if } c = c' \wedge \ b(x) = \text{true} \\ 0 & \text{otherwise.} \end{cases} \tag{7.1}$$

If the feature value is larger than zero, the feature function is said to be *active*; otherwise it is called *inactive*. Due to [Darroch & Ratcliff 72] the solution of the ME approach amounts to a log-linear or exponential model:

$$p_\Lambda(c|x) = \frac{1}{Z(x)} \exp\left[\sum_i \lambda_i f_i(x,c)\right], \tag{7.2}$$

Here, $\Lambda = \{\lambda_i\}$ denotes the set of model parameters, and $Z(x)$ is the normalization term:

$$Z(x) = \sum_{c'} \exp\left[\sum_i \lambda_i f_i(x,c')\right]. \tag{7.3}$$

For a sequence of labeled training samples (x_n, c_n), $(n = 1, ..., N)$, the objective function for the ME criterion is defined as:

$$G(\Lambda) \coloneqq \sum_{n=1}^{N} \log p_\Lambda(c_n|x_n) \tag{7.4}$$

$$= \sum_{x,c} N(x,c) \log p_\Lambda(c|x), \tag{7.5}$$

where $N(x,c)$ denotes the frequency of occurrences of (x,c) in the training data. The parameter set which maximizes Eq. (7.5) can be determined by derivating G wrt. λ_i:

$$\frac{\partial G}{\partial \lambda_i} = N_i - Q_i(\Lambda) \stackrel{!}{=} 0, \tag{7.6}$$

where

$$N_i := \sum_{x,c} N(x,c) \cdot f_i(x,c), \tag{7.7}$$

$$Q_i(\Lambda) := \sum_x N(x) \sum_c p_\Lambda(c|x) \cdot f_i(x,c). \tag{7.8}$$

Note that the counts $\{N_i\}$ are not necessarily integers since α_c may be any positive real number. Eq. (7.5) is a sum of convex functions, and hence, $G(\Lambda)$ is convex, too. Because of the convexity, there is exactly one global maximum which can effectively be determined by means of the GIS algorithm.

Algorithm 4 Generalized Iterative Scaling (GIS) algorithm. The parameter ε is a small positive value that controls the terminating condition of the outer loop

$\forall i$: compute N_i, init. $\Lambda_0 = \{\lambda_i^0\}$; $j := 0$
loop
  $G(\Lambda_j) := 0, \quad \forall i\colon\ Q_i(\Lambda_j) := 0$
  for each sample $n = 1, \ldots, N$ do
    $G(\Lambda_j) = G(\Lambda_j) + \log p_{\Lambda_j}(c_n|x_n)$
    for each class index $c = 1, \ldots, C$ do
      for each active feature i do
        $Q_i(\Lambda_j) := Q_i(\Lambda_j) + p_{\Lambda_j}(c|x_n)$
  $\forall i\colon\ \lambda_i^{j+1} := \lambda_i^j + \frac{1}{F} \log\left(N_i / Q_i(\Lambda_j)\right)$
  if $G(\Lambda_{j+1})/N < \varepsilon$ stop; else $j := j + 1$;

A prerequisite of the GIS algorithm is that the number of active feature functions is constant in each iteration. This constraint is enforced by adding a *complement feature* f_0:

$$f_0(x, c) := F - \sum_i f_i(x, c), \quad F := \max_{x,c} \sum_i f_i(x, c). \tag{7.9}$$

The complement feature fills the gap between the maximum over the sum of all feature functions, F, and the sum over the active feature functions. The quantity F has to be determined in advance on the training corpus. According to [Pietra & Pietra$^+$ 97], the parameter update $\Delta\lambda_i$ for a feature i results from solving the equation:

$$Q_i(\Lambda) \cdot \exp\left[\Delta\lambda_i F\right] = N_i, \tag{7.10}$$

which amounts to the GIS algorithm as shown in Algorithm 4.

7.1.1 Converting a Gaussian HMM into a Log-linear Model

Let $r = 1, \ldots, R$ denote a sequence of training utterances, where each utterance shall be given as a sequence of acoustic observation vectors $X_r = x_{r1}, ..., x_{rT_r}$ together with a reference transcription $W_r = w_{r1}, ..., w_{rN_r}$. Since most speech recognizers employ cepstral features, the feature functions $f_{s'd}$ are defined as a set of projections that map a D-dimensional observation vector $x \in \mathbb{R}^D$ onto its dth component if and only if the predicted Markov state s' corresponds with the state s the observation is aligned with. To ensure that the feature functions are positive and fulfill the sum-to-one constraint, the observations are affinely transformed with a diagonal scaling matrix $\mathbf{A}$ plus an offset (*bias*) b such that the following constraint holds for all training samples $\{x_t\}$:

$$y_t := \mathbf{A}^\top \cdot x_t + b \ \text{ with } \ y_{td} > 0 \quad \text{and} \quad \sum_{r=1}^{R} \sum_{t=1}^{T_r} y_t = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} \tag{7.11}$$

The affine transformation pertains the convexity of the optimization problem, and therefore it is safe to enforce the positivity and the normalization constraint through the above transformation of the data. The set of feature functions is then defined as:

$$f_{s'd}(x, s) := \begin{cases} y_d & \text{if } s = s' \\ 0 & \text{otherwise.} \end{cases} \tag{7.12}$$

The emission probability of a Markov state s given an acoustic observation vector x is modeled via a Gaussian distribution $p(x|\theta_s) = \mathcal{N}(x|\mu_s, \mathbf{\Sigma})$ with $\theta_s = \{\mu_s, \mathbf{\Sigma}\}$ comprising a state dependent mean vector $\mu_s \in \mathbb{R}^D$ and a globally pooled covariance matrix $\mathbf{\Sigma} \in \mathbb{R}^{D \times D}$. Using a single, globally pooled covariance matrix has the advantage that the terms quadratic in x in the density function of the Gaussian distribution cancel due to the explicit re-normalization. As a consequence, we obtain a log-linear model for the emission probabilities:

$$p(W_r|X_r) \underset{\text{Viterbi}}{=} \frac{p(W_r) \cdot \max\limits_{s_1^{T_r}|W_r} \left\{ \prod\limits_{t=1}^{T_r} p(s_t|s_{t-1}) \cdot \mathcal{N}(x_t|\mu_{s_t}, \Sigma) \right\}}{\sum\limits_{W \in \mathcal{M}_r} p(W) \cdot \max\limits_{s_1^{T_r}|W} \left\{ \prod\limits_{t=1}^{T_r} p(s_t|s_{t-1}) \cdot \mathcal{N}(x_t|\mu_{s_t}, \Sigma) \right\}} \tag{7.13}$$

$$= \frac{p(W_r) \cdot \max\limits_{s_1^{T_r}|W_r} \left\{ \prod\limits_{t=1}^{T_r} p(s_t|s_{t-1}) \cdot \exp(\mu_{s_t}^\top \mathbf{\Sigma}^{-1} x - \frac{1}{2} \mu_{s_t}^\top \mathbf{\Sigma}^{-1} \mu_{s_t}) \right\}}{\sum\limits_{W} p(W) \cdot \max\limits_{s_1^{T_r}|W} \left\{ \prod\limits_{t=1}^{T_r} p(s_t|s_{t-1}) \cdot \exp(\mu_{s_t}^\top \mathbf{\Sigma}^{-1} x - \frac{1}{2} \mu_{s_t}^\top \mathbf{\Sigma}^{-1} \mu_{s_t}) \right\}} \tag{7.14}$$

$$= \frac{p(W_r) \cdot \max\limits_{s_1^{T_r}|W_r} \left\{ \prod\limits_{t=1}^{T_r} p(s_t|s_{t-1}) \, e^{\sum_{d=1}^{D+1} \widehat{\lambda}_{s_t,d} \cdot f_{s_t,d}(\widehat{x}, s_t)} \right\}}{\sum\limits_{W \in \mathcal{M}_r} p(W) \cdot \max\limits_{s_1^{T_r}|W} \left\{ \prod\limits_{t=1}^{T_r} p(s_t|s_{t-1}) \, e^{\sum_{d=1}^{D+1} \widehat{\lambda}_{s_t,d} \cdot f_{s_t,d}(\widehat{x}, s_t)} \right\}}. \tag{7.15}$$

Here, $\mathcal{M}_r$ denotes a set of competing word sequences, and

$$\widehat{\lambda}_s = \left[\eta_s^\top \mathbf{\Sigma}^{-1} \mathbf{A}^{-\top}, \; -\frac{1}{2} \big(\log\det(2\pi\mathbf{\Sigma}) + \eta_s^\top \mathbf{\Sigma}^{-1} \eta_s \big) \right]^\top \in \mathbb{R}^{D+1}, \tag{7.16}$$

with $\eta_s = \mu_s + \mathbf{A}^{-\top} b$ and the augmented observation vector $\widehat{x} = [x^\top, 1]^\top \in \mathbb{R}^{D+1}$. Ideally, the set $\mathcal{M}_r$ would contain all possible word sequences. In practice, $\mathcal{M}_r$ is obtained through a recognition pass and can be represented by word lattices or N-best lists (cf. Chapter 5).

Although the ME objective function corresponds with the *Maximum Mutual Information* (MMI) criterion, we use the symbol $\mathcal{F}_{\text{ME}}$ to emphasize the use of the log-linear model:

$$\mathcal{F}_{\text{ME}}(\Lambda) = \sum_{r=1}^{R} \log \frac{p(W_r) \cdot \rho(X_r|W_r)}{\sum\limits_{W \in \mathcal{M}_r} p(W) \cdot \rho(X_r|W)}, \tag{7.17}$$

where

$$\rho(X_r|W) = \max_{s_1^{T_r}|W} \left\{ \prod_{t=1}^{T_r} p(s_t|s_{t-1}) \, e^{\sum_{d=1}^{D+1} \widehat{\lambda}_{s_t,d} \cdot f_{s_t,d}(\widehat{x}, s_t)} \right\}. \tag{7.18}$$

In contrast to $p(W_r|X_r)$, $\rho(X_r|W)$ is not a real distribution because it does not satisfy the normalization constraint. Nevertheless, the quantity can still be used to decode a spoken utterance since for a recognizer it suffices that scores are comparable.

To maximize the objective function, we first determine the partial derivative of $\mathcal{F}_{\text{ME}}$ with respect to λ_{sd}:

$$\frac{\partial \mathcal{F}_{\text{ME}}(\Lambda)}{\partial \lambda_{s,d}} = \sum_{r=1}^{R} \sum_{t=1}^{T_r} \Big[\gamma_{rt}(s|W_r) - \gamma_{rt}(s) \Big] \cdot y_{rtd}. \tag{7.19}$$

Here, $\gamma_{rt}(s|W_r)$ denotes the forward-backward (FB) probability of hypothesizing state s at time frame t given the transcription W_r, and $\gamma_{rt}(s)$ is the generalized FB probability of hypothesizing state s at time frame t independent of any word sequence. As described in Chapter 5, the FB probabilities can be estimated efficiently on word lattices or state graphs.

7.2 Maximum Entropy in Automatic Speech Recognition

So far, we have not yet addressed the problem of aligning the acoustic observations with the states of a Markov chain. As mentioned before, the alignment cannot be observed directly but has to be described via a hidden variable. Moreover, in general the alignment is not a fixed quantity but may change in the course of the training process. However, we can make the assumption that the alignment of a spoken word sequence (i.e., the $N_{s,d}$ statistics) remains unaltered over a small number of GIS iterations. This assumption is of course risky and bears a certain wrongness since an update of the parameter set Λ might also change the optimal alignment of a spoken sentence. To alleviate this problem, the training utterances are re-aligned occassionally to ensure that the $N_{s,d}$ statistics remain reliable. As a consequence, the $N_{s,d}$ counts do now depend on the parameter set Λ, and the statistics from Eqs. (7.7) and (7.8) have to be redefined accordingly:

$$N_{s,d}(\Lambda) = \sum_{r=1}^{R} \sum_{t=1}^{T_r} \gamma_{r,t}(s|W_r) \cdot y_{r,t,d}, \tag{7.20}$$

$$Q_{s,d}(\Lambda) = \sum_{r=1}^{R} \sum_{t=1}^{T_r} \gamma_{r,t}(s) \cdot y_{r,t,d}. \tag{7.21}$$

Due to the required re-alignments, the optimization problem is no longer convex, and as a consequence, the GIS algorithm is not guaranteed to converge to a globally optimal solution. Nevertheless, the gradients of the resulting update rules can still be used to optimize the parameters of the log-linear acoustic model.

$$\overline{\lambda}_{s,d} = \lambda_{s,d} + \frac{1}{F} \log \Big[N_{s,d}(\Lambda) / Q_{s,d}(\Lambda) \Big]. \tag{7.22}$$

7.2.1 Comparison with other Discriminative Training Methods

In contrast to Eq. (7.22), the re-estimation formulae in a 'standard' discriminative training lead to update rules that depend on the logarithm of the *difference* between the $N_{s,d}$ and $Q_{s,d}$ counts rather than their ratio. To verify this, we review a standard optimization technique for discriminative training based on the *Extended Baum* (EB) algorithm. Although the objective functions are identical for the ME and the MMI criterion, the MMI objective function is introduced as a separate definition to emphasize the functional form of the model distribution (i.e., a Gaussian model vs. a log-linear model) and the optimization method applied (i.e., EB vs. GIS):

$$\mathcal{F}_{\mathrm{MMI}}(\theta) = \sum_{r=1}^{R} \log \frac{p(W_r) \cdot p_\theta(X_r \,|\, W_r)}{\sum\limits_{W \in \mathcal{M}_r} p(W) \cdot p_\theta(X_r \,|\, W)}. \tag{7.23}$$

The derivative of $\mathcal{F}_{\mathrm{MMI}}(\theta)$ wrt. θ_s yields:

$$\frac{\partial \mathcal{F}_{\mathrm{MMI}}(\theta)}{\partial \theta_s} = \Gamma_s \left(\frac{\partial \log p(x \,|\, \theta_s)}{\partial \theta_s} \right), \tag{7.24}$$

where Γ_s denotes the discriminative average for state s which is defined by (cf. Eq. (4.41)):

$$\Gamma_s\big(g(X)\big) := \sum_{r=1}^{R}\sum_{t=1}^{T_r}\Big[\gamma_{r,t}(s\,|\,W_r) \;-\; \gamma_{r,t}(s)\Big]\cdot g(x_{r,t}). \tag{7.25}$$

The EB method leads to the following re-estimation equations (cf. Eq. (4.62)):

$$\overline{\mu}_{s,d} = \mu_{s,d} + \epsilon_{s,d}\cdot\Big[\Gamma_{s,d}(x) - \mu_{s,d}\cdot\Gamma_s(1)\Big], \tag{7.26}$$

with $\epsilon_{s,d} = 1/\big(\Gamma_s(1) - D_s\big)$ and D_s denoting a state-specific iteration constant that controls the convergence rate of the training process. The choice of D_s has to ensure that $\epsilon_{s,d}$ is positive which can be accomplished by setting

$$D_s = h\cdot\max\big\{D_s^{\min},\; 1/\beta - \Gamma_s(1)\big\}.$$

Here, $D_s^{\min}$ is a constant that guarantees the positivity of all variances. The parameter $\beta > 0$ is chosen such as to prevent overflows caused by low-valued denominators, and $h > 1$ is a scaling factor that controls the step size of the gradient (cf. Section 4.3). With the decomposition of the discriminative averages Γ_s into the correct model Γ_s^{num} and the competing model Γ_s^{den},

$$\Gamma_s^{\mathsf{num}}\big(g(X)\big) = \sum_{r=1}^{R}\sum_{t=1}^{T_r}\gamma_{r,t}(s\,|\,W_r)\cdot g(x_{r,t}) \tag{7.27}$$

$$\Gamma_s^{\mathsf{den}}\big(g(X)\big) = \sum_{r=1}^{R}\sum_{t=1}^{T_r}\gamma_{r,t}(s)\cdot g(x_{r,t}), \tag{7.28}$$

Eq. (7.22) can be re-written as follows, and we obtain the expressions for the re-estimation formulae for both the GIS algorithm and the EB method:

$$\text{GIS:}\quad \overline{\lambda}_{s,d} = \lambda_{s,d} + \frac{1}{F}\cdot\log\Big[\Gamma_{s,d}^{\mathsf{num}}(y)\,/\,\Gamma_{s,d}^{\mathsf{den}}(y)\Big], \tag{7.29}$$

$$\text{EB:}\quad \overline{\mu}_{s,d} = \mu_{s,d} + \epsilon_{s,d}\cdot\Big[\Gamma_{s,d}(x) - \mu_{s,d}\cdot\Gamma_{s,d}(1)\Big]. \tag{7.30}$$

Comparing both gradients, one may expect that the convergence of the GIS algorithm will turn out to be moderately slow since the update rule depends on the logarithm of the ratio of the correct model Γ_s^{num} and the competing model Γ_s^{den} whereas the gradient of the EB algorithm is based on the difference $\Gamma_s^{\mathsf{num}} - \Gamma_s^{\mathsf{den}}$.

7.2.2 Integration of Linear Feature Transformations

The log-linear form of the emission probabilities allows for initializing the parameter set Λ with a linear feature transformation as, for instance, a *Linear Discriminant Analysis* (LDA). Let $\mathbf{H}$ denote the LDA transformation matrix. If $\mathbf{H}$ has full rank, then the mean vector m and the covariance matrix $\mathbf{S}$ in the $(\mathbf{A}+b)$-transformed feature space can be computed from the parameters of the Gaussian distribution $\mathcal{N}(\mathbf{H}^\top x|\mu_z, \mathbf{\Sigma}_z)$ in the original, LDA-transformed feature space:

$$m = \mathbf{A}\mathbf{H}^{-\top}\mu_z + b, \tag{7.31}$$

$$\mathbf{S} = \mathbf{A}\mathbf{H}^{-\top}\mathbf{\Sigma}_z\mathbf{H}^{-1}\mathbf{A}^\top. \tag{7.32}$$

Thus, we obtain the following expression for λ_s which can be used to bootstrap the free parameters of the log-linear model:

$$\lambda_s = \begin{bmatrix} \mathbf{S}^{-1}m_s \\ -\frac{1}{2}\big(\log\det(2\pi\mathbf{\Sigma}_z) + m_s^\top\mathbf{S}^{-1}m_s\big) \end{bmatrix}. \tag{7.33}$$

7.3 Experiments

Experiments were conducted on the *SieTill* corpus for telephone line recorded, continuously spoken German connected digit strings. The corpus comprises approximately 13k sentences for both the training and the test set. Table 7.1 summarizes some corpus statistics. A more detailed description of the recognition task is given in Table 6.1 on page 78.

The recognition system used for the experiments employs gender-dependent whole-word HMMs with continuous emission densities. For each gender, 214 distinct states plus one state for silence were used. The observation vectors consist of 12 cepstral features with their first derivatives and the second derivative of the first component. The baseline recognizer was trained under the ML criterion in the Viterbi approximation and achieves a *word error rate* (WER) of 4.59% (cf. Table 7.2). Both the ME training and the EB training were bootstrapped with the ML trained acoustic model. Whenever the objective function increased by more than 10% relative, the GIS algorithm was interrupted and the training utterances were re-aligned before the GIS optimization was continued. After 15 re-alignments, the ME-based system achieved a WER of 3.52% which is a relative improvement of 23% compared to the ML-trained baseline system. Using the same number of re-alignments for the EB algorithm, the standard discriminative approach achieved a WER of 4.11%. Though this is a substantial improvement of the ME training over the EB algorithm, it is important to note that the GIS algorithm performs thousands of iterations between two adjacent re-alignment phases to yield the reported gain. The ability of the ME criterion to generalize to unseen data is, however, remarkable.

Using a Linear Discriminant Analysis (LDA), the relative performance gain decreases, but is still significant. Thus the relative improvement between the baseline result and the ME approach is 14%. Other than the systems trained under the ML criterion or the MMI criterion using the EB algorithm, the parameters of the ME-trained system were initialized with the LDA transformation matrix according to Eq. (7.33). Because of that, the GIS algorithm operates on the untransformed high-dimensional feature space and has therefore the possibility to extract more information from the data whereas the ML-trained system and the EB-optimized system operate on the LDA transformed features. Due to the comparatively small number of re-alignments, the ME based approach has not yet reached its optimum and additional training iterations might be necessary in order to achieve similar performance gains as have been obtained with the optimization using the EB algorithm.

Table 7.1: Corpus statistics for the SieTill corpus.

corpus	female		male	
	sent.	digits	sent.	digits
test	6176	20205	6938	22881
train	6113	20115	6835	22463

Table 7.2: Word error rates on the SieTill test corpus for different optimization criteria.

method	LDA	# re-align	WER[%]	SER[%]
ML	no	30	4.59	11.34
ME		15	**3.52**	**9.17**
EB		15	4.11	10.36
ML	yes	30	3.78	9.74
ME		10	3.24	8.44
EB		20	**2.95**	**7.56**

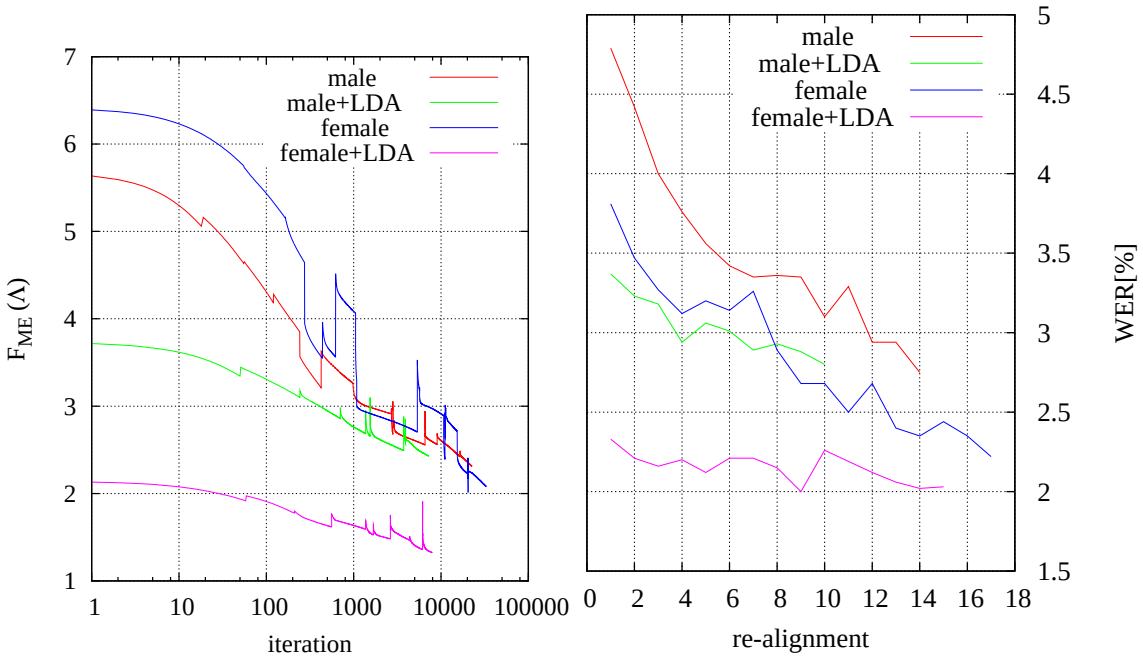


Figure 7.1: Evolution of the ME objective function in the course of the iteration process (left) and word error rates at the re-alignment points (right) on the SieTill training corpus.

7.4 Recent Developments

By now, direct posterior models for automatic speech recognition have become of increasing interest in the research community. In [Heigold & Schlüter$^+$ 07] it is shown that Gaussian HMMs (GHMMs) are equivalent to Gaussian HMM-like *Hidden Conditional Random Fields*. While conventional GHMMs are usually estimated with a criterion on a segment level, hybrid approaches are typically based on a formulation of the criterion on the frame level. The study shows that improvements of HCRFs over GHMMs found in literature are not due to a refined acoustic modeling but rather from the more robust formulation of the underlying optimization problem or spurious local optima. In [Heigold & Deselaers$^+$ 08a], the work presented in this chapter has been continued by extending GIS such that it does not only allow for training log-linear models with hidden variables but also enables the optimizatio of discriminative training criteria other than the Maximum Mutual Information criterion as, for instance, the Minimum Phone Error criterion.

A completely different approach to the training problem of linear models is suggested in the following chapter which extends the *Minimum Error Rate Training* (MERT) algorithm for N-best lists as suggested by [Och 03] for *statistical machine translation* to word lattices. The power of the extended MERT algorithm is that it allows for efficiently constructing and exploring the exact error surface over all sentence hypotheses that are encoded in a word lattice under virtually any automated evaluation criterion that is used in natural language processing.

7.5 Conclusions

While *Maximum Entropy* (ME) based learning procedures have been successfully applied to text based natural language processing, there are only few investigations on using the ME framework

for acoustic modeling in automatic speech recognition. In this chapter, it was shown that the *Generalized Iterative Scaling* (GIS) algorithm can be used as an optimization algorithm to discriminatively train the parameters of an automatic speech recognizer based on *Hidden Markov Models* (HMMs) with continuous Gaussian densities. The ME approach was compared analytically and experimentally with both a conventional *Maximum Likelihood* (ML) training and a standard approach to discriminative training under the *Maximum Mutual Information* (MMI) criterion based on the *Extended Baum* (EB) algorithm. Experiments conducted on a recognition task for continuously spoken connected German digit strings achieved a relative improvement of up to 23% over the ML trained system, and more than 14% over the MMI criterion trained with the EB algorithm. In combination with a linear discriminant analysis, the EB algorithm performed better and outperformed the ME approach by 9% relative.

Chapter 8

Minimum Error Rate Training

Minimum Error Rate Training (MERT) is an effective means to estimate the feature function weights of a linear model such that an automated evaluation criterion for measuring system performance can directly be optimized in training. To accomplish this, the training procedure determines for each feature function its exact error surface on a given set of sentence hypotheses. The feature function weights are then adjusted by traversing the error surface combined over all sentences and picking those values for which the resulting error count reaches a minimum. Typically, candidates in MERT are represented as N-best lists which contain the N most probable sentence hypotheses produced by a decoder. This chapter presents a novel algorithm that allows for efficiently constructing and representing the exact error surface of *all* sentence hypotheses that are encoded in a word lattice. Compared to N-best MERT, the number of sentence hypotheses thus taken into account increases by several orders of magnitudes. The proposed method can be used to train the feature function weights of a log-linear combination of feature functions and multiple knowledge sources.

The remainder of this chapter is organized as follows. Section 8.1 motivates the general concept behind the MERT criterion. Section 8.2 briefly reviews N-best MERT and introduces some basic concepts that are used in order to develop the line optimization algorithm for word lattices in Section 8.3. Section 8.4 presents an upper bound on the complexity of the unsmoothed error surface for the sentence hypotheses represented in a word lattice. This upper bound is used to prove the space and runtime efficiency of the suggested algorithm. The chapter concludes with a summary in Section 8.5.

8.1 Introduction

Many statistical methods in natural language processing aim at minimizing the probability of sentence errors. In practice, however, system quality is often measured based on error metrics that assign non-uniform costs to classification errors and thus go far beyond counting the number of wrong decisions. Examples are the mean average precision for ranked retrieval, the F-measure for parsing, and the *word error rate* in automatic speech recognition. A class of training criteria that provides a tighter connection between the decision rule and the final error metric is known as *Minimum Error Rate Training* (MERT) and has been suggested in the context of statistical machine translation in [Och 03].

MERT aims at estimating the model parameters such that the decision under the zero-one loss function maximizes some end-to-end performance measure on a development corpus. In combination with log-linear models, the training procedure allows for a direct optimization of the unsmoothed error count. The criterion can be derived from Bayes' decision rule as follows: Let $X_r = x_{r1}, ..., x_{rT_r}$ denote a sequence of acoustic observation vectors together with the corresponding spoken word sequence $W_r = w_{r1}, ..., w_{rN_r}$. Under the zero-one loss function, the sentence hypothesis which maximizes the *a posteriori* probability is chosen:

$$\hat{W} = \arg\max_{W} \left\{ \Pr(W|X_r) \right\} \tag{8.1}$$

Since the true posterior distribution is unknown, $\Pr(W|X_r)$ is modeled via a log-linear model which combines some feature functions $h_m(W, X)$ with feature function weights λ_m, $m = 1, ..., M$:

$$\Pr(W|X_r) = p_{\lambda_1^M}(W|X_r) \tag{8.2}$$

$$= \frac{\exp\left[\sum_{m=1}^{M} \lambda_m h_m(W, X_r)\right]}{\sum_{W'} \exp\left[\sum_{m=1}^{M} \lambda_m h_m(W', X_r)\right]} \tag{8.3}$$

The feature function weights are the parameters of the model, and the objective of the MERT criterion is to find a parameter set $\hat{\lambda}_1^M$ that minimizes the error count on a representative set of training sentences. More precisely, let $(\mathcal{X}, \mathcal{W}) := (X_r, W_r)_{r=1,...,R}$ denote the training utterances of a speech corpus, each consisting of a sequence of acoustic observation vectors $X_r = x_{r1}, ..., x_{rT_r}$ together with the corresponding spoken word sequence $W_r = w_{r1}, ..., w_{rN_r}$. Assuming that the corpus-based error count for some sentence hypotheses $\mathcal{M}_r = \{W(X_r) \,|\, r = 1, ..., R\}$ is additively decomposable into the error counts of the individual sentences, i.e., $E(\mathcal{W}, \mathcal{V}) = \sum_{r=1}^{R} E(W_r, W(X_r))$, the MERT criterion is given as:

$$\hat{\lambda}_1^M = \arg\min_{\lambda_1^M} \left\{ \sum_{r=1}^{R} E\big(W_r, \hat{W}(X_r; \lambda_1^M)\big) \right\} \tag{8.4}$$

$$= \arg\min_{\lambda_1^M} \left\{ \sum_{r=1}^{R} \sum_{W \in \mathcal{M}_r} E(W_r, W) \delta\big(\hat{W}(X_r; \lambda_1^M), W\big) \right\} \tag{8.5}$$

with

$$\hat{W}(X_r; \lambda_1^M) = \arg\max_{W} \left\{ \sum_{m=1}^{M} \lambda_m h_m(W, X_r) \right\} \tag{8.6}$$

In [Och 03], it was shown that linear models can effectively be trained under the MERT criterion using a special line optimization algorithm. This line optimization determines for each feature function h_m and sentence X_r the exact error surface on a set of sentence hypotheses $\mathcal{M}_r$. The feature function weights are then adjusted by traversing the error surface combined over all sentences in the training corpus and moving the weights to a point where the resulting error reaches a minimum.

Sentence hypotheses in MERT are typically represented as N-best lists which contain the N most probable sentence hypotheses. A downside of this approach is, however, that N-best lists can only capture a very small fraction of the search space. As a consequence, the line optimization algorithm needs to repeatedly decode the development corpus and enlarge the candidate repositories with newly found hypotheses in order to avoid overfitting on $\mathcal{M}_r$ and preventing the optimization procedure from stopping in a poor local optimum.

This chapter presents a novel algorithm that allows for efficiently constructing and representing the unsmoothed error surface for *all* sentence hypotheses that are encoded in a word lattice. The number of sentence alternatives thus taken into account increases by several orders of magnitudes compared to N-best MERT.

8.2 Minimum Error Rate Training on N-best Lists

The goal of MERT is to find a weights set that minimizes the unsmoothed error count on a representative training corpus (cf. Eq. (8.4)). This can be accomplished through a sequence of line minimizations along some vector directions $\{d_1^M\}$. Starting from an initial point λ_1^M, computing

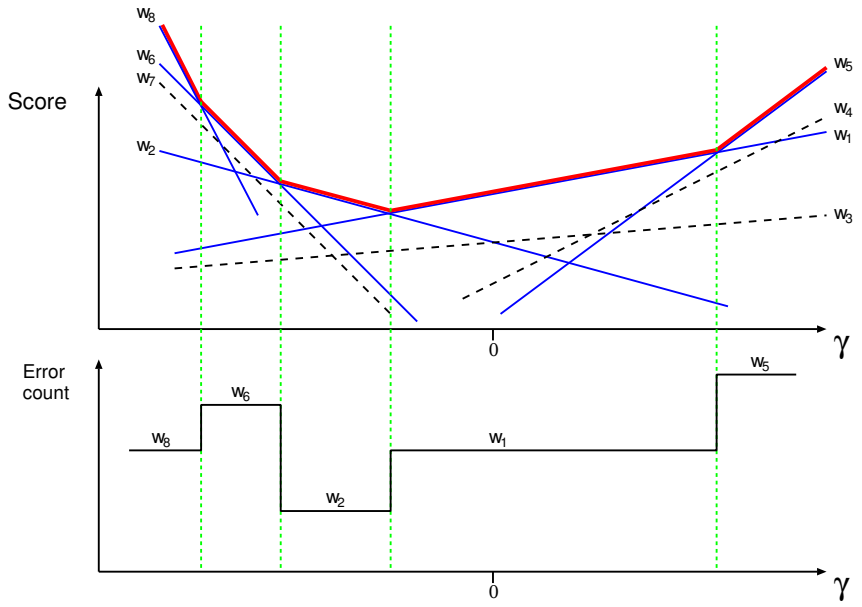


Figure 8.1: The upper envelope (bold, red curve) for a set of lines is the convex hull which consists of the topmost line segments. Each line corresponds to a sentence hypothesis and is thus related to a certain error count. Envelopes can efficiently be computed with Algorithm 5.

the most probable sentence hypothesis out of a set of K sentence hypotheses $\mathcal{M}_r = \{W_1, ..., W_K\}$ along the line $\lambda_1^M + \gamma \cdot d_1^M$ results in the following optimization problem [Och 03]:

$$\begin{aligned}
\hat{W}(X_r;\gamma) &= \arg\max_{W \in \mathcal{M}_r} \left\{ (\lambda_1^M + \gamma \cdot d_1^M)^\top \cdot h_1^M(W, X_r) \right\} \\
&= \arg\max_{W \in \mathcal{M}_r} \left\{ \underbrace{\sum_m \lambda_m h_m(W, X_r)}_{=a(W,X_r)} + \gamma \cdot \underbrace{\sum_m d_m h_m(W, X_r)}_{=b(W,X_r)} \right\} \\
&= \arg\max_{W \in \mathcal{M}_r} \underbrace{\left\{ a(W, X_r) + \gamma \cdot b(W, X_r) \right\}}_{(*)}
\end{aligned} \tag{8.7}$$

Hence, the total score $(*)$ for any sentence hypothesis corresponds to a line in the plane with γ as the independent variable. For any particular choice of γ, the decoder seeks that word sequence which yields the largest score and therefore corresponds to the topmost line segment.

Overall, the candidate repository $\mathcal{M}_r$ defines K lines where each line may be divided into at most K line segments due to possible intersections with the other $K-1$ lines. The sequence of the topmost line segments constitute the *upper envelope* which is the pointwise maximum over all lines induced by $\mathcal{M}_r$. The upper envelope is a convex hull and can be inscribed with a convex polygon whose edges are the segments of a piecewise linear function in γ [Papineni 99, Och 03]:

$$\mathsf{Env}(X) = \max_{W \in \mathcal{M}} \left\{ a(W, X) + \gamma \cdot b(W, X) : \gamma \in \mathbb{R} \right\} \tag{8.8}$$

The importance of the upper envelope is that it provides a compact encoding of all possible outcomes that a rescoring of $\mathcal{M}_r$ may yield if the parameter set λ_1^M is moved along the chosen

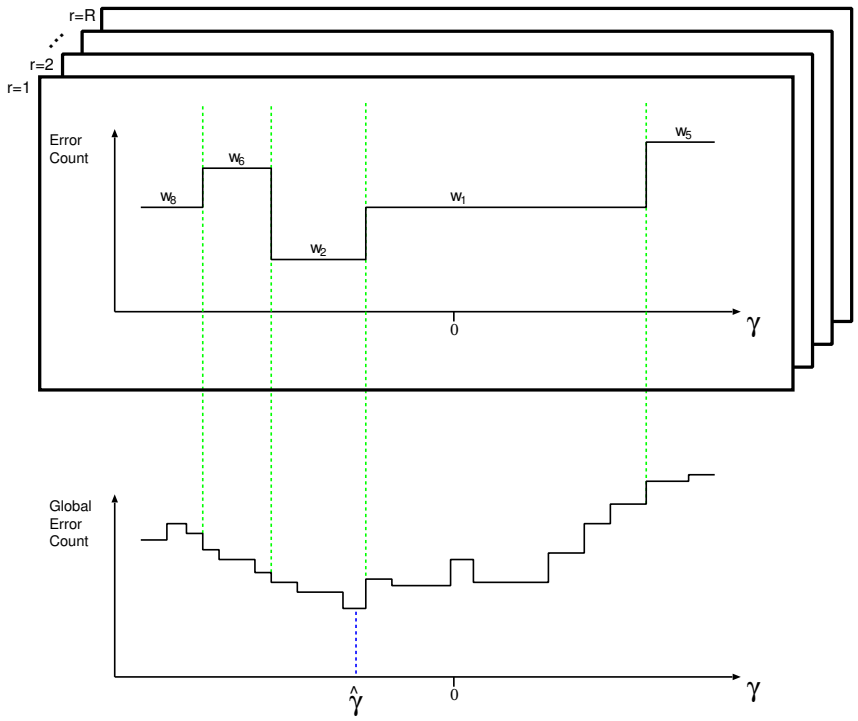


Figure 8.2: Global error surface obtained by merging the utterance specific error surfaces.

direction. Once the upper envelope has been determined, its constituent line segments can be projected onto the error counts of the corresponding sentence hypotheses (cf. Figure 8.1). This projection is independent of *how* the envelope is generated and can therefore be applied to any set of line segments[1].

An effective means to compute the upper envelope is a *sweep line* algorithm which is often used in computational geometry to determine the intersection points of a sequence of lines or line segments [Bentley & Ottmann 79]. The idea is to shift ("sweep") a vertical ray from $-\infty$ to $+\infty$ over the plane while keeping track of those points where two or more lines intersect. Since the upper envelope is fully specified by the topmost line segments, it suffices to store the following components for each line object ℓ: the x-intercept $\ell.x$ with the left-adjacent line, the slope $\ell.m$, and the y-intercept $\ell.y$; a fourth component, $\ell.W$, is used to store the sentence hypothesis. Algorithm 5 shows the pseudo code for a sweep line algorithm which reduces an input array `lvert a[0..K-1]`lvert consisting of the K line objects of the candidate repository $\mathcal{M}_r$ to its upper envelope. By construction, the upper envelope consists of at most K line segments. The endpoints of each line segment define the interval boundaries at which the decision made by the decoder will change. Hence, as γ increases from $-\infty$ to $+\infty$, we will see that the most probable sentence hypothesis will change whenever γ passes an intersection point.

Let $\gamma_1^r < \gamma_2^r < ... < \gamma_{N_r}^r$ denote the sequence of interval boundaries and let $\Delta E_1^r, \Delta E_2^r, ..., \Delta E_{N_r}^r$ denote the corresponding sequence of changes in the error count where ΔE_n^r is the amount by which the error count will change if γ is moved from a point in $[\gamma_{n-1}^r, \gamma_n^r)$ to a point in $[\gamma_n^r, \gamma_{n+1}^r)$. Both sequences together provide an exhaustive representation of the unsmoothed error surface for the sentence r along the line $\lambda_1^M + \gamma \cdot d_1^M$. The error surface for the whole training corpus is obtained

[1] For lattice MERT, it will therefore suffice to find an efficient way to compute the upper envelope over all sentence hypotheses that are encoded in a word graph.

Algorithm 5 SweepLine

input: array `a[0..K-1]` containing lines
output: upper envelope of `a`

```
sort(a:m);
j = 0; K = size(a);
for (i = 0; i < K; ++i) {
  ℓ = a[i];
  ℓ.x = -∞;
  if (0 < j) {
    if (a[j-1].m == ℓ.m) {
      if (ℓ.y <= a[j-1].y) continue;
      --j;
    }
    while (0 < j) {
      ℓ.x = (ℓ.y - a[j-1].y)/
            (a[j-1].m - ℓ.m);
      if (a[j-1].x < ℓ.x) break;
      --j;
    }
    if (0 == j) ℓ.x = -∞;
    a[j++] = ℓ;
  } else a[j++] = ℓ;
}
a.resize(j);
return a;
```

by merging the interval boundaries (and their corresponding error counts) over all sentences in the training corpus (cf. Figure 8.2). The optimal γ can then be found by traversing the merged error surface and choosing a point from the interval where the total error reaches its minimum.

After the parameter update, $\hat{\lambda}_1^M = \lambda_1^M + \gamma_{\text{opt}} \cdot d_1^M$, the decoder may find new sentence hypotheses which are merged into the candidate repositories if they are ranked among the top N candidates. The relation $K = N$ holds therefore only in the first iteration. From the second iteration on, K is usually larger than N. The sequence of line optimizations and decodings is repeated until (1) the candidate repositories remain unchanged and (2) $\gamma_{\text{opt}} = 0$.

8.3 Minimum Error Rate Training on Lattices

In this section, the algorithm for computing the upper envelope on N-best lists is extended to word lattices. For a description of how to generate lattices, see Section 5.1.6.

Formally, a word lattice for a sentence r is defined as a connected, directed acyclic graph $\mathcal{G}_r = (\mathcal{V}_r, \mathcal{E}_r)$ with vertice set $\mathcal{V}_r$, unique source and sink nodes $s, t \in \mathcal{V}_r$, and a set of arcs $\mathcal{E}_r \subset \mathcal{V}_r \times \mathcal{V}_r$. Each arc is labeled with a word w_{ij} and the (local) feature function values $h_1^M(w_{ij}, X_r)$. A path $\pi = (v_0, e_0, v_1, e_1, ..., e_{n-1}, v_n)$ in $\mathcal{G}_r$ (with $e_i \in \mathcal{E}_r$ and $v_i, v_{i+1} \in \mathcal{V}_r$ as the tail and head of e_i, $0 \leqslant i < n$) defines a partial sentence hypothesis W_π which is the concatenation of all word hypotheses along this path. The corresponding feature function values are obtained by summing

over the arc-specific feature function values:

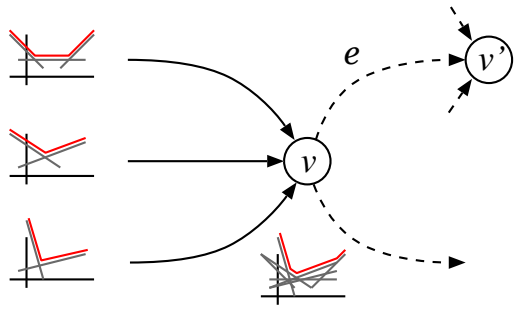
$$\pi : \bullet \xrightarrow[h_1^M(w_{0,1},X)]{w_{0,1}} \bullet \xrightarrow[h_1^M(w_{1,2},X)]{w_{1,2}} \cdots \xrightarrow[h_1^M(w_{n-1,n},X)]{w_{n-1,n}} \bullet$$
$$\begin{matrix} v_0 & & v_1 & & & & v_n \end{matrix}$$

$$W_\pi = \bigcirc_{\substack{i,j: \\ v_i \to v_j \in \pi}} w_{ij} = w_{0,1} \circ ... \circ w_{n-1,n}$$

$$h_1^M(W_\pi, X) = \sum_{\substack{i,j: \\ v_i \to v_j \in \pi}} h_1^M(w_{ij}, X)$$

In the following, we use the notation $\mathsf{in}(v)$ and $\mathsf{out}(v)$ to refer to the set of incoming and outgoing arcs for a node $v \in \mathcal{V}_r$. Similarly, $\mathsf{head}(e)$ and $\mathsf{tail}(e)$ denote the head and tail of $e \in \mathcal{E}_r$.

To develop the algorithm for computing the upper envelope of *all* sentence hypotheses that are encoded in a word lattice, we first consider a node $v \in \mathcal{V}_r$ with some incoming and outgoing arcs:



Each path that starts at the source node s and ends in v defines a partial sentence hypothesis which can be represented as a line (cf. Eq. (8.7)). We now assume that the upper envelope for these partial sentence hypotheses is known. The lines that constitute this envelope shall be denoted by $f_1, ..., f_N$. Next we consider continuations of these partial sentence hypotheses by following one of the outgoing arcs $e \in \mathsf{out}(v)$. Each such arc defines another line denoted by $g(e)$. If we add the slope and y-intercept of $g(e)$ to each line in the set $\{f_1, ..., f_N\}$, then the upper envelope will be constituted by segments of $f_1 + g(e), ..., f_N + g(e)$. This operation neither changes the number of line segments nor their relative order in the envelope, and therefore it preserves the structure of the convex hull. As a consequence, we can propagate the resulting envelope over an outgoing arc e to a successor node $v' = \mathsf{head}(e)$. Other incoming arcs for v' may be associated with different upper envelopes, and all that remains is to merge these envelopes into a single combined envelope. This is, however, easy to accomplish since the combined envelope is simply the convex hull of the union over the line sets which constitute the individual envelopes. Thus, by merging the arrays that store the line segments for the incoming arcs and applying Algorithm 5 to the resulting array we obtain the combined upper envelope for *all* partial sentence hypotheses that are associated with paths starting at the source node s and ending in v'. The correctness of this procedure is based on the following two observations:

(1) A single sentence hypothesis cannot constitute multiple line segments of the same envelope. This is because sentence hypotheses associated with different line segments are path-disjoint.

(2) Once a partial sentence hypotheses has been discarded from an envelope because its associated line $\tilde{f}$ is completely covered by the topmost line segments of the convex hull, there is no path continuation that could bring back $\tilde{f}$ into the upper envelope again. Proof: Suppose that such a continuation exists, then this continuation can be represented as a line g, and since $\tilde{f}$ has been discarded from the envelope, the path associated with g must also be a valid continuation for the line segments $f_1, ..., f_N$ that constitute the envelope. Thus it follows that $\max(f_1 + g, ..., f_N + g) =$

Algorithm 6 Lattice Envelope

input: a word lattice $\mathcal{G}_r = (\mathcal{V}_r, \mathcal{E}_r)$
output: upper envelope of $\mathcal{G}_r$

```
a = ∅;
L = ∅;
TopSort(G_X);
for v = s to t do {
  a = SweepLine( ⋃_{e∈in(v)} L[e]);
  foreach (e ∈ in(v))
    L.delete(e);
  foreach (e ∈ out(v)) {
    L[e] = a;
    for (i = 0; i < a.size(); ++i) {
      L[e][i].m = a[i].m + Σ_m d_m h_m(e,X);
      L[e][i].y = a[i].y + Σ_m λ_m h_m(e,X);
      L[e][i].p = a[i].p ∘ w_{v,head(e)};
    }
  }
}
return a;
```

$\max(f_1, ..., f_N) + g < \tilde{f} + g$ for some $\gamma \in \mathbb{R}$. This, however, is in contradiction with the premise that $\tilde{f} < \max(f_1, ..., f_N)$ for all $\gamma \in \mathbb{R}$.

To keep track of the word expansions when propagating an envelope over an outgoing arc $e \in \mathsf{tail}(v)$, the word label $w_{v,\mathsf{head}(e)}$ has to be appended from the right to all partial sentence hypotheses in the envelope. The complete algorithm then works as follows: First, all nodes in the word lattice are sorted in topological order. Starting with the source node, we combine for each node v the upper envelopes that are associated with v's incoming arcs by merging their respective line arrays and reducing the merged array into a combined upper envelope using Algorithm 5. The combined envelope is then propagated over the outgoing arcs by associating each $e \in \mathsf{out}(v)$ with a copy of the combined envelope. This copy is modified by adding the parameters (slope and y-intercept) of the line $g(e)$ to the envelope's constituent line segments. The envelopes of the incoming arcs are no longer needed and can be deleted in order to release memory. The envelope computed at the sink node is by construction the convex hull over all sentence hypotheses represented in the lattice, and it compactly encodes those candidates which maximize the decision rule Eq. (8.1) for any point along the line $\lambda_1^M + \gamma \cdot d_1^M$. Algorithm 6 shows the pseudo code. Note that the component $\ell.x$ does not change and therefore requires no update.

It remains to verify that the suggested algorithm is efficient in both running time and memory. For this purpose, we first analyze the complexity of Algorithm 5 and derive from it the running time of Algorithm 6.

After sorting, each line object in Algorithm 5 is visited at most three times. The first time is when it is picked by the outer loop. The second time is when it either gets discarded or when it terminates the inner loop. Whenever a line object is visited for the third time, it is irrevocably removed from the envelope. The runtime complexity is therefore dominated by the initial sorting and amounts to $\mathcal{O}(K \log K)$

Topological sort on a word lattice $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ can be performed in time $\Theta(|\mathcal{V}|+|\mathcal{E}|)$. As will be shown in Section 8.4, the size of the upper envelope for $\mathcal{G}$ can never exceed the size of the arc set $\mathcal{E}$. The same holds for any subgraph $\mathcal{G}_{[s,v]}$ of $\mathcal{G}$ which is induced by the paths that connect the

source node s with $v \in \mathcal{V}$. Since the envelopes propagated from the source to the sink node can only increase linearly in the number of previously processed arcs, the total running time amounts to a worst case complexity of $\mathcal{O}(|\mathcal{V}|\cdot|\mathcal{E}|\log|\mathcal{E}|)$.

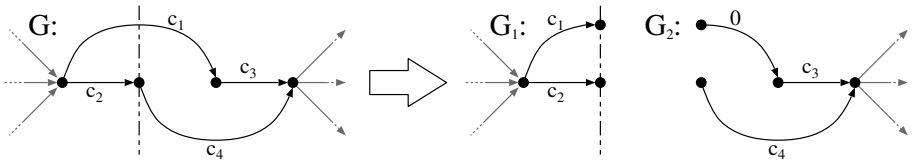
8.4 Upper Bound for Size of Envelopes

The memory efficiency of the suggested algorithm results from the following theorem which provides a novel upper bound for the number of cost minimizing paths in a directed acyclic graph with arc-specific affine cost functions. The bound is not only meaningful for proving the space efficiency of lattice MERT, but it also provides deeper insight into the structure and complexity of the unsmoothed error surface induced by log-linear models. Since we are examining a special class of shortest paths problems, we will invert the sign of each local feature function value in order to turn the feature scores into corresponding costs. Hence, the objective of finding the best sentence hypothesis in a word lattice becomes the problem of finding all cost-minimizing paths in a graph with affine cost functions.

Theorem: *Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a connected directed acyclic graph with vertex set $\mathcal{V}$, unique source and sink nodes $s, t \in \mathcal{V}$, and an arc set $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ in which each arc $e \in \mathcal{E}$ is associated with an affine cost function $c_e(\gamma) = a_e \cdot \gamma + b_e$, $a_e, b_e \in \mathbb{R}$. Counting ties only once, the cardinality of the union over the sets of all cost-minimizing paths for all $\gamma \in \mathbb{R}$ is then upper-bounded by $|\mathcal{E}|$:*

$$\Big| \bigcup_{\gamma \in \mathbb{R}} \big\{ \pi \,:\, \pi = \pi(\mathcal{G}; \gamma) \text{ is a cost-minimizing path in } \mathcal{G} \text{ given } \gamma \big\} \Big| \;\leqslant\; |\mathcal{E}| \tag{8.9}$$

Proof: The proposition holds for the empty graph as well as for the case that $\mathcal{V} = \{s, t\}$ with all arcs $e \in \mathcal{E}$ joining the source and sink node. Let $\mathcal{G}$ therefore be a larger graph. Then we perform an s-t cut and split $\mathcal{G}$ into two subgraphs $\mathcal{G}_1$ (left subgraph) and $\mathcal{G}_2$ (right subgraph). Arcs spanning the section boundary are duplicated (with the costs of the copied arcs in $\mathcal{G}_2$ being set to zero) and connected with a newly added head or tail node:



The zero-cost arcs in $\mathcal{G}_2$ that emerged from the duplication process are contracted, which can be done without loss of generality because zero-cost arcs do not affect the total costs of paths in the lattice. The contraction essentially amounts to a removal of arcs and is required in order to ensure that the sum of edges in both subgraphs does not exceed the number of edges in $\mathcal{G}$. All nodes in $\mathcal{G}_1$ with out-degree zero are then combined into a single sink node t_1. Similarly, nodes in $\mathcal{G}_2$ whose in-degree is zero are combined into a single source node s_2. Let N_1 and N_2 denote the number of arcs in $\mathcal{G}_1$ and $\mathcal{G}_2$, respectively. By construction, $N_1 + N_2 = |\mathcal{E}|$. Both subgraphs are smaller than $\mathcal{G}$ and thus, due to the induction hypothesis, their lower envelopes consist of at most N_1 and N_2 line segments, respectively. We further notice that either envelope is a convex hull whose constituent line segments inscribe a convex polygon, in the following denoted by $\mathcal{P}_1$ and $\mathcal{P}_2$. Now, we combine both subgraphs into a single graph $\mathcal{G}'$ by merging the sink node t_1 in $\mathcal{G}_1$ with the source node s_2 in $\mathcal{G}_2$. The merged node is an *articulation point* whose removal would disconnect both subgraphs, and hence, all paths in $\mathcal{G}'$ that start at the source node s and stop in the sink node t lead through this articulation point. The graph $\mathcal{G}'$ has at least as many cost minimizing paths as $\mathcal{G}$, although these paths as well as their associated costs might be different from those in $\mathcal{G}$. The additivity of the cost function and the articulation point allow us to split the costs for any path from s to t into

two portions: the first portion can be attributed to $\mathcal{G}_1$ and must be a line inside $\mathcal{P}_1$; the remainder can be attributed to $\mathcal{G}_2$ and must therefore be a line inside $\mathcal{P}_2$. Hence, the total costs for any path in $\mathcal{G}'$ can be bounded by the convex hull of the superposition of $\mathcal{P}_1$ and $\mathcal{P}_2$. This convex hull is again a convex polygon which consists of at most $N_1 + N_2$ edges, and therefore, the number of cost minimizing paths in $\mathcal{G}'$ (and thus also in $\mathcal{G}$) is upper bounded by $N_1 + N_2$. □

Corollary: *The upper envelope for a word lattice $\mathcal{G}_r = (\mathcal{V}_r, \mathcal{E}_r)$ consists of at most $|\mathcal{E}_r|$ line segments.* This bound can even be refined and one obtains $|\mathcal{E}|-|\mathcal{V}|+2$. Both bounds are tight.

This result may seem somewhat surprising as it states that, independent of the choice of the direction along which the line optimization is performed, the structure of the error surface is far less complex than one might expect based on the huge number of alternative sentence hypotheses that are represented in the lattice and thus contribute to the error surface. In fact, this result is a consequence of using a log-linear model which constrains how costs (or scores, respectively) can evolve due to hypothesis expansion. If instead quadratic cost functions were used, the size of the envelopes could not necessarily be limited in the same way. The above theorem does not, however, provide any additional guidance that would help to choose more promising directions in the line optimization algorithm to find better local optima.

A study which is complementary to the upper bound on the size of envelopes derived in Section 8.4 is provided in [Elizalde & Woods 06] which shows that the number of inference functions of any graphical model as, for instance, Bayesian networks and Markov random fields is polynomial in the size of the model if the number of parameters is fixed.

8.5 Summary

This chapter presented a novel algorithm that allows for efficiently constructing and representing the unsmoothed error surface over all sentence hypotheses that are represented in a word lattice. The proposed algorithm can be used to train the feature function weights of a log-linear model for automatic speech recognition under the *Minimum Error Rate Training* (MERT) criterion.

Chapter 9

Discriminative Linear Transformations

Linear transformations are extensively used in automatic speech recognition as a means for feature normalization and speaker adaptation. Commonly used feature transformations include the *Linear Discriminant Analysis* (LDA) [Haeb-Umbach & Ney 92b] and the *Maximum Likelihood Linear Transform* (MLLT) [Gopinath 98].

The LDA is a common data driven method that seeks a linear transformation which maps a high dimensional input vector into a lower dimensional subspace while retaining a maximum amount of class discrimination information. The statistics needed for computing the LDA are collected from the class density functions and their class prior probabilities. The LDA is based on the assumption that each class density function can be adequately modeled with a single multivariate Gaussian. The Gaussians are assumed to be isotropic, that is, they are supposed to share a common, class-independent covariance matrix. Although the actual goal is to improve the recognition rate of words or phonemes, classes for LDA in automatic speech recognition are usually identified with the Markov states or the indices of the mixture components. The reason for this is due to the inherent model assumption that the observations are supposed to be generated by a stochastic process based on which the elements scatter normally distributed around their class centroids. This assumption is more valid (and better suited to describe the data) if classes are identified with Markov states rather than with word or phoneme models for which the scatter pattern is a trajectory defined by the non-linear time alignment.

Unlike LDA, the MLLT is a feature transformation which aims at reducing the likelihood loss if *class-specific* covariance matrices are to be estimated under a diagonal modeling constraint. The MLLT tries to decorrelate the feature space by transforming the training samples such that the model assumption of class specific diagonal covariances is more valid in the transformed space than in the original feature space. If only two covariance matrices are involved, this problem has an exact solution: both matrices can be diagonalized simultaneously through a rotation matrix that is composed from the eigenvectors of the corresponding generalized eigenvalue problem. However, if more than two covariance matrices need to be decorrelated, there is in general no linear solution to this problem, and one has to search for an approximation that alleviates the mismatch between full and diagonal covariance matrices as best as possible. A solution to this problem is provided by the MLLT which is a special case of Kumar's *Heteroscedastic Linear Discriminant Analysis* (HLDA) [Kumar & Andreou 98b]. The HLDA extends the LDA to multi-class problems with non-uniform class-specific variances. If no axes in the original feature space are discarded, the HLDA is equivalent with the MLLT.

Motivated by its successful application to automatic speech recognition, several modifications have been proposed for the MLLT resulting in new transformations as, for example, *Semi-tied Covariances* [Gales 97], the *Extended Maximum Likelihood Linear Transform* (EMLLT) [Huang & Goel$^+$ 02], and the SPAM model [Goel & Axelrod$^+$ 03]. The latter models the inverse of the covariance matrix (the so-called precision matrix) from a set of basis matrices that span the space of precision matrices. Depending on how many basis matrices are combined, the full covariance matrix can be modeled arbitrarily exact.

This chapter presents a novel framework which allows for estimating linear feature transforma-

tions in a discriminative manner. Thereby, the MLLT as proposed in [Gopinath 98] is used as a case study. In Section 9.1, the MLLT is introduced and discussed from a likelihood perspective. Section 9.2 presents a robust optimization procedure for estimating the MLLT. Section 9.3 shows how the MLLT can be embedded into the discriminative framework and suggests an auxiliary function that is suitable for optimizing the transformation. Section 9.4 presents experimental results conducted on the HUB-4 corpus. A summary presented in Section 9.5 concludes this chapter.

Notation and Style

Throughout this chapter, the following notation for vectors and matrices are used. The convention is that vectors are always *column* vectors denoted by small letters, e.g., a, b, c, x. Row vectors are denoted by the transpose of a column vector: $a^\top$. Matrices are denoted by bold capital letters, e.g. $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, $\mathbf{\Sigma}$. Given a matrix $\mathbf{A}$, the element in row i and column j is denoted by a_{ij}. The dth column vector of a matrix $\mathbf{A}$ is written as $(\mathbf{A})_d$. To specify the dth row vector, the matrix $\mathbf{A}$ needs to be transposed in advance: $\left(\mathbf{A}^\top\right)_d$. If not stated otherwise, matrices are always composed by their respective column vectors. Therefore, matrices have to be transposed before they can be left-multiplied to a vector x, e.g., $\mathbf{A}^\top x$. Similarly, a matrix $\mathbf{A}$ which is multiplied from left to a matrix $\mathbf{\Sigma}$ has to be transposed in advance. Hence, the notation $\mathbf{A}^\top \Sigma \mathbf{A}$ is used in the following sections rather than $\mathbf{A}\mathbf{\Sigma}\mathbf{A}^\top$.

9.1 Maximum Likelihood Based Linear Feature Transformations

Let (x_n, k_n) $(n = 1, \ldots, N)$ denote a sequence of N training observations x_n with corresponding class labels k_n. The objective of the MLLT is to find a linear transformation $\widehat{\mathbf{A}}$ which maximizes the following objective function:

$$\widehat{\mathbf{A}} = \arg\max_{\mathbf{A}} |\mathbf{A}|^N \prod_{k=1}^{K} \left|\mathrm{diag}(\mathbf{A}^\top \mathbf{\Sigma}_k \mathbf{A})\right|^{-\frac{N_k}{2}}. \tag{9.1}$$

Here, N_k denotes the number of observations assigned with class k, and $\mathbf{\Sigma}_k$ is the full sample covariance matrix for class k. The logarithm of Eq. (9.1) defines the MLLT objective function:

$$\mathcal{F}_{\mathrm{MLLT}}(\mathbf{A}) = N \log|\mathbf{A}| - \sum_{k=1}^{K} \frac{N_k}{2} \log\left|\mathrm{diag}(\mathbf{A}^\top \mathbf{\Sigma}_k \mathbf{A})\right|. \tag{9.2}$$

To verify that $\widehat{\mathbf{A}}$ does indeed reduce the mismatch between full and diagonal variances, one can inspect *Hadamard's inequality* which provides an upper bound to the determinant of a non-singular matrix $\mathbf{C}$:

$$\left|\det(\mathbf{C})\right| \leqslant \sqrt{\prod_i \left(\sum_j c_{ij}^2\right)}. \tag{9.3}$$

Hence, for the special case of a regular matrix $\mathbf{\Sigma}$ (e.g., a covariance or a precision matrix), it becomes clear that the determinant of the full matrix is always a lower bound to the determinant of the same matrix with its off-diagonals set to zero:

$$\det(\mathbf{\Sigma}) \leqslant \det\left(\mathrm{diag}(\mathbf{\Sigma})\right) = \prod_d \sigma_{dd}^2. \tag{9.4}$$

Thus, the second term in Eq. (9.2) will be a minimum if $\mathbf{A}$ diagonalizes $\mathbf{\Sigma}_k$ for all classes k. Notice that $\mathbf{A}$ will in general not be able to fully diagonalize all covariance matrices simultaneously, and

a certain mismatch may therefore always remain. However, if the residuals in the optimization problem are weighted with their empirical class priors, the expectation of the remaining loss will be minimal.

Derivating $\mathcal{F}_{\text{MLLT}}$ with respect to $\mathbf{A}$ yields:

$$\frac{\partial \mathcal{F}_{\text{MLLT}}}{\partial \mathbf{A}} = N \cdot \frac{\mathbf{A}^{-\top}}{|\mathbf{A}|} \cdot |\mathbf{A}| - \sum_{k=1}^{K} \frac{N_k}{2} \cdot \frac{\partial \log \left|\text{diag}(\mathbf{A}^\top \boldsymbol{\Sigma} \mathbf{A})\right|}{\partial \mathbf{A}}. \tag{9.5}$$

This can be transformed into the following expression (the detailed calculations are shown in Section B.3.2):

$$\frac{\partial \mathcal{F}_{\text{MLLT}}}{\partial \mathbf{A}} = N \cdot \mathbf{A}^{-\top} - \sum_{k=1}^{K} N_k \cdot \underbrace{\boldsymbol{\Sigma}_k \cdot \mathbf{A} \cdot \text{diag}(\mathbf{A}^\top \boldsymbol{\Sigma}_k \mathbf{A})^{-1}}_{(*)} \stackrel{!}{=} 0 \tag{9.6}$$

9.1.1 Application to Automatic Speech Recognition

In automatic speech recognition, the classes k are typically identified with the Markov states. Let $\mathcal{S} = \{1, \ldots, S\}$ denote a set of Markov states. Each state $s \in \mathcal{S}$ shall be associated with a Gaussian mixture distribution:

$$p(x|\theta_s) = \sum_{l=1}^{L(s)} c_{sl} \cdot \mathcal{N}(x \,|\, \mu_{sl}, \boldsymbol{\Sigma}_{sl}). \tag{9.7}$$

Here, $L(s)$ denotes the number of mixture components, c_{sl} refers to the density weights, and μ_{sl} and $\boldsymbol{\Sigma}_{sl}$ denote the mean and covariance matrix. As in previous chapters, $\theta_s = \big\{(c_{sl}, \mu_{sl}, \boldsymbol{\Sigma}_{sl}) \,|\, l = 1, \ldots, L(s)\big\}$ comprises the set of all distribution parameters of mixture components associated with state s. In the following, the density index l is omitted to simplify the notation. Moreover, we do not strictly distinguish between a Markov state s and its associated mixture distribution.

In general, incorporating a global feature transform $\mathbf{A}$ changes the normalization term of a Gaussian distribution as follows:

$$\mathcal{N}(\mathbf{A}^\top x \,|\, \mu_s, \boldsymbol{\Sigma}_s) = \frac{|\mathbf{A}|}{\sqrt{\det(2\pi \boldsymbol{\Sigma}_s)}} \exp\left\{ -\frac{1}{2} \big(\mathbf{A}^\top x - \mu_s\big)^\top \cdot \boldsymbol{\Sigma}_s^{-1} \cdot \big(\mathbf{A}^\top x - \mu_s\big) \right\} \tag{9.8}$$

Now let (X_r, W_r) $(r = 1, \ldots, R)$ denote a sequence of R training utterances where each pair consists of a sequence of acoustic observations $X_r = x_{r1}, \ldots, x_{rT_r}$ together with a transcription of the spoken word sequence $W_r = w_{r1}, \ldots, w_{rN_r}$. Furthermore, let $\gamma_{rt}(s|W_r) = p_\theta(s|X_r, W_r)$ denote the state occupancy probability as defined in Eq. (2.22). Then the MLLT objective function is given by:

$$\mathcal{F}_{\text{MLLT}}(\mathbf{A}) = \sum_{r=1}^{R} \sum_{t=1}^{T_r} \sum_{s=1}^{S} \gamma_{rt}(s|W_r) \cdot \left[\log|\mathbf{A}| - \frac{1}{2} \cdot \log \left|\text{diag}(\mathbf{A}^\top \boldsymbol{\Sigma}_s \mathbf{A})\right| \right]. \tag{9.9}$$

9.1.2 Related Transformations and their Auxiliary Functions

The MLLT is a special case of *Kumar's* HLDA which, in contrast to the well known LDA transformation, does not make the assumption that all class-specific variances have to be isotropic. Let $\boldsymbol{\Sigma}_0$ denote the total scatter matrix, which is equal to the sum over the (class-independent) within class scatter matrix $\mathbf{W}$ and the (class-independent) between class scatter matrix $\mathbf{B}$:

$$\boldsymbol{\Sigma}_0 = \mathbf{W} + \mathbf{B}. \tag{9.10}$$

with

$$\mathbf{W} = \frac{1}{T} \sum_{r=1}^{R} \sum_{t=1}^{T_r} \sum_{s=1}^{S} \gamma_{rt}(s|W) \cdot (x_{rt} - \mu_s) \cdot (x_{rt} - \mu_s)^\top \tag{9.11}$$

$$\mathbf{B} = \frac{1}{T} \sum_{r=1}^{R} \sum_{t=1}^{T_r} \sum_{s=1}^{S} \gamma_{rt}(s|W) \cdot (\mu - \mu_s) \cdot (\mu - \mu_s)^\top) \tag{9.12}$$

Then the HLDA auxiliary function is given by:

$$\mathcal{Q}_{\text{HLDA}}(\mathbf{A}, \theta) = \frac{1}{2} \sum_{s=1}^{S} \sum_{l=1}^{L_s} \sum_{r=1}^{R} \sum_{t=1}^{T_r} \gamma_{rt}(s, l|W_r) \cdot \log \left(\frac{|\mathbf{A}|^2}{\left|\text{diag}(\mathbf{A}_{[p]}^\top \cdot \mathbf{\Sigma}_{sl} \cdot \mathbf{A}_{[p]})\right| \cdot \left|\text{diag}(\mathbf{A}_{[n-p]}^\top \cdot \mathbf{\Sigma}_0 \cdot \mathbf{A}_{[n-p]})\right|} \right) \tag{9.13}$$

Here, $\mathbf{A}_{[p]}$ denotes the first p row vectors of $\mathbf{A}$, while $\mathbf{A}_{[n-p]}$ denotes its last $(n-p)$ row vectors. Similarly, the objective functions of the LDA and the MLLT can be written as:

$$\mathcal{Q}_{\text{LDA}}(\mathbf{A}, \theta) = \frac{1}{2} \sum_{s=1}^{S} \sum_{l=1}^{L_s} \sum_{r=1}^{R} \sum_{t=1}^{T_r} \gamma_{rt}(s, l|W_r) \cdot \log \left(\frac{|\mathbf{A}|^2}{\left|\text{diag}(\mathbf{A}_{[p]}^\top \cdot \mathbf{\Sigma} \cdot \mathbf{A}_{[p]})\right| \cdot \left|\text{diag}(\mathbf{A}_{[n-p]}^\top \cdot \mathbf{\Sigma}_0 \cdot \mathbf{A}_{[n-p]})\right|} \right) \tag{9.14}$$

$$\mathcal{Q}_{\text{MLLT}}(\mathbf{A}, \theta) = \frac{1}{2} \sum_{s=1}^{S} \sum_{l=1}^{L_s} \sum_{r=1}^{R} \sum_{t=1}^{T_r} \gamma_{rt}(s, l|W_r) \cdot \log \left(\frac{|\mathbf{A}|^2}{\left|\text{diag}(\mathbf{A}_{[n]}^\top \cdot \mathbf{\Sigma}_{sl} \cdot \mathbf{A}_{[n]})\right|} \right) \tag{9.15}$$

Figure 9.1 illustrates the different decisions made by the HLDA and the LDA transformation.

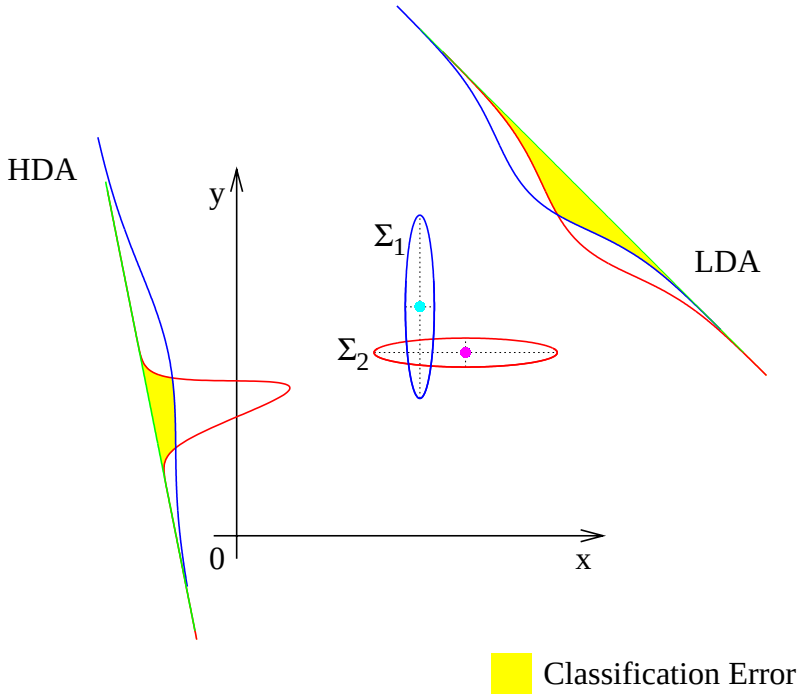


Figure 9.1: Different decisions made by the HLDA and the LDA transformation.

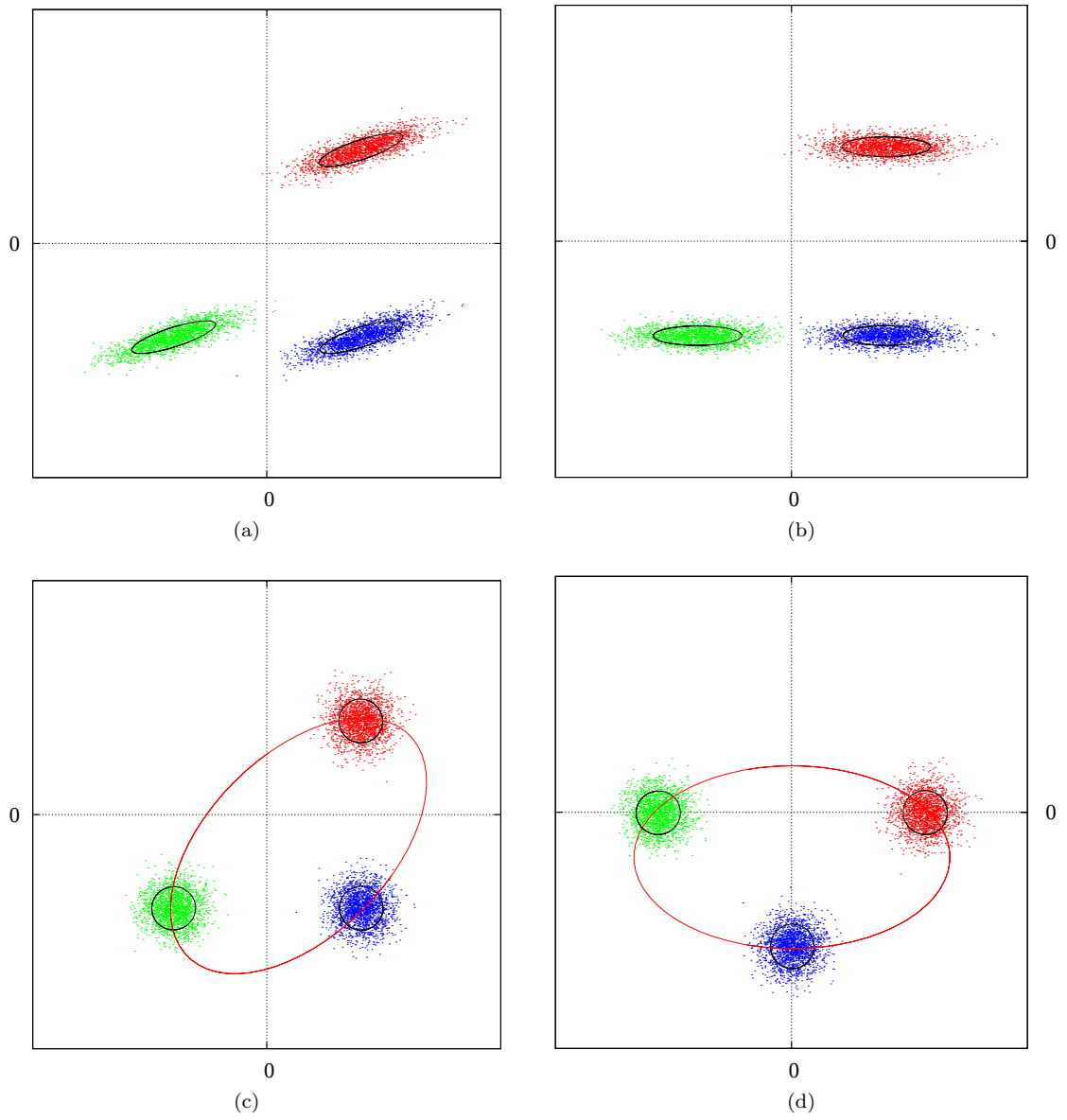


Figure 9.2: Effect of the LDA transformation on four uniformly distributed classes. All classes are Gaussian distributions with isotropic covariance matrices (a). The LDA can be decomposed into a sequence of transformations $\mathbf{A} = \mathbf{R}_2\mathbf{\Lambda}^{-\frac{1}{2}}\mathbf{R}_1$ which first rotates the within class scatter matrices using transformation $\mathbf{R}_1$ such that their principal components are parallel to the coordinate axes (b). Scaling the coordinate axes with the inverse of the root of the corresponding eigenvalues, $\mathbf{\Lambda}^{-\frac{1}{2}}$, whitens the hyper ellipsoids (c). Finally, the between class scatter matrix is diagonalized using $\mathbf{R}_2$, resulting in a configuration for which projecting the sample points onto the abscissa increases class separability (d).

9.2 A Robust Optimization Procedure

A robust optimization procedure for estimating the MLLT is proposed in [Gales 98]. The optimization scheme makes use of the adjoint (or adjugate) of $\mathbf{A}$ which allows for decomposing the initially non-linear optimization problem into an easy to manage row-by-row optimization problem. Starting point is the auxiliary function from Eq. (9.15) and its derivative with respect to $\mathbf{A}$:

$$\frac{\partial \mathcal{Q}_{\text{MLLT}}(\mathbf{A},\theta)}{\partial \mathbf{A}} = \sum_{s=1}^{S}\sum_{l=1}^{L_s}\sum_{r=1}^{R}\sum_{t=1}^{T_r}\gamma_{rt}(s,l|W_r)\cdot\left(\mathbf{A}^{-\top} - \mathbf{\Sigma}_{sl}\cdot\mathbf{A}\cdot\text{diag}(\mathbf{A}^{\top}\mathbf{\Sigma}_{sl}\mathbf{A})^{-1}\right) \tag{9.16}$$

To simplify the problem of finding the matrix $\widehat{\mathbf{A}}$ which maximizes Eq. (9.15), we define a set of *sufficient statistics* given by the following quantities:

$$\text{total count:} \qquad T = \sum_{s=1}^{S}\sum_{l=1}^{L_s}\sum_{r=1}^{R}\sum_{t=1}^{T_r}\gamma_{rt}(s,l|W_r) \tag{9.17}$$

$$d\text{th component:} \qquad \sigma^2_{sld} = \text{diag}(\mathbf{A}^{\top}\mathbf{\Sigma}_{sl}\mathbf{A})_d \tag{9.18}$$

$$\text{variance statistics:} \qquad \mathbf{G}^{(d)} = \sum_{s=1}^{S}\sum_{l=1}^{L_s}\sum_{r=1}^{R}\sum_{t=1}^{T_r}\frac{\gamma_{rt}(s,l|W_r)}{\sigma^2_{sld}}\cdot\mathbf{\Sigma}_{sl} \tag{9.19}$$

Since a direct maximization of Eq. (9.16) is numerically difficult to handle, the matrix $\mathbf{A}^{-\top}$ is re-written by its cofactors. The cofactors have the following important property:

$$(\mathbf{A}^{-\top})_d = \frac{c_d}{a_d^{\top}\cdot c_d}, \qquad c_d^{\top} = \big[\text{cof}(a_{1d}),\ldots,\text{cof}(a_{Dd})\big], \qquad \text{cof}(a_{de}) = (-1)^{d+e}|\mathbf{A}_{de}| \tag{9.20}$$

Thus, Eq. (9.16) can be re-written as follows:

$$\left(\frac{\partial \mathcal{Q}_{\text{MLLT}}(\mathbf{A},\theta)}{\partial \mathbf{A}}\right)_d = T\cdot\frac{c_d}{a_d^{\top}\cdot c_d} - \mathbf{G}^{(d)}\cdot a_d \tag{9.21}$$

Setting Eq. (9.21) to 0 and solving the resulting equations leads to the following update equations for the column vectors a_d of the MLLT transformation matrix:

$$a_d = \sqrt{\frac{T}{c_d^{\top}\cdot\mathbf{G}^{(d)^{-1}}\cdot c_d}}\cdot\mathbf{G}^{(d)^{-1}}\cdot c_d \tag{9.22}$$

The detailed calculations can be found in Section B.3.3. Algorithm 7 shows the major steps involved for computing the MLLT. Figure 9.3 illustrates the effect of the MLLT optimization procedure for the example of four, normally distributed classes. The smaller ellipses in Figure 9.3a show the contours of the corresponding covariance matrices for constant probability while the larger ellipses show the contours if the covariance matrices are estimated under a diagonal modeling constraint. Figure 9.3b shows the same ellipses in the MLLT-transformed feature space. Here, the mismatch between full and diagonal covariance matrices is substantially reduced which leads to a more reliable estimation of the variance parameters and thus may help to improve class separability.

In principal, the MLLT optimization procedure can also be applied to maximize the LDA objective function, although there are more efficient optimization methods available which take advantage of the fact that the LDA is based on the assumption that all classes are uniformly distributed. The solution of the LDA objective function is a rotation matrix which simultaneously diagonalizes both the within class scatter matrix and the between class scatter matrix. Figure 9.2 (a-d) illustrates the effect of the LDA transformation on uniformly distributed classes (a) by first whitening the within class scatter matrices (b-c) followed by a diagonalization of the between class scatter matrix (d).

Algorithm 7 Maximum Likelihood Linear Transform

Compute Sufficient Statistics:

total count: $$T = \sum_{s=1}^{S} \sum_{l=1}^{L_s} \sum_{r=1}^{R} \sum_{t=1}^{T_r} \gamma_{rt}(s, l|W_r)$$

full covariance matrices: $$\mathbf{\Sigma}_{sl} = \sum_{r=1}^{R} \sum_{t=1}^{T_r} \gamma_{rt}(s, l|W_r) \cdot \frac{(x_{rt} - \mu_{sl}) \cdot (x_{rt} - \mu_{sl})^\top}{\Gamma_{sl}^{\mathsf{num}}(1)}$$

Optimization Procedure:

$\mathbf{A} \coloneqq \mathbf{I}$
repeat
  $r \coloneqq 0$
  $\mathbf{C} \coloneqq \mathbf{A}^{-\top}$
  for $(d = 1, ..., D)$
    $\mathbf{G}^{(d)} \coloneqq \mathbf{0}$
    $a_d \coloneqq (\mathbf{A})_d$
    $c_d \coloneqq (\mathbf{C})_d$
    for $(s = 1, ..., S)$
      for $(l = 1, ..., L_s)$
        $\sigma_{sld}^2 \coloneqq a_d^\top \cdot \mathbf{\Sigma}_{sl} \cdot a_d$
        $\mathbf{G}^{(d)} \mathbin{+}= \frac{\gamma_{rt}(s, lW_r)}{\sigma_{sld}^2} \cdot \mathbf{\Sigma}_{sl}$
    $a_d \coloneqq \sqrt{\frac{T}{c_d^\top \cdot \mathbf{G}^{(d)^{-1}} \cdot c_d}} \cdot \mathbf{G}^{(d)^{-1}} \cdot c_d \; - \; (\mathbf{A})_d$
    $(\mathbf{A})_d \mathbin{+}= a_d$
    $r \mathbin{+}= ||a_d||_2$
until $(r < \tau)$

9.3 Discriminative Estimation of Linear Transformations

The MLLT objective function aims at minimizing the likelihood loss if the covariance matrices are to be estimated under a diagonal modeling constraint. The objective function does not, however, take class separability into account, and therefore it might not exploit the full potential of this transformation. To compensate for that, the MLLT is embedded into a discriminative framework in this section. The resulting transformation is called *Discriminative Linear Transform* (DLT). The objective of the DLT is to minimize the likelihood loss preferably for those classes where retaining as much information of the full covariance matrix as possible is beneficial to discriminate between different classes. As a consequence, a larger loss in terms of likelihood is only allowed for those classes which are already sufficiently well separable and therefore prone to retain their ability to discriminate between different classes even if the diagonal modeling constraint results in a likelihood loss.

Incorporating a class separability criterion into the MLLT objective function can be accomplished by subtracting the generalized FB probabilities from the state occupancy probabilities in Eq. (9.15) and Eq. (9.16), respectively. This leads to the following objective function for the DLT and its gradient with respect to the transformation matrix $\mathbf{A}$:

$$\mathcal{F}_{\mathrm{DLT}}(\mathbf{A}, \theta) = \frac{1}{2} \sum_{s=1}^{S} \sum_{l=1}^{L_s} \sum_{r=1}^{R} \sum_{t=1}^{T_r} \Big[\gamma_{rt}(s, l|W_r) - \gamma_{rt}(s, l) \Big] \cdot \log \left(\frac{|\mathbf{A}|^2}{\left|\mathrm{diag}(\mathbf{A}_{[n]}^\top \cdot \mathbf{\Sigma}_{sl} \cdot \mathbf{A}_{[n]})\right|} \right) \quad (9.23)$$

$$\frac{\partial \mathcal{F}_{\mathrm{DLT}}(\mathbf{A},\theta)}{\partial \mathbf{A}} = \sum_{s=1}^{S}\sum_{l=1}^{L_s}\sum_{r=1}^{R}\sum_{t=1}^{T_r}\Big[\gamma_{rt}(s,l|W_r) - \gamma_{rt}(s,l)\Big]\cdot\Big(\mathbf{A}^{-\top} - \boldsymbol{\Sigma}_{sl}\cdot\mathbf{A}\cdot\mathrm{diag}(\mathbf{A}^{\top}\boldsymbol{\Sigma}_{sl}\mathbf{A})^{-1}\Big) \tag{9.24}$$

Note that Eq. (9.23) will be in a local optimum if $\gamma_{rt}(s,l) = \gamma_{rt}(s,l|W_r)$ holds for all (s,l), in which case Eq. (9.24) will become zero. However, this does not necessarily imply that $\widehat{\mathbf{A}}$ would be able to fully diagonalize all covariance matrices involved.

Since a direct optimization of Eq. (9.23) is not feasible, we use an auxiliary function that allows for optimizing Eq. (9.23) iteratively. According to Section 4.5, this auxiliary function can be constructed as follows:

$$\mathcal{G}_{\mathrm{DLT}}(\mathbf{A},\theta) = \mathcal{F}_{\mathrm{DLT}}(\mathbf{A},\theta) \;+\; \mathcal{G}_0\big((\mathbf{A},\theta),\,(\widehat{\mathbf{A}},\widehat{\theta})\big). \tag{9.25}$$

Here, $\mathcal{G}_0$ is a smoothing function which needs to be chosen such that the differential of $\mathcal{G}_0\big((\mathbf{A},\theta),\,(\mathbf{A},\theta)\big)$ at $(\widehat{\mathbf{A}},\widehat{\theta}) = (\mathbf{A},\theta)$ is equal to $\mathbf{0}$. The following choice of the smoothing function meets this requirement:

$$\mathcal{G}_0\big((\mathbf{A},\theta),\,(\widehat{\mathbf{A}},\widehat{\theta})\big) = \int p(x|\theta_s^{(i)})\cdot\nabla_{\mathbf{A}}\log p(x|\theta_s)\big|_{\theta_s=\theta_s^{(i)}}\,dx \tag{9.26}$$

The maximization of Eq. (9.26) can be performed using an extension of the optimization algorithm described in Section 9.2. Hence, re-writing Eq. (9.24) with its cofactors and re-arranging terms yields:

$$\underbrace{\sum_{s=1}^{S}\sum_{l=1}^{L_s}\Big(\Gamma_{sl}(1) + D_{sl}\Big)}_{=:T}\cdot\frac{c_d}{\bar{a}_d^{\top}\cdot c_d} = \underbrace{\sum_{s=1}^{S}\sum_{l=1}^{L_s}\Big[\mathbf{G}_{sl}^{(d)} \;-\; \mathbf{H}_{sl}^{(d)}\Big]}_{=:\,\mathbf{J}^{(d)}}\cdot\bar{a}_d \tag{9.27}$$

$$\Longleftrightarrow \qquad \bar{a}_d = \sqrt{\frac{T}{c_d^{\top}\cdot{\mathbf{J}^{(d)}}^{-1}\cdot c_d}}\cdot{\mathbf{J}^{(d)}}^{-1}\cdot c_d \tag{9.28}$$

with

$$T = \sum_{s=1}^{S}\sum_{l=1}^{L(s)}\Big[\Gamma_{sl}(1) \;+\; D_{sl}\Big] \tag{9.29a}$$

$$\mathbf{J}^{(d)} = \sum_{s=1}^{S}\sum_{l=1}^{L(s)}\Big[\mathbf{G}_{sl}^{(d)} \;-\; \mathbf{H}_{sl}^{(d)}\Big] \tag{9.29b}$$

$$\mathbf{G}_{sl}^{(d)} = \frac{1}{\sigma_{sld}^2}\cdot\Gamma_{sl}(1)\cdot\boldsymbol{\Sigma}_{sl} \tag{9.29c}$$

$$\mathbf{H}_{sl}^{(d)} = \frac{D_{sl}}{\sigma_{sld}^2}\cdot\Big[\boldsymbol{\Sigma}_{sl} + \boldsymbol{\Xi}_{sl} - \frac{1}{\sigma_{sld}^2}\cdot\boldsymbol{\Sigma}_{sl}\cdot\bar{\mathbf{A}}\cdot\bar{\mathbf{A}}^{\top}\cdot\boldsymbol{\Xi}_{sl}\Big] \tag{9.29d}$$

$$\boldsymbol{\Xi}_{sl} = \mathbf{A}^{-\top}\cdot\mathrm{diag}(\mathbf{A}^{\top}\boldsymbol{\Sigma}_{sl}\mathbf{A})\cdot\mathbf{A}^{-1} \qquad \text{(old transformation)} \tag{9.29e}$$

$$\sigma_{sld}^2 = \mathrm{diag}(\bar{\mathbf{A}}\boldsymbol{\Sigma}_{sl}\bar{\mathbf{A}})_d \tag{9.29f}$$

Figure 9.3 illustrates the different decisions made by the MLLT and the DLT for a four-class problem with non-uniformly distributed classes. Figure 9.3-a shows the four classes in the original (untransformed) feature space. Both the *red* and the *green* class have a large overlap if the variance parameters are estimated under a diagonal modeling constraint. However, since the number of data

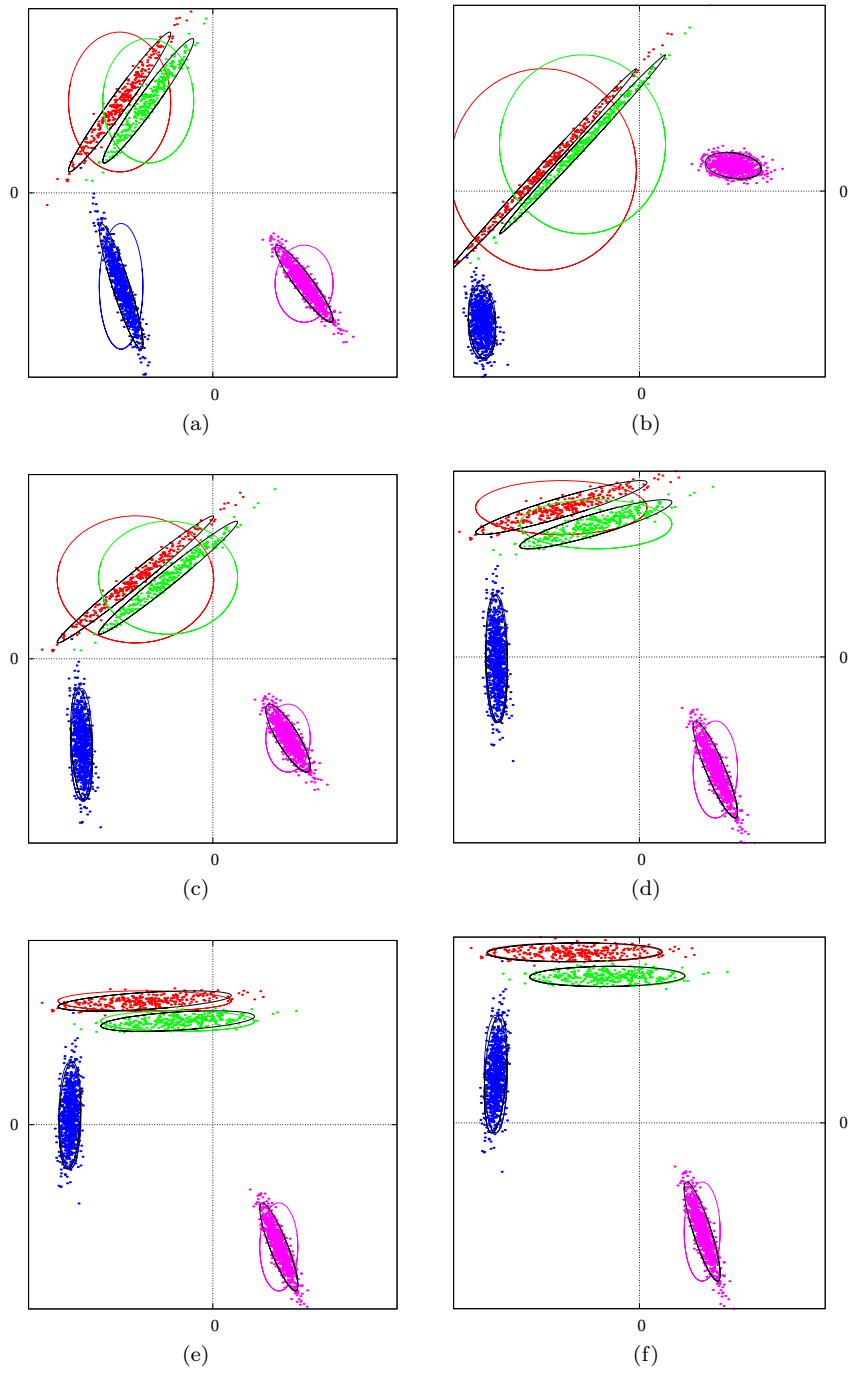


Figure 9.3: Illustration of the different decisions made by the MLLT and the DLT for a four-class problem with non-uniform distributions. Figure (a) shows each class in the original (untransformed) feature space. Figure (b) shows the decision made by the MLLT and Figures (c-f) show the decision made by the DLT after one (c), two (d), three (e), and four (f) iterations, respectively.

points assigned to either class is much smaller than the number of data points that belong to the *blue* and the *purple* class, the MLLT concentrates on reducing the mismatch between full and diagonal covariance matrices for the *blue* and the *purple* class at the expense of the *green* and the *red* class as shown in Figure 9.3-b. The DLT instead increases the weight of those classes for which the numerator statistics $\Gamma^{\mathsf{num}}(1)$ differ significantly from the denominator statistics $\Gamma^{\mathsf{den}}(1)$, and thus, the DLT aims at enhancing class separability. Figures 9.3-c and 9.3-f show the effect of the DLT transformation after 1, 2, 3, and 4 iterations.

9.4 Experiments

Experiments were conducted on the male speaker portion of the HUB-4 corpus. In order to estimate the MLLT transformation matrix, a set of single Gaussian models with full covariances matrices was estimated using the alignment information (time alignment path) obtained from the best ML-trained models. The MLLT matrix was then trained according to Algorithm 7 by executing 1000 iterations. The DLT was trained in a similar way except that it includes the time alignment information obtained from the denominator model. Figure 9.4 shows the evolution of the MLLT and the DLT objective function on the male speaker portion of the HUB-4 corpus in the course of the iteration index. The y-axis shows the residuals, that is, the sum over the Euclidean norms of the changes in the column vectors of the transformation matrix. After each iteration, the resulting transformation matrix is normalized such that its determinant is equal to 1.0 which is accomplished through a row-wise normalization. Table 9.1 shows the performance gain obtained with the MLLT and the DLT transformation matrix on top of the best ML-trained baseline system. Even though the DLT is able to improve on top of the MLLT, the performance gains are rather small. One reason for this is that mixture densities are able to compensate for some of the adverse effects caused by imposing the diagonal modeling constraint onto the covariance matrices. Under the assumption that the training samples scatter normally distributed around their class centers, a larger number of mixture components seems to be able to approximate the full covariance case sufficiently well by padding the corresponding hyper ellipsoid of the full covariance matrix with a set of smaller hyper ellipsoids induced by the Gaussians with the diagonal modeling constraint.

Table 9.1: Error rates obtained with the *Maximum Likelihood Linear Transformation* (MLLT) and the *Discriminative Linear Transform* (DLT) on top of a *Linear Discriminant Analysis* (LDA) estimated on the male speaker portion of the HUB-4 corpus.

	HUB-4, male speaker portion mix.-spec. variances		
transformation	PER[%]	WER[%]	SER[%]
LDA	15.90	23.43	93.18
LDA+MLLT	15.51	22.85	92.68
LDA+DLT	15.50	22.75	92.42

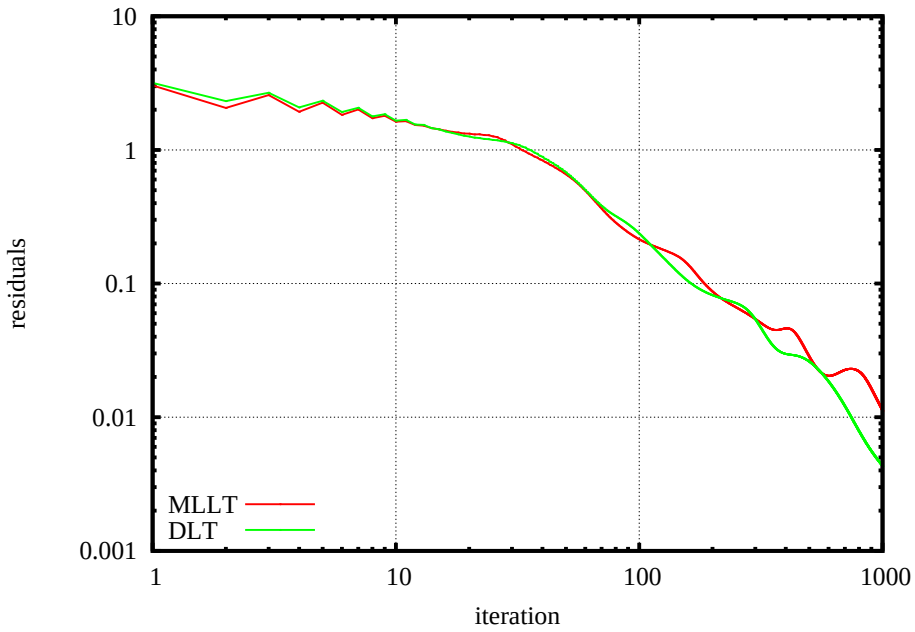


Figure 9.4: Evolution of the MLLT and the DLT objective function on the male speaker portion of the HUB-4 corpus in the course of the iteration index. After each iteration, the respective transformation matrices are normalized such that their determinants are equal to one.

9.5 Conclusions

This chapter presented a discriminative extension of the *Maximum Likelihood Linear Transformation* (MLLT) called *Discriminative Linear Transform* (DLT). The DLT aims at finding a feature space in which a diagonal modeling constraint for covariance matrices is more valid than in the original feature space. In contrast to the MLLT, the DLT objective function also focuses on class separability by taking both the numerator and the denominator statistics into account. Experiments conducted on the male speaker portion of the HUB-4 corpus showed a small improvement in terms of word error rate for DLT over the MLLT.

Chapter 10

Scientific Contributions

The work presented in this thesis addressed several aspects of discriminative training for automatic speech recognition. In Section 2.6.1, the discriminative approach was motivated through a simple two-class problem which illustrated how the discriminative approach can lead to improved recognition performance compared to the Maximum Likelihood estimator if the family of model distributions does not contain the true distribution. In Section 4.1.1, an *extended unifying approach* for a class of discriminative training criteria was presented that, besides the *Maximum Mutual Information* criterion and the *Minimum Classification Error* criterion, also contains criteria more recently proposed, including the *Minimum Word Error* criterion and the closely related *Minimum Phone Error* criterion. The extended unifying approach provides a single framework in which several discriminative training criteria can be analyzed analytically and experimentally. Monte Carlo experiments conducted for some of the criteria yielded upper bounds to the Bayes classification error rate that turned out to be tighter than the bounds that can be found in literature. The parameter optimization problem was discussed for the case that Gaussian mixture models with *full* covariance matrices and arbitrary tying schemes are used. It was shown that the problem of choosing the iteration constants leads for this setting to the solution of a *quadratic eigenvalue problem* (QEP). The QEP decomposes into a system of quadratic inequalities if a diagonal modeling constraint is imposed onto the covariance matrices. For the problem of controlling the convergence speed, two novel methods on setting the iteration constants in discriminative training were developed. In particular, the suggested Λ method turned out to be the most robust method among the investigated optimization schemes, leading to an enhanced recognition performance that is independent of whether a system employs a single, globally pooled covariance matrix or if it uses density specific variance models. A set of parameterized re-estimation equations was derived that provides update equations for the parameters of a Gaussian mixture model for all criteria contained in the extended unifying approach. Some general rules for building auxiliary functions to optimize the distribution parameters in a discriminative framework were derived in Section 4.5.

In Chapter 5, details of the lattice-based discriminative training were discussed. Section 5.1 introduced the concept of *Hidden Markov Model* (HMM) *networks* which were used in order to generate lattices for the correct and the competing model. HMM networks can be considered as a hybrid approach between lexical prefix trees and finite state machines. The global search based on word-dependent tree copies was extended such that it employs the more general concept of word-dependent graph copies. A set of graph transformations was developed which allow for using the language model look-ahead pruning in combination with HMM networks. Two forward-backward algorithms for estimating posterior probabilities and word or phoneme accuracies were presented. In addition, an improved forward-backward pruning method was suggested that, compared to [Sixtus & Ortmanns 99], has the advantage that it does not require a pruning threshold; instead, it suffices to specify the desired graph density. In Section 5.4, the first successful implementation of a large-scale *Minimum Classification Error* training was presented. New methods for accelerating the process of generating lattices, estimating sufficient statistics, and updating the model parameters were developed and implemented. Among these methods is a refined technique to dynamically adjust pruning thresholds, a constrained recognition scheme, and a client-server model for

distributed computing on machine clusters with inhomogeneous architectures. The complete code base implemented for the discriminative system in this thesis comprises more than 70 000 lines of ANSI conform C++ code. An extensive use of the *standard template library* (STL) as well as the strict omission of third-party software guarantee the portability of the developed software package to many different platforms and architectures.

Experimental results conducted on several speech corpora with different levels of difficulty, ranging from elementary digit recognition (SieTill) over the recognition of read speech (Wsj-5k, Nab-20k, Nab-65k) up to the automatic transcription of broadcasts (Hub-4), were reported in Chapter 6. Questions pursued addressed how the performance gains achieved by a discriminative training are affected by increasing the model size. In addition, the effectiveness of the training procedure was analyzed for the case that Gaussian mixture models with different variance tying schemes are used. Compared to a system that employs a single, globally pooled variance vector, the use of density specific variances could be shown to consistently outperform the pooled variance system in terms of both training time and system performance. It was experimentally shown that re-estimating the variance parameters does not seem to be beneficial in discriminative training. While this result was to be expected for the case that a single, globally pooled covariance matrix is used, it is not obvious that fixing the density-specific variance parameters also prevails over the unconstrained re-estimation procedure. A dominant aspect of the investigation presented in Chapter 6 was the comparison of various discriminative training criteria contained in the extended unifying approach. The comparison was carried out on several speech corpora for different systems, one of which using a globally pooled variance vector and the other one using density specific variances. Criteria investigated in the extended unifying approach comprise the *Maximum Mutual Information* (MMI) criterion; its maximum approximation, the *Corrective Training*; the *Minimum Classification Error* (MCE) criterion together with its maximum approximation, the *Falsifying Training*; the *Diversity Index* with its two special instances, the *Shannon* criterion and the *Gini* criterion; the *Jeffreys* criterion; the *Chernoff* criterion; and the *Minimum Word Error* (MWE) criterion together with the closely related *Minimum Phone Error* (MPE) criterion. While both the MWE and the MPE criterion achieved good results in combination with density-specific variances, they were not able to outperform the MMI or the Gini criterion. Substantial improvements for the pooled variance system could also be achieved with the Jeffreys criterion. Overall, the MCE criterion turned out to be the most robust and best performing criterion in terms of word error rate among all investigated criteria. Additional performance gains could be obtained by using unigram-decoded lattices in training. Unigram-decoded lattices extend the idea proposed in [Schlüter & Müller$^+$ 99] according to which lattices used in discriminative training should be re-scored with a low-order m-gram model.

In Chapter 7, a novel approach was presented which models the posterior probability directly as a log-linear model. The direct model was shown to follow the *Maximum Entropy* principle and was effectively trained using the *Generalized Iterative Scaling* (GIS) algorithm. Both the direct model and its optimization via the GIS algorithm were compared analytically and experimentally with the MMI criterion and the *Extended Baum* algorithm.

Chapter 8 extended the *Minimum Error Rate Training* (MERT) criterion for N-best lists as suggested in [Och 03] to word lattices. The suggested algorithm allows for efficiently constructing, representing, and exploring the exact and unsmoothed error surface for *all* sentence hypotheses that are encoded in a word lattice under virtually any automated error criterion used in natural language processing. A line minimization method was described that can be used to train the parameters of a linear model such that the error rate is directly optimized on training data. The size of any such error surface was shown to be linearly bounded in the number of lattice arcs. This bound holds for the case that all possible parameter settings are considered simultaneously that lie along a line in the parameter space for a log-linear model.

Chapter 9 presented a novel framework that allows for estimating the *Maximum Likelihood Linear Transform* (MLLT) in a discriminative manner, leading to the so-called *Discriminative*

Linear Transformation (DLT). Experiments conducted on the male-speaker portion of the HUB-4 corpus showed small improvements for the DLT over the MLLT using the best ML-trained system with density-specific variances.

10.1 Conclusions

This thesis investigated a large number of discriminative training criteria for automatic speech recognition. The investigation was performed within an extended unifying approach that allows for comparing different criteria consistently with respect to both the theoretical and the experimental analysis. The investigated criteria were applied to various speech recognition tasks with different level of difficulty. Within the scope of this investigation the effect of using a single, globally pooled variance vector versus density-specific variances was studied. The results clearly show that the use of density-specific variances leads to faster convergence rates and better recognition accuracies than the use of a single, globally pooled variance. The MLLT was shown to be an essential part in a multi variance system. Empirical evidence was given that the variance parameters in a discriminative training procedure do not need to be re-estimated; instead they can be fixed to their *Maximum Likelihood* estimates. This result is independent of whether the acoustic model employs a single, globally pooled variance vector or whether density specific variances are used. Though variance parameters still play an important role when choosing the step size in the *Extended Baum* algorithm, their discriminative estimation does not provide room to further enhance class separability.

This thesis also showed that the MCE criterion yields similar or even better performance gains than both the MPE and the MWE criterion. Moreover, the MCE criterion proved to be very robust on large vocabulary tasks and emerged as the recommended criterion in discriminative training.

The concept of HMM networks was introduced and shown to be very powerful, resulting in fast training procedures and small network sizes particularly if very large pronunciation dictionaries with high-order language models are used. Unlike finite state transducers, HMM networks are less memory consumptive since the composition between the acoustic and the language model network is not required. Instead, the language model can be implemented as an external distributed service which enables HMM networks to benefit from language models that are too large to fit into the memory of a single machine.

The log-linear approach suggested in Chapter 7 showed a large potential for automatic speech recognition. Hence, using a fairly simple set of cepstral feature functions, the log-linear model trained with the *Generalized Iterative Scaling* algorithm was able to outperform a system based on Gaussian HMMs that was trained under the *Maximum Mutual Information* criterion.

The novel bound derived in Chapter 8 on the complexity of error surfaces provided new insights into the structure of error minimizing training criteria for log-linear models. While the number of sentence hypotheses encoded in a word lattice is exponential in the number of lattice arcs, it turned out that error surfaces are always linearly bounded in the lattice size. This implies that the majority of sentence hypotheses "cannot be reached" by error minimizing training criteria.

10.2 Outlook

Based on the findings in this thesis, possible future directions that could be investigated may cover the following topics:

Smoothed Error Criteria for MPE and MWE

The use of the logistic regression in the MCE criterion is essential to yield performance gains over other criteria as, for example, the MMI criterion or the MPE and the MWE criterion. A similar

smoothing function can be applied to re-weight competing hypothesis in the MPE and the MWE criterion.

Effect of MPE and MWE on Minimum Bayes Risk Decoding

Both the MPE and the MWE criterion are risk based criteria that aim at maximizing the expectation of a predefined accuracy function. An interesting follow-up question is whether these criteria may lead to additional performance gains if instead of the 0-1 loss function the *Minimum Bayes Risk* decision rule is used.

Boosted MCE

Recently, improvements over MPE and MWE have been reported using a boosted version of the MMI criterion [Povey & Kanevsky$^+$ 08]. The MMI criterion is enhanced by first computing an approximated accuracy measure similar to the MPE or MWE criterion for hypotheses in the denominator lattice and then by giving those hypotheses a higher weight for which the accuracy is low. A similar extension can easily be developed for the MCE criterion. Closely related to [Povey & Kanevsky$^+$ 08] is the work by [Qiang & Juang 07]. A much clearer reference to the underlying margin concept is given in [Qiang & Juang 07]. In [Heigold & Deselaers$^+$ 08b], it is shown how training criteria that are common in automatic speech recognition such as the MPE criterion and the MMI criterion can be extended to allow for incorporating margin terms which makes the criteria equivalent to Support Vector Machines.

Log-linear Models

Log-linear models are often simpler to train and allow for incorporating a large number of different features into the objective function. While this thesis showed that a log-linear model based on *Mel frequency cepstral coefficients* can already outperform a discriminatively trained system based on Gaussian HMMs, the approach may further benefit from exploring and incorporating new features that take larger contexts and long-range dependencies into account. A particularly appealing and promising approach to train the parameters of the log-linear model could become the lattice-based *Minimum Error Rate Training* framework [Macherey & Och$^+$ 08]. The line minimization method presented in this thesis allows for efficiently computing, representing, and exploring the exact and unsmoothed error surface of virtually any automated error metric used in natural language processing. While MERT has been shown to be tremendously powerful in statistical machine translation, its application to automatic speech recognition seems self-evident.

Appendix A

Symbols and Acronyms

The following two sections contain a comprehensive list of the symbols and acronyms used in this thesis.

A.1 Mathematical Symbols

θ	set of all parameters of the acoustic model
f	smoothing function
f'	derivative of the smoothing function
α	weighting exponent
ϱ	slope of a sigmoidal smoothing function
r	index of speech utterance
t	time frame index
T_r	number of time frames in utterance r
X_r	sequence of acoustic observations vectors $x_{r1}, ..., x_{rT_r}$
D	dimension of the acoustic observation vectors
N_r	number of spoken words in utterance r
W_r	sequence of spoken words $w_{r1}, ..., w_{rN_r}$ of utterances r
$p_\theta(X_r\|W_r)$	acoustic emission probability density for utterance r given the spoken word sequence W_r
$p_\theta(W_r\|X_r)$	*a-posteriori* probability for the spoken word sequence of utterance r
w, v	word indices
W, V	word sequences of any length
$L(W, V)$	loss function between two word sequences W and V
h	history of a word: a sequence of $m-1$ words $h_1, ..., h_{m-1}$ preceding a given word

$\mathcal{M}_r$	set of alternative (competing) word sequences of utterance r
$\mathcal{F}$	unified discriminative criterion
$\mathcal{F}_{\text{MMI}}$	*Maximum Mutual Information* (MMI) criterion
$\mathcal{F}_{\text{MCE}}$	*Minimum Classification Error* (MCE) criterion
$\mathcal{F}_{\text{CT}}$	*Corrective Training* (CT) criterion
$\mathcal{F}_{\text{FT}}$	*Falsifying Training* (FT) criterion
$\mathcal{F}_{\text{MWE}}$	*Minimum Word Error* (MWE) criterion
$\mathcal{F}_{\text{MPE}}$	*Minimum Phone Error* (MPE) criterion
$\mathcal{F}_{\text{Jeff}}$	*Jeffrey* criterion
$\mathcal{F}_{\text{MSE}}$	*Minimum Squared Error* (MSE) criterion
$\mathcal{F}_{\text{ME}}$	*Maximum Entropy* (ME) criterion
$\delta(i,j)$	Kronecker delta, equals 1 for $i=j$, and 0 otherwise
s	state of a *Hidden Markov Model* (HMM)
l	density index of a mixture density
c_{sl}	mixture weight for density l in state s
μ_{sl}	mean vector parameter of a single Gaussian density l in state s
Σ_{sld}	covariance component of a single Gaussian density l in state s
σ^2_{sl}	variance vector of a single Gaussian density l in state s (diagonal covariance)
θ_{sl}	all parameters $\{c_{sl}, \mu_{sl}, \Sigma_{sl}\}$ of a single Gaussian probability density
$p(x_{rt}\|\mu_{sl}, \Sigma_{sl})$	single Gaussian emission probability density
$p_\theta(x_{rt}\|s)$	mixture Gaussian emission probability density conditioned on state s
$s_t(X_r, W)$	state of the optimal *Viterbi* alignment path at time t for utterance r given a word sequence W
$l(x,s)$	index of the mixture density component that maximizes the emission probability for state s given the acoustic observation x
$\gamma_{rt}(s\|W)$	*forward-backward* (FB) probability to observe state s at time t for utterance r given a word sequence W
$\gamma_{rt}(s)$	generalized *forward-backward* (FB) probability to observe state s at time t for utterance r given all alternative word sequences
$g(X_r)$	any (usually polynomial) function of the acoustic observations
$\Gamma_{sl}\big(g(X_r)\big)$	discriminative averages over function g of the acoustic observations with respect to density l in state s

$\Gamma_{sl}^{\text{num}}\big(g(x)\big)$	averages over function g of the acoustic observations with respect to density l in state s for the spoken word sequence (numerator statistics)
$\Gamma_{sl}^{\text{den}}\big(g(x)\big)$	averages over function g of the acoustic observations with respect to density l in state s for all alternative word sequences (denominator statistics)
$\mathcal{S}(\theta, \overline{\theta})$	auxiliary function of the extended *Baum* (EB) algorithm
D, D_s	globally pooled or state specific iteration constants of the EB algorithm
$p_\theta(s_1^{T_r}, x_{r1}^{T_r})$	joint probability for the acoustic observations of utterance r and a state sequence $s_1^{T_r}$
$\mathcal{N}(x\|\mu, \Sigma)$	single Gaussian density with mean μ and covariance matrix Σ
$\mathbf{A}$	linear acoustic feature transformation matrix
$\text{const}(x)$	a class of functions that are constant in their argument x
$\mathcal{K}$	set of equivalence classes
$k \in \mathcal{K}$	set of indices of all Gaussians that share a common parameter
$K(s)$	equivalence class for state s
$\gamma_{rt}(s, l\|W; \theta)$	*forward-backward* (FB) probability to observe density l in state s at time t for utterance r given a word sequence W for the acoustic model as specified by θ
$\gamma_{rt}(s, l\|\theta)$	generalized *forward-backward* (FB) probability to observe density l in state s at time t for utterance r given all alternative word sequences for the acoustic model as specified by θ
$\Omega_{sl}\big(g(x)\big)$	expectation of a (usually polynomial) function of the acoustic observations under the probability density function $p(x\|\theta_{sl})$
$\text{diag}(\mathbf{A})$	same matrix as $\mathbf{A}$ but with all off-diagonal elements set to zero
a_{ij}	matrix element at row i and column j
$\|\mathbf{A}\|$	determinant of matrix $\mathbf{A}$
$\mathbf{A}_{ij}$	matrix obtained by stripping row i and column j of $\mathbf{A}$
$\text{cof}(a_{ij})$	cofactor of an element a_{ij} which is $(-1)^{i+j}\|\mathbf{A}_{ij}\|$
$\mathbf{A}^*$	adjoint or adjugate matrix of $\mathbf{A}$
$\text{tr}(\mathbf{A})$	trace of matrix $\mathbf{A}$
$\mathcal{L}(X_1^R)$	likelihood of all training samples
$\mathbf{I}$	unity matrix
$\mathbf{0}$	zero matrix

A.2 Acronyms

ARPA **A**dvanced **R**esearch **P**rojects **A**gency

BW **B**aum **W**elch

CT ***C**orrective **T**raining*

DLT ***D**iscriminative **L**inear **T**ransform*

EB ***E**xtended **B**aum* algorithm

EM ***E**xpectation **M**aximization*

FB ***F**orward-**B**ackward*

FT ***F**alsifying **T**raining*

GER **g**raph **e**rror **r**ate

HLDA ***H**eteroscedastic **L**inear **D**iscriminant **A**nalysis*

HMM **H**idden ***M**arkov* **M**odel

LDA ***L**inear **D**iscriminant **A**nalysis*

LM **l**anguage **m**odel

MCE ***M**inimum **C**lassification **E**rror*

ME ***M**aximum **E**ntropy*

MERT ***M**inimum **E**rror **R**ate **T**raining*

MFCC **M**el **F**requency **C**epstral **C**oefficients

ML ***M**aximum **L**ikelihood*

MLLT ***M**aximum **L**ikelihood **L**inear **T**ransform*

MMI ***M**aximum **M**utual **I**nformation*

MPE ***M**inimum **P**hone **E**rror*

MSE ***M**inimum **S**quared **E**rror*

MWE ***M**inimum **W**ord **E**rror*

PER **p**hone **e**rror **r**ate

PLP **p**erceptual **l**inear **p**rediction

PP **p**er**p**lexity

QEP **q**uadratic **e**igenvalue **p**roblem

SER **s**entence **e**rror **r**ate

WER **w**ord **e**rror **r**ate

WSJ **W**all **S**treet **J**ournal - a speech corpus provided by ARPA

del **del**etions

dev **dev**elopment corpus

eval **eval**uation corpus

ins **ins**ertions

sub **sub**stitutions

Appendix B

Detailed Calculations

This chapter presents definitions, proofs, and intermediate steps for some calculations discussed in this thesis. The section titles refer to the chapter in which the calculations occur.

B.1 Detailed Calculations for Chapter 2

This section contains detailed intermediate steps for calculations used in Chapter 2: "Automatic Speech Recognition".

B.1.1 Derivative of the Acoustic Probability

The following equations show the derivative of the logarithm of the acoustic probability with respect to a state-specific parameter set θ_s:

$$\begin{aligned}\frac{\partial \log p_\theta(x_1^T|W)}{\partial \theta_s} &= \frac{1}{p_\theta(x_1^T|W)} \frac{\partial}{\partial \theta_s} \sum_{[s_1^T]|W} \prod_{t=1}^{T} p(x_t|\theta_{s_t}) \cdot p(s_t|s_{t-1}) \\ &= \frac{1}{p_\theta(x_1^T|W)} \sum_{[s_1^T]|W} \sum_{t=1}^{T} \left[\frac{\partial}{\partial p(x_t|\theta_s)} \prod_{\tau=1}^{T} p(x_\tau|\theta_{s_\tau}) \cdot p(s_\tau|s_{\tau-1}) \right] \frac{\partial p(x_t|\theta_s)}{\partial \theta_s}\end{aligned} \tag{B.1}$$

Evaluating the partial differentiation and rearranging terms yields:

$$\begin{aligned}&= \frac{1}{p_\theta(x_1^T|W)} \sum_{t=1}^{T} \frac{\partial p(x_t|\theta_s)}{\partial \theta_s} \cdot \sum_{[s_1^T]|W} \delta_{s,s_t} \cdot \left[p(s_t|s_{t-1}) \prod_{\substack{\tau=1 \\ \tau \neq t}}^{T} p(x_\tau|\theta_{s_\tau}) \cdot p(s_\tau|s_{\tau-1}) \right] \\ &= \frac{1}{p_\theta(x_1^T|W)} \sum_{t=1}^{T} \frac{1}{p(x_t|\theta_s)} \frac{\partial p(x_t|\theta_s)}{\partial \theta_s} \cdot \sum_{[s_1^{t-1}, s_t=s, s_{t+1}^T]|W} \left[\prod_{\tau=1}^{T} p(x_\tau|\theta_{s_\tau}) \cdot p(s_\tau|s_{\tau-1}) \right] \\ &= \sum_{t=1}^{T} \frac{\sum_{[s_1^{t-1}, s_{t+1}^T]|W} p_\theta(x_1^T, s_1^{t-1}, s_t=s, s_{t+1}^T)}{p_\theta(x_1^T|W)} \frac{\partial \log p(x_t|\theta_s)}{\partial \theta_s} \\ &= \sum_{t=1}^{T} p_\theta(s_t=s|x_1^T, W) \cdot \frac{\partial \log p(x_t|\theta_s)}{\partial \theta_s}\end{aligned} \tag{B.2}$$

Hence, the derivative of the log acoustic probability with respect to θ_s decomposes into a sum of partial derivatives of log-emission probabilities that are weighted with respect to their

corresponding state posterior probabilities. The state posterior probabilities can efficiently be estimated by means of the forward-backward algorithm [Baum 72, Rabiner & Juang 86].

In a similar way, the logarithm of the acoustic probability can be differentiated with respect to a state transition probability:

$$\frac{\partial \log p_\theta(x_1^T|W)}{\partial p(s|\sigma)} = \frac{1}{p_\theta(x_1^T|W)} \frac{\partial}{\partial p(s|\sigma)} \sum_{[s_1^T]|W} \prod_{t=1}^{T} p(x_t|\theta_{s_t}) \cdot p(s_t|s_{t-1}) \tag{B.3}$$

$$= \frac{1}{p_\theta(x_1^T|W)} \sum_{[s_1^T]|W} \sum_{t=1}^{T} \left[\frac{\partial}{\partial p(s|\sigma)} \prod_{\tau=1}^{T} p(x_\tau|\theta_{s_\tau}) \cdot p(s_\tau|s_{\tau-1}) \right] \tag{B.4}$$

$$= \frac{1}{p_\theta(x_1^T|W)} \sum_{t=1}^{T} \sum_{[s_1^T]|W} \delta_{\sigma,s_{t-1}} \cdot \delta_{s,s_t} \cdot \left[p(x_t|\theta_{s_t}) \cdot \prod_{\substack{\tau=1 \\ \tau \neq t}}^{T} p(x_\tau|\theta_{s_\tau}) \cdot p(s_\tau|s_{\tau-1}) \right] \tag{B.5}$$

$$= \frac{1}{p_\theta(x_1^T|W)} \sum_{t=1}^{T} \frac{1}{p(s|\sigma)} \sum_{[s_1^{t-2}, s_{t-1}=\sigma, s_t=s, s_{t+1}^T]|W} \left[\prod_{\tau=1}^{T} p(x_\tau|\theta_{s_\tau}) \cdot p(s_\tau|s_{\tau-1}) \right] \tag{B.6}$$

$$= \frac{1}{p(s|\sigma)} \sum_{t=1}^{T} \frac{p_\theta(x_1^T, s_{t-1}{=}\sigma, s_t{=}s|W)}{\underbrace{p_\theta(x_1^T|W)}_{=:\gamma_{rt}(\sigma,s)}} \tag{B.7}$$

$$= \frac{1}{p(s|\sigma)} \sum_{t=1}^{T} \gamma_{rt}(\sigma, s|W) \tag{B.8}$$

B.1.2 Auxiliary Function for Optimizing Gaussian HMMs

The following equations show two forms of an auxiliary function for maximizing the parameters of a Gaussian HMM under the ML criterion:

$$\mathcal{Q}(\bar{\theta}|\theta) = \sum_{r=1}^{R} \sum_{[s_1^{T_r}]|W_r} p_\theta(s_1^{T_r}|X_r) \cdot \log p_{\bar{\theta}}(X_r, s_1^{T_r}) \tag{B.9}$$

$$= \sum_{r=1}^{R} \sum_{[s_1^{T_r}]|W_r} p_\theta(s_1^{T_r}|X_r) \sum_{t=1}^{T_r} \log \left[p_{\bar{\theta}}(x_{rt}|\bar{\theta}_{s_t}) \cdot p(s_t|s_{t-1}) \right] \tag{B.10}$$

$$= \sum_{r=1}^{R} \sum_{s,\sigma} \sum_{[s_1^{t-2}, s_{t-1}=\sigma, s_t=s, s_{t+1}^{T_r}]|W_r} p_\theta(s_1^{T_r}|X_r) \sum_{t=1}^{T_r} \log \left[p_{\bar{\theta}}(x_{rt}|\bar{\theta}_{s_t}) \cdot p(s_t|s_{t-1}) \right] \tag{B.11}$$

$$= \sum_{s} \sum_{r=1}^{R} \sum_{t=1}^{T_r} \Big[p_\theta(s_t{=}s|X_r, W_r) \cdot \log p_{\bar{\theta}(x_{rt}|\bar{\theta}_{s_t})} + \sum_{\sigma} p_\theta(s_{t-1}{=}\sigma, s_t{=}s|X_r, W_r) \cdot \log p(s|\sigma) \Big] \tag{B.12}$$

$$= \sum_{s} \sum_{r=1}^{R} \sum_{t=1}^{T_r} \Big[\gamma_{rt}(s|W_r) \cdot \log p(x_{rt}|\bar{\theta}_{s_t}) \; + \; \sum_{\sigma} \gamma_{rt}(\sigma, s|W_r) \cdot \log p(s|\sigma) \Big] \tag{B.13}$$

B.2 Detailed Calculations for Chapter 4

This section contains detailed calculations for some formulae used in Chapter 4: "Discriminative Training". Section B.2.1 shows a derivation of the MMI auxiliary function. The derivation follows the technical report [Gunawardana 01].

B.2.1 Derivation of the MMI Objective Function

The derivation of the auxiliary function used in this thesis relies on two proposition, the first of which was used in the proof of the EB algorithm [Gopalakrishnan & Kanevsky$^+$ 91], and the second of which is a straightforward application of Jensen's inequality.

Proposition 1: Let $P(\hat{\theta}) = \dfrac{Q(\hat{\theta})}{R(\hat{\theta})}$ be the ratio of two positive real valued functions Q and R on Θ. Given a value $\theta \in \Theta$ and a real constant D, define the function

$$F(\hat{\theta}|\theta) = Q(\hat{\theta}) \; - \; P(\theta) \cdot R(\hat{\theta}) \; + \; D. \tag{B.14}$$

Then

$$F(\hat{\theta}|\theta) \geqslant F(\theta|\theta) \quad \Rightarrow \quad P(\hat{\theta}) \geqslant P(\theta). \tag{B.15}$$

Proof: Note that $F(\theta|\theta) = D$:

$$\begin{aligned} F(\theta|\theta) &= Q(\theta) - P(\theta) \cdot R(\theta) + D && \text{(B.16)} \\ &= Q(\theta) - \frac{Q(\theta)}{R(\theta)} \cdot R(\theta) + D && \text{(B.17)} \\ &= D. && \text{(B.18)} \end{aligned}$$

Thus we obtain:

$$\begin{aligned} & F(\hat{\theta}|\theta) \;\geqslant\; F(\theta|\theta) && \text{(B.19)} \\ \Leftrightarrow \quad & \underbrace{Q(\hat{\theta})}_{=P(\hat{\theta})\cdot R(\hat{\theta})} - P(\theta) \cdot R(\hat{\theta}) + D \;\geqslant\; \underbrace{Q(\theta)}_{P(\theta)\cdot R(\theta)} - P(\theta) \cdot R(\theta) + D && \text{(B.20)} \\ \Leftrightarrow \quad & P(\hat{\theta}) \cdot R(\hat{\theta}) - P(\theta) \cdot R(\hat{\theta}) + D \;\geqslant\; D && \text{(B.21)} \\ \Leftrightarrow \quad & \big[P(\hat{\theta}) - P(\theta)\big] \cdot \underbrace{R(\hat{\theta})}_{>0} \;\geqslant\; 0 && \text{(B.22)} \\ \Leftrightarrow \quad & P(\hat{\theta}) - P(\theta) \;\geqslant\; 0 && \text{(B.23)} \\ \Leftrightarrow \quad & P(\hat{\theta}) \;\geqslant\; P(\theta). && \text{(B.24)} \end{aligned}$$

Proposition 2: Suppose μ is a (Lebesgue) measure on some space $\mathcal{X}$. Suppose f is positive real valued function on $\mathcal{X} \times \Theta$, and

$$F(\hat{\theta}) = \int_{\mathcal{X}} f(x, \hat{\theta}) \; d\mu. \tag{B.25}$$

Then

$$\int_{\mathcal{X}} f(x,\theta) \log f(x,\hat{\theta}) \; d\mu \;\geqslant\; \int_{\mathcal{X}} f(x,\theta) \log f(x,\theta) \; d\mu \qquad \Rightarrow \qquad F(\hat{\theta}) \;\geqslant\; F(\theta). \tag{B.26}$$

Proof: Note that $\frac{f(x,\theta)}{F(\theta)}$ is a probability density with respect to μ. Thus, we can apply *Jensen*'s inequality for convex functions to get the desired result. Since log is a concave function, $-\log$ is a convex function, and hence the requirements for applying Jensen's inequality are fulfilled. To simplify the notation we will use the following abbreviations:

$$g_i \equiv f(x,\theta) \tag{B.27a}$$
$$f_i \equiv f(x,\hat{\theta}) \tag{B.27b}$$
$$G \equiv F(\theta) \tag{B.27c}$$
$$F \equiv F(\hat{\theta}) \tag{B.27d}$$

Thus we can write:

$$\int_{\mathcal{X}} f(x,\theta)\log f(x,\hat{\theta})\,d\mu \;\geqslant\; \int_{\mathcal{X}} f(x,\theta)\log f(x,\theta)\,d\mu \tag{B.28}$$
$$\Leftrightarrow \quad \sum_i g_i \log f_i \;\geqslant\; \sum_i g_i \log g_i \tag{B.29}$$
$$\Leftrightarrow \quad \sum_i g_i \log \frac{f_i}{g_i} \;\geqslant\; 0 \tag{B.30}$$
$$\Leftrightarrow \quad \sum_i \frac{g_i}{G} \log \frac{f_i}{g_i} \;\geqslant\; 0 \tag{B.31}$$
$$\Leftrightarrow \quad 0 \;\geqslant\; -\sum_i \frac{g_i}{G} \log \frac{f_i}{g_i} \;\geqslant\; -\log \sum_i \frac{g_i}{G} \cdot \frac{f_i}{g_i} \qquad \text{(Jensen inequality)} \tag{B.32}$$
$$\Leftrightarrow \quad 0 \;\leqslant\; \log \sum_i \frac{g_i}{G} \cdot \frac{f_i}{g_i} \tag{B.33}$$
$$= \log \sum_i \frac{f_i}{G} \tag{B.34}$$
$$= \log \frac{F}{G} \tag{B.35}$$
$$\Leftrightarrow \quad F \;\geqslant\; G \tag{B.36}$$

Eq. (B.32) applies the *Jensen* inequality for convex functions which is defined as follows (cf. [Teubner 1996, p. 39]): Let p_k be real valued, non-negative coefficients with $\sum_{k=1}^{K} p_k = 1$ and $x_k \in \mathbb{R}^N$ for all k. Moreover let $F:\ \mathbb{R}^N \to \mathbb{R}$ be convex. Then the following inequality holds:

$$F\left(\sum_{k=1}^{K} p_k \cdot x_k\right) \;\leqslant\; \sum_{k=1}^{K} p_k \cdot F(x_k) \tag{B.37}$$

A similar inequality exists for integrals:

$$F\left(\frac{\int_{\mathcal{X}} p(x)\cdot g(x)\,dx}{\int_{\mathcal{X}} p(x)\,dx}\right) \leqslant \frac{\int_{\mathcal{X}} p(x)\cdot F\big(g(x)\big)\,dx}{\int_{\mathcal{X}} p(x)\,dx} \tag{B.38}$$

Now we can derive the EB algorithm as follows:

$$F(\hat{\theta},\theta) = \underbrace{p_{\hat{\theta}}(X,W)}_{Q(\hat{\theta})} - \underbrace{p_{\theta}(W|X)}_{P(\theta)} \cdot \underbrace{p_{\hat{\theta}}(X)}_{R(\hat{\theta})} + D \tag{B.39}$$

$$= \sum_{s_1^T} \Big[p_{\hat{\theta}}(W, s_1^T, X) - p_{\theta}(W|X) \cdot p_{\hat{\theta}}(X, s_1^T) \Big] + D \tag{B.40}$$

$$= \sum_{s_1^T} \Big[p(W, s_1^T) - p_{\theta}(W|X) \cdot \underbrace{p(s_1^T)}_{=p_{\hat{\theta}}(X,s_1^T)} \Big] \cdot p_{\hat{\theta}}(X|s_1^T) + D \tag{B.41}$$

$$= \sum_{s_1^T} \left(\Big[p(W, s_1^T) - p_{\theta}(W|X) \cdot p(s_1^T) \Big] \cdot p_{\hat{\theta}}(X|s_1^T) + d(s_1^T) \right) \quad \text{with } \sum_{s_1^T} d(s_1^T) = D \tag{B.42}$$

$$= \sum_{s_1^T} \underbrace{\int_{\mathcal{X}} \underbrace{\Big[p(W, s_1^T) \cdot 1_{\{X\}}(Y) - p_{\theta}(W|X) \cdot p(s_1^T) \cdot 1_{\{X\}}(Y) + d(s_1^T) \Big]}_{(*)} \cdot p_{\hat{\theta}}(Y|s_1^T)\, dY}_{=: f(W, s_1^T, X, Y, \hat{\theta}|\theta)} \tag{B.43}$$

$$= \sum_{s_1^T} \int_{\mathcal{X}} f(W, s_1^T, X, Y, \hat{\theta}|\theta)\, dY \tag{B.44}$$

Here the following identities were used:

$$p_{\hat{\theta}}(X, W) = \sum_{s_1^T} p_{\hat{\theta}}(W, s_1^T, X) \tag{B.45}$$

$$= \sum_{s_1^T} p_{\hat{\theta}}(X|s_1^T, W) \cdot p_{\hat{\theta}}(W, s_1^T) \tag{B.46}$$

$$= \sum_{s_1^T} p_{\hat{\theta}}(X|s_1^T) \cdot p_{\hat{\theta}}(W, s_1^T) \tag{B.47}$$

Here we make the assumption that the joint probability $p_{\hat{\theta}}(W, s_1^T)$ is equal to zero if the state sequence s_1^T is not consistent with the word sequence W. With propositions 1 and 2 we then obtain:

$$\begin{aligned} &\sum_{s_1^T} \int_{\mathcal{X}} f(W, s_1^T, X, Y, \theta|\theta) \log f(W, s_1^T, X, Y, \hat{\theta}|\theta)\, dY \\ &\qquad \geqslant \sum_{s_1^T} \int_{\mathcal{X}} f(W, s_1^T, X, Y, \theta|\theta) \log f(W, s_1^T, X, Y, \theta|\theta)\, dY \end{aligned} \tag{B.48}$$

$$\underset{\text{Prop. 2}}{\Rightarrow} F(\hat{\theta}|\theta) \;\geqslant\; F(\theta|\theta) \tag{B.49}$$

$$\underset{\text{Prop. 1}}{\Rightarrow} P(\hat{\theta}|\theta) \;\geqslant\; P(\theta|\theta) \tag{B.50}$$

$$\Rightarrow \quad p_{\hat{\theta}}(W|X) \geqslant p_{\theta}(W|X) \tag{B.51}$$

It remains to maximize the following expression with respect to $\theta^{(i+1)}$:

$$\theta^{(i+1)} \in \operatorname*{argmax}_{\theta\in\Theta} \sum_{s_1^T} \int_{\mathcal{X}} f(W, s_1^T, X, Y, \theta^{(i)}|\theta^{(i)}) \log f(W, s_1^T, X, Y, \theta|\theta^{(i)})\, dY \tag{B.52}$$

$$= \operatorname*{argmax}_{\theta\in\Theta} \sum_{s_1^T} \int_{\mathcal{X}} f(W, s_1^T, X, Y, \theta^{(i)}|\theta^{(i)}) \log p_\theta(Y|s_1^T)\, dY \qquad \text{since } (*) \text{ does not depend on } \theta \tag{B.53}$$

$$= \operatorname*{argmax}_{\theta\in\Theta} \sum_{s_1^T} \int_{\mathcal{X}} \Big[p(W, s_1^T) \cdot 1_{\{X\}}(Y) - p_{\theta^{(i)}}(W|X) \cdot p(s_1^T) \cdot 1_{\{X\}}(Y) + d(s_1^T)\Big] \cdot \\ \cdot\, p_{\theta^{(i)}}(Y|s_1^T) \cdot \log p_\theta(Y|s_1^T)\, dY \tag{B.54}$$

$$= \operatorname*{argmax}_{\theta\in\Theta} \sum_{s_1^T} \Big[\underbrace{p(W, s_1^T) \cdot p_{\theta^{(i)}}(X|s_1^T)}_{=p_{\theta^{(i)}}(W, s_1^T, X)} - p_{\theta^{(i)}}(W|X) \cdot p_{\theta^{(i)}}(X, s_1^T)\Big] \cdot \log p_\theta(X|s_1^T) \quad \Big| \quad \cdot \frac{1}{p_{\theta^{(i)}}(X, W)} \\ + \sum_{s_1^T} d(s_1^T) \int_{\mathcal{X}} p_{\theta^{(i)}}(Y|s_1^T) \cdot \log p_\theta(Y|X)\, dY \tag{B.55}$$

$$= \operatorname*{argmax}_{\theta\in\Theta} \sum_{s_1^T} \Big[p_{\theta^{(i)}}(s_1^T|X, W) - \underbrace{\frac{p_{\theta^{(i)}}(W|X) \cdot \sum_V p_{\theta^{(i)}}(X, V, s_1^T)}{p_{\theta^{(i)}}(X, W)}}_{= \frac{p_{\theta^{(i)}}(X,W)}{p_{\theta^{(i)}}(X)} \cdot \frac{\sum_V p_{\theta^{(i)}}(s_1^T|X,V) \cdot p_{\theta^{(i)}}(V|X) \cdot p_{\theta^{(i)}}(X)}{p_{\theta^{(i)}}(X,W)}}\Big] \cdot \log p_\theta(X|s_1^T) \\ + \sum_{s_1^T} d(s_1^T) \int_{\mathcal{X}} p_{\theta^{(i)}}(Y|s_1^T) \cdot \log p_\theta(Y|s_1^T)\, dY \tag{B.56}$$

$$= \operatorname*{argmax}_{\theta\in\Theta} \sum_{s_1^T} \Big[p_{\theta^{(i)}}(s_1^T|X, W) \; - \sum_V p_{\theta^{(i)}}(V|X) \cdot p_{\theta^{(i)}}(s_1^T|X, V)\Big] \cdot \log p_\theta(X|s_1^T) \\ + \sum_{s_1^T} d(s_1^T) \int_{\mathcal{X}} p_{\theta^{(i)}}(Y|s_1^T) \cdot \log p_\theta(Y|s_1^T)\, dY \tag{B.57}$$

The last equation can be transformed into the following expression:

$$= \operatorname*{argmax}_{\theta\in\Theta} \sum_s \sum_{t=1}^T \Big[\gamma_t(s|W) - \overbrace{\sum_V p_{\theta^{(i)}}(V|X) \cdot \gamma_t(s|V)}^{=:\gamma_t(s)}\Big] \cdot \log p_\theta(x_t|s) \\ + \sum_{\sigma,s} \sum_{t=1}^T \Big[\gamma_t(\sigma, s|W) - \underbrace{\sum_V p_{\theta^{(i)}}(V|X) \cdot \gamma_t(\sigma, s|V)}_{=:\gamma_t(\sigma,s)}\Big] \cdot \log p_\theta(s|\sigma) \\ + \sum_{s_1^T} d(s_1^T) \sum_{t=1}^T \int_{\mathbb{R}} p_{\theta^{(i)}}(x|s_t) \cdot \log p_\theta(x|s_t)\, dx \tag{B.58}$$

$$
\begin{aligned}
= \operatorname*{argmax}_{\theta\in\Theta} & \sum_{\sigma,s}\sum_{t=1}^{T}\Big[\gamma_t(\sigma,s|W) \;-\; \gamma_t(\sigma,s)\Big]\cdot\log p_\theta(x_t,s|\sigma) \\
& + \sum_s D_s \int_{\mathbb{R}} p_{\theta^{(i)}}(x|s_t)\cdot\log p_\theta(x|s_t)
\end{aligned}
\tag{B.59}
$$

Proof:

$$
\sum_{s_1^T} p_{\theta^{(i)}}(s_1^T|X,W)\cdot\log p_\theta(X|s_1^T) \tag{B.60}
$$

$$
= \sum_{s_1^T} \frac{p_{\theta^{(i)}}(s_1^T,X|W)}{p_{\theta^{(i)}}(X|W)}\cdot\log\prod_{t=1}^{T} p_\theta(x_t|s_t)\cdot p_\theta(s_t|s_{t-1}) \tag{B.61}
$$

$$
= \sum_{t=1}^{T}\sum_{\sigma,s}\sum_{[s_1^{t-2},s_{t-1}=\sigma,s_t=s,s_{t+1}^T]} \frac{p_{\theta^{(i)}}(X,s_1^T|W)}{p_{\theta^{(i)}}(X|W)}\cdot\Big[\log p_\theta(x_t|s)\cdot p_\theta(s|\sigma)\Big] \tag{B.62}
$$

$$
\begin{aligned}
= & \sum_{t=1}^{T}\sum_{s}\sum_{[s_1^{t-1},s_t=s,s_{t+1}^T]} \frac{p_{\theta^{(i)}}(X,s_1^T|W)}{p_{\theta^{(i)}}(X|W)}\cdot\log p_\theta(x_t|s) \\
& + \sum_{t=1}^{T}\sum_{\sigma,s}\sum_{[s_1^{t-2},s_{t-1}=\sigma,s_t=s,s_{t+1}^T]} \frac{p_{\theta^{(i)}}(X,s_1^T|W)}{p_{\theta^{(i)}}(X|W)}\cdot\log p_\theta(s|\sigma)
\end{aligned}
\tag{B.63}
$$

$$
\begin{aligned}
= & \sum_{t=1}^{T}\sum_{s} p_{\theta^{(i)}}(s_t{=}s|X,W)\cdot\log p_\theta(x_t|s) \\
& + \sum_{t=1}^{T}\sum_{\sigma,s} p_{\theta^{(i)}}(s_{t-1}{=}\sigma,\, s_t{=}s|X,W)\cdot\log p_\theta(s|\sigma)
\end{aligned}
\tag{B.64}
$$

$$
= \sum_{s}\sum_{t=1}^{T}\Big[\gamma_t(s|W)\cdot\log p_\theta(x_t|s) \;+\; \sum_{\sigma}\gamma_t(\sigma,s|W)\cdot\log p_\theta(s|\sigma)\Big] \tag{B.65}
$$

Here, we used the following identities:

$$
\sum_{s_1^T} d(s_1^T)\int_{\mathcal{X}} p(X|s_1^T)\cdot\log q(X|s_1^T)\;dX \tag{B.66}
$$

$$
= \sum_{s_1^T} d(s_1^T)\int_{\mathbb{R}_1}\cdots\int_{\mathbb{R}_T} p(x_1^T|s_1^T)\cdot\log q(x_1^T|s_1^T)\;dx_1\ldots dx_T \tag{B.67}
$$

$$
= \sum_{s_1^T} d(s_1^T)\int_{\mathbb{R}_1}\cdots\int_{\mathbb{R}_T}\left[\prod_{t=1}^{T} p(x_t|s_t)\right]\cdot\log\left[\prod_{t=1}^{T} q(x_t|s_t)\right] dx_1\ldots dx_T \tag{B.68}
$$

$$
= \sum_{s_1^T} d(s_1^T)\sum_{t=1}^{T}\int_{\mathbb{R}_1}\cdots\int_{\mathbb{R}_T}\underbrace{\left[\prod_{\substack{\tau=1\\ \tau\neq t}}^{T} p(x_\tau|s_\tau)\right]}_{\text{const}(x_t)}\cdot p(x_t|s_t)\cdot\log q(x_t|s_t)\;dx_1\ldots dx_T \tag{B.69}
$$

$$= \sum_{s_1^T} d(s_1^T) \sum_{t=1}^{T} \int_{\mathbb{R}_t} p(x_t|s_t) \cdot \log q(x_t|s_t)\ dx_t \cdot$$

$$\cdot \underbrace{\int_{\mathbb{R}_1} \cdots \int_{\mathbb{R}_{t-1}} \int_{\mathbb{R}_{t+1}} \cdots \int_{\mathbb{R}_T} \prod_{\substack{\tau=1 \\ \tau \neq t}}^{T} p(x_\tau|s_\tau)\ dx_1 \ldots dx_{t-1} dx_{t+1} \ldots dx_T}_{= \prod_{\substack{\tau=1 \\ \tau \neq t}}^{T} \int_{x_\tau} p(x_\tau|s_\tau)\ dx_\tau = 1} \tag{B.70}$$

$$= \sum_{s_1^T} d(s_1^T) \sum_{t=1}^{T} \int_{\mathbb{R}} p(x|s_t) \cdot \log q(x|s_t)\ dx \tag{B.71}$$

$$= \sum_{s} \underbrace{\sum_{t=1}^{T} \sum_{[s_1^{t-1}, s_t = s, s_{t+1}^T]} d(s_1^T)}_{=: D_s} \int_{\mathbb{R}} p(x|s) \cdot \log q(x|s)\ dx \tag{B.72}$$

$$= \sum_{s} D_s \int_{\mathbb{R}} p(x|s) \cdot \log q(x|s)\ dx \tag{B.73}$$

B.2.2 The Logistic Function and its Derivative

Due to

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{B.74}$$

$$= \frac{e^{2x} - 1}{e^{2x} + 1} \tag{B.75}$$

$$= \frac{e^{2x} + 1 - 2}{e^{2x} + 1} \tag{B.76}$$

$$= 1 - \frac{2}{1 + e^{2x}} \tag{B.77}$$

the logistic function can be expressed via the hyperbolic tangent:

$$\frac{1}{1 + e^{2x}} = \frac{1}{2} \cdot \Big[1 - \tanh(x)\Big]. \tag{B.78}$$

With

$$f(x) = -\frac{1}{1 + e^{2\rho x}} \tag{B.79}$$

and

$$\frac{\mathrm{d}\tanh(x)}{\mathrm{d}x} = \frac{1}{\cosh^2(x)} \tag{B.80}$$

$$= 1 - \tanh^2(x) \tag{B.81}$$

the derivative of the sigmoid function yields:

$$\frac{\mathrm{d}f(x)}{\mathrm{d}x} = \frac{1}{2} \cdot \frac{1}{\cosh^2(\rho x)} \cdot \rho \tag{B.82}$$

$$= \frac{\rho}{2} \cdot \Big[1 - \tanh^2(\rho x)\Big] \tag{B.83}$$

B.2.3 Minimum Word Error and Minimum Phone Error Criterion

Both the *Minimum Word Error* (MWE) criterion and the *Minimum Phone Error* (MPE) criterion aim at minimizing the expectation value of an error related loss function on training data. If this loss function is local, both criteria can be computed efficiently on word or phoneme lattices. This section contains some detailed calculations for deriving the MWE and the MPE criterion with respect to a state-specific parameter set θ_s. We first note that the derivative of the general discriminative approach as proposed in Chapter 4 can be reduced to the derivative of the local discriminative criterion:

$$\mathcal{F} = \frac{1}{R} \sum_{r=1}^{R} f\big(\log \mathcal{F}_r\big). \tag{B.84}$$

This gives:

$$\frac{\partial \mathcal{F}}{\partial \theta_{sl}} = \frac{1}{R} \sum_{r=1}^{R} f'\big(\log \mathcal{F}_r\big) \cdot \frac{1}{\mathcal{F}_r} \cdot \frac{\partial \mathcal{F}_r}{\partial \theta_{sl}} \tag{B.85}$$

We also note that the logarithm of the acoustic probability, $\log p_\theta(x_1^T|w_1^N)$, for a word sequence w_1^N can be represented by a corresponding sequence of phonemes (or any other subword units) h_1^M. Hence, the derivative of the log probability with respect to θ_s yields:

$$\frac{\partial p_\theta^\kappa(x_1^T|h_1^M)}{\partial \theta_{sl}} = \kappa \cdot p_\theta^\kappa(x_1^T|h_1^M) \cdot \frac{\partial}{\partial \theta_{sl}} \log p_\theta(x_1^T|h_1^M) \tag{B.86}$$

$$= \kappa \cdot p_\theta^\kappa(x_1^T|h_1^M) \cdot \sum_{m=1}^{M} \frac{\partial}{\partial \theta_{sl}} \log p_\theta(x_{\tau_m}^{t_m}|h_m) \tag{B.87}$$

$$= \kappa \cdot p_\theta^\kappa(x_1^T|h_1^M) \cdot \sum_{m=1}^{M} \sum_{t=\tau_m}^{t_m} p_\theta(s_t{=}s|x_{\tau_m}^{t_m}, h_m) \cdot \frac{\partial \log p(x_t|\theta_s)}{\partial \theta_s} \tag{B.88}$$

Both equations lead to the following derivative of the local discriminative criterion for both the MWE and the MPE criterion:

$$\begin{aligned}
\frac{\partial \mathcal{F}_r}{\partial \theta_{sl}} &= \kappa \cdot \frac{\sum\limits_{h_1^M} p^\kappa(h_1^M) \cdot p_\theta^\kappa(X_r|h_1^M) \sum\limits_{m'=1}^{M} \ell(h_{m'}, W_r) \cdot \sum\limits_{m=1}^{M} \frac{\partial}{\partial \theta_{sl}} \log p_\theta(x_r{}_{\tau_m}^{t_m}|h_m)}{\sum\limits_{\bar{h}_1^{\bar{M}}} p^\kappa(\bar{h}_1^{\bar{M}}) \cdot p_\theta^\kappa(X_r|\bar{h}_1^{\bar{M}})} \\
&\quad \kappa \cdot \frac{\sum\limits_{h_1^M} p^\kappa(h_1^M) \cdot p_\theta^\kappa(X_r|h_1^M) \cdot \sum\limits_{m'=1}^{M} \ell(h_{m'}, W_r)}{\Big[\sum\limits_{\bar{h}_1^{\bar{M}}} p^\kappa(\bar{h}_1^{\bar{M}}) \cdot p_\theta^\kappa(\bar{h}_1^{\bar{M}})\Big]^2} \cdot \\
&\qquad \cdot \sum_{\tilde{h}_1^{\tilde{M}}} p^\kappa(\tilde{h}_1^{\tilde{M}}) \cdot p_\theta^\kappa(X_r|\tilde{h}_1^{\tilde{M}}) \cdot \sum_{m=1}^{M} \frac{\partial}{\partial \theta_{sl}} \log p_\theta(x_r{}_{\tau_m}^{t_m}|\tilde{h}_m) \\
&= \kappa \cdot \sum_h \Bigg(\sum_{h_1^M : \exists m\ h_m = h} p_\theta(h_1^M|X_r) \cdot \sum_{m=1}^{M} \ell(h_m, W_r) \\
&\qquad - \sum_{\bar{h}_1^{\bar{M}}} p_\theta(\bar{h}_1^{\bar{M}}|X_r) \cdot \sum_{\bar{m}=1}^{\bar{M}} \ell(\bar{h}_{\bar{m}}, W_r) \cdot \sum_{h_1^M : \exists m\ h_m = h} p_\theta(h_1^M|X_r) \Bigg) \cdot \\
&\qquad \cdot \frac{\partial}{\partial \theta_{sl}} \log p_\theta(x_r{}_{\tau(h)}^{t(h)}|h) \\
&= \kappa \cdot \sum_h \Bigg(\frac{1}{p_\theta(h|X_r)} \cdot \sum_{h_1^M : \exists m\ h_m = h} p_\theta(h_1^M|X_r) \cdot \sum_{m=1}^{M} \ell(h_m, W_r) \\
&\qquad - \sum_{h_1^M} p_\theta(h_1^M|X_r) \cdot \sum_{m=1}^{M} \ell(h_m, |W_r) \Bigg) \cdot p_\theta(h|X_r) \cdot \\
&\qquad \cdot \frac{\partial}{\partial \theta_{sl}} \log p_\theta(x_r{}_{\tau(h)}^{t(h)}|h) \\
&= \kappa \cdot \sum_h \Bigg(\underbrace{\frac{1}{p_\theta(h|X_r)} \cdot \sum_{h_1^M : \exists m\ h_m = h} p_\theta(h_1^M|X_r) \cdot \sum_{m=1}^{M} \ell(h_m, W_r)}_{=: c_r(h)} \\
&\qquad \underbrace{- \sum_{h_1^M} p_\theta(h_1^M|X_r) \cdot \sum_{m=1}^{M} \ell(h_m, |W_r)}_{=: \mathcal{F}_r(\theta)} \Bigg) \cdot p_\theta(h|X_r) \cdot \\
&\qquad \cdot \sum_{t=\tau(h)}^{t(h)} p_\theta(s_t{=}s|x_r{}_{\tau(h)}^{t(h)}, h) \cdot \frac{\partial}{\partial \theta_{sl}} \log p(x_{rt}|\theta_s) \qquad \text{(B.89)}
\end{aligned}$$

The term $c_r(h)$ is basically the criterion itself. To provide an efficient estimation scheme for this term, $c_r(h)$ is decomposed such that the concept of dynamic programming can be applied. To accomplish this, $c_r(h)$ is split into a portion $\alpha'_r(h)$ that reflects the forward accuracy and a portion

$\beta'_r(h)$ reflecting the backward accuracy:

$$c_r(h) = \frac{\sum\limits_{h_1^M:\exists m\ h_m=h} p_\theta(h_1^M|X_r)\cdot\sum\limits_{m=1}^{M}\ell(h_m,W_r)}{\sum\limits_{h_1^M:\exists m\ h_m=h} p_\theta(h_1^M|X_r)} \tag{B.90}$$

$$= \frac{\sum\limits_{h_1^M:\exists m\ h_m=h} p^\kappa(h_1^M)\cdot p_\theta^\kappa(X_r|h_1^M)\cdot\sum\limits_{m=1}^{M}\ell(h_m,W_r)}{\sum\limits_{h_1^M:\exists m\ h_m=h} p_\theta(h_1^M|X_r)} \tag{B.91}$$

$$= \frac{\sum\limits_{h_1^N,h,\bar{h}_1^{\bar{N}}} p^\kappa(h_1^N,h)\cdot p_\theta^\kappa({x_r}_1^{t(h)}|h_1^N,h)\cdot p^\kappa(\bar{h}_1^{\bar{N}})\cdot p_\theta^\kappa({x_r}_{\tau(\bar{h}_1)}^{T_r}|\bar{h}_1^{\bar{N}})\cdot\mathcal{L}(h_1^N,h,\bar{h}_1^{\bar{N}};W_r)}{\sum\limits_{h_1^N,h,\bar{h}_1^{\bar{N}} p^\kappa(h_1^N,h)\cdot p_\theta^\kappa({x_r}_1^{t(h)}|h_1^N,h)\cdot p^\kappa(\bar{h}_1^{\bar{N}})\cdot p_\theta^\kappa({x_r}_{\tau(\bar{h}_1)}^{T_r}|\bar{h}_1^{\bar{N}})} \tag{B.92}$$

$$\text{with}\qquad \mathcal{L}(h_1^N,h,\bar{h}_1^{\bar{N}};W_r) = \left[\sum_{n=1}^{N}\ell(h_n,W_r)\ +\ \ell(h,W_r)\ +\ \sum_{\bar{n}=1}^{\bar{N}}\ell(\bar{h}_{\bar{n}},W_r)\right]$$

$$= \underbrace{\frac{\sum\limits_{h_1^N} p^\kappa(h_1^N,h)\cdot p_\theta^\kappa({x_r}_1^{t(h)}|h_1^N,h)\cdot\left[\sum\limits_{n=1}^{N}\ell(h_n,W_r)+\ell(h,W_r)\right]\cdot\left[\sum\limits_{\bar{h}_1^{\bar{N}}} p^\kappa(\bar{h}_1^{\bar{N}})\cdot p_\theta^\kappa({x_r}_{\tau(\bar{h}_1)}^{T_r}|\bar{h}_1^{\bar{N}})\right]}{\sum\limits_{h_1^N} p^\kappa(h_1^N,h)\cdot p_\theta^\kappa({x_r}_1^{t(h)}|h_1^N,h)\cdot\left[\sum\limits_{\bar{h}_1^{\bar{N}}} p^\kappa(\bar{h}_1^{\bar{N}})\cdot p_\theta^\kappa({x_r}_{\tau(\bar{h}_1)}^{T_r}|\bar{h}_1^{\bar{N}})\right]}}_{=:\alpha'_r(h)}$$

$$+ \underbrace{\frac{\left[\sum\limits_{h_1^N} p^\kappa(h_1^N,h)\cdot p_\theta^\kappa({x_r}_1^{t(h)}|h_1^N,h)\right]\cdot\sum\limits_{\bar{h}_1^{\bar{N}}} p^\kappa(\bar{h}_1^{\bar{N}}|h)\cdot p_\theta^\kappa({x_r}_{\tau(\bar{h}_1)}^{T_r}|\bar{h}_1^{\bar{N}})\cdot\sum\limits_{\bar{n}=1}^{\bar{N}}\ell(\bar{h}_{\bar{n}},W_r)}{\left[\sum\limits_{h_1^N} p^\kappa(h_1^N,h)\cdot p_\theta^\kappa({x_r}_1^{t(h)}|h_1^N,h)\right]\cdot\sum\limits_{\bar{h}_1^{\bar{N}}} p^\kappa(\bar{h}_1^{\bar{N}}|h)\cdot p_\theta^\kappa({x_r}_{\tau(\bar{h}_1)}^{T_r}|\bar{h}_1^{\bar{N}})}}_{=:\beta'_r(h)} \tag{B.93}$$

$$= \alpha'_r(h)\ +\ \beta'_r(h) \tag{B.94}$$

Transforming the expressions for $\alpha'_r(h)$ and $\beta'_r(h)$ into corresponding recursive formulae provide an efficient computation scheme for estimating the forward-backward accuracies by means of dynamic programming. To simplify the notation, the index r is omitted in the following, i.e., we write

$x_t \equiv x_{rt}$, $\alpha'(h) \equiv \alpha'_r(h)$, and $\beta'(h) = \beta'_r(h)$. Then, $\alpha'(h)$ can be re-written as follows:

$$\alpha'_r(h) = \frac{\sum_{h_1^N} p^\kappa(h_1^N, h) \cdot p_\theta^\kappa({x_r}_1^{t(h)} | h_1^N, h) \cdot \left[\sum_{n=1}^{N} \ell(h_n, W_r) + \ell(h, W_r) \right]}{\sum_{h_1^N} p^\kappa(h_1^N, h) \cdot p_\theta^\kappa({x_r}_1^{t(h)} | h_1^N, h) \cdot} \tag{B.95}$$

$$= \frac{\sum_{\tilde{h}=h_N} \sum_{h_1^{N-1}} p^\kappa(h_1^N) \cdot p_\theta^\kappa({x_r}_1^{t(h_N)} | h_1^N) \cdot \sum_{n=1}^{N} \ell(h_n, W_r) \cdot p^\kappa(h|h_N) \cdot p_\theta^\kappa({x_r}_{\tau(h)}^{t(h)} | h)}{\sum_{\tilde{h}=h_N} \sum_{h_1^{N-1}} p^\kappa(h_1^N) \cdot p_\theta^\kappa({x_r}_1^{t(h_N)} | h_1^N) \cdot p^\kappa(h|h_N) \cdot p_\theta^\kappa({x_r}_{\tau(h)}^{t(h)} | h)} \tag{B.96}$$

$$+ \ell(h, W_r)$$

$$= \frac{\sum_{\tilde{h}=h_N} \sum_{h_1^{N-1}} p^\kappa(h_1^N) \cdot p_\theta^\kappa({x_r}_1^{t(h_N)} | h_1^N) \cdot \sum_{n=1}^{N} \ell(h_n, W_r) \cdot p^\kappa(h|h_N) \cdot}{\sum_{\tilde{h}=h_N} \underbrace{\sum_{h_1^{N-1}} p^\kappa(h_1^N) \cdot p_\theta^\kappa({x_r}_1^{t(h_N)} | h_1^N) \cdot p^\kappa(h|h_N)}_{=:\alpha_r(\tilde{h})}} + \ell(h, W_r) \tag{B.97}$$

$$= \frac{\overbrace{\sum_{\tilde{h}=h_N} \sum_{h_1^{N-1}} p^\kappa(h_1^{N-1}, h_N) \cdot p_\theta^\kappa({x_r}_1^{t(h_N)} | h_1^{N-1}, h_N) \cdot \left[\sum_{n=1}^{N} \ell(h_n, W_r) + \ell(h_N, W_r) \right]}^{=\alpha'_r(\tilde{h}) \cdot \alpha_r(\tilde{h})} \cdot p^\kappa(h|h_N)}{\sum_{\tilde{h}=h_N} \alpha_r(\tilde{h}) \cdot p^\kappa(h|h_N)}$$

$$+ \ell(h, W_r) \tag{B.98}$$

$$= \frac{\sum_{\tilde{h}} \alpha'_r(\tilde{h}) \cdot \alpha_r(\tilde{h}) \cdot p^\kappa(h|\tilde{h})}{\sum_{\tilde{h}} \alpha'_r(\tilde{h}) \cdot \alpha_r(\tilde{h})} + \ell(h, W_r) \tag{B.99}$$

Here, $\sum_{\tilde{h}}$ runs over all predecessors $\tilde{h}$ of h. The backward accuracy is rewritten in a similar way:

$$\beta'_r(h) = \frac{\sum_{h_1^N} p^\kappa(h_1^N | h) \cdot p_\theta^\kappa({x_r}_{\tau(h_1)}^{T_r}) \cdot \sum_{n=1}^{N} \ell(h_n, W_r)}{\sum_{h_1^N} p^\kappa(h_1^N | h) \cdot p_\theta^\kappa({x_r}_{\tau(h_1)}^{T_r})} \tag{B.100}$$

$$= \frac{\sum_{\tilde{h}=h_1} p^\kappa(h_1|h) \cdot p_\theta^\kappa({x_r}_{\tau(h_1)}^{t(h_1)} | h_1) \cdot \sum_{h_2^N} p^\kappa(h_2^N | h_1) \cdot p({x_r}_{\tau h_2}^{T_r} | h_2^N) \cdot \left[\ell(h_1|W_r) + \sum_{n=2}^{N} \ell(h_n, W_r) \right]}{\sum_{\tilde{h}=h_1} p^\kappa(h_1|h) \cdot p_\theta^\kappa({x_r}_{\tau(h_1)}^{t(h_1)} | h_1) \cdot \underbrace{\sum_{h_2^N} p^\kappa(h_2^N | h_1) \cdot p({x_r}_{\tau h_2}^{T_r} | h_2^N)}_{=:\beta_r(\tilde{h})}} \tag{B.101}$$

$$= \frac{\sum\limits_{\tilde{h}=h_1} p^{\kappa}(h_1|h) \cdot p_{\theta}^{\kappa}(x_{r\,\tau(h_1)}^{t(h_1)}|h_1) \cdot \overbrace{\sum\limits_{h_2^N} p^{\kappa}(h_2^N|h_1) \cdot p(x_{r\,\tau h_2}^{T_r}|h_2^N) \cdot \sum\limits_{n=2}^{N} \ell(h_n, W_r)}^{=\beta_r'(\tilde{h}) \cdot \beta_r(\tilde{h})}}{\sum\limits_{\tilde{h}=h_1} p^{\kappa}(h_1|h) \cdot p_{\theta}^{\kappa}(x_{r\,\tau(h_1)}^{t(h_1)}|h_1) \cdot \beta_r(\tilde{h})} \tag{B.102}$$

$$+ \frac{\sum\limits_{\tilde{h}=h_1} p^{\kappa}(h_1|h) \cdot p_{\theta}^{\kappa}(x_{r\,\tau(h_1)}^{t(h_1)}|h_1) \cdot \ell(h_1, W_r) \cdot \sum\limits_{h_2^N} p^{\kappa}(h_2^N|h_1) \cdot p(x_{r\,\tau h_2}^{T_r}|h_2^N)}{\sum\limits_{\tilde{h}=h_1} p^{\kappa}(h_1|h) \cdot p_{\theta}^{\kappa}(x_{r\,\tau(h_1)}^{t(h_1)}|h_1) \cdot \beta_r(\tilde{h})}$$

$$= \frac{\sum\limits_{\tilde{h}=h_1} p^{\kappa}(h_1|h) \cdot p_{\theta}^{\kappa}(x_{r\,\tau(h_1)}^{t(h_1)}|h_1) \cdot \beta_r'(\tilde{h}) \cdot \beta_r(\tilde{h})}{\sum\limits_{\tilde{h}=h_1} p^{\kappa}(h_1|h) \cdot p_{\theta}^{\kappa}(x_{r\,\tau(h_1)}^{t(h_1)}|h_1) \cdot \beta_r(\tilde{h})} \tag{B.103}$$

$$+ \frac{\sum\limits_{\tilde{h}=h_1} p^{\kappa}(h_1|h) \cdot p_{\theta}^{\kappa}(x_{r\,\tau(h_1)}^{t(h_1)}|h_1) \cdot \ell(h_1, W_r) \cdot \beta_r(\tilde{h})}{\sum\limits_{\tilde{h}=h_1} p^{\kappa}(h_1|h) \cdot p_{\theta}^{\kappa}(x_{r\,\tau(h_1)}^{t(h_1)}|h_1) \cdot \beta_r(\tilde{h})}$$

$$= \frac{\sum\limits_{\tilde{h}=h_1} p^{\kappa}(h_1|h) \cdot p_{\theta}^{\kappa}(x_{r\,\tau(h_1)}^{t(h_1)}|h_1) \cdot \beta_r(\tilde{h}) \cdot \Big[\beta_r'(\tilde{h}) + \ell(h_1, W_r)\Big]}{\beta_r(h)} \tag{B.104}$$

$$\text{with} \quad \beta_r(h) = \sum_{h_1^N} p^{\kappa}(h_1^N|h) \cdot p_{\theta}^{\kappa}(x_{r\,\tau(h_1)}^{T_r}|h_1^N)$$

Here, $\sum_{\tilde{h}}$ runs over all successors $\tilde{h}$ of h.

B.3 Detailed Calculations for Chapter 9

This section contains detailed calculations for results used in Chapter 9: "Discriminative Linear Transformations".

B.3.1 The MLLT from a Likelihood Perspective

The following description is a formal derivation of the MLLT objective function from a maximum likelihood perspective. The derivation follows [Gopinath 98]. To emphasize the key issues, the MLLT is first presented for a one class problem.

Let x_1^N denote the training data. Using a Gaussian distribution with mean μ and variance $\mathbf{\Sigma}$, the likelihood of the training samples is given by Eq. (B.105) (a proof is shown at the end of this

section):

$$\begin{aligned}
\mathcal{L}(x_1^N) &= \prod_{n=1}^{N} \mathcal{N}(x_n|\mu,\mathbf{\Sigma}) \\
&= \prod_{n=1}^{N} \frac{1}{\sqrt{\det(2\pi\mathbf{\Sigma})}} \cdot \exp\left\{ -\frac{1}{2}(x_n-\mu)^\top \mathbf{\Sigma}^{-1}(x_n-\mu)\right\} \\
&= (2\pi)^{-\frac{ND}{2}} \cdot \exp\left\{ -\frac{1}{2}\sum_{n=1}^{N}\Big[(x_n-\mu)^\top \mathbf{\Sigma}^{-1}(x_n-\mu) + \log|\mathbf{\Sigma}|\Big]\right\} \\
&= (2\pi)^{-\frac{ND}{2}} \cdot \exp\left\{ -\frac{1}{2}N\Big[(\bar{\mu}-\mu)^\top \mathbf{\Sigma}^{-1}(\bar{\mu}-\mu) + tr(\mathbf{\Sigma}^{-1}\bar{\mathbf{\Sigma}}) + \log|\mathbf{\Sigma}|\Big]\right\}
\end{aligned} \tag{B.105}$$

Here, $\bar{\mu}$ and $\bar{\mathbf{\Sigma}}$ denote the sample mean and covariance, respectively. Eq. (B.105) is maximized by the ML estimates $\widehat{\mu} = \bar{\mu}$ and $\widehat{\mathbf{\Sigma}} = \bar{\mathbf{\Sigma}}$. Hence, the likelihood value of the training data is

$$\begin{aligned}
\mathcal{L}(x_1^N) &= (2\pi)^{-\frac{ND}{2}} \cdot \exp\left\{ -\frac{1}{2}N\Big[(\bar{\mu}-\bar{\mu})^\top \bar{\mathbf{\Sigma}}^{-1}(\bar{\mu}-\bar{\mu}) + tr(\bar{\mathbf{\Sigma}}^{-1}\bar{\mathbf{\Sigma}}) + \log|\bar{\mathbf{\Sigma}}|\Big]\right\} \\
&= (2\pi)^{-\frac{ND}{2}} \cdot \exp\left\{ -\frac{1}{2}N\Big[\, tr(\mathbf{I}) \;+\; \log|\bar{\mathbf{\Sigma}}| \,\Big]\right\} \\
&= (2\pi e)^{-\frac{ND}{2}} \cdot |\,\bar{\mathbf{\Sigma}}\,|^{-\frac{N}{2}},
\end{aligned} \tag{B.106}$$

and on average each sample contributes $|\bar{\mathbf{\Sigma}}|^{-\frac{1}{2}}$ to the ML value $\mathcal{L}(x_1^N)$.

Now we consider a global linear feature transformation $\mathbf{A}$ of the training data. With $y = \mathbf{A}^\top x$ denoting the training samples in the transformed space, the ML values in the two feature spaces are related as:

$$\mathcal{L}(x_1^N) = \mathcal{L}(y_1^N) \cdot |\mathbf{A}| \tag{B.107}$$

To verify this relation, it suffices to show Eq. (B.107) for two normal distributions. The proposition then follows from Eq. (B.105). Let $y := \mathbf{A}^\top x$, $\mu_y := \mathbf{A}^\top \mu_x$, and $\mathbf{\Sigma}_y := \mathbf{A}^\top \mathbf{\Sigma}_x \mathbf{A}$. Then the following sequence of identities holds:

$$\begin{aligned}
\mathcal{N}(y\,|\,\mu_y,\mathbf{\Sigma}_y)\cdot|\mathbf{A}| &= \frac{|\mathbf{A}|}{\sqrt{|2\pi\mathbf{\Sigma}_y|}}\, e^{-\frac{1}{2}(y-\mu_y)^\top \mathbf{\Sigma}_y^{-1}(y-\mu_y)} \\
&= \frac{|\mathbf{A}|}{\sqrt{|2\pi\mathbf{A}^\top\mathbf{\Sigma}_x\mathbf{A}|}}\, e^{-\frac{1}{2}\left[\mathbf{A}^\top(x-\mu_x)\right]^\top\left(\mathbf{A}^\top\mathbf{\Sigma}_x\mathbf{A}\right)^{-1}\left[\mathbf{A}^\top(x-\mu_x)\right]} \\
&= \frac{|\mathbf{A}|}{|\mathbf{A}|\sqrt{|2\pi\mathbf{\Sigma}_x|}}\, e^{-\frac{1}{2}(x-\mu_x)^\top\mathbf{A}\mathbf{A}^{-1}\mathbf{\Sigma}_x^{-1}\mathbf{A}^{-\top}\mathbf{A}^\top(x-\mu_x)} \\
&= \mathcal{N}(x\,|\,\mu_x,\mathbf{\Sigma}_x)
\end{aligned} \tag{B.108}$$

With the results from Eqs. (B.105-B.107) we can now derive the actual MLLT objective function. For this purpose, we consider a special case where the covariances in the original space, $\mathbb{R}_x$, are modeled as full matrices while a diagonal modeling constraint is imposed on the covariances in the transformed space, $\mathbb{R}_y$. Due to the *Hadamard* inequality (cf. Eq. (9.4)), this constraint may cause a likelihood loss in the transformed space. Therefore, our goal is to find a feature space where the likelihood loss is as small as possible. In other words, we search for a feature space where the model assumption of diagonal covariances is most valid. In contrast to the former considerations,

we now investigate a multiple class problem. Under the assumption that each training sample is assigned with exactly one class, there is no interaction between the classes, and the ML estimates of the parameters for each Gaussian solely depend on the training data for that class. Again let x_1^N and y_1^N denote the training data in the two feature spaces $\mathbb{R}_x$ and $\mathbb{R}_y$. Furthermore, let m_k and $\mathbf{S}_k$ denote the mean and variance in feature space $\mathbb{R}_y$. Then the likelihood in the original feature space $\mathbb{R}_x$ is given by:

$$\mathcal{L}(x_1^N) = \prod_{n=1}^{N} \mathcal{N}(y_n | m_{k_n}, \mathbf{S}_{k_n}) \cdot |\mathbf{A}| \tag{B.109}$$

$$= \prod_{n=1}^{N} \frac{|\mathbf{A}|}{\sqrt{|2\pi \mathbf{S}_{k_n}|}} \cdot \exp\left\{ -\frac{1}{2}(y_n - m_{k_n})^\top \cdot \mathbf{S}_{k_n}^{-1} \cdot (y_n - m_{k_n}) \right\} \tag{B.110}$$

$$= \prod_{n=1}^{N} \frac{|\mathbf{A}|}{\sqrt{|2\pi \cdot \operatorname{diag}\big(\mathbf{A}^\top \mathbf{\Sigma}_{k_n} \mathbf{A}\big)|}} \cdot \exp\left\{ -\frac{1}{2}\big[\mathbf{A}^\top (x_n - \mu_{k_n})\big]^\top \cdot \operatorname{diag}\big(\mathbf{A}^\top \mathbf{\Sigma}_{k_n} \mathbf{A}\big)^{-1} \cdot \big[\mathbf{A}^\top (x_n - \mu_{k_n})\big] \right\} \tag{B.111}$$

$$= \prod_{k=1}^{K} \left(\frac{|\mathbf{A}|}{\sqrt{\big|\operatorname{diag}\big(\mathbf{A}^\top \mathbf{\Sigma}_k \mathbf{A}\big)\big|}} \right)^{N_k} \cdot (2\pi)^{-N_k D/2} \cdot \prod_{n_k=1}^{N_k} \exp\left\{ -\frac{1}{2}\big(x_{n_k} - \mu_k\big)^\top \mathbf{A} \operatorname{diag}\big(\mathbf{A}^\top \mathbf{\Sigma}_k \mathbf{A}\big)^{-1} \mathbf{A}^\top \big(x_{n_k} - \mu_k\big) \right\} \tag{B.112}$$

$$= \prod_{k=1}^{K} \left(\frac{|\mathbf{A}|}{\sqrt{\big|\operatorname{diag}\big(\mathbf{A}^\top \mathbf{\Sigma}_k \mathbf{A}\big)\big|}} \right)^{N_k} \cdot (2\pi)^{-N_k D/2} \cdot \exp\left\{ -\frac{1}{2} N_k \underbrace{\Big[\underbrace{(\mu_k - \mu_k)^\top \mathbf{A} \operatorname{diag}\big(\mathbf{A}^\top \mathbf{\Sigma}_k \mathbf{A}\big)^{-1} \mathbf{A}^\top (\mu_k - \mu_k)}_{=0} + \underbrace{tr\big(\operatorname{diag}(\mathbf{A}^\top \mathbf{\Sigma}_k \mathbf{A})^{-1} \cdot \mathbf{S}_k\big)}_{=tr(\mathbf{I})} \Big]}_{=D} \right\} \tag{B.113}$$

$$= \prod_{k=1}^{K} \left(\frac{|\mathbf{A}|}{\sqrt{\big|\operatorname{diag}\big(\mathbf{A}^\top \mathbf{\Sigma}_k \mathbf{A}\big)\big|}} \right)^{N_k} \cdot (2\pi e)^{-\frac{N_k D}{2}} \tag{B.114}$$

$$= (2\pi e)^{-\frac{ND}{2}} \cdot |\mathbf{A}|^N \cdot \prod_{k=1}^{K} \left(\frac{1}{\sqrt{\big|\operatorname{diag}\big(\mathbf{A}^\top \mathbf{\Sigma}_k \mathbf{A}\big)\big|}} \right)^{N_k} \tag{B.115}$$

Thus we finally obtain:

$$\mathcal{L}(x_1^N) = (2\pi e)^{-\frac{ND}{2}} \cdot |\mathbf{A}|^N \cdot \prod_{k=1}^{K} \left(\frac{1}{\sqrt{\big|\operatorname{diag}(\mathbf{A}^\top \mathbf{\Sigma}_k \mathbf{A})\big|}} \right)^{N_k}$$

Proof of Eq. (B.105)

With the definition of the sample mean $\bar{\mu} = \frac{1}{N}\sum_n x_n$ and the sample covariance matrix $\bar{\mathbf{\Sigma}} = \left(\frac{1}{N}\sum_n x_n\, x_n^\top\right) - \bar{\mu}\,\bar{\mu}^\top$, Eq. (B.105) results from the following identities:

$$\sum_n (x_n-\mu)^\top \mathbf{\Sigma}^{-1}(x_n-\mu) = \Big(\sum_n x_n^\top \mathbf{\Sigma}^{-1} x_n\Big) \;-\; 2N\bar{\mu}^\top\mathbf{\Sigma}^{-1}\mu \;+\; N\mu^\top\mathbf{\Sigma}^{-1}\mu \tag{B.116}$$

$$= N\cdot\Bigg[\underbrace{\frac{1}{N}\Big(\sum_n x_n^\top\mathbf{\Sigma}^{-1}x_n\Big)}_{(*)} \;-\; 2\bar{\mu}^\top\mathbf{\Sigma}^{-1}\mu \;+\; \mu^\top\mathbf{\Sigma}^{-1}\mu\Bigg] \tag{B.117}$$

$$= N\cdot\Bigg[\underbrace{tr\big(\mathbf{\Sigma}^{-1}\bar{\mathbf{\Sigma}}\big) \;+\; \bar{\mu}^\top\mathbf{\Sigma}^{-1}\bar{\mu}}_{\equiv(*)} \;-\; 2\bar{\mu}^\top\mathbf{\Sigma}^{-1}\mu \;+\; \mu^\top\mathbf{\Sigma}^{-1}\mu\Bigg] \tag{B.118}$$

$$= N\cdot\Big[(\bar{\mu}-\mu)^\top\mathbf{\Sigma}^{-1}(\bar{\mu}-\mu) \;+\; tr\big(\mathbf{\Sigma}^{-1}\bar{\mathbf{\Sigma}}\big)\Big] \tag{B.119}$$

To complete the proof it remains to show that the underbraced term in Eq. (B.118) is identical to $(*)$:

$$tr\big(\mathbf{\Sigma}^{-1}\bar{\mathbf{\Sigma}}\big) = tr\Bigg(\mathbf{\Sigma}^{-1}\Bigg[\frac{1}{N}\sum_n x_n\, x_n^\top - \bar{\mu}\,\bar{\mu}^\top\Bigg]\Bigg) \tag{B.120}$$

$$= \frac{1}{N}\sum_n tr\big(\mathbf{\Sigma}^{-1}x_n\, x_n^\top\big) \;-\; tr\big(\mathbf{\Sigma}^{-1}\bar{\mu}\,\bar{\mu}^\top\big) \tag{B.121}$$

$$= \frac{1}{N}\sum_n tr\Big(\underbrace{\mathbf{\Sigma}^{-\frac{1}{2}}}_{A}\,\underbrace{\mathbf{\Sigma}^{-\frac{1}{2}}}_{B}\,\underbrace{x_n\, x_n^\top}_{C}\Big) \;-\; tr\Big(\underbrace{\mathbf{\Sigma}^{-\frac{1}{2}}}_{A'}\,\underbrace{\mathbf{\Sigma}^{-\frac{1}{2}}}_{B'}\,\underbrace{\bar{\mu}\,\bar{\mu}^\top}_{C'}\Big) \tag{B.122}$$

Using the identity $tr(ABC) \equiv tr(BCA)$ it follows

$$= \frac{1}{N}\sum_n tr\Big(\underbrace{\mathbf{\Sigma}^{-\frac{1}{2}}x_n}_{a}\cdot\underbrace{x_n^\top\mathbf{\Sigma}^{-\frac{1}{2}}}_{a^\top}\Big) \;-\; tr\Big(\underbrace{\mathbf{\Sigma}^{-\frac{1}{2}}\bar{\mu}}_{b}\,\underbrace{\bar{\mu}^\top\mathbf{\Sigma}^{-\frac{1}{2}}}_{b^\top}\Big) \tag{B.123}$$

Finally, applying $tr(a\cdot a^\top) \equiv a^\top\cdot a$ gives

$$= \frac{1}{N}\sum_n \big(x_n^\top\mathbf{\Sigma}^{-1}x_n\big) \;-\; \bar{\mu}^\top\mathbf{\Sigma}^{-1}\bar{\mu} \tag{B.124}$$

Now adding $\bar{\mu}^\top\mathbf{\Sigma}^{-1}\bar{\mu}$ to the left side of Eq. (B.120) and to the right side of Eq. (B.124) yields $(*)$. □

B.3.2 Formal Derivative of the MLLT Objective Function

The derivative of $\mathcal{F}_{\text{MLLT}}$ with respect to $\mathbf{A}$ yields:

$$\frac{\partial\mathcal{F}_{\text{MLLT}}}{\partial\mathbf{A}} = N\cdot\frac{\mathbf{A}^{-\top}}{|\mathbf{A}|}\cdot|\mathbf{A}| - \sum_{k=1}^{K}\frac{N_k}{2}\cdot\frac{\partial\log\big|\text{diag}(\mathbf{A}^\top\mathbf{\Sigma}\mathbf{A})\big|}{\partial\mathbf{A}}. \tag{B.125}$$

Due to the following identities

$$\frac{\partial\log|\mathbf{A}^\top\mathbf{\Sigma}\mathbf{A}|}{\partial\mathbf{A}} = \frac{1}{|\mathbf{A}^\top\mathbf{\Sigma}\mathbf{A}|}\cdot|\mathbf{A}^\top\mathbf{\Sigma}\mathbf{A}|\cdot(\mathbf{\Sigma}+\mathbf{\Sigma}^\top)\cdot\mathbf{A}\cdot(\mathbf{A}^\top\mathbf{\Sigma}\mathbf{A})^{-1} \tag{B.126}$$

$$= 2\cdot\mathbf{\Sigma}\cdot\mathbf{A}\cdot(\mathbf{A}^\top\mathbf{\Sigma}\mathbf{A})^{-1} \tag{B.127}$$

the derivative of $\log\left|\operatorname{diag}(\mathbf{A}^\top\boldsymbol{\Sigma}\mathbf{A})\right|$ with respect to $\mathbf{A}$ can be transformed into:

$$\frac{\partial \log\left|\operatorname{diag}(\mathbf{A}^\top\boldsymbol{\Sigma}\mathbf{A})\right|}{\partial\mathbf{A}} = \frac{1}{\left|\operatorname{diag}(\mathbf{A}^\top\boldsymbol{\Sigma}\mathbf{A})\right|}\cdot\left|\operatorname{diag}(\mathbf{A}^\top\boldsymbol{\Sigma}\mathbf{A})\right|\cdot(\boldsymbol{\Sigma}+\boldsymbol{\Sigma}^\top)\cdot\mathbf{A}\cdot\operatorname{diag}(\mathbf{A}^\top\boldsymbol{\Sigma}\mathbf{A})^{-1} \tag{B.128}$$

$$= 2\cdot\boldsymbol{\Sigma}\cdot\mathbf{A}\cdot\operatorname{diag}(\mathbf{A}^\top\boldsymbol{\Sigma}\mathbf{A})^{-1} \tag{B.129}$$

This finally gives:

$$\frac{\partial\mathcal{F}_{\text{MLLT}}}{\partial\mathbf{A}} = N\cdot\mathbf{A}^{-\top} - \sum_{k=1}^{K} N_k\cdot\underbrace{\boldsymbol{\Sigma}_k\cdot\mathbf{A}\cdot\operatorname{diag}(\mathbf{A}^\top\boldsymbol{\Sigma}_k\mathbf{A})^{-1}}_{(*)} \stackrel{!}{=} 0 \tag{B.130}$$

B.3.3 Detailed Calculations for the Robust Optimization Used in Section 9.2

Eq. (9.16) can be rewritten as follows:

$$\left(\frac{\partial\mathcal{Q}_{\text{MLLT}}(\mathbf{A},\theta)}{\partial\mathbf{A}}\right)_d = \sum_{s=1}^{S}\sum_{l=1}^{L_s}\sum_{r=1}^{R}\sum_{t=1}^{T_r}\gamma_{rt}(s,l|W_r)\cdot\left(\frac{c_d}{a_d^\top\cdot c_d} - \boldsymbol{\Sigma}_{sl}\cdot a_d\cdot\underbrace{\left(a_d^\top\boldsymbol{\Sigma}_{sl}a_d\right)^{-1}}_{=:\sigma_{sld}^2}\right) \tag{B.131}$$

$$\stackrel{!}{=} 0 \tag{B.132}$$

$$\iff \quad T\cdot\frac{c_d}{a_d^\top\cdot c_d} = \sum_{s=1}^{S}\sum_{l=1}^{L_s}\sum_{r=1}^{R}\sum_{t=1}^{T_r}\frac{\gamma_{rt}(s,l|W_r)}{\sigma_{sld}^2}\cdot\boldsymbol{\Sigma}_{sl}\cdot a_d \tag{B.133}$$

$$\iff \quad T\cdot\frac{c_d}{a_d^\top\cdot c_d} = \mathbf{G}^{(d)}\cdot a_d \qquad \text{with } \mathbf{G}^{(d)} = \sum_{s=1}^{S}\sum_{l=1}^{L_s}\sum_{r=1}^{R}\sum_{t=1}^{T_r}\frac{\gamma_{rt}(s,l|W_r)}{\sigma_{sld}^2}\cdot\boldsymbol{\Sigma}_{sl} \tag{B.134}$$

$$\iff \quad \lambda T c_d = \mathbf{G}^{(d)}\cdot a_d \qquad \text{with } \lambda = \frac{1}{a_d^\top\cdot c_d} \tag{B.135}$$

$$\iff \quad \lambda T\mathbf{G}^{(d)^{-1}}\cdot c_d = a_d^\top \tag{B.136}$$

Plugging Eq. (B.136) back into Eq. (B.134) gives

$$Tc_d = \mathbf{G}^{(d)}\cdot a_d\cdot(a_d^\top\cdot c_d) \tag{B.137}$$

$$= \mathbf{G}^{(d)}\cdot\lambda T\cdot\mathbf{G}^{(d)^{-1}}\cdot c_d\cdot\left(\left[\lambda T\cdot\mathbf{G}^{(d)^{-1}}\cdot c_d\right]^\top\cdot c_d\right) \tag{B.138}$$

$$= \lambda T\cdot\mathbf{G}^{(d)}\cdot\mathbf{G}^{(d)^{-1}}\cdot c_d\cdot c_d^\top\cdot\lambda T\cdot\mathbf{G}^{(d)^{-1}}\cdot c_d \tag{B.139}$$

$$= \lambda T\cdot c_d\cdot c_d^\top\cdot\lambda T\cdot\mathbf{G}^{(d)^{-1}}\cdot c_d \tag{B.140}$$

$$= \alpha^2\cdot c_d\cdot c_d^\top\cdot\mathbf{G}^{(d)^{-1}}\cdot c_d \qquad \text{with } \alpha := \lambda T \tag{B.141}$$

Finally we obtain:

$$\alpha^2\cdot c_d\cdot c_d^\top\cdot\mathbf{G}^{(d)^{-1}}\cdot c_d - T\cdot c_d = 0 \tag{B.142}$$

$$\iff \quad \alpha^2\cdot c_d^\top\cdot\mathbf{G}^{(d)^{-1}}\cdot c_d - T = 0 \tag{B.143}$$

which leads to the root

$$\alpha_0 = \sqrt{\frac{T}{c_d^\top\cdot\mathbf{G}^{(d)^{-1}}\cdot c_d}} \tag{B.144}$$

The transformation columns of the MLLT matrix are thus given by:

$$a_d = \alpha_0 \cdot \mathbf{G}^{(d)^{-1}} \cdot c_d \tag{B.145}$$

$$= \sqrt{\frac{T}{c_d^\top \cdot \mathbf{G}^{(d)^{-1}} \cdot c_d}} \cdot \mathbf{G}^{(d)^{-1}} \cdot c_d \tag{B.146}$$

Bibliography

[Alleva & Huang+ 96] P. Alleva, X.D. Huang, M.Y. Hwang: Improvements on the Pronunciation Prefix Tree Search Organization. Proc. *1996 Int. Conf. on Acoustics, Speech and Signal Processing*, Vol. 1, pp. 133–136, Atlanta, GA, May 1996.

[Axelrod & Goel+ 04] S. Axelrod, V. Goel, R. Gopinath, P. Olsen, K. Visweswariah: Discriminative Estimation of Subspace Constrained Gaussian Mixture Models for Speech Recognition. *IEEE Trans. on Speech and Audio Processing*, Vol. 12, March 2004.

[Bahl & Brown+ 86] L.R. Bahl, P.F. Brown, P.V. de Souza, R.L. Mercer: Maximum Mutual Information Estimation of Hidden Markov Model Parameters for Speech Recognition. Proc. *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Vol. 1, pp. 49–52, Tokyo, Japan, May 1986.

[Bahl & Brown+ 88] L.R. Bahl, P.F. Brown, P.V. de Souza, R.L. Mercer: A New Algorithm for the Estimation of Hidden Markov Model Parameters. Proc. *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, pp. 493–496, New York, NY, April 1988.

[Bahl & Jelinek+ 83] L.R. Bahl, F. Jelinek, R.L. Mercer: A Maximum Likelihood Approach to Continuous Speech Recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 5, pp. 179–190, March 1983.

[Baker 75] J.K. Baker: Stochastic Modeling for Automatic Speech Understanding. In D.R. Reddy, editor, *Speech Recognition*, pp. 512–542. Academic Press, New York, NY, USA, 1975.

[Bakis 76] R. Bakis: Continuous Speech Word Recognition via Centisecond Acoustic States. Proc. *Proc. ASA Meeting*, Washington, DC, April 1976.

[Bauer 97] J.G. Bauer: Enhanced Control and Estimation of Parameters for a Telephone Based Isolated Digit Recognizer. Proc. *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Vol. 2, pp. 1531–1534, Munich, Germany, April 1997.

[Bauer 98] J. Bauer: Application of Discriminative Methods for Isolated Word Recognition. In I. Balderjahn, R. Mathar, M. Schader, editors, *Classification, Data Analysis, and Data Highways*, pp. 287–294. Springer-Verlag, Berlin - Heidelberg, 1998.

[Baum 72] L.E. Baum: An Inequality and Associated Maximization Technique in Statistical Estimation for Probabilistic Functions of Markov Processes. In O. Shisha, editor, *Inequalities*, Vol. 3, pp. 1–8. Academic Press, New York, NY, 1972.

[Baum & Eagon 67] L.E. Baum, 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*, Vol. 73, pp. 360–363, 1967.

[Baum & Petrie+ 70] L.E. Baum, T. Petrie, G. Soules, N. Weiss: A Maximization Technique Occuring in the Statistical Analysis of Probabilistic Functions of Markov Chains. *Annals of Mathematical Statistics*, Vol. 41, pp. 164–171, 1970.

[Baum & Sell 68] L.E. Baum, G. Sell: Growth Transformations for Functions on Manifolds. *Pacific Journal of Mathematics*, Vol. 27, pp. 211–227, 1968.

[Bender & Macherey$^+$ 03] O. Bender, K. Macherey, F.J. Och, H. Ney: Comparison of Alignment Templates and Maximum Entropy Models for Natural Language Understanding. Proc. *Proc. of the 10th Conf. of the Europ. Chapter of the Assoc. for Computational Linguistics*, pp. 148–152, Budapest, Hungary, April 2003.

[Bentley & Ottmann 79] J.L. Bentley, T.A. Ottmann: Algorithms for Reporting and Counting Geometric Intersections. *IEEE Trans. on Computers*, Vol. C-28, No. 9, pp. 643–647, 1979.

[Beulen & Ortmanns$^+$ 99] K. Beulen, S. Ortmanns, C. Elting: Dynamic Programming Search Techniques for Across-Word Modeling in Speech Recognition. Proc. *International Conference on Acoustics, Speech, and Signal Processing*, pp. 609–612, Phoenix, AZ, USA, March 1999.

[Beyerlein 98] P. Beyerlein: Discriminative Model Combination. Proc. *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Vol. 1, pp. 481–484, Seattle, WA, USA, May 1998.

[Bhattacharyya 46] A. Bhattacharyya: On a measure of divergence between two multinomial populations. *Sankhya, Series A*, Vol. 7, No. 4, pp. 401–406, 1946.

[Biem & Katagiri 93] A. Biem, S. Katagiri: Feature Extraction Based on Minimum Classification Error/Generalized Probabilistic Descent Method. Proc. *IEEE International Conference on Acoustics, Speech and Signal Processing*, Vol. 2, pp. 275–278, Minneapolis, Minnesota, USA, April 1993.

[Bisani & Ney 04] M. Bisani, H. Ney: Boostrap Estimates for Confidence Intervals in ASR Performance Evaluation. Proc. *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Vol. 1, pp. 409–412, Montreal, May 2004.

[Bishop 95] C.M. Bishop: *Neural Networks for Pattern Recognition*, chapter The Multilayer Perceptron, pp. 116–160. Clarendon Press, Oxford, 1995.

[Brants & Popat$^+$ 07] T. Brants, A. Popat, P. Xu, F. Och, J. Dean: Large Language Models in Machine Tranlation. Proc. *Proceedings of the 2007 Conference on Empirical Methods in Natural Language Processing*, N.N., Prague, Czech Republic, June 2007. Association for Computational Linguistics.

[Bridle & Brown$^+$ 82] J. Bridle, M. Brown, R. Chamberlain: An Algorithm for Connected Word Recognition. Proc. *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 899–902, Paris, France, 1982.

[Cardin & Normandin$^+$ 93] R. Cardin, Y. Normandin, E. Millien: Inter-Word Coarticulation Modeling and MMIE Training for Improved Connected Digit Recognition. Proc. *IEEE International Conference on Acoustics, Speech and Signal Processing*, Vol. 2, pp. 243–246, Minneapolis, MN, USA, April 1993.

[Chernoff 52] H. Chernoff: A Measure of Asymptotic Efficiency of Tests of a Hypothesis Based on the Sum of Observations. *Annals of Mathematical Statistics*, Vol. 23, pp. 493–507, 1952.

[Chou & Juang$^+$ 92] W. Chou, B.H. Juang, C.H. Lee: Segmental GPD Training of HMM Based Speech Recognizer. Proc. *1992 Int. Conf. on Acoustics, Speech and Signal Processing*, Vol. 1, pp. 473–476, San Francisco, CA, March 1992.

[Chou & Lee+ 93] W. Chou, C.H. Lee, B.H. Juang: Minimum Error Rate Training based on N-Best String Models. Proc. *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Vol. 2, pp. 652–655, Minneapolis, Minnesota, USA, April 1993.

[Chou & Lee+ 94] W. Chou, C.H. Lee, B.H. Juang: Minimum Error Rate Training of Inter-Word Context Dependent Acoustic Model Units in Speech Recognition. Proc. *1994 Int. Conf. on Speech and Language Processing*, Vol. 2, pp. 439–442, Yokohama, Japan, September 1994.

[Chow 90] Y.L. Chow: Maximum Mutual Information Estimation of HMM Parameters for Continuous Speech Recognition Using the N-best Algorithm. Proc. *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, pp. 701–704, Albuquerque, NM, April 1990.

[Darroch & Ratcliff 72] J.N. Darroch, D. Ratcliff: Generalized iterative scaling for log-linear models. *Annals of Mathematical Statistics*, Vol. 43, No. 5, pp. 1470–1480, 1972.

[Davis & Mermelstein 80] S.B. Davis, P. Mermelstein: Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. 28, No. 4, pp. 357–366, August 1980.

[Dempster & Laird+ 77] A.P. Dempster, N.M. Laird, D.B. Rubin: Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal Royal Statistical Society*, Vol. 39, No. 1, pp. 1–38, 1977.

[Duda & Hart 73] R.O. Duda, P.E. Hart: *Pattern Classification and Scene Analysis*. John Wiley & Sons, New York, NY, USA, 1973.

[Efron 79] B. Efron: Bootstrap Methods: Another Look at the Jackknife. *Annals of Mathematical Statistics*, Vol. 7, pp. 1–26, 1979.

[Eisele & Haeb-Umbach+ 96] T. Eisele, R. Haeb-Umbach, D. Langmann: A Comparative Study of Linear Feature Transformation Techniques for Automatic Speech Recognition. Proc. *Fourth International Conference on Spoken Language Processing*, Vol. 1, pp. 252–255, Philadelphia, PA, USA, Oktober 1996.

[Elizalde & Woods 06] S. Elizalde, K. Woods: Bounds on the Number of Inference Functions of a Graphical Model, pp. 1–19, Oct. 2006. `arXiv:math/0610233v1`.

[Gales 97] M.J.F. Gales: Semi-Tied Full-Covariance Matrices for Hidden Markov Models. Technical Report CUED/F-INFENG/TR287, University of Cambridge, April 1997. (ftp://svr-ftp.eng.cam.ac.uk/pub/reports/gales_tr298.ps.gz).

[Gales 98] M.J.F. Gales: Maximum Likelihood Linear Transformations for HMM-based Speech Recognition. *Computer Speech and Language*, Vol. 12, No. 2, pp. 75–98, April 1998.

[Garofolo & Fiscus+ 97] J.S. Garofolo, J.G. Fiscus, W.M. Fisher: Design and Preparation of the 1996 HUB-4 Broadcast News Benchmark Test Corpora. Proc. *Speech Recognition Workshop*, pp. 15–21, San Francisco, CA, February 1997.

[Goel & Axelrod+ 03] V. Goel, S. Axelrod, R.A. Gopinath, P. Olsen, K. Visveswariah: Discriminative estimation of subspace precision and mean (SPAM) models. Proc. *Europ. Conf. on Speech Communication and Technology*, Vol. 4, pp. 2617–2620, Geneva, Switzerland, Sep. 2003.

[Goel & Kumar+ 04] V. Goel, S. Kumar, W. Byrne: Segmental minimum Bayes-risk decoding for automatic speech recognition. *IEEE Transactions on Speech and Audio Processing*, Vol. 12, No. 3, pp. 234–249, May 2004.

[Gopalakrishnan & Kanevsky$^+$ 89] P.S. Gopalakrishnan, D. Kanevsky, A. Nádas, D. Nahamoo: A Generalization of the Baum Algorithm to Rational Objective Functions. Proc. *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Vol. 1, pp. 631–634, Glasgow, GB, May 1989.

[Gopalakrishnan & Kanevsky$^+$ 91] P.S. Gopalakrishnan, D. Kanevsky, A. Nádas, D. Nahamoo: An Inequality for Rational Functions with Applications to Some Statistical Estimation Problems. *IEEE Transactions on Information Theory*, Vol. 37, No. 1, pp. 107–113, January 1991.

[Gopinath 98] R.A. Gopinath: Maximum Likelihood Modeling with Gaussian Distributions for Classification. Proc. *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Vol. 2, pp. 661–664, Seattle, WA, May 1998.

[Gunawardana 01] A. Gunawardana: Maximum Mutual Information Estimation of Acoustic HMM Emission Densities. Technical Report CLSP Research Note No. 40, Center for Language and Speech Processing, Johns Hopkins University, Baltimore, MD, USA, pp. 1–6, June 2001.

[Gunawardana & Byrne 01] A. Gunawardana, W. Byrne: Discriminative speaker adaptation with conditional maximum likelihood linear regression. Proc. *Europ. Conf. on Speech Communication and Technology*, Vol. 2, pp. 1203–1206, Aalborg, Denmark, September 2001.

[Haeb-Umbach & Ney 92a] R. Haeb-Umbach, H. Ney: Linear Discriminant Analysis for Improved Large Vocabulary Continuous Speech Recognition. Proc. *IEEE International Conference on Acoustics, Speech and Signal Processing*, Vol. 1, pp. 13–16, San Francisco, CA, USA, March 1992.

[Haeb-Umbach & Ney 92b] R. Haeb-Umbach, H. Ney: Linear Discriminant Analysis for Improved Large Vocabulary Continuous Speech Recognition. Proc. *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Vol. 1, pp. 13–16, San Francisco, CA, USA, March 1992.

[Hamming 89] R.W. Hamming: *Digital Filters*. Dover Books on Engineering. Dover Publications, Inc., Mineola, NY, USA, 3rd edition, 1989.

[Heigold & Deselaers$^+$ 08a] G. Heigold, T. Deselaers, R. Schlüter, H. Ney: GIS-like Estimation of Log-Linear Models with Hidden Variables. Proc. *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, pp. 4045–4048, Las Vegas, NV, USA, April 2008.

[Heigold & Deselaers$^+$ 08b] G. Heigold, T. Deselaers, R. Schlüter, H. Ney: Modified MMI/MPE: A Direct Evaluation of the Margin in Speech Recognition. Proc. *International Conference on Machine Learning (ICML)*, pp. 384–391, Helsinki, Finland, July 2008.

[Heigold & Macherey$^+$ 05] G. Heigold, W. Macherey, R. Schlüter, H. Ney: Minimum Exact Word Error Training. Proc. *Automatic Speech Recognition and Understanding*, pp. 186–190, San Juan, Puerto Rico, November 2005.

[Heigold & Schlüter$^+$ 07] G. Heigold, R. Schlüter, H. Ney: On the Equivalence of Gaussian HMM and Gaussian HMM-like Hidden Conditional Random Fields. Proc. *Interspeech*, pp. 1721–1724, Antwerp, Belgium, August 2007.

[Hermansky 89] H. Hermansky: Perceptual Linear Predictive (PLP) Analysis of Speech. *Journal Acoustical Society of America*, Vol. 87, No. 4, pp. 1738–1752, April 1989.

[Huang & Goel$^+$ 02] J. Huang, V. Goel, R. Gopinath, B. Kingsbury, P. Olsen, K. Visweswariah: Large Vocabulary Conversational Speech Recognition with the Extended Maximum Likelihood Linear Transformation (EMLLT) Model. Proc. *Int. Conf. on Spoken Language Processing*, pp. 2597–2600, Denver, CO, Sept. 2002.

[Huang & Jack 89] X.D. Huang, M.A. Jack: Semi-Continuous Hidden Markov Models for Speech Signals. *Computer Speech and Language*, Vol. 3, No. 3, pp. 329–252, 1989.

[Jeffreys 46] H. Jeffreys: An Invariant Form for the Prior Probabilities in Estimation Problems. *Proceedings of the Royal Society of London, Series A*, Vol. 186, pp. 453–461, 1946.

[Jelinek 76] F. Jelinek: Continous Speech Recognition by Statistical Method. *Proceedings of the IEEE*, Vol. 64, No. 4, pp. 532–557, April 1976.

[Juang & Katagiri 92] B.H. Juang, S. Katagiri: Discriminative Learning for Minimum Error Classification. *IEEE Transactions on Signal Processing*, Vol. 40, No. 12, pp. 3043–3054, December 1992.

[Kaiser & Horvat$^+$ 00] J. Kaiser, B. Horvat, Z. Kačič: A Novel Loss Function for the Overall Risk Criterion Based Discriminative Training of HMM Models. Proc. *Int. Conf. on Spoken Language Processing*, Vol. 2, pp. 887–890, Beijing, China, Oct. 2000.

[Kaiser & Horvat$^+$ 02] J. Kaiser, B. Horvat, Z. Kačič: Overall Risk Criterion Estimation of Hidden Markov Model Parameters. *Speech Communication*, Vol. 38, No. 3–4, pp. 365–382, Nov. 2002.

[Kanevsky 95] D. Kanevsky: A Generalization of the Baum Algorithm to Functions on Non-Linear Manifolds. Proc. *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Vol. 1, pp. 473–476, Detroit, MI, May 1995.

[Kanevsky 03] D. Kanevsky: Extended Baum Transformations for General Functions. Technical Report RC22919 (W0309-163), IBM Research Devision, Thomas J. Watson Research Center, Yorktown Heights, NY, USA, Sept. 2003. (http://domino.research.ibm.com/library/cyberdig.nsf/).

[Kanevsky 04] D. Kanevsky: Extended Baum Transformations for General Functions. Proc. *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Vol. 1, pp. 821–824, Montreal, Quebec, Canada, May 2004.

[Kanevsky 05] D. Kanevsky: Extended Baum Transformations for General Functions, II. Technical Report RC23645 (W0506-120), IBM Research Devision, Thomas J. Watson Research Center, Yorktown Heights, NY, USA, June 2005. (http://domino.research.ibm.com/library/cyberdig.nsf/).

[Kapadia 98] S. Kapadia: *Discriminative Training of Hidden Markov Models*. Ph.d. thesis, Cambridge University Engineering Department, Cambrdige, UK, 1998.

[Kingsbury & Rayner 71] N.G. Kingsbury, P.J.W. Rayner: Digital Filtering Using Logarithmic Arithmetic. *Electronics Letters*, Vol. 7, pp. 56–58, April 1971.

[Koehn 04] P. Koehn: Statistical Significance Tests for Machine Translation Evaluation. Proc. *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pp. 388–395, Barcelona, Spain, Aug. 2004. Association for Computational Linguistics.

[Kullback & Leibler 51] S. Kullback, A. Leibler: On Information and Sufficiency. *Annals of Mathematical Statistics*, Vol. 22, pp. 79–86, 1951.

[Kumar & Andreou 98a] N. Kumar, A.G. Andreou: Heteroscedastic Discriminant Analysis And Reduced Rank HMMs For Improved Speech Recognition. *Speech Communication*, Vol. 26, No. 4, pp. 283–297, Dec. 1998.

[Kumar & Andreou 98b] N. Kumar, A.G. Andreou: Heteroscedastic Discriminant Analysis And Reduced Rank HMMs For Improved Speech Recognition. *Speech Communication*, Vol. 26, No. 4, pp. 283–297, Dec. 1998.

[Kumar & Byrne 02] S. Kumar, W. Byrne: Risk Based Lattice Cutting for Segmental Minimum Bayes Risk Decoding. Proc. *Int. Conf. on Spoken Language Processing*, Denver, CO, Sept. 2002.

[Kuo & Gao 03] H.K.J. Kuo, Y. Gao: Maximum Entropy Direct Models for Speech Recognition. Proc. *Automatic Speech Recognition and Understanding Workshop*, St. Thomas, Virgin Islands, December 2003.

[Lee & Mahajan 89] K.F. Lee, S. Mahajan: Corrective and Reinforcement Learning for Speaker-Independent Continuous Speech Recognition. Proc. *ISCA Europ. Conf. on Speech Communication and Technology*, Vol. 1, pp. 490–493, Paris, France, Sept. 1989.

[Levenshtein 65] V.I. Levenshtein: Binary Codes Capable of Correcting Deletions. *Doklady Akademii nauk SSSR (in Russian)*, Vol. 163, No. 4, pp. 845–848, 1965. Also in Cybernetics and Control Theory, 10(8), pp 707-710.

[Levinson & Rabiner$^+$ 83] S.E. Levinson, L.R. Rabiner, M.M. Sondhi: An Introduction to the Application of the Theory of Probabilistic Functions of a Markov Process to Automatic Speech Recognition. *Bell System Technical Journal*, Vol. 62, No. 4, pp. 1035–1074, April 1983.

[Likhododev & Gao 02] A. Likhododev, Y. Gao: Direct Models for Phoneme Recognition. Proc. *Int. Conference on Acoustics, Speech, and Signal Processing*, Vol. 1, pp. 89–92, Orlando, FL, May 2002.

[Liu & Gales 04] X. Liu, M.J.F. Gales: Model Complexity Control and Compression Using Discriminiative Growth Functions. Proc. *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Vol. 1, pp. 797–800, Montreal, Quebec, Canada, May 2004.

[Lowerre 76] B. Lowerre: *A Comparative Performance Analysis of Speech Understanding Systems*. Ph.d. thesis, Carnegie-Mellon University, Pittsburgh, PA, 1976.

[Macherey 98] W. Macherey: Implementation and Comparison of Discriminative Training Methods for Automatic Speech Recognition. Master's thesis, Chair of Computer Science VI, Computer Science Department, RWTH Aachen University, Aachen, Germany, Nov. 1998.

[Macherey & Bender$^+$ 09] K. Macherey, O. Bender, H. Ney: Applications of Statistical Machine Translation Approaches To Spoken Language Understanding. *IEEE Trans. on Audio, Speech, and Language Processing*, Vol. 17, No. 4, pp. 803–818, May 2009.

[Macherey & Haferkamp$^+$ 05] W. Macherey, L. Haferkamp, R. Schlüter, H. Ney: Investigations on Error Minimizing Training Criteria for Discriminative Training in Automatic Speech Recognition. Proc. *ISCA Europ. Conf. on Speech Communication and Technology*, pp. 2133–2136, Lisboa, Portugal, Sept. 2005.

[Macherey & Ney 02] W. Macherey, H. Ney: Towards Automatic Corpus Preparation for a German Broadcast News Transcription System. Proc. *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Vol. 1, pp. 733–736, Orlando, FL, USA, May 2002.

[Macherey & Och$^+$ 08] W. Macherey, F. Och, I. Thayer, J. Uszkoreit: Lattice-based Minimum Error Rate Training for Statistical Machine Translation. Proc. *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pp. 725–734, Honolulu, Hawaii, USA, Oct. 2008.

[Macherey & Schlüter+ 04] W. Macherey, R. Schlüter, H. Ney: Discriminative Training with Tied Covariance Matrices. Proc. *8th Int. Conf. on Spoken Language Processing*, Vol. 1, pp. 681–684, Jeju Island, Korea, October 2004.

[Macherey & Viechtbauer+ 02] W. Macherey, J. Viechtbauer, H. Ney: Probabilistic Retrieval Based On Document Representations. Proc. *Int. Conf. on Spoken Language Processing*, Vol. 3, pp. 1481–1484, Denver, CO, September 2002.

[Macherey & Viechtbauer+ 03] W. Macherey, J. Viechtbauer, H. Ney: Probabilistic Aspects in Spoken Document Retrieval. *EURASIP Journal on Applied Signal Processing*, Vol. 2003, pp. 115–127, Feb. 2003. "Special issue on "Unstructured Information Management from Multimedia Data Sources".

[Matushita 56] K. Matushita: Decision Rule, Based on Distance, for the Classification Problem. *Annals of the Institute of Statistical Mathematics*, Vol. 8, pp. 67–77, 1956.

[Matusov & Kanthak+ 05] E. Matusov, S. Kanthak, , H. Ney: On the Integration of Speech Recognition and Statistical Machine Translation. Proc. *ISCA Europ. Conf. on Speech Communication and Technology*, pp. 3177–3180, Lisboa, Portugal, Sept. 2005.

[McCallum & Freitag+ 00] A. McCallum, D. Freitag, F. Pereira: Maximum Entropy Markov Models for Information Extraction and Segmentation. Proc. *17th International Conf. on Machine Learning*, pp. 591–598. Morgan Kaufmann, San Francisco, CA, 2000.

[McDermott & Hazen 04] E. McDermott, T.J. Hazen: Minimum Classification Error Training of Landmark Models for Real-Time Continuous Speech Recognition. Proc. *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, Vol. 1, pp. 937–940, Montreal, Canada, May 2004.

[McDermott & Katagiri 97] E. McDermott, S. Katagiri: String-Level MCE for Continuous Phoneme Recognition. Proc. *1997 Europ. Conf. on Speech Communication and Technology*, Vol. 1, pp. 123–126, Rhodes, Greece, September 1997.

[McDermott & Katagiri 05] E. McDermott, S. Katagiri: Minimum classification error for large scale speech recognition tasks using weighted finite state transducers. Proc. *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, Vol. 1, pp. 113–116, Philadelphia, PA, March 2005.

[McDonough & Waibel 02] J. McDonough, A. Waibel: On Maximum Mutual Information Speaker-Adapted Training. Proc. *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Vol. 1, pp. 601–604, Orlando, FL, USA, May 2002.

[Merialdo 88] B. Merialdo: Phonetic Recognition Using Hidden Markov Models and Maximum Mutual Information Training. Proc. *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, pp. 111–114, New York, NY, April 1988.

[Na & Jeon+ 95] K. Na, B. Jeon, D. Chang, S. Chae, S. Ann: Discriminative Training of Hidden Markov Models Using Overall Risk Criterion and Reduced Gradient Method. Proc. *ISCA Europ. Conf. on Speech Communication and Technology*, Vol. 1, pp. 97–100, 1995.

[Nádas 83] A. Nádas: A Decision Theoretic Formulation of a Training Problem in Speech Recognition And a Comparison of Training by Unconditional Versus Conditional Maximum Likelihood. *IEEE Trans. on Acoustics, Speech, and Signal Processing*, Vol. 31, No. 4, pp. 814–817, 1983.

[Nádas & Nahamoo+ 88] A. Nádas, D. Nahamoo, M.A. Picheny: On a Model-Robust Training Method for Speech Recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. 36, No. 11, pp. 1432–1436, September 1988.

[Ney 84] H. Ney: The Use of a One-Stage Dynamic Programming Algorithm for Connected Word Recognition. *IEEE Trans. on Speech and Audio Processing*, Vol. 32, No. 2, pp. 263–271, April 1984.

[Ney 90] H. Ney: Acoustic Modeling of Phoneme Units for Continuous Speech Recognition. Proc. M.A.L. L. Torres, E. Masgrau, editor, *Signal Processing V: Theories and Applications, Fifth Europ. Signal Processing Conference*, pp. 65–72, Barcelona, Spain, September 1990. Elsevier Science Publishers B. V.

[Ney 03] H. Ney: On the Relationship between Classification Error Bounds and Training Criteria in Statistical Pattern Recognition. Proc. *1st Iberian Conference on Pattern Recognition and Image Analysis*, Vol. LNCS 2652 of Lecture Notes in Computer Science, pp. 636–645, Puerto de Andratx, Spain, June 2003. Springer-Verlag.

[Ney & Aubert 94] H. Ney, X. Aubert: A Word Graph Algorithm for Large Vocabulary Continuous Speech Recognition. Proc. *Int. Conf. on Spoken Language Processing*, Vol. 3, pp. 1355–1358, Yokohama, Japan, Sept. 1994.

[Ney & Aubert 96] H. Ney, X. Aubert: Dynamic Programming Search Strategies: From Digit Strings to Large Vocabulary Word Graphs. In C.H. Lee, F.K. Soong, K.K. Paliwal, editors, *Automatic Speech and Speaker Recognition Advanced Topics*, chapter 16, pp. 385–411. Kluwer Academic Publishers, Norwell, Massachusetts, USA, 1996.

[Ney & Mergel$^+$ 87] H. Ney, D. Mergel, A. Noll, A. Paeseler: A Data-Driven Organization of the Dynamic Programming Beam Search for Continuous Speech Recognition. Proc. *1987 Int. Conf. on Acoustics, Speech and Signal Processing*, pp. 833–836, Dallas, TX, April 1987.

[Noreen 89] E.W. Noreen: *Computer-Intensive Methods for Testing Hypotheses*. John Wiley & Sons, Canada, 1989.

[Normandin 91] Y. Normandin: *Hidden Markov Models, Maximum Mutual Information Estimation, and the Speech Recognition Problem*. Ph.D. thesis, Department of Electrical Engineering, McGill University, Montreal, Canada, June 1991.

[Normandin 95] Y. Normandin: Optimal Splitting of HMM Gaussian Mixture Components with MMIE Training. Proc. *1995 Int. Conf. on Acoustics, Speech and Signal Processing*, Vol. 1, pp. 449–452, Detroit, MI, May 1995.

[Normandin & Morgera 91] Y. Normandin, S.D. Morgera: An Improved MMIE Training Algorithm for Speaker-Independent, Small Vocabulary, Continuous Speech Recognition. Proc. *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Vol. 1, pp. 537–540, Toronto, Canada, May 1991.

[Och 03] F.J. Och: Minimum Error Rate Training in Statistical Machine Translation. Proc. *41st Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 160–167, Sapporo, Japan, July 2003.

[Och & Ney 02] F.J. Och, H. Ney: Discriminative Training and Maximum Entropy Models for Statistical Machine Translation. Proc. *40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 295–302, Philadelphia, PA, USA, July 2002.

[Odell & Valtchev$^+$ 94] J.J. Odell, V. Valtchev, P.C. Woodland, S.J. Young: A One-Pass Decoder Design for Large Vocabulary Recognition. Proc. *ARPA Spoken Language Technology Workshop*, pp. 405–410, Plainsboro, NJ, March 1994.

[Ortmanns & Ney 95] S. Ortmanns, H. Ney: An Experimental Study of the Search Space for 20000-Word Speech Recognition. Proc. *1995 Europ. Conf. on Speech Communication and Technology*, Vol. 2, pp. 901–904, Madrid, Spain, September 1995.

[Ortmanns & Ney$^+$ 96a] S. Ortmanns, H. Ney, A. Eiden: Language-Model Look-Ahead for Large Vocabulary Speech Recognition. Proc. *1996 Int. Conf. on Spoken Language Processing*, pp. 2095–2098, Philadelphia, PA, October 1996.

[Ortmanns & Ney$^+$ 96b] S. Ortmanns, H. Ney, A. Eiden: Language-Model Look-Ahead for Large Vocabulary Speech Recognition. Proc. *Int. Conf. on Spoken Language Processing*, Vol. 4, pp. 2095–2098, Philadelphia, PA, USA, Oct. 1996.

[Ortmanns & Ney$^+$ 97] S. Ortmanns, H. Ney, X. Aubert: A Word Graph Algorithm for Large Vocabulary Continuous Speech Recognition. *Computer Speech and Language*, Vol. 11, No. 1, pp. 43–72, Jan. 1997.

[Paliwal & Bacchiani$^+$ 95] K.K. Paliwal, M. Bacchiani, Y. Sagisaka: Minimum Classification Error Training Algorithm for Feature Extractor and Pattern Classifier in Speech Recognition. Proc. *ESCA 3rd European Conference On Speech Communication And Technology*, Vol. 1, pp. 541–544, Rhodes, Greece, September 1995.

[Papineni 99] K.A. Papineni: Discriminative Training via Linear Programming. Proc. *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Vol. 2, pp. 561–564, Phoenix, AZ, March 1999.

[Patil & Taillie 82] G.P. Patil, C. Taillie: Diversity as a Concept and its Measurement. *Journal of the American Statistical Association*, Vol. 77, pp. 548–561, 1982.

[Pietra & Pietra$^+$ 97] S.D. Pietra, V.D. Pietra, J. Lafferty: Inducing features of random fields. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 19, No. 4, pp. 380–393, April 1997.

[Povey 04] D. Povey: *Discriminative training for large vocabulary speech recognition.* Ph.D. dissertation, Dept. of Eng., Cambridge Univ., Cambridge, August 2004.

[Povey & Kanevsky$^+$ 08] D. Povey, D. Kanevsky, B. Kingsbury, B. Ramabhadran, G. Saon, K. Visweswariah: Boosted MMI for Model and Feature-Space Discriminative Training. Proc. *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, pp. 4057–4060, Las Vegas, NV, April 2008.

[Povey & Woodland 99] D. Povey, P.C. Woodland: Frame Discrimination Training of HMMs for Large Vocabulary Speech Recognition. Proc. *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Vol. 1, pp. 333–336, Phoenix, AZ, May 1999.

[Povey & Woodland 00] D. Povey, P.C. Woodland: An Investigation of Frame Discrimination for Continuous Speech Recognition. Technical Report CUED/F-INFENG/TR332, University of Cambridge, May 2000. (ftp://svr-ftp.eng.cam.ac.uk/pub/reports/povey_tr332.ps.gz).

[Povey & Woodland 02] D. Povey, P.C. Woodland: Minimum phone error and I-smoothing for improved discriminative training. Proc. *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Vol. 1, pp. 105–108, Orlando, FL, May 2002.

[Press & Teukolsky$^+$ 02] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery: *Numerical Recipes in C++.* Cambridge University Press, Cambridge, UK, second edition, 2002.

[Qiang & Juang 07] F. Qiang, B.H. Juang: Automatic Speech Recognition Based on Weighted Minimum Classification Error (W-MCE) Training Method. Proc. *Automatic Speech Recognition and Understanding*, pp. 278–283, Kyoto, Japan, December 2007.

[Rabiner 89] L.R. Rabiner: A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proceedings of the IEEE*, Vol. 77, No. 2, pp. 257–286, Feb. 1989.

[Rabiner & Juang 86] L. Rabiner, B.H. Juang: An Introduction to Hidden Markov Models. *IEEE Trans. on Speech and Audio Processing*, Vol. 3, No. 1, pp. 4–16, 1986.

[Rabiner & Juang 93] L.R. Rabiner, B.H. Juang: *Fundamentals of Speech Recognition*. Prentice Hall PTR, Englewood Cliffs, NJ, 1st edition, 1993.

[Radnaparkhi 96] A. Radnaparkhi: A maximum entropy model for part-of-speech tagging. Proc. *Proc. of the Conf. on Empirical Methods in Natural Language Processing*, pp. 133–142, Univ. of Pennsylvania, May 1996.

[Reichl & Ruske 95] W. Reichl, G. Ruske: Discriminative Training for Continuous Speech Recognition. Proc. *ESCA 3rd European Conference On Speech Communication And Technology*, Vol. 1, pp. 537–540, Madrid, Spanien, September 1995.

[Rosenfeld 96] R. Rosenfeld: A maximum entropy approach to adaptive statistical language modeling. *Computer, Speech, and Language*, Vol. 10, No. 3, pp. 187–228, July 1996.

[Sakoe 79] H. Sakoe: Two-Level DP-Matching - A Dynamic Programming-Based Pattern Matching Algorithm for Connected Word Recognition. *IEEE Trans. on Speech and Audio Processing*, Vol. 27, pp. 588–595, Dec. 1979.

[Saon & Padmanabhan$^+$ 00] G. Saon, M. Padmanabhan, R. Gopinath, S. Chen: Maximum Likelihood Discriminant Feature Spaces. Proc. *IEEE International Conference on Acoustics, Speech and Signal Processing*, Vol. 2, pp. 1129–1132, Istanbul, Turkey, June 2000.

[Saul & Rahim 00] L.K. Saul, M.G. Rahim: Maximum Likelihood and Minimum Classification Error Factor Analysis for Automatic Speech Recognition. *IEEE Transactions on Speech and Audio Processing*, Vol. 8, No. 2, pp. 115–125, March 2000.

[Schlüter 00] R. Schlüter: *Investigations on Discriminative Training Criteria*. Ph.D. thesis, RWTH Aachen University, Aachen, Germany, Sept. 2000.

[Schlüter & Keysers$^+$ 06] R. Schlüter, D. Keysers, H. Ney: Tight Error Bounds for a Class of Statistical Criteria. personal communication, 2006.

[Schlüter & Macherey$^+$ 97] R. Schlüter, W. Macherey, S. Kanthak, H. Ney, L. Welling: Comparison of Optimization Methods for Discriminative Training Criteria. Proc. *ISCA Europ. Conf. on Speech Communication and Technology*, Vol. 1, pp. 15–18, Rhodes, Greece, September 1997.

[Schlüter & Macherey 98] R. Schlüter, W. Macherey: Comparison of Discriminative Training Criteria. Proc. *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Vol. 1, pp. 493–496, Seattle, WA, May 1998.

[Schlüter & Macherey$^+$ 99] R. Schlüter, W. Macherey, B. Müller, H. Ney: A Combined Maximum Mutual Information and Maximum Likelihood Approach for Mixture Density Splitting. Proc. *1999 European Conference on Speech Communication and Technology*, Vol. 4, pp. 1715–1718, Budapest, Hungary, September 1999.

[Schlüter & Macherey+ 01] R. Schlüter, W. Macherey, B. Müller, H. Ney: Comparison of Discriminative Training Criteria and Optimization Methods for Speech Recognition. *Speech Communication*, Vol. 34, No. 3, pp. 287–310, May 2001.

[Schlüter & Müller+ 99] R. Schlüter, B. Müller, F. Wessel, H. Ney: Interdependence of Language Models and Discriminative Training. Proc. *1999 Automatic Speech Recognition and Understanding (ASRU) Workshop*, Vol. 1, pp. 119–122, Keystone, CO, December 1999.

[Schlüter & Ney 00] R. Schlüter, H. Ney: Model Based MCE Bound to the True *Bayes*' Error. Proc. *IEEE Signal Processing Letters*, April 2000.

[Shannon 48] C.E. Shannon: A Mathematical Theory of Communication. *Bell System Technical Journal*, Vol. 27, pp. 379–423, July 1948.

[Sixtus 03] A. Sixtus: *Across-Word Phoneme Models for Large Vocabulary Continuous Speech Recognition*. Ph.D. thesis, RWTH Aachen, Jan. 2003.

[Sixtus & Ortmanns 99] A. Sixtus, S. Ortmanns: High Quality Word Graphs Using Forward-Backward Pruning. Proc. *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Vol. 2, pp. 593–596, Phoenix, AZ, USA, March 1999.

[Steinbiss & Tran+ 94] V. Steinbiss, B.H. Tran, H. Ney: Improvements in Beam Search. Proc. *1994 Int. Conf. on Speech and Language Processing*, pp. 2143–2146, Yokohama, Japan, September 1994.

[Tsakalidis & Doumpiotis+ 02] S. Tsakalidis, V. Doumpiotis, W. Byrne: Dicriminative Linear Transforms for Feature Normalization and Speaker Adaptation in HMM Estimation. Proc. *Int. Conf. on Spoken Language Processing*, pp. 2585–2588, Denver, CO, Sept. 2002.

[Tsakalidis & Doumpiotis+ 05] S. Tsakalidis, V. Doumpiotis, W. Byrne: Discriminative Linear Transforms for Feature Normalization and Speaker Adaptation in HMM Estimation. *IEEE Trans. on Speech and Audio Processing*, Vol. 13, No. 3, pp. 367–376, May 2005.

[Valtchev 95] V. Valtchev: *Discriminative Methods in HMM-based Speech Recognition*. Ph.D. thesis, St. John's College, University of Cambridge, Cambridge, UK, March 1995.

[Valtchev & Odell+ 96] V. Valtchev, J.J. Odell, P.C. Woodland, S.J. Young: Lattice-Based Discriminative Training For Large Vocabulary Speech Recognition. Proc. *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Vol. 2, pp. 605–608, Atlanta, GA, USA, May 1996.

[Vintsyuk 71] T.K. Vintsyuk: Elementwise Recognition of Continuous Speech Composed of Words from a Specified Dictionary. *Cybernetics*, Vol. 7, pp. 133–143, March/April 1971.

[Viterbi 67] A.J. Viterbi: Error Bounds for Convolutional Codes and an Asymptotically Optimal Decoding Algorithm. *IEEE Transactions on Information Theory*, Vol. 13, pp. 260–269, April 1967.

[Wakita 73] H. Wakita: Direct Estimation of the Vocal Tract Shape by Inverse Filtering of Acoustic Speech Waveform. *IEEE Transactions on Audio Electroacoustics*, Vol. 21, pp. 417–427, 1973.

[Welling 99] L. Welling: *Merkmalsextraktion in Spracherkennungssystemen für großen Wortschatz*. Ph.d. thesis, Chair of Computer Science VI, Computer Science Department, RWTH Aachen University, Aachen, Germany, January 1999.

[Wessel 02] F. Wessel: *Word Posterior Probabilities for Large Vocabulary Continuous Speech Recognition*. Ph.D. thesis, RWTH Aachen, Aachen, Germany, Aug. 2002.

[Woodland & Povey 00] P.C. Woodland, D. Povey: Large Scale Discriminative Training for Speech Recognition. Proc. *ISCA ITRW ASR2000, Automatic Speech Recognition, Challenges for the new millenium*, pp. 7–16, Paris, France, Sept. 2000.

[Woodland & Povey 02] P.C. Woodland, D. Povey: Large scale discriminative training of hidden Markov models for speech recognition. *Computer Speech & Language*, Vol. 16, No. 1, pp. 25–48, 2002.

[Young 92] S.J. Young: The General Use of Tying in Phoneme Based HMM Recognisers. Proc. *1992 Int. Conf. on Acoustics, Speech and Signal Processing*, Vol. 1, pp. 569–572, San Francisco, CA, March 1992.

[Young & Evermann$^+$ 06] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. Liu, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, P. Woodland: *The HTK Book (for HTK Version 3.4)*. Cambridge University Engineering Department, 2006.

Resume (April 6, 2009)

Dipl.-Inform. (M.S.) Wolfgang Macherey

Personal Data

date of birth:	March 29th, 1973
place of birth:	Düren-Birkesdorf
nationality:	German

Profession/Education

06/2009 – present	Senior Research Scientist at Google Inc., Mountain View, CA, USA
10/2006 – 05/2009	Research Scientist at Google Inc., Mountain View, CA, USA
03/2006 – 09/2006	Research Scientist at Google Inc., Zurich, Switzerland
1999 – 2006	Research Assistant and Ph.D. student at Computer Science Department of RWTH Aachen University (Prof. Dr.-Ing. Hermann Ney)
1999	Diploma's degree in Computer Science with honors.
1992 – 1999	Studies of computer science at RWTH Aachen University. Major field of study: statistical pattern recognition. Minor field of study: physical chemistry, thermodynamics. Diploma's thesis title: "Implementation and Comparison of Discriminative Criteria for Small Vocabulary Continuous Speech Recognition."
06/1992	Final secondary-school examinations (Abitur). Awarded by the Chemical Industry.
1983 – 1992	Secondary school Burgau Gymnasium Düren (Abitur).
1991	Qualified for the 3rd of four rounds of the International Chemistry Olympiad (IChO).
1979 – 1983	Primary school.

Experience/Profession

06/2009 – present	Senior Research Scientist at Google Inc.
03/2006 – 05/2009	Research Scientist at Google Inc.
01/2005 – 09/2005	Internship at Google Mountain View, CA, in the *Rosetta* machine translation group. Participated with the *Rosetta* team in the NIST-2005 machine translation evaluation.
07/2002 – 09/2002	Summer student at T.J. Watson Research Center in Yorktown Heights, NY.
04/2001 – 12/2002	Scientific consultant of Aixplain Corporation. Development and implementation of ASR systems for industrial partners.
05/1999 – 03/2006	Research Assistant and Ph.D. student at RWTH Aachen University. Responsibilities comprise research in statistical pattern recognition with the focus on acoustic modeling in ASR, project management, and teaching activities (supervision of diploma theses, organization and conducting of seminars and laboratory courses for undergraduate students).
11/1996 – 04/1999	Undergraduate scientist in the speech recognition group of Prof. Dr. Ing. H. Ney (domain discriminative Training). Tutor for several lectures in computer science.

Awards

2004	**EURASIP Best Paper Award** for the journal article "Comparison of Discriminative Training Criteria and Optimization Methods for Speech Recognition," published in Speech Communications.
2000	*Springorum Medal* for outstanding Master thesis.
1992	Award of the German Chamber of Chemical Industry.