A Log-Linear Discriminative Modeling Framework for Speech Recognition

Von der Fakultät für Mathematik, Informatik und Naturwissenschaften der Rheinisch-Westfälischen Technischen Hochschule Aachen zur Erlangung des akademischen Grades eines Doktors der Naturwissenschaften genehmigte Dissertation

vorgelegt von

Diplom–Physiker Georg Heigold

aus

Luzern, Schweiz

Berichter:
Professor Dr.–Ing. Hermann Ney
Professor Dr. Dietrich Klakow

Tag der mündlichen Prüfung: 29. Juni 2010

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. In particular, I would like to thank:

Prof. Dr.-Ing. Hermann Ney for the opportunity for doing research in this interesting and challenging area. This work would have not been possible without his continuous interest, advice, and support.

Prof. Dr. Dietrich Klakow from Saarland University, Germany, for kindly taking over the task of the co-referee for this thesis.

Dr. rer.-nat. Ralf Schlüter for the introduction to speech recognition and discriminative training, and his continuous constructive advice.

Patrick Lehnen and Stefan Hahn for the introduction to part-of-speech tagging and their assistance with the experiments.

Thomas Deselaers and Philippe Dreuw for their support with the experiments in handwriting recognition.

Muhammad Ali Tahir for performing the experiments with the discriminative feature transforms.

Christian Gollan, Thomas Deselaers, Björn Hoffmeister, Patrick Lehnen, Wolfgang Macherey, András Zolnay, and all other people from the Chair of Computer Science 6 for the interesting discussions on various speech recognition-related topics.

Oliver Bender, Thomas Deselaers, Mirko Kohns, Stefan Koltermann, Christian Plahl, and David Rybach for their excellent support with the computing equipment without which I could not have done so many experiments.

Stefan Hahn, Björn Hoffmeister, Patrick Lehnen, Markus Nußbaum, Christian Plahl, Muhammad Tahir, and Simon Wiesler for the proof-reading.

Volker Steinbiß, Gisela Gillmann, Jessica Kikum, Annette Kopp, Renate Linzenich, Ira Storms, and Andreas Wergen for their support in financial issues.

Annette, Frederik, Thierry, Rebekka, and Christoph for their encouragment in the evenings and at the weekends.

This work was partly funded by the European Commission under the integrated projects TC-STAR (FP6-506738) and LUNA (FP6-033549), this work was partly realized as part of the Quaero Programme, funded by OSEO, French State agency for innovation, and this work is partly based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. HR001-06-C-0023. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the DARPA.

Abstract

Conventional speech recognition systems are based on Gaussian hidden Markov models (HMMs). Discriminative techniques such as log-linear modeling have been investigated in speech recognition only recently. This thesis establishes a log-linear modeling framework in the context of discriminative training criteria, with examples from continuous speech recognition, part-of-speech tagging, and handwriting recognition. The focus will be on the theoretical and experimental comparison of different training algorithms.

Equivalence relations for Gaussian and log-linear models in speech recognition are derived. It is shown how to incorporate a margin term into conventional discriminative training criteria like for example minimum phone error (MPE). This permits to evaluate directly the utility of the margin concept for string recognition. The equivalence relations and the margin-based training criteria lead to a unified view of three major training paradigms, namely Gaussian HMMs, log-linear models, and support vector machines (SVMs). Generalized iterative scaling (GIS) is traditionally used for the optimization of log-linear models with the maximum mutual information (MMI) criterion. This thesis suggests an extension of GIS to log-linear models including hidden variables, and to other training criteria (*e.g.* MPE). Finally, investigations on convex optimization in speech recognition are presented. Experimental results are provided for a variety of tasks, including the European Parliament plenary sessions task and Mandarin broadcasts.

Zusammenfassung

Konventionelle Spracherkennungssysteme basieren auf Gaußschen HMMs. Diskriminative Techniken wie log-lineare Modellierung werden erst seit kurzem in der Spracherkennung untersucht. Diese Dissertation führt einen log-linearen Formalismus im Kontext der diskriminativen Trainings-Kriterien ein - mit Beispielen aus der kontinuierlichen Spracherkennung, dem Part-of-Speech-Tagging und der Handschrifterkennung. Der theoretische und experimentelle Vergleich von verschiedenen Trainings-Algorithmen bildet den Schwerpunkt dieser Arbeit.

Äquivalenzrelationen für Gaußsche und log-lineare Modelle in der Spracherkennung werden hergeleitet. Es wird gezeigt, wie ein Margin-Term in konventionellen diskriminativen Trainings-Kriterien wie zum Beispiel Minimum Phone Error (MPE) eingebaut werden kann, wodurch wir den Nutzen des Margin-Konzepts für die Erkennung von Strings direkt messen können. Die Äquivalenz-Relationen und die margin-basierten Trainings-Kriterien führen zu einer Vereinheitlichung drei wichtiger Trainingsparadigmen (Gaußsche HMMs, log-linearen Modelle und Support-Vektor-Maschinen (SVMs)). Generalized Iterative Scaling (GIS) wird traditionellerweise eingesetzt, um log-lineare Modelle mit dem Maximum Mutual Information (MMI)-Kriterium zu optimieren. Diese Dissertation schlägt eine Erweiterung von GIS für log-lineare Modelle mit verborgenen Variablen und für andere Trainings-Kriterien (zum Beispiel MPE) vor. Zum Schluss wird konvexe Optimierung in der Spracherkennung untersucht. Experimentelle Ergebnisse werden für eine Vielfalt von Aufgaben gezeigt, einschließlich der European-Parliament-Plenary-Sessions-Aufgabe und Mandarin Broadcasts.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Speech is one of the most natural means of human communication. Therefore, automatic speech recognition is a convenient basis for the development of human-machine interfaces, telecommunication services, and multimedia tools. Speech recognition can be used as a stand-alone tool (*e.g.* data entry and document preparation). It can also serve as the input for further natural language processing like for example spoken language translation or spoken language understanding.

Automatic speech recognition is the process of converting an acoustic signal (speech) to written text (recognized words) by a machine. Throughout this work, automatic speech recognition is investigated in the framework of statistical decision theory. Structured statistical models are used to reduce the complexity of the task. Conventionally, the statistical model is decomposed into the *language model* and the *acoustic model*. The latter model assumes *acoustic features* which are generated from the acoustic signal in a preprocessing step. In general, the word error is used to evaluate the performance of speech recognition systems.

The considered acoustic models have a huge number of free model parameters. These parameters are estimated using a suitable training criterion. Traditionally, the acoustic model has been represented by generative models. Discriminative techniques are based on a more direct approach and attempt to optimize directly the performance, *i.e.,* the word error of the speech recognition system.

1.1 Statistical Speech Recognition

In recent years, the statistical approach to speech recognition has prevailed over other approaches. Given a sequence of acoustic observations $x_1^T = x_1, \ldots, x_T$, that word sequence $w_1^N = w_1, \ldots, w_N$ should be chosen according to Bayes' decision rule which maximizes the *a posteriori* probability [Bayes 63]:

$$\begin{aligned} [w_1^N]_{\text{opt}} &= \underset{w_1^N}{\operatorname{argmax}} \left\{ p(w_1^N | x_1^T) \right\} \\ &= \underset{w_1^N}{\operatorname{argmax}} \left\{ p(x_1^T | w_1^N) \cdot p(w_1^N) \right\}. \end{aligned} \tag{1.1}$$

Equation (1.1) defines the two basic stochastic models that are involved in automatic speech recognition. The acoustic model $p(x_1^T|w_1^N)$ denotes the probability of observing the sequence of feature vectors x_1^T given a word sequence w_1^N. The language model $p(w_1^N)$ provides an *a priori* probability for a word sequence w_1^N. The basic architecture of a statistical speech recognition system is depicted in Figure 1.1 [Ney 90]. The system consists of four main components which will be described in detail in the following sections:

- The *signal analysis* (Section 1.1.1) module aims at extracting acoustic features from the input speech signal. It provides the speech recognizer with a sequence of acoustic vectors x_1^T.
- The *acoustic model* (Section 1.1.2) consists of statistical models for the smallest sub-word units to be distinguished by the speech recognizer, *e.g.* phonemes, syllables or whole words, and a pronunciation lexicon which defines the composition of an acoustic model for a given word from the sub-word units.
- The *language model* (Section 1.1.3) provides the *a priori* probability of a hypothesized word sequence based on the syntax, semantics and pragmatics of the language to be recognized.
- The *search module* (Section 1.1.4) finally combines the two knowledge sources acoustic model and language model to determine the word sequence that maximizes Equation (1.1). The search space for continuous speech recognition consists of all word sequences produced by a (finite) vocabulary.

This thesis will focus on discriminative techniques for the acoustic model $p(x_1^N|w_1^N)$. In the conventional generative approach (*e.g.* maximum likelihood), this component can be considered independent of the other components. This simplification holds no longer for the discriminative techniques that model directly the posterior $p(w_1^N|x_1^T)$, the basic quantity in the Bayes rule in Equation (1.1). In particular, discriminative training also takes the language model $p(w_1^N)$ into consideration and in fact, does not provide an estimate for the acoustic model $p(x_1^T|c_1^N)$. This different viewpoint typically leads to a significant increase in complexity because not only the correct but all competing word sequences as well enter the optimization.

1.1.1 Signal analysis/feature extraction

The signal analysis module aims at providing the speech recognition system with a sequence of acoustic vectors. The acoustic vectors build a parameterization of the speech waveform observed at the microphone. The signal analysis should remove as much information irrelevant for the speech recognition process as possible, for instance intensity, background noise, speaker identity, and only retain the information relevant for the *content* of the utterance. The signal analysis of today's state-of-the-art speech recognition systems is based on a short term spectral analysis [Rabiner & Schafer 78], usually a Fourier analysis. Three procedures for further processing and smoothing are widely used: Mel frequency cepstral coefficients (MFCC) [Davis & Mermelstein 80] and perceptual linear prediction (PLP) [Hermansky 90]. These features are motivated by the models of the human auditory system. Beside features

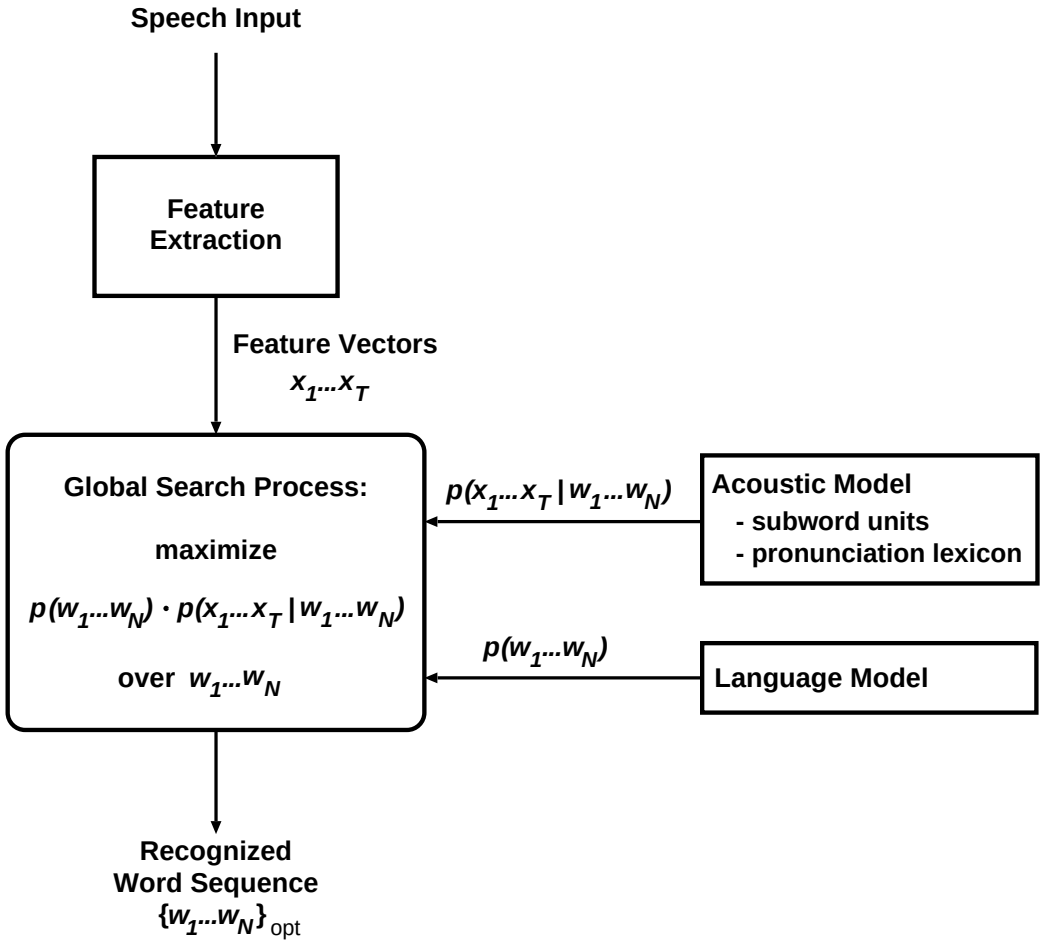


Figure 1.1: Basic architecture of a statistical automatic speech recognition system [Ney 90].

derived form the short-term power spectrum, several alternative acoustic features have been developed in recent years, including the TANDEM approach [Hermansky & Ellis+ 00a].

A commonly used method to include dynamic information is augmenting the original feature vector with the first and second derivatives yielding a high dimensional vector. A more general approach is based on the linear discriminant analysis (LDA) applied to concatenated feature vectors of neighboring time frames [Fisher 36, Duda & Hart+ 01]. The LDA is a linear transformation which projects a feature space into a lower dimensional subspace such that the class separability for distributions with equal variances is maximized.

In particular, the demand for speaker independence on the acoustic vectors is hard to meet. The above mentioned MFCC and PLP features for instance, are also used for speaker identification tasks [Doddington & Przybocki+ 00]. This means that there is still plenty of information of the given speaker contained in these features. Several methods have been developed to cope with the speaker dependency of the acoustic feature vectors: *speaker normalization*, which tries to reduce the speaker dependency by transforming the acoustic feature vectors, and *speaker adaptation*, which tries to adjust the model parameters of the speech recognition system to the characteristics of the given speaker. In [Pitz 05], a comprehensive comparison of these methods is presented along with a unified view of speaker-dependent transformations.

1.1.2 Acoustic modeling

The aim of acoustic modeling is to provide a statistical model $p(x_1^T|w_1^N)$ for the realization of a sequence of acoustic vectors x_1^T given a word sequence w_1^N. The acoustic model is a concatenation of the acoustic models for the basic sub-word units that the speech recognition system utilizes, according to a pronunciation lexicon.

Depending on the amount of training data and the desired model complexity, the sub-word units are whole words, syllables, phonemes, or phonemes in context. Smaller units than words enable the speech recognition system to recognize words which have not been seen in the training data and to ensure that enough instances of each unit have been observed in training to allow a reliable parameter estimation. In large vocabulary speech recognition (LVCSR), the most commonly used sub-word units are phonemes in the context of one or two adjacent phonemes, so-called triphones and quinphones, respectively. Context-dependent phonemes (allophones) are used to account for the different pronunciations of a phoneme depending on the surrounding phonemes.

The acoustic realizations of a sub-word unit differ significantly depending on the speaking rate. To model the variations in speaking rate, hidden Markov models (HMM) have been established as a *de-facto* standard for speech recognition systems [Baker 75, Rabiner 89]. An HMM is a stochastic finite state automaton consisting of a number of states and transitions between the states. The probability $p(x_1^T|w_1^N)$ is extended by unobservable (hidden) random variables representing the states:

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

The sum is over all possible state sequences s_1^T for a given word sequence w_1^N. Using Bayes' identity, this can be rewritten as

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

This equation can be further simplified by applying the first order Markov assumption [Duda & Hart+ 01]. The probabilities $p(x_t|x_1^{t-1}, s_1^t, w_1^N)$ and $p(s_t|x_1^{t-1}, s_1^{t-1}, w_1^N)$ are assumed not to depend on previous observations but only on the states (Markov) and on the immediate predecessor state only (first-order):

$$p(x_1^T|w_1^N) = \sum_{s_1^T} \prod_{t=1}^{T} p(x_t|s_t, w_1^N) \cdot p(s_t|s_{t-1}, w_1^N) \ . \tag{1.2}$$

Thus, the probability $p(x_1^T|w_1^N)$ is split into the *emission probability* $p(x_t|s_t, w_1^N)$ denoting the probability to observe an acoustic vector x_t while being in state s_t, and the *transition probability* $p(s_t|s_{t-1}, w_1^N)$ for a transition from state s_{t-1} to state s_t. Usually, the sum in Equation (1.2) is approximated by the maximum.

$$p(x_1^T|w_1^N) \approx \max_{s_1^T} \left\{ \prod_{t=1}^{T} p(x_t|s_t, w_1^N) \cdot p(s_t|s_{t-1}, w_1^N) \right\} \ . \tag{1.3}$$

This approximation is called Viterbi or maximum approximation [Ney 90]. Equations (1.2) and (1.3) can be solved efficiently using the forward-backward algorithm [Baum 72, Rabiner & Juang 86], which is an example of dynamic programming [Bellman 57, Viterbi 67, Ney 84].

An example of an HMM for a part of the word "seven" is shown in Figure 1.2. The topology used in this work has been introduced by Bakis [Bakis 76]: the basic HMM consists of six subsequent states where each two successive states are identical. Only transitions from a state to itself (*loop*), the next state (*forward*), and the next to next state (*skip*) are allowed. Using a frame-shift of 10ms, the path through the HMM with forward transitions only amounts to 60ms. This is close to the average duration of phonemes for most languages. This 6-state HMM has a minimum duration of 30ms (only skip transitions). This has been found to be too long for fast conversational speech, *e.g.* on the Verbmobil II corpus [Molau 03]. In this case, a 3-state model is used where the two identical states are merged into a single one. This reduces the minimum length of the HMM.

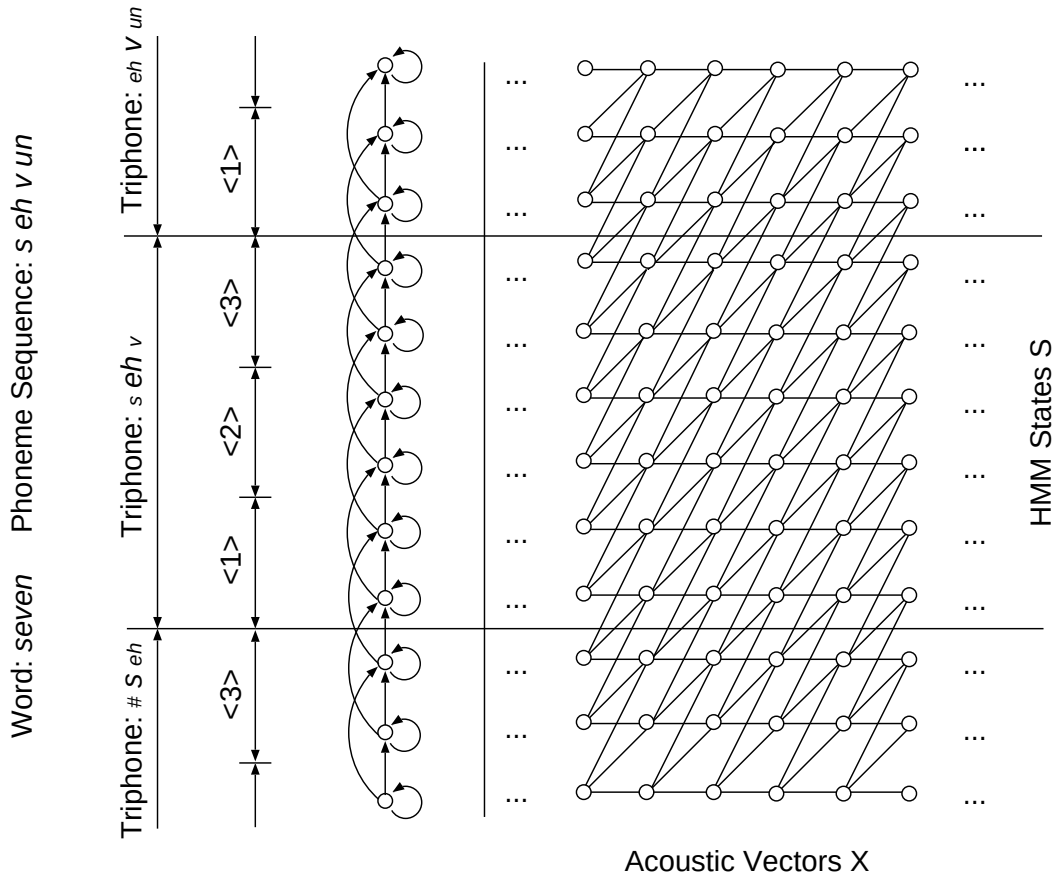


Figure 1.2: 6-state hidden Markov model in Bakis topology for the triphone ${}_{s}eh_{v}$ in the word "seven". The HMM segments are denoted by <1>, <2>, and <3>.

The emission probabilities $p(x_t|s_t, w_1^N)$ of an HMM can be modeled by discrete probabilities [Jelinek 76], semi-continuous probabilities [Huang & Jack 89] or continuous probability distributions [Levinson & Rabiner$^+$ 83]. A commonly used model for continuous probability distributions are Gaussian mixture models (GMMs). Assuming GMMs, the emission probabil-

ities read

$$p(x|s, w_1^N) = \sum_{l=1}^{L_s} c_{sl}\, \mathcal{N}(x|\mu_{sl}, \Sigma, w_1^N) \tag{1.4}$$

where c_{sl} denotes the non-negative mixture weights subject to the constraint $\sum_{l=1}^{L_s} c_{sl} = 1$, and $\mathcal{N}(x|\mu, \Sigma)$ denotes the Gaussian density with mean μ and covariance matrix Σ. In the RWTH system, a single globally pooled and diagonal covariance matrix is used. This choice is made to avoid problems caused by data sparseness, and due to efficiency reasons. Diagonal covariances assume decorrelated features. The feature decorrelation can be done, for instance, by LDA in a preprocessing step. Conventionally, the set of parameters $\Lambda = \{\{\mu_{sl}\}, \{c_{sl}\}, \Sigma\}$ is estimated according to the maximum likelihood (ML) training criterion in combination with the expectation-maximization (EM) algorithm [Dempster & Laird+ 77].

The number of distinct allophone states as basic sub-word units increases exponentially with the context length. Thus, a large number of allophones will have no or too few observations for a reliable parameter estimation. Therefore, several states are tied together yielding *generalized* allophone models [Young 92]. Decision tree-based state clustering (*e.g.* CART) is used in almost all LVCSR systems. The main advantage of this top-down clustering method is that no back-off models need to be trained and unseen allophones will be assigned to an appropriate HMM state. Details of the state clustering in the RWTH system can be found in [Beulen & Ortmanns+ 99]. As the pronunciation of a phoneme depends on the surrounding phonemes, a phoneme at a word boundary is pronounced differently depending on the predecessor and successor words. This coarticulation effect is modeled explicitly using *across-word* allophones [Hon & Lee 91, Odell & Valtchev+ 94], which take respectively into account the ending and beginning phonemes of the adjacent words as a left and right context. Details of the across-word model implementation for the RWTH system can be found in [Sixtus 03].

1.1.3 Language modeling

The language model $p(w_1^N)$ provides an *a priori* probability for a word sequence $w_1^N = w_1, \ldots, w_N$. The syntax, semantics and pragmatics of the language to be recognized are implicitly covered by this statistical model. Due to the unlimited number of possible word sequences, further model assumptions have to be applied in order to estimate a reliable model. For LVCSR, m-gram language models [Bahl & Jelinek+ 83] have become widely accepted. The m-gram language models assume that the word sequence follows an $(m-1)$-th order Markov process, *i.e.,* the probability of the word w_n only depends on the $(m-1)$ predecessor words. Thus, the probability $p(w_1^N)$ factorizes into

$$\begin{aligned} p(w_1^N) &= \prod_{n=1}^{N} p(w_n|w_1^{n-1}) \\ &\overset{\text{model assumption}}{=} \prod_{n=1}^{N} p(w_n|w_{n-m+1}^{n-1}) \,. \end{aligned} \tag{1.5}$$

The word sequence $h_n = w_{n-m+1}^{n-1}$ is denoted as history of length m of the word w_n with the definitions $h := w_1^{n-1}$ if $n < m$ and $h := \emptyset$ if $n - 1 < n - m + 1$, *e.g.* at the boundary $p(w_1|w_1^0) =$

$p(w_1)$.

A commonly used measure for the evaluation of language models is the *perplexity* PP

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

The log-perplexity is equal to the entropy of the model and can be interpreted as the average number of choices to continue a word sequence w_{n-m+1}^{n-1} at position n. When using the perplexity as optimization criterion for training the language model, closed form solutions for $p(w|h)$ can be derived which are equal to the relative frequency of the word sequence on the training corpus. Also, the word error rate has recently been used in ASR and SMT for the evaluation of language models. The number of possible m-grams increases exponentially with the history length m. Thus, for a large vocabulary V, a considerable amount of m-grams will not be seen in training or has too few observations for a reliable estimation of $p(w|h)$, even for very large training corpora. Therefore, smoothing methods have to be applied. The smoothing is based on discounting in combination with backing-off or interpolation [Katz 87, Ney & Essen$^+$ 94, Generet & Ney$^+$ 95, Ney & Martin$^+$ 97]. Discounting subtracts probability mass from seen events which is then distributed over all unseen events (backing-off) or over all events (interpolation), usually in combination with a language model with shorter history. The parameters of the smoothed language model can be estimated using a cross-validation scheme like leaving-one-out [Ney & Essen$^+$ 94].

1.1.4 Search

The search module of the speech recognition system combines the two knowledge sources, which are acoustic model and language model as depicted in Figure 1.1. The objective of the search is to find the word sequence that maximizes the *a posteriori* probability for a given sequence x_1^T of acoustic feature vectors according to Equation (1.1)

$$\begin{aligned}[w_1^N]_{\text{opt}} &= \operatorname*{argmax}_{w_1^N} \left\{p(w_1^N|x_1^T)\right\} \\ &= \operatorname*{argmax}_{w_1^N} \left\{p(w_1^N) \cdot p(x_1^N|w_1^N)\right\} . \end{aligned} \tag{1.6}$$

If the language model is given by the m-gram model in Equation (1.5) and the acoustic model is an HMM as given in Equation (1.2), the following optimization problem has to be solved by the search module:

$$\begin{aligned}[w_1^N]_{\text{opt}} &= \operatorname*{argmax}_{w_1^N} \left\{\left[\prod_{n=1}^{N} p(w_n|w_{n-m+1}^{n-1})\right] \cdot \left[\sum_{s_1^T} \prod_{t=1}^{T} p(x_t|s_t, w_1^N) \cdot p(s_t|s_{t-1}, w_1^N)\right]\right\} \\ &\overset{\text{Viterbi approx.}}{=} \operatorname*{argmax}_{w_1^N} \left\{\left[\prod_{n=1}^{N} p(w_n|w_{n-m+1}^{n-1})\right] \cdot \left[\max_{s_1^T} \prod_{t=1}^{T} p(x_t|s_t, w_1^N) \cdot p(s_t|s_{t-1}, w_1^N)\right]\right\} . \end{aligned} \tag{1.7}$$

In the second step, the Viterbi approximation is applied to the HMM. This reduces significantly the complexity of the optimization problem. Equation (1.7) can be solved efficiently using

dynamic programming [Bellman 57]. Dynamic programming exploits the mathematical structure and divides the problem into sub-instances. Like in all search problems, the search can be organized in two different ways: a depth-first and breadth-first search. The depth-first strategy is used by the A^*-search or stack-decoding algorithm. Here, the state hypotheses are expanded time-asynchronously depending on a heuristic estimate of the costs to complete the path [Jelinek 69, Paul 91].
The breadth-first search design is used by the Viterbi search where all state hypotheses are expanded time-synchronously [Vintsyuk 71, Baker 75, Sakoe 79, Ney 84]. In this approach, the probabilities of all hypotheses up to a given time frame are computed and thus can be compared to each other. This allows to reduce the search space significantly by pruning unlikely hypotheses early in the search process. Especially in the breadth-first approach, an efficient pruning is necessary as the number of possible word sequences with maximum length N grows exponentially with N. Thus, a full optimization of Equation (1.7) is only feasible for small vocabulary sizes $|W|$. For large vocabulary sizes approximations have to be made. Instead of finding the exact optimal solution of Equation (1.7) the goal is changed to find a sufficiently good solution with much less effort. In the so-called beam-search, only that fraction of the hypotheses is expanded whose likelihood is sufficiently close to that of the best hypothesis of the given time frame [Lowerre 76, Ney & Mergel$^+$ 87, Ortmanns & Ney 95]. Beam-search does not guarantee to find the globally best word sequence. This optimal sequence may have been pruned at an intermediate search stage due to a poor likelihood. However, if the pruning parameters are adjusted properly no significant search errors occur and the search effort is reduced considerably.

Several other methods can be applied to reduce further the computational complexity of the Viterbi or beam-search, including lexical prefix tree [Ney & Häb-Umbach$^+$ 92], look-ahead [Steinbiss & Ney$^+$ 93, Häb-Umbach & Ney 94, Odell & Valtchev$^+$ 94, Alleva & Huang$^+$ 96, Ortmanns & Ney$^+$ 96], and fast likelihood computation [Ramasubramansian & Paliwal 92, Fritsch 97, Bocchieri 93, Ortmanns & Ney$^+$ 97b, Ortmanns 98, Kanthak & Schütz$^+$ 00]. More advanced algorithms involving search (*e.g.* discriminative training) use N-best lists [Schwartz & Chow 90, Schwartz & Austin 91] or word lattices [Ney & Aubert 94, Ortmanns & Ney$^+$ 97a, Macherey 10] to reduce the search space.

1.2 Discriminative Techniques: State of the Art

Conventional speech recognition systems in ASR are based on generative Gaussian HMMs (GHMMs) [Rabiner & Juang 97]. Traditionally, these GHMMs are optimized using a generative training criterion, *e.g.* maximum likelihood (ML) [Rabiner & Juang 86, Rabiner 89]. In many state-of-the-art systems, the generatively estimated GHMMs are reestimated with a discriminative training criterion like for example maximum mutual information (MMI) in a postprocessing step [Bahl & Brown$^+$ 86, Juang & Katagiri 92, Normandin 96, Valtchev & Odell$^+$ 97]. Numerical optimization techniques are employed for the discriminative training, *e.g.* extended Baum Welch (EBW) [Normandin & Morgera 91] and general gradient descent (GD) [Katagiri & Juang$^+$ 98]. A vast number of refinements have been proposed and discussed in the literature, both concerning the training criteria (Section 1.2.1) and the optimization algorithms (Section 1.2.6). Word lattices have proved to be useful in this context.

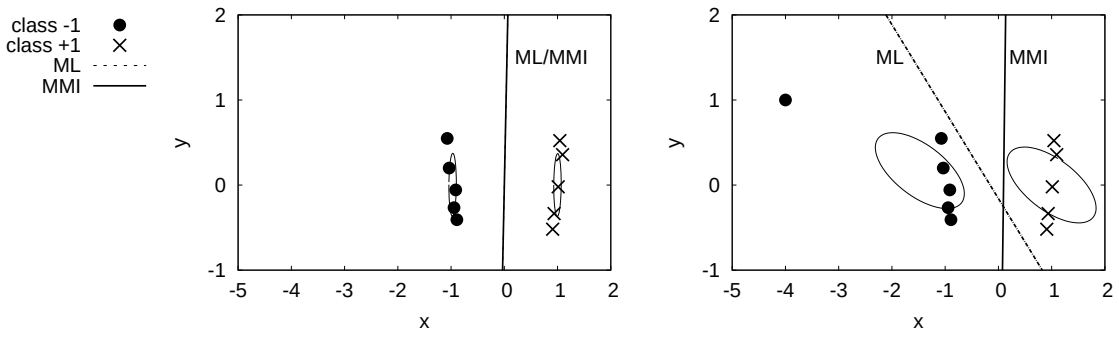


Figure 1.3: Comparison of decision boundaries induced by ML and MMI under different conditions. Each of the two classes is modeled by a Gaussian distribution with a full but shared covariance matrix. A uniform prior is used. The estimated covariance is indicated by ellipses. Left: data and model match. Right: data and model do not match, see outlier at (-4.0,1.0).

Conventional lattice-based training can be regarded as an example for the transducer-based training (Section 1.2.2).

More recently, discriminative models replacing the conventional GHMMs have been investigated for speech recognition (Section 1.2.3). Also, there has been a growing interest in training algorithms with additional theoretical properties. Regularization techniques [Hastie & Tibshirani+ 01] and the margin concept [Vapnik 95] aim at increasing the generalization ability (Section 1.2.5). Optimization algorithms using growth transformations [Gopalakrishnan & Kanevsky+ 91] and convex optimization techniques [Boyd & Vandenberghe 04] lead to stronger convergence results (Section 1.2.6).

Few theoretical work has been done so far to compare the generative and discriminative training criteria. The Cramer-Rao lower bound guarantees that if the model is correct, the lowest variance estimate of the model parameters will be obtained with ML. The work in [Nádas 83, Nádas & Nahamoo+ 88] shows that MMI performs no worse than ML. Figure 1.3 illustrates this asymptotic result. The situation for finite training data is different where ML outperforms MMI for sufficiently little (relative to model complexity) data [Ng & Jordan 02]. The robustness of estimators was studied in general in [Huber 81, Hampel 86].

1.2.1 Discriminative training criteria

The training criteria can be classified in probabilistic and error-based training criteria. The probabilistic training criteria include ML (generative) and MMI [Bahl & Brown+ 86, Chow 90, Kapadia & Valtchev+ 93, Cardin & Normandin+ 93, Bahl & Padmanabhan+ 96, Bahl & Padmanabhan 98, Normandin 91, Normandin & Morgera 91, Normandin & Cardin+ 94, Normandin & Lacouture+ 94, Normandin 96, Valtchev 95, Valtchev & Odell+ 96, Valtchev & Odell+ 97, Merialdo 88, Schlüter 00, Woodland & Povey 00, Woodland & Povey 02]. Similar to the hybrid approach (Section 1.2.3), a variant of MMI for frame discrimination was proposed in [Povey & Woodland 99, Povey & Woodland 02]. The error-based training criteria try to optimize directly the classification error. Two prominent examples of this class

of training criteria are the minimum classification error (MCE) [Juang & Katagiri 92, McDermott & Katagiri 97, McDermott & Katagiri 05, McDermott & Hazen+ 07, Macherey & Haferkamp+ 05, Macherey 10], and minimum word/phone error (MWE/MPE) [Povey & Woodland 02, Povey 04]. Earlier work on word error-based training can be found in [Chou & Lee+ 93, Chou & Lee+ 94, Kaiser & Horvat+ 00, Bauer 01, Kaiser & Horvat+ 02]. The MWE/MPE training criterion is generalized in the minimum Bayes risk (MBR) training framework [Kaiser & Horvat+ 00, Kaiser & Horvat+ 02, Doumpiotis & Byrne 04, Doumpiotis & Byrne 05, Gibson & Hain 06, Gibson 08]. This framework also includes variants of MWE/MPE, *e.g.* minimum phone frame error (MPFE) [Zheng & Stolcke 05b], minimum divergence-based discriminative training [Du & Liu+ 06], or non-uniform error cost functions [Fu & Juang 08]. Finally, training criteria incorporating a margin term have been proposed. The discussion of these training criteria is deferred until Section 1.2.5.

Such discriminatively refined GHMMs have proved to outperform the generatively optimized GHMMs, not only on tasks of low complexity [Chow 90, Juang & Katagiri 92, Cardin & Normandin+ 93, Chou & Lee+ 94, Kaiser & Horvat+ 00, Bauer 01, Kapadia & Valtchev+ 93, Normandin 96, McDermott & Katagiri 97, Valtchev & Odell+ 97, Bahl & Padmanabhan 98, Merialdo 88, Schlüter 00], but also for large vocabulary continuous speech recognition (LVCSR) systems [Woodland & Povey 00, Povey 04, Doumpiotis & Byrne 05, McDermott & Katagiri 05, Zheng & Stolcke 05b, Gibson 08, Macherey & Haferkamp+ 05], some of them trained on thousands of hours of audio data [Evermann & Chan+ 05]. The earliest discriminative training algorithms used N-best lists to approximate the search space. Lattices have been used instead since [Valtchev & Odell+ 96], particularly in LVCSR.

Thorough comparisons of the different training criteria have been done [Schlüter 00, Schlüter & Macherey+ 01, Macherey & Haferkamp+ 05, Povey & Kingsbury 07, Macherey 10]. Starting with [Reichl & Ruske 95], the "training criteria zoo" has finally been described in the unified training criterion [Schlüter & Macherey+ 97, Schlüter 00, Schlüter & Macherey+ 01, Macherey & Haferkamp+ 05, He & Deng+ 08, Nakamura & McDermott+ 09].

Ideally, speech recognition systems are optimized by minimizing the empirical risk using the (exact) word error. There are a couple of practical problems with this ideal training criterion. First, no efficient algorithm is known to the author to calculate the word error for all possible word sequences, even if restricted to lattices. For that reason, several approximations to the exact loss function have been investigated: exact word error on N-best lists [Kaiser & Horvat+ 00, Kaiser & Horvat+ 02] or pinched lattices [Doumpiotis & Byrne 04], and approximate word error rates on lattices [Schlüter 00, Povey & Woodland 02, Povey 04, Zheng & Stolcke 05b]. Second, the exact empirical risk is a non-differentiable function which is replaced by a smooth approximation in practice (Section 3.8.2). The exact empirical risk could be optimized using grid search techniques or the approach for statistical machine translation (SMT) suggested in [Och 03, Macherey & Och+ 08]. This, however, has not been done for acoustic models so far.

The above mentioned training criteria were originally designed for the reestimation of the Gaussian HMM parameters in a supervised manner. These training criteria have also been applied to model adaptation [Zheng & Stolcke 05a], lightly-supervised acoustic model

training [Chan & Woodland 04], the optimization of linear feature transforms like for example the linear discriminant analysis (LDA) [Omar & Hasegawa-Johnson 03] and feature-space MPE (fMPE) [Povey & Kingsbury+ 05], speaker adaptation [Gunawardana 01, Wang 06, Lööf & Schlüter+ 07], precision matrix models [Sim & Gales 06], or handwriting recognition [Nopuswanchai & Povey 03].

1.2.2 Transducer-based discriminative training

Weighted finite-state transducer (WFST) methods proved to solve elegantly many difficult problems in the field of natural language processing. An overview of the basic WFST algorithms is given in [Mohri 04]. Several WFST toolkits are publicly available, *e.g.* AT&T FSA Library™ [Mohri & Pereira+ 00a], or FSA [Kanthak & Ney 04]. Non-trivial applications of these WFST algorithms include a full and lazy compilation of the search network for speech recognition [Mohri & Pereira+ 00b], integrated speech translation [Vidal 97, Matusov & Kanthak+ 05], and parameter estimation [Eisner 01, Lin & Yvon 05, McDermott & Katagiri 05, Kuo & Zweig+ 07, Li & Eisner 09] to mention but a few.

State-of-the-art discriminative acoustic model training uses lattices to approximate the combinatorial search space. Therefore, the training can be considered an example for transducer-based training. For a few important training criteria, efficient algorithms are known to calculate efficiently the accumulation statistics. MMI and MCE rely on the forward/backward (FB) probabilities (*cf.* Baum algorithm) [Rabiner 89] and MWE/MPE uses Povey's recursion formula [Povey & Woodland 02]. An elegant framework for general transducer-based training was proposed in [Eisner 01] (MMI), and more recently in [Li & Eisner 09] (MWE/MPE-like training criteria). The complexity of this algorithm scales with the number of model parameters used in the transducer. This can be done more efficiently as shown in Chapter 3.

Some of the loss metrics used in speech recognition fit into the transducer-based framework. The most important example is the calculation of the word error rate for a single reference transducer [Ristad & Yianilos 98a] and for a set of reference transducers [Mohri 03]. The latter problem is typical of MWE/MPE-like training criteria.

In this work (Chapter 3), transducer-based training is used to optimize graphical models, *e.g.* conditional random fields (CRFs) [Lafferty & McCallum+ 01, Sutton & McCallum 07, Gunawardana & Mahajan+ 05] to be discussed next.

1.2.3 Discriminative models & parameterization

Generative models define the class posteriors indirectly through the joint probabilities. In contrast, discriminative models directly provide a posterior model - hence also known as direct models. Prominent examples of discriminative models include the log-linear models (or logistic regression), conditional random fields (CRFs), support vector machines (SVMs), and neural networks (NN). The focus of this thesis shall be on discriminative models based on a log-linear parameterization [Ney 09].

Log-linear models/maximum entropy models. The maximum entropy principle motivates the maximum entropy models, also known as log-linear models due to their functional form [Jaynes 03]. Log-linear models are not new to pattern recognition. These models have been employed for discriminative language modeling [Rosenfeld 94], natural language processing (NLP) [Berger & Della Pietra+ 96], discriminative model combination (DMC) [Beyerlein 97, Beyerlein 98, Beyerlein 00], SMT [Och & Ney 02] *etc.* So far, only few work has been done on direct log-linear acoustic modeling [Hifny & Renals+ 05]. The work in [Layton & Gales 06, Layton & Gales 07] is related to the log-linear approach. Log-linear models have been specialized as to capture better the specifics of sequential data.

Maximum entropy Markov models (MEMMs). MEMMs were first described in [McCallum & Freitag+ 00] in the context of information extraction and segmentation. This discriminative model was studied for speech recognition in [Likhododev & Gao 02, Kuo & Gao 06]. MEMMs may suffer from the *label bias* problem [Bottou 91, Lafferty & McCallum+ 01]. CRFs, for example, solve this problem.

Conditional random fields (CRFs). CRFs are a framework for graphical sequential models. Originally, CRFs have been proposed for NLP [Lafferty & McCallum+ 01, Sutton & McCallum 07, Cohn 07]. Recently, CRFs have also been applied to acoustic modeling in speech recognition [Macherey & Ney 03, Gunawardana & Mahajan+ 05, Abdel-Haleem 06, Fosler-Lussier & Morris 08, Hifny & Renals 09, Morris & Fosler-Lussier 09]. Various acoustic representations for the log-linear models have been tested: conventional MFCC features [Macherey & Ney 03, Gunawardana & Mahajan+ 05], rank-based features [Kuo & Gao 06], posterior-based features [Hifny & Gao 08] or spline-based features [Yu & Deng+ 09]. Also, more sophisticated detector-based features like for example MLP features [Fosler-Lussier & Morris 08] and nearest neighbor based spotter features [Heigold & Li+ 09] have been studied.

Hybrid architectures. The hybrid approach combines the advantages of HMMs and discriminative classifiers [Bourlard & Morgan 94]. In the past, various static classifiers were employed: neural networks (NN) [Robinson & Fallside 91, Robinson & Hochberg+ 96, Kershaw & Robinson+ 96, Rigoll & Willett 98, Stadermann 06], (discriminatively optimized) Gaussian mixture models (GMMs) [Povey & Woodland 99, Povey & Woodland 00], support vector machines (SVMs) [Ganapathisraju 02], and maximum entropy models [Hifny & Renals+ 05]. The experimental results reported in [Kingsbury 09] suggest that speech recognition probably goes beyond simple frame discrimination.

Reparameterization of generative models. It has often been demonstrated in the literature that GMMs and GHMMs can be represented as log-linear models [Jebara 02, Macherey & Schlüter+ 04, Gunawardana & Mahajan+ 05, Abdel-Haleem 06]. This observation inspired the reparameterization of GMMs and GHMMs to derive optimization algorithms with better expected numerical properties [Sim & Gales 06, Sha & Saul 07a, Sha & Saul 07b]. The Gaussian models impose parameter constraints (*e.g.* positive variances) and HMMs are directed models with local normalization constraints. CRFs do not constrain the parameters and are

undirected models with a single global normalization constraint. For these reasons, it is not obvious how to transform a log-linear model into an equivalent generative model.

it is believed that the opposite is not true (*i.e.,* not every log-linear model can be represented as an equivalent generative model) [Lafferty & McCallum+ 01, Saul & Lee 02, Sha & Saul 07a, Gunawardana & Mahajan+ 05, Cohn 07]. Chapter 4 establishes equivalence relations for Gaussian and log-linear models. These equivalence relations are based on the degeneracy in the relationship between a Gaussian mixture model (GMM) and the *a posteriori* class probability functions that it induces [Ristad & Yianilos 98b].

1.2.4 Equivalence relations for generative and log-linear models

Equivalence relations have been established for general (the only restriction is that the distributions are non-zero) directed and undirected models obeying certain conditional independence assumptions, see for example [Lauritzen & Dawid+ 90]. Chapter 4 focuses on a few restricted model classes (*e.g.* GHMMs) of practical interest that are small subsets of those general model classes. In the terminology of [Ng & Jordan 02], equivalent generative and discriminative models are called a generative/discriminative pair. Only a couple of generative/discriminative pairs appear to be known in the literature. As for the transformation from a discriminative into a generative model, however, the statements are not always clear, different statements may be conflicting, and explicit transformation rules are missing.

The log-linear and Gaussian-based discriminant analysis, for example, have been thoroughly studied in the literature. The work in [Anderson 82] shows that the Gaussian-based discriminant analysis is a subset of the log-linear discriminant analysis. However, it remains unclear if the transformation from the log-linear to the Gaussian-based discriminant analysis is always possible. According to [Ng & Jordan 02] (without proof), Gaussian-based and logistic discriminant analysis form a generative/discriminative pair. This result is supported indirectly by the analysis of the discriminant functions in [Duda & Hart+ 01, pp.19]. In contrast, [Saul & Lee 02] clearly states that the log-linear discriminant analysis is more expressive than the Gaussian-based.

The situation for the more complex HMMs is similar. The authors in [Sutton & McCallum 07] claim that the transformation is possible, without giving any details to support their claim. Assuming a weighted finite-state transducer (WFST) with non-negative arc weights, weight pushing produces an equivalent stochastic WFST [Mohri 09, p.242]. This implies that the transformation is possible, at least under suitable boundary conditions. The detailed analysis in [Jaynes 03, pp.646] suggests that the stationarity of the transition probabilities is violated for finite sequences. According to [Gunawardana & Mahajan+ 05], the transformation is impossible in general due to the parameter constraints. From the statements in [Cohn 07], it is unclear whether the local normalization constraints reduce the model flexibility, or only make the modeling less convenient.

1.2.5 Generalization ability

Various techniques have been proposed in the literature to prevent the parameters from overfitting. The most important approaches are discussed now.

Regularization. Regularization techniques including the maximum *a posteriori* (MAP) approach [Gauvain & Lee 94] and smoothing, are employed to avoid overfitting. In discriminative training of GHMMs, the H-criterion [Gopalakrishnan & Kanevsky+ 88] and I-smoothing [Povey & Woodland 02, Povey & Gales+ 03, Povey 04] are the most popular smoothing techniques. Log-linear models have been optimized using a Gaussian prior, *i.e.,* the ℓ_2-regularization [Chen & Rosenfeld 99] and the ℓ_2-regularization around a non-uniform initial model [Li 07] for regularization.

Large margin classifiers/SVMs. Probably approximately correct (PAC) generalization bounds were derived in [Vapnik 95]. The design of new training algorithms like for example the large margin classifiers are motivated by these theoretical results. Probably the best known large margin classifier is the SVM [Vapnik 95]. Multi-class formulations of SVMs do also exist [Weston & Watkins 99, Altun & Tsochantaridis+ 03, Taskar & Guestrin+ 03]. There is a close relationship of SVMs and logistic regression [Jaakkola & Meila+ 99, Zhang & Jin+ 03]. In speech recognition, SVMs have been tested in hybrid architectures, *e.g.* [Ganapathisraju 02]. In recent years, novel training algorithms for speech recognition have been designed to incorporate a margin term, see below for the literature.

Margin-based training in ASR. The first approaches to margin-based training in speech recognition used SVMs [Vapnik 95] in a hybrid architecture, *e.g.* [Ganapathisraju 02]. The hidden Markov SVMs [Altun & Tsochantaridis+ 03] and max-margin Markov networks [Taskar & Guestrin+ 03] might be more suitable for string recognition but have not been tested in the context of speech recognition. One of the first papers on direct margin-based training in speech recognition was [Liu & Jiang+ 05]. The authors demonstrate the utility of the suggested maximum relative separation margin on the ISOLET database. The training criterion can be refined to large margin estimation (LME) such that the optimization problem can be solved with semidefinite programming [Li & Jiang 06, Jiang & Li 07] or the more efficient second-order cone programming [Yin & Jiang 07]. Experimental results are presented for the TIDIGITS database. Soft margin error (SME) including extensions is introduced in [Li & Yuan+ 06, Li & Yan+ 07, Li & Yan+ 08]. An experimental comparison with conventional training criteria is provided up to a small LVCSR task (WSJ5k). The Gaussian parameters are reparameterized (*cf.* Section 1.2.3) in [Sha & Saul 06, Sha & Saul 07a, Sha & Saul 07b] to derive a convex optimization problem, accompanied with tests on the TIMIT database. Similar to SVMs and other conventional large margin classifiers, all these training criteria are based on the hinge loss function. Only [Yu & Deng+ 08] use the smoothed classification error from MCE for the loss function. The authors report on experimental results for a telephony speech task [Yu & Deng+ 06, Yu & Deng+ 07, Yu & Deng+ 08] and for spoken utterance classification [Yaman & Deng+ 07]. Instead of a single margin parameter, the integral over an interval of margin parameters can be used to establish relations between MMI and

MPE [McDermott & Watanabe+ 09]. Boosted MMI [Povey & Kanevsky+ 08] is motivated by the boosting technique [Bishop 06]. This training criterion was tested on several LVCSR tasks together with refinements for EBW [Povey & Kanevsky+ 08]. This variant of MMI can be interpreted as a margin-based approach [Saon & Povey 08]. Similarly, boosted MPE can be defined [McDermott & Nakamura 08]. It does not only applies to MMI but to other conventional training criteria, including MPE and MCE as well. The work presented in Chapter 5 was developed independently of boosted MMI and MPE.

1.2.6 Numerical optimization

Numerical optimization techniques are essential for discriminative training. The optimization of state-of-the-art acoustic models is a non-trivial task due to the complexity and large-scale nature of speech recognition. Therefore, much effort has been spent on developing efficient optimization algorithms. Here, the algorithms are distinguished by their properties (*e.g.* growth transformation and convexity).

Most groups employ highly tuned versions of EBW [Merialdo 88, Schlüter 00, Woodland & Povey 02, Povey 04, Macherey & Schlüter+ 04, Macherey 10] to discriminatively reestimate GHMMs. EBW is motivated by a growth transformation [Normandin & Morgera 91] to be further discussed below. Log-linear models have often been optimized using GIS which also defines a growth transformation [Darroch & Ratcliff 72]. Recently, GIS is replaced more and more with more efficient [Malouf 02, Minka 03] gradient-based optimization algorithms, *e.g.* QProp [Fahlman 88], Rprop [Riedmiller & Braun 93, Anastasiadis & Magoulas+ 05] and L-BFGS [Nocedal & Wright 99].

Gradient-based optimization. A good overview on gradient-based optimization algorithms can be found in [Nocedal & Wright 99]. Most of these optimization algorithms are shown to converge towards a local optimum, although at different convergence rates. These optimization algorithms can be used in batch or online mode. In speech recognition, several of these algorithms have proved to converge reasonably fast in practice. Gradient descent (GD) is mainly used in earlier work on discriminative training [Chou & Juang+ 92, Valtchev 95, Katagiri & Juang+ 98, Bauer 01, McDermott & Katagiri 97]. Experimental comparisons of Rprop, QProp, and L-BFGS can be found in [McDermott & Katagiri 05, McDermott & Hazen+ 07, Gunawardana & Mahajan+ 05, Mahajan & Gunawardana+ 06]. As shown in [Schlüter 00], EBW and GD are closely related for a suitable choice of step sizes.

Growth transformations. Growth transformations are iterative optimization algorithms that are not only convergent but also guarantee to increase the training criterion in each iteration. Although introduced with a different terminology, the simplest and most general growth transformation probably goes back to [Armijo 66]. In particular, it applies to GHMMs with floored variances and any type of log-linear models. The expectation-maximization (EM) algorithm is based on the inequality derived in [Baum 72, Dempster & Laird+ 77]. The typical application of EM is the ML training of GHMMs. EBW [Normandin & Morgera 91] may be considered the discriminative counterpart of EM. The existence of finite iteration constants was first proved for discrete-valued distributions for MMI [Normandin & Morgera 91,

Gopalakrishnan & Kanevsky+ 91, Gunawardana 01] and extended to other training criteria in the rational form, *e.g.* MCE and MWE/MPE [He & Deng+ 06]. This result was extended to real-valued densities (*e.g.* GHMMs) in [Kanevsky 04, Axelrod & Goel+ 07] (without constructive proof). Already much earlier, [Ristad & Yianilos 98b] showed the possibility of EM-style algorithms for MMI optimization of GMMs. The iteration constants guaranteeing an increase of the objective function in each iteration are expected to be too large, leading to unacceptable slow convergence. Therefore, many heuristics have been discussed how to determine good iteration constants in practice [Valtchev & Odell+ 97, Merialdo 88, Schlüter 00, Woodland & Povey 02, Povey 04, Macherey & Schlüter+ 04, Macherey 10, Hifny & Gao 08, Hsiao & Tam+ 09]. The reverse Jensen inequality leads to update rules similar to the EBW update rules [Jebara 02]. Many heuristics in setting the iteration constants in ASR can be justified with this growth transformation [Afify 05]. GIS [Darroch & Ratcliff 72] is the best known growth transformation for log-linear models. Improved iterative scaling (IIS) [Berger & Della Pietra+ 96] is a more efficient variant of GIS. The convergence properties of these algorithms are studied in [Wu 83]. After a reparameterization according to Section 1.2.3, GMMs can be optimized with a GIS-like algorithm [Saul & Lee 02]. An extension for MMI from incomplete data (*e.g.* log-linear mixtures) was proposed in [Riezler 98, Riezler & Kuhn+ 00, Wang & Schuurmans+ 02] for natural language processing with discrete-valued feature functions. Finally, many optimization problems can be solved with generalized EM (GEM) [Bishop 06, Wang & Schuurmans+ 02, p.454] by decomposing the problem into simpler subproblems and alternating optimization of these subproblems, *e.g.* mixtures of experts [Jordan & Jacobs 94]. Chapter 6 proposes two novel growth transformations, the one for log-linear models and the other for Gaussian models.

Convex optimization. Convex optimization is an important subfield of numerical optimization. It assumes convex training criteria such that any local optimum is also a global optimum. Many problems can be described in a natural way as a convex optimization problem, *e.g.* SVMs with the hinge loss function. The hidden variables of conventional acoustic models (*e.g.* HMM state sequences) make the construction of convex training criteria harder. Nevertheless, examples of convex training criteria do exist for GHMMs [Sha & Saul 06, Sha & Saul 07a, Sha & Saul 07b, Chang & Luo+ 08]. Under the same assumptions, the convexity of standard CRFs can be maintained [Kuo & Gao 06, Abdel-Haleem 06, Fosler-Lussier & Morris 08]. Similar investigations on convex optimization can be found in Chapter 7.

Chapter 2

Scientific Goals

Conventional speech recognition systems are based on Gaussian HMMs. A major conceptual point of criticism of this approach is the indirect modeling of the class posteriors, which are the key quantity in statistical pattern recognition. Log-linear models are motivated by the maximum entropy principle [Jaynes 03]. These models provide a direct parameterization of the class posteriors and thus, are expected to be more suitable for pattern recognition. The utility of log-linear models like for example conditional random fields (CRFs) has been shown in many fields of pattern recognition. So far, only little work has been done to investigate log-linear techniques for speech recognition.

The objective of this thesis is to establish a log-linear modeling framework in the context of discriminative training criteria, with examples from automatic speech recognition (ASR), part-of-speech tagging, and handwriting recognition. The theoretical and experimental goals of this work address the different aspects of a training algorithm: the choice of the model/parameterization, the training criterion, and the optimization algorithm. Namely, these include:

A comparison of Gaussian and log-linear HMMs (Chapter 4). Gaussian HMMs are generative models where the class posteriors are determined by the joint probabilities. Log-linear HMMs are discriminative models which avoid this indirection by defining directly the class posteriors. In the past, it was shown that the Gaussian models induce log-linear class posteriors. Yet, log-linear models are fully unconstrained models while Gaussian models are constrained models, *e.g.* positivity of variances or local normalization constraints of HMMs. Due to the absence of such parameter constraints, several authors have suggested that log-linear HMMs are more flexible than Gaussian HMMs. Experimental investigations on phoneme classification and recognition tasks seem to support this claim.
The present thesis establishes equivalence relations for the conventional (discriminatively) estimated Gaussian HMMs and the corresponding log-linear HMMs with first- and second-order features. Particularly for complex ASR tasks, not all requirements for an exact equivalence are typically fulfilled in practice, and the numerical stability may be an issue. For these reasons, this thesis also provides an experimental comparison of Gaussian HMMs and log-linear HMMs for various speech recognition tasks of completely different complexity.

An evaluation of the utility of the margin concept for string recognition (Chapter 5). Large margin classifiers like for example the support vector machine (SVM) are motivated by the generalization bounds from statistical learning theory [Vapnik 95]. They are the *de-facto* standard in statistical machine learning. Conventional training criteria in ASR are loss-based and do not include a margin term. To the author's best knowledge, no comprehensive study on the utility of the margin concept for string recognition has been done so far.
This thesis addresses two open issues in this context: the definition of an *efficient* margin-based training algorithm for string recognition tasks with focus on large vocabulary continuous speech recognition (LVCSR), and the *direct* evaluation of the utility of the margin term for string recognition. More precisely, the conventional training criteria including maximum mutual information (MMI), minimum classification error (MCE), and minimum phone error (MPE) are slightly modified to incorporate a margin term. To our best knowledge, this is the first approach to large margin MPE. It is shown that the resulting training criteria for log-linear models are differentiable approximations to the SVM with the respective loss function. The training criteria modified in this way are used to evaluate the utility of a margin term for string recognition across different tasks. The experimental study includes examples from ASR (with tasks from LVCSR trained on up to 1,500h audio data), part-of-speech tagging, and handwriting recognition.

An EM/GIS-style optimization algorithm for HCRFs (Chapter 6). The standard training criterion for log-linear models is MMI, *i.e.,* the log-posteriors. Traditionally, this training criterion has been optimized using generalized iterative scaling (GIS). Compared with other optimization algorithms, GIS has the additional property to improve the training criterion in each iteration.
In speech recognition, the acoustic modeling typically includes hidden variables (*e.g.* through the HMM), or MMI is not the choice of training criterion (*e.g.* MPE). These are two examples of practical interest that are not covered by standard GIS. This thesis suggests an extension of GIS to include such applications. The effectivity of the proposed optimization algorithm is tested on an optical character recognition (OCR) and a digit string speech recognition task.

Investigations on convex optimization in speech recognition (Chapter 7). Conventional training criteria in ASR are non-convex and thus, can get stuck in spurious local optima. Strictly speaking, this makes the fair comparison of different methods questionable. In addition, the conventional discriminative training in speech recognition uses many approximations and heuristics. All this leads much engineering work and expertise to make the discriminative training work well in practice.
Convex optimization appears to be a principled way to avoid such difficulties. This thesis introduces a couple of convex training criteria for speech recognition. Based on first comparative experimental results on a simple digit string recognition task, the potential of more "fool-proof" training algorithms in ASR is discussed.

Development of a transducer-based discriminative framework (Chapter 3). Standard implementations for discriminative training in speech recognition use word lattices annotated with language and acoustic model scores. In general, different training criteria use different

algorithms to calculate efficiently the gradient (*e.g.* MMI and MPE). These implementations are not suitable for the variety of string-based applications considered here.
This thesis proposes a unified implementation based on the concept of weighted finite-state transducers. The basic implementation can be used with little effort for different models (*e.g.* Gaussian and log-linear HMMs, CRFs), different training criteria (*e.g.* MMI, MCE, MPE), and across different tasks (*e.g.* ASR, part-of-speech tagging, handwriting recognition). The salient feature of our transducer-based discriminative framework is that the efficient calculation of the gradient of a broad class of training criteria including MMI, MCE, and MPE is based on the same algorithm used with different semirings. As an example, the transducer-based framework allows for a convenient implementation of the word errors on a word lattice. This result is used to compare minimum word error (MWE) using an approximate and the exact word error.

Chapter 3

Discriminative Training: A Transducer-Based Framework

This chapter provides the general setting of the transducer-based discriminative training used throughout this work. Conventional discriminative training in ASR uses word lattices, which can be represented as weighted finite-state transducers (WFSTs). The presented transducer-based framework includes word lattices but goes beyond conventional lattice-based training in a good way. Although not well established in ASR, the proposed transducer-based framework is not completely novel. Similar ideas can be found for conventional lattice-based discriminative training [He & Deng+ 08], CRFs [Lafferty & McCallum+ 01, Sutton & McCallum 07], HCRFs [Gunawardana & Mahajan+ 05], or the learning of WFSTs [Eisner 01]. Our approach implements a variety of training criteria including the well-known MMI, MCE, and MPE training criteria based on the *same* standard forward/backward (FB) algorithm [Rabiner & Juang 86]. The optimization of probabilistic and error-based training criteria merely differs in the choice of the semiring. This resembles the approach in [Eisner 01, Li & Eisner 09] where the expectation is computed with the expectation semiring, the covariance is computed with the covariance semiring *etc.* In contrast, our approach uses the probability semiring to compute the expectation, the expectation semiring to compute the covariance *etc.* This is an important difference that leads to a substantial reduction in complexity. First results within this framework are given at the end of the chapter.

3.1 Weighted Finite-State Transducers (WFSTs)

The basic definitions and concepts related to finite-state transducers (FSTs) are introduced in this section. We distinguish three major concepts in this context:

FST/WFST Definition of the set of valid strings, WFSTs are annotated with scores in addition (Section 3.1.1).

Semiring Definition of basic operations, representing *e.g.* abstract multiplication and addition (Section 3.1.2).

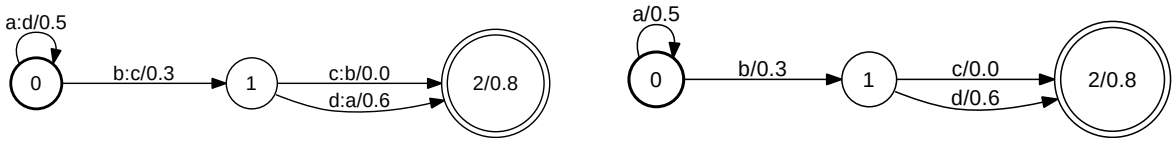


Figure 3.1: Left: WFST on the input and output alphabet $\Sigma_{in} = \Sigma_{out} = \{a, b, c, d\}$. Right: acceptor on the input alphabet $\Sigma_{in} = \{a, b, c, d\}$.

Algorithm Definition of complex operations on WFSTs, parameterized by the semiring (Section 3.1.3).

In general, different WFSTs, semirings, and algorithms are used to solve the different tasks.

3.1.1 WFSTs

We start with the basic definition of WFSTs. Here, $\mathbb{N}$ stands for the natural numbers and $\mathbb{K}$ denotes a field.

Definition 1. *A weighted finite-state transducer (WFST) is a 7-tuple*

$$T := \left(\Sigma_{in}, \Sigma_{out}, (\mathbb{K}, \oplus, \otimes, \bar{0}, \bar{1}), S, I, F, E\right)$$

where Σ_{in} is the input alphabet, Σ_{out} is the output alphabet, $(\mathbb{K}, \oplus, \otimes, \bar{0}, \bar{1})$ denotes the semiring, $S \subset \mathbb{N}$ are the states, $I \in S \times \mathbb{K}$ is the unique initial state, $F \subset S \times \mathbb{K}$ are the final states, and $E \subset S \times \{\Sigma_{in} \cup \epsilon\} \times \{\Sigma_{out} \cup \epsilon\} \times \mathbb{K} \times S$ are the edges.

Note that an acceptor is a simplified WFST which discards the output alphabet Σ_{out}. For this reason, WFST and acceptor shall not be distinguished explicitly. A few simple examples are shown in Figure 3.1. States and edges are represented by circles and arrows, respectively. The bold circle indicates the initial state and the double circles the final states. An edge is labeled with the input and output symbol, and the edge weight, *input*:*output*/*weight*.

3.1.2 Semirings

A semiring extends a field $\mathbb{K}$. In particular, it defines the basic operations for manipulating the WFSTs.

Definition 2. $(\mathbb{K}, \oplus, \otimes, \bar{0}, \bar{1})$ *is a semiring, iff*

1. $(\mathbb{K}, \oplus, \bar{0})$ *is a commutative monoid,* i.e., *i)* $(x \oplus y) \oplus z = x \oplus (y \oplus z)$*, ii)* $\bar{0} \oplus x = x \oplus \bar{0} = x$*, and iii)* $x \oplus y = y \oplus x$*;*
2. $(\mathbb{K}, \otimes, \bar{1})$ *is a monoid,* i.e., *i)* $(x \otimes y) \otimes z = x \otimes (y \otimes z)$*, and ii)* $\bar{1} \otimes x = x \oplus \bar{1} = x$*;*
3. $\otimes$ *distributes over* $\oplus$*,* i.e., *i)* $x \otimes (y \oplus z) = (x \otimes y) \oplus (x \otimes z)$*, and ii)* $(x \oplus y) \otimes z = (x \otimes z) \oplus (y \otimes z)$*;*
4. $\bar{0}$ *is an annihilator for* $\otimes$*,* i.e., *i)* $\bar{0} \otimes x = x \otimes \bar{0} = \bar{0}$.

Table 3.1: Semirings over $\mathbb{R}$ in ASR.

Semiring	$\mathbb{K}$	$x \oplus y$	$x \otimes y$	$\bar{0}$	$\bar{1}$	$\mathsf{inv}(x)$
probability	$\mathbb{R}^+$	$x + y$	$x \cdot y$	0	1	$\frac{1}{x}$
log	$\mathbb{R} \cup \{-\infty, +\infty\}$	$-\log(\exp(-x) + \exp(-y))$	$x + y$	$+\infty$	0	$-x$
tropical	$\mathbb{R} \cup \{-\infty, +\infty\}$	$\min\{x, y\}$	$x + y$	$+\infty$	0	$-x$

Table 3.2: Expectation semiring over $\mathbb{R}^+ \times \mathbb{R}$.

Semiring	$\mathbb{K}$	$(p, v) \oplus (p', v')$	$(p, v) \otimes (p', v')$	$\bar{0}$	$\bar{1}$	$\mathsf{inv}(p, v)$
expectation	$\mathbb{R}^+ \times \mathbb{R}$	$(p + p', v + v')$	$(p \cdot p', p \cdot v' + p' \cdot v)$	$(0, 0)$	$(1, 0)$	$\left(\frac{1}{p}, -\frac{v}{p^2}\right)$

The most important semirings over $\mathbb{R}$ in ASR are introduced in Table 3.1. Some algorithms require the definition of the inverse in addition. The inverse has the property that $\mathsf{inv}(x) \otimes x = \bar{1}$ for any $x \in \mathbb{K}$. Due to the commutativity of the semiring, $x \otimes \mathsf{inv}(x) = \bar{1}$ also holds true. Note that the log semiring is equivalent to the probability semiring in the negated log space. Another semiring that will become important is the expectation semiring. This semiring was proposed in [Eisner 01] to efficiently calculate expectations in the context of transducer-based MMI training. The definition of this vector semiring can be found in Table 3.2. The intuition behind this definition is that the p-component defines a probability semiring in the usual way while the v-component takes account of an additive random variable (*e.g.* word error).

Finally, a path $\pi \in E \times \cdots \times E$ is defined to connect two states by a sequence of connected edges. Here, two edges are connected iff the starting state of the one edge is identical to the ending state of the other edge. The path weight is obtained by extension of the respective edge weights, $w(\pi) := \bigotimes_{e \in \pi} w(e)$. The collected weight of different paths is defined as $\bigoplus_{\pi} w(\pi)$. Typically, transducer-based algorithms are defined on the path level. The efficient algorithms, however, are implemented locally on the edge level by making use of the properties of semirings, *e.g.* associativity and distributivity. This idea is illustrated in the next section by introducing some basic algorithms.

3.1.3 Algorithms

There is a variety of standard algorithms for transducers [Mohri 04]. WFST toolkits like for example FSA [Kanthak & Ney 04] provide implementations of these algorithms, *e.g.* composition, determinization, ϵ-removal, or union. Here, the focus shall be on a few algorithms which are relevant in the context of discriminative training, see Table 3.3 for a summary.

Composition. The composition assumes two input WFSTs, T_l and T_r. The output is also a WFST. The path weights of the resulting WFST $T_l \circ T_r$ are defined as

$$w_{T_l \circ T_r}(w_1^N, v_1^M) := \bigoplus_{u_1^L} w_{T_l}(w_1^N, u_1^L) \otimes w_{T_r}(u_1^L, v_1^M). \tag{3.1}$$

The paths are denoted by the label sequence w_1^N, v_1^M. This means that the output of the left WFST T_l must match the input of the right WFST T_r. As a consequence, the composition

realizes a mapping from sequences in the input alphabet of the left WFST to sequences in the output alphabet of the right WFST. The composed path weights are obtained by extension of the two separate path weights. For this reason, the composition can also be employed to combine different knowledge sources, *e.g.* the combination of the language model and the acoustic model scores. Applying the composition to acceptors results in the intersection of the two input acceptors because non-matching paths are discarded. An efficient implementation of this algorithm exists with complexity $O(|E_l|+|S_l|)(|E_r|+|S_r|)$ where $|E|$ and $|S|$ denote the number of edges and states [Mohri 04]. In general, this implementation only provides the correct result if one of the two input WFSTs is deterministic. Otherwise, the algorithm introduces duplicate paths which lead to incorrect edge weights in case of the log semiring, for example. In case of the tropical semiring, this duplication of paths is not critical.

Transposition. The transpose of a WFST is obtained by reversing the direction of all edges. The (single) input state is declared as the final state. A new initial state is added that has ϵ-edges to all final states.

Forward/backward (FB) scores & posteriors. The forward/backward (FB) probabilities are the basic quantities in efficient implementations of shortest path algorithms. The forward scores (also known as state potentials) of an acyclic transducer are defined as

$$\alpha(\mathit{init}) := \bar{1} \qquad \alpha(s) := \bigoplus_{\pi=(\mathit{init},s)} w(\pi) \tag{3.2}$$

where the collection is over all partial paths $\pi = (\mathit{init}, s)$ from the initial state *init* to the state s. The backward scores are defined similarly on the transposed WFST. Assuming a topological ordering, these quantities can be calculated efficiently in a recursive manner (*cf.* dynamic programming)

$$\alpha(\mathit{init}) := \bar{1} \qquad \alpha(s) := \bigoplus_{s':(s',s)\in E} \alpha(s') \otimes w(s', s). \tag{3.3}$$

Here, the collection is over all states s' such that the edge (s', s) is an edge of the WFST. This recursive implementation results in a complexity of $O(|E|)$. The backward score $\beta(\mathit{init})$ in the initial state is the collection over all path weights of the WFST under consideration. In case of the probability semiring, this quantity is identical to the sum over all path weights (*cf.* normalization constant). In case of the tropical semiring, this quantity corresponds with the shortest path score. In the first example, the backward score can be used for the normalization of the path weights, $w(\pi) \otimes \mathsf{inv}(\beta(\mathit{init}))$.

The posterior WFST is based on these FB scores. Assuming a WFST P, the edge weights of the induced posterior WFST Q are defined as

$$w_Q(e) \quad := \quad \bigoplus_{\pi\in P:e\in\pi} w_P(\pi) \otimes \mathsf{inv}\left(\bigoplus_{\pi\in P} w_P(\pi)\right) \tag{3.4}$$

which is the collection of all paths going through the edge $e \in E$, including the normalization. In terms of the FB scores, the posteriors for edge $e = (s', s)$ read

$$w_Q(e) \quad = \quad \alpha(s') \otimes w_P(e) \otimes \beta(s) \otimes \mathsf{inv}(\beta(\mathit{init})). \tag{3.5}$$

Hence, the posterior WFST Q can be calculated in $O(|E|)$. For the probability semiring, these posteriors coincide with the posterior probabilities, *e.g.* expectation-maximization (EM) for HMMs [Baum 72, Rabiner & Juang 86]. For the tropical semiring, the edge posteriors represent the shortest distance of a path through the edge under consideration and can be used to calculate the best/shortest path of WFST P, $\mathsf{best}(P)$. For other semirings like for example the expectation semiring, however, the interpretation of the posteriors may not be obvious (Section 3.5).

Pruning. The full WFSTs of interest are usually prohibitively large in ASR applications (*e.g.* word lattices). The WFSTs are then pruned to a reasonable size. FB pruning is probably the most popular approach in the context of discriminative training to reduce the size of the WFST. FB pruning discards all edges with an edge posterior (calculated with the tropical semiring) below some predefined threshold [Sixtus & Ortmanns 99]. For acyclic WFSTs (*e.g.* word lattices), the implementation based on FB scores has linear complexity.

Projection. The projection transforms a WFST into an acceptor by discarding the input or output labels.

Epsilon removal. The epsilon removal replaces a WFST by an *equivalent* WFST without any ϵ-edges. Two WFSTs are equivalent if they define the same set of (weighted) paths [Mohri 01]. The current implementation works only on acceptors.

Determinization. The determinization replaces a weighted acceptor with an equivalent weighted acceptor such that no state has two outgoing edges with the same input label. Determinization should be avoided in general because the worst case complexity is exponential [Mohri & Riley 97].

Minimization. The minimization replaces the deterministic input WFST with an equivalent deterministic WFST with the minimal number of states. The implementation assumes a deterministic input WFST [Mohri & Riley 97]. The complexity is $O(|E| \log |S|)$ for general WFSTs and $O(|E|)$ for acyclic WFSTs.

Scaling of weights. The scores w in ASR are usually scaled with some factor $\gamma \in \mathbb{R}$. This scaling of the edge weights of WFST P is performed by a utility function. The resulting WFST has the edge weights $\gamma \cdot w_P(e)$, $\forall e \in E$. For the tropical and log semirings, this produces the desired scaled probabilities, p^γ.

Traversing. In FSA, WFSTs are typically traversed with a depth first search (DFS). Specific actions can be implemented for each step of DFS.

Weight pushing Assuming the path weights, the edge weights are not uniquely defined in general. Weight pushing redistributes the edge weights of a WFST without changing the path

Table 3.3: WFST algorithms from the toolkit FSA [Kanthak & Ney 04]. WFSTs are denoted by T. Complexities are given for connected WFSTs in terms of the number of edges $|E|$ and states $|S|$.

Algorithm	Assumption	Description	Complexity
$\circ$	T_1, T_2 on same semiring with $\Sigma_{1,out} = \Sigma_{2,in}$ (in general: T_1 or T_2 deterministic)	composition of T_1 and T_2	$O(\|E_1\|\|E_2\|)$
transpose	T	reversion of all paths in T	$O(\|E\|)$
best	T on tropical semiring	best/shortest path	$O(\|E\|)$
posterior	acyclic T	(generalized) edge posteriors	$O(\|E\|)$
prune	acyclic T	elimination of edges with low posterior	$O(\|E\|)$
project$_2$	T	mapping of transducer to acceptor by discarding input labels	$O(\|E\|)$
remove-epsilon	T (acceptor)	equivalent WFST without ϵ-edges	$O(\|S\|\|E\|)$
determinize	(ϵ-free) T with *e.g.* twins property	equivalent deterministic WFST	exponential
minimize	deterministic T	equivalent deterministic WFST with minimal number of states	$O(\|E\| \log \|S\|)$ $O(\|E\|)$ (acyclic)
multiply	T on tropical or log semiring, $\gamma \in \mathbb{R}$	multiplication of edge weights with γ	$O(\|E\|)$
push-weights	T on *e.g.* tropical semiring	normalization of distribution of weights	$O(\|E\| + \|S\| \log \|S\|)$ $O(\|E\|)$ (acyclic)

weights. This might have a critical impact on the efficiency in many applications. It can be shown that a WFST after weight pushing is probabilistic, *i.e.,* the collection of the outgoing edge weights of any state is unity [Mohri 09].

3.2 Word Lattices

Word lattices represent a subspace of the full search space with the most "promising" word sequences. Compared with N-best lists, word lattices provide a compact representation of combinatorially many word sequences which can be often processed efficiently (*e.g.* WFST algorithms in Table 3.3). The word lattices can be represented as acyclic WFSTs over the lemma pronunciation alphabet. The states are annotated with word boundary information including the time frame and the acoustic context in case of across word modeling. The edge weights are set to the language model, the acoustic model, or the combined negated log-scores.

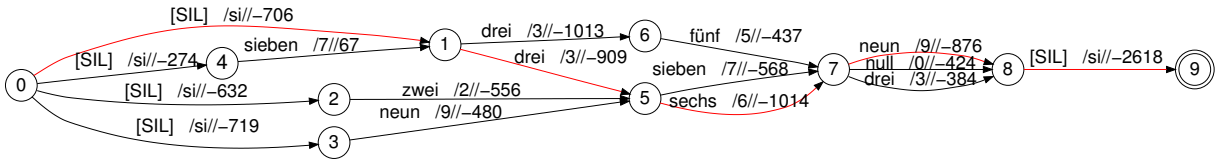


Figure 3.2: Example word lattice from SieTill (without word boundaries). The spoken digit string is "drei sechs neun" (marked in red).

Usually, the acoustic model scores only include the score of the best HMM state sequences (*cf.* Viterbi approximation). The default semiring is the tropical semiring. Depending on the task, however, it can also be a different semiring, *e.g.* the log semiring for the calculation of the FB probabilities. Word-conditioned lattices have the additional property that each state has a unique language model history. Under the given assumptions, the edge weights are well-defined. Figure 3.2 shows a real example word lattice from the digit string recognition task SieTill. The lattices are generated by a word-conditioned-tree search where the (pruned) search space is stored as WFST. The reader is referred to existing literature for the technical details of the lattice generation [Ney & Aubert 94, Ortmanns & Ney+ 97a, Macherey 10]. Throughout this work, the lattices were generated with the RWTH Aachen University speech recognition toolkit [Rybach & Gollan+ 09].

Discriminative training typically involves the summation over all competing hypotheses. For efficiency reasons, this summation space is approximated with word lattices for conventional discriminative training in ASR. Special attention must be paid to "duplicate" hypotheses which can have an impact on the discriminative training. This is the motivation for preprocessing steps like for example the filtering of silence and noise edges [Wessel & Schlüter+ 01, Wessel 02, Hoffmeister & Klein+ 06]. This is a subtle but important difference between the word lattices used for the search (*e.g.* language model rescoring) and for the training.

3.3 Unified Training Criterion

An important class of training criteria is discussed in this section. It is based on the *unified training criterion* introduced in [Schlüter & Macherey+ 01, Macherey & Haferkamp+ 05, He & Deng+ 08, Macherey 10]. For $r = 1, \ldots, R$ training utterances, the variant in [Macherey 10, Chapter 4.1] can be written as

$$\mathcal{F}(\Lambda) = \sum_{r=1}^{R} f\left(\frac{\sum\limits_{W \in \Sigma^\star} [p(W)p_\Lambda(X_r|W)]^\gamma A(W, W_r)}{\sum\limits_{W \in \Sigma^\star} [p(W)p_\Lambda(X_r|W)]^\gamma B(W, W_r)}\right). \tag{3.6}$$

Here, $f : \mathbb{R} \to \mathbb{R}$, $u \mapsto f(u)$ is some smoothing function including $\frac{1}{\gamma} \log u$ (*cf.* [Macherey 10]), $\gamma \in \mathbb{R}^+$ is some scaling factor, $\Sigma^\star$ denotes the set of word sequences assuming the vocabulary Σ, and $A, B : \Sigma^\star \times \Sigma^\star \to \mathbb{R}$ are two weight functions. Unlike in [Macherey 10, Chapter 4.1], the word sequences filter is realized by the weight function B in our formulation. Typical of the discriminative training criteria is that they involve sums over all competing word sequences,

$W \in \Sigma^\star$. This is not feasible in general. For this reason, the summation space is usually restricted to the sequences in the word lattice, see Section 3.2.

In this thesis, a transducer-based formulation shall be used. The basic quantity is the (abstract) expectation of the random variable A w.r.t. the (probabilistic) WFST P

$$E_P[A] \quad := \quad \frac{\sum\limits_{\pi \in P} w_P(\pi) w_A(\pi)}{\sum\limits_{\pi \in P} w_P(\pi)}. \tag{3.7}$$

To avoid convergence issues, acyclic WFSTs are assumed. For simplicity, WFSTs A and P shares the topology, *i.e.,* the two WFSTs only differ in the edge weights. WFSTs with different topologies can be preprocessed by intersection (implemented with composition) to satisfy this assumption. In general, P is a *pseudo* probabilistic WFST (*i.e.,* non-negative weights but without normalization). This is why the definition in Equation (3.7) includes the normalization constant. Including the dependency of P on the model parameters Λ, the unified training in Equation (3.6) can be rewritten

$$\mathcal{F}(\Lambda) \quad = \quad \sum_{r=1}^{R} f\left(\frac{E_{P_{\Lambda r}}[A_r]}{E_{P_{\Lambda r}}[B_r]}\right). \tag{3.8}$$

The random variable B w.r.t. WFST P share the topology with WFSTs A, P. This formulation of the unified training criterion is identical to the original formulation in Equation (3.6) because the normalization constant cancels. The optimal model parameters are determined by

$$\hat{\Lambda} \quad = \quad \operatorname*{argmax}_{\Lambda} \{\mathcal{F}(\Lambda)\}. \tag{3.9}$$

In Section 5.2.3, it will be shown how this unified training criterion can be extended to incorporate a margin term.

For the remainder of this chapter, a simplified variant of the unified training criterion in Equation (3.8) shall be used to keep the notational complexity at a minimum

$$\mathcal{F}(\Lambda) \quad = \quad \sum_{r=1}^{R} f\left(E_{P_{\Lambda r}}[A_r]\right). \tag{3.10}$$

Table 3.4 illustrates how the most common training criteria in ASR can be represented within the unified training criterion in Equation (3.10).

Maximum mutual information (MMI). In Table 3.4, $\mathbf{1}_{\mathsf{spk}}$ stands for the indicator function of the spoken hypothesis. The indicator function has the value 1 at points of the set spk and 0 otherwise. The logarithmic function is chosen for the smoothing function.

Power approximation (POW). MMI is based on the logarithm which diverges for vanishing probabilities, $\log u \overset{u \to 0}{\to} \infty$. This might cause problems with outliers. To avoid this divergence, the power identity

$$\log u \quad = \quad \lim_{\kappa \to 0} \frac{u^\kappa - 1}{\kappa}$$

Table 3.4: Important probabilistic and error-based training criteria in ASR as instances of the unified training criterion in Equation (3.10), L_Λ is defined and used only in Section 3.4.

Identifier	A	$f(u)$	$f'(u)$	L_Λ
MMI	$\mathbf{1}_{\text{spk}}$	$\log u$	$\frac{1}{u}$	$E_{p_\Lambda}[\mathbf{1}_{\text{spk}}]^{-1}\mathbf{1}_{\text{spk}}$
POW	$\mathbf{1}_{\text{spk}}$	$\frac{u^\kappa - 1}{\kappa}$	$u^{\kappa-1}$	$E_{p_\Lambda}[\mathbf{1}_{\text{spk}}]^{\kappa-1}\mathbf{1}_{\text{spk}}$
MCE	$\mathbf{1}_{\text{spk}}$	$\sigma_\beta(u)$	$\frac{\sigma_\beta(u)(1-\sigma_\beta(u))}{u(1-u)}$	$f'(E_{p_\Lambda}[\mathbf{1}_{\text{spk}}])\mathbf{1}_{\text{spk}}$
MWE	A_{word}	u	1	A_{word}
MPE	A_{phone}	u	1	A_{phone}

is used to approximate the logarithm. This approximation is termed *power approximation* (POW). In contrast to the logarithm, the power approximation is bounded below for $u > 0$. Although derived from a probabilistic training criterion, the power approximation resembles an error-based training criterion.

Minimum word/phoneme error (MWE/MPE). Like minimum Bayes risk (MBR) training in general [Kaiser & Horvat+ 02, Doumpiotis & Byrne 05, Gibson & Hain 06], the MWE/MPE training criterion is the expectation of some error measure. The approximate word/phoneme accuracy according to [Povey 04] are denoted by $A_{\text{word}}/A_{\text{phone}}$ and define MWE/MPE. The smoothing function is set to the identity function.

Minimum classification error (MCE). In this training criterion, the sigmoid function $\sigma_\beta : \mathbb{R} \to [0,1],\ u \mapsto \frac{u^\beta}{u^\beta + (1-u)^\beta}$ is used for the smoothing function. It is used to approximate the step function representing the ideal classification error. The parameter $\beta \in \mathbb{R}^+$ controls the smoothness of the approximation [Juang & Katagiri 92, McDermott & Katagiri 97, Schlüter & Macherey+ 01, Macherey & Haferkamp+ 05].

In all these examples, P is set to the (scaled) joint probabilities. The training criteria from Table 3.4 are plotted in Figure 3.3 (left-hand side) as a function of $p(c_n|x_n)$ for a binary classification problem.

3.4 Gradient of Unified Training Criterion

The training criteria in ASR are typically optimized with a gradient-based optimization algorithm. For this reason, it is important that the gradient of the unified training criterion in Equation (3.10) can be efficiently calculated on WFSTs. Define the covariance between the two random variables X and Y (represented as WFSTs) as

$$Cov_P(X,Y) := \frac{\sum_{\pi \in P} w_P(\pi)\,(w_X(\pi) - E_P[X]) \cdot (w_Y(\pi) - E_P[Y])}{\sum_{\pi \in P} w_P(\pi)}. \tag{3.11}$$

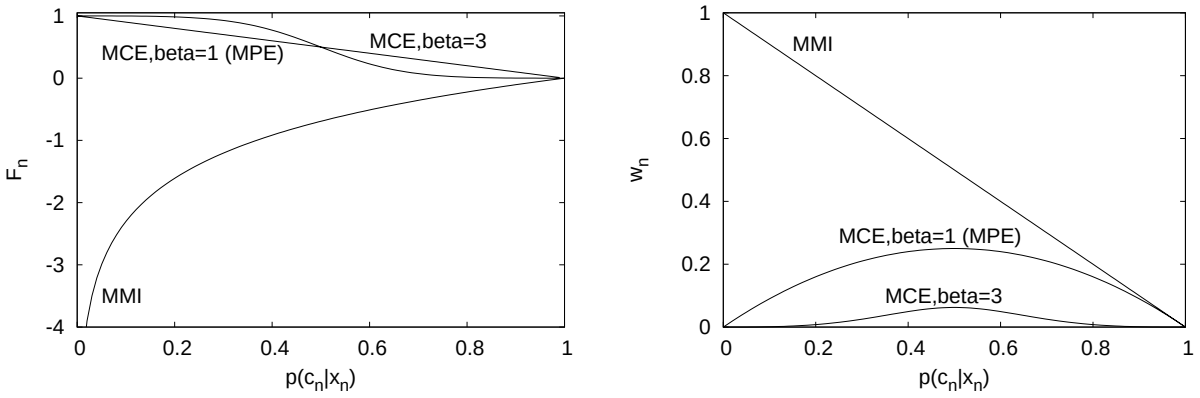


Figure 3.3: Illustration of a few training criteria for binary classification and i.i.d. data. Left: training criterion *vs.* $p(c_n|x_n)$. Right: accumulation weight w_n *vs.* the posterior of the correct class $p(c_n|x_n)$. The competing class has the same weight but with opposite sign. ML uses uniform accumulation weights, independent of $p(c_n|x_n)$.

Together with the shortcut $L_\Lambda := f'(E_{P_\Lambda}[A])A$, the gradient of the unified training criterion in Equation (3.10) w.r.t. the (free) model parameters Λ can be written as

$$\nabla\mathcal{F}(\Lambda) \quad = \quad \sum_{r=1}^{R} Cov_{P_{\Lambda r}}(L_{\Lambda r}, \nabla \log P_{\Lambda r}). \tag{3.12}$$

In this identity, $\nabla \log P_\Lambda$ stands for the WFST sharing the topology with P_Λ but with the gradient of $\log P_\Lambda$ as the edge weights, *e.g.* $\nabla \log P_\Lambda(e) \leftarrow \sum_t \nabla \log p_\Lambda(x_t, s_t)$.

The unified training criterion can be interpreted as a weighted maximum likelihood (ML) accumulation with the weights w_π defined by

$$\begin{aligned} \nabla\mathcal{F}(\Lambda) &\overset{\text{Equation (3.12)}}{=} \sum_{\pi\in P} f'(E_P[A]) \frac{w_P(\pi)}{\sum_{\pi'\in P} w_P(\pi')} (w_A(\pi) - E_P[A]) \cdot w_{\nabla \log P}(\pi) \\ &=: \sum_{\pi\in P} w_\pi \cdot w_{\nabla \log P}(\pi) \end{aligned}$$

(dependency on r and Λ are dropped for simplicity). This allows for a different illustration of the training criteria, providing additional insight into the differences of the different training criteria. Figure 3.3 (right-hand side) plots the accumulation weight w_n - subscript π substituted with n to indicate independent and identically-distributed (i.i.d.) observations - *vs.* the posterior of the correct class $p(c_n|x_n)$. The quantities f, f' and A, L can be found in Table 3.4.

The problem of calculating the gradient has been reduced to the calculation of a transducer-based covariance. Obviously, the efficient calculation of the covariance can be also used for more complex expectation-based training criteria than in Equation (3.10) (*e.g.* MCE on state level). Moreover, the covariance is a basic quantity in statistics that occurs in many different contexts. For this reason, this is a useful feature of any probabilistic transducer library. Last

but not least, the unified training criterion can be generalized to incorporate a margin term, see Chapter 5.

The next section shows how n-th order statistics for probabilistic transducers and random variables represented as WFSTs can be calculated efficiently.

3.5 Efficient Calculation of N-th Order Statistics

[Eisner 01] proposed an elegant way for the network-based optimization using MMI. The algorithm is based on the expectation semiring and the following identity.

Proposition 3. *Assume an acyclic WFST P over the probability semiring, and a WFST X over the log semiring. P and X share the topology. Define the acyclic WFST Z to have the same topology as P, X and the edge weights $w_Z(e) = (w_Z(e)[p], w_Z(e)[v])$ with $w_Z(e)[p] := w_P(e)$ and $w_Z(e)[v] := w_P(e)w_X(e)$ over the expectation semiring. Then,*

$$E_P[X] = \frac{\beta(init)[v]}{\beta(init)[p]}.$$

The $p-$ and v-components of the backward score over the expectation semiring in the initial state based on Z are denoted by $\beta(init)[p]$ and $\beta(init)[v]$ as introduced in Section 3.1.3.

The proof of this lemma can be found in [Eisner 01].

For training, an expectation is calculated for each segment and each (active) feature. In ASR, the accumulation of the MMI statistics is based on another identity for the expectation where the sum over all paths in the WFST, $\pi \in P$, is replaced by a sum over all edges, $e \in P$. This leads to a more efficient calculation of the gradient, *e.g.* [Schlüter 00].

Proposition 4. *Assume an acyclic WFST P over the probability semiring, and a WFST X over the log semiring. P and X share the topology. Let $Q(P)$ be the posterior WFST induced by P as defined in Equation (3.4). Then,*

$$E_P[X] = \sum_{e \in P} w_X(e) w_{Q(P)}(e).$$

Proof. The identity is proved by rearranging terms

$$
\begin{aligned}
E_P[X] &\overset{\text{Equation (3.7)}}{:=} \sum_{\pi\in P} \frac{w_P(\pi)}{\sum_{\pi'\in P} w_P(\pi')} w_X(\pi) \\
&\overset{\text{additivity of } X}{=} \sum_{\pi\in P} \frac{w_P(\pi)}{\sum_{\pi'\in P} w_P(\pi')} \sum_{e\in\pi} w_X(e) \\
&\overset{\sum_{e\in\pi} = \sum_{e\in P} \delta(e\in\pi)}{=} \sum_{\pi\in P} \sum_{e\in P} \delta(e\in\pi) \frac{w_P(\pi)}{\sum_{\pi'\in P} w_P(\pi')} w_X(e) \\
&= \sum_{e\in P} \sum_{\pi\in P} \delta(e\in\pi) \frac{w_P(\pi)}{\sum_{\pi'\in P} w_P(\pi')} w_X(e) \\
&= \sum_{e\in P} \underbrace{\sum_{\pi\in P: e\in\pi} \frac{w_P(\pi)}{\sum_{\pi'\in P} w_P(\pi')}}_{\overset{\text{Equation (3.4)}}{=:} w_{Q(P)}(e)} w_X(e) \\
&= \sum_{e\in P} w_{Q(P)}(e) w_X(e).
\end{aligned}
$$

□

Interesting about this identity is that the sum over the paths can be replaced by a sum over the edges. The goal of this section consists of deriving a similar identity for the covariance. For this purpose, Proposition 4 is extended to the expectation semiring. Keep in mind that for the p-component, the previous proposition is recovered because the p-component is identical to the probability semiring.

Proposition 5. *Assume an acyclic WFST P over the probability semiring, and WFSTs X and Y over the log semiring. P, X, and Y share the topology. Define the WFST Z over the* expectation semiring *and assign the weights $w_Z(e) = (w_Z(e)[p], w_Z(e)[v])$ with $w_Z(e)[p] := w_P(e)$ and $w_Z(e)[v] := w_P(e) w_X(e)$ to the edges of Z. Then,*

$$
Cov_P(X, Y) = \sum_{e\in Y} w_Y(e) w_{Q(Z)}(e)[v].
$$

In other words, the expectation semiring is used to calculate efficiently the covariance in this identity. This contrasts Proposition 3 where the expectation semiring is used for the calculation of the expectation instead.

Proof. It can be shown that the covariance transforms into

$$
Cov_P(X, Y) = \sum_{e\in Y} w_Y(e) \sum_{\pi\in Y: e\in\pi} \frac{w_P(\pi)}{\beta(init)[p]} \left(w_X(\pi) - \frac{\beta(init)[v]}{\beta(init)[p]} \right).
$$

Observe that the normalization constant and the expectation are expressed in terms of the backward score in the initial state, see Section 3.1.3 and Proposition 3 for further details. The

proof of this identity is similar to the proof of Proposition 4. Hence, it suffices to show that the inner sum of the right-hand side of this equation equals the edge posterior of Z, $w_{Q(Z)}(e)[v]$

$$\sum_{\pi \in Z : e \in \pi} \frac{w_P(\pi)}{\beta(init)[p]} \left(w_X(\pi) - \frac{\beta(init)[v]}{\beta(init)[p]} \right) \overset{\text{definition of } Z}{=} \sum_{\pi \in Z : e \in \pi} \frac{w_Z(\pi)[p]}{\beta(init)[p]} \left(\frac{w_Z(\pi)[v]}{w_Z[\pi][p]} - \frac{\beta(init)[v]}{\beta(init)[p]} \right)$$

$$= \frac{\bigoplus_{\pi \in Z : e \in \pi} w_Z(\pi)[v]}{\beta(init)[p]} - \frac{\bigoplus_{\pi \in Z : e \in \pi} w_Z(\pi)[p] \cdot \beta(init)[v]}{\beta(init)[p]^2}.$$

Applying the identity $((p_1, v_1) \otimes \mathsf{inv}(p_2, v_2))[v] = \frac{v_1}{p_2} - \frac{p_1 v_2}{p_2^2}$ with $(p_1, v_1) := \bigoplus_{\pi \in Z : e \in \pi} w_Z(\pi)$ and $(p_2, v_2) := \beta(init)$ to the last expression, leads to

$$\frac{\bigoplus_{\pi \in Z : e \in \pi} w_Z(\pi)[v]}{\beta(init)[p]} - \frac{\bigoplus_{\pi \in Z : e \in \pi} w_Z(\pi)[p] \cdot \beta(init)[v]}{\beta(init)[p]^2} = \left(\bigoplus_{\pi \in Z : e \in \pi} w_Z(\pi) \otimes \mathsf{inv}(\beta_{init}) \right)[v]$$

$$= w_{Q(Z)}(e)[v].$$

This concludes the proof. □

In practice and similar to the semiring pair probability/log, the expectation semiring is replaced by an equivalent but numerically more stable formulation. For ASR word lattices, this variant reduces to the recursion formula introduced in [Povey & Woodland 02] and used for MWE/MPE.

In summary, the value of expectation-based training criteria can be computed efficiently with the probability semiring. Similar relations for the gradient of the training criterion (*i.e.,* the covariance and the expectation semiring) were established. In general, n-th order derivatives of the training criterion include $n + 1$-st order statistics which can be calculated efficiently by a n-th order semiring similar to the expectation semiring. In numerical optimization, for instance, advanced algorithms such as conjugate gradient (CG) and Newton methods [Nocedal & Wright 99] use the Hessian matrix (*i.e.,* the second derivatives) for refining the step sizes. The "covariance semiring" would be the appropriate semiring in this case.

3.6 Transducer-Based Implementation

Now, we are in the position to describe our transducer-based implementation for the discriminative training. The implementation is based on the WFST library FSA [Kanthak & Ney 04]. Special about this implementation is that the training criteria represented by the unified training criterion in Equation (3.10) including MMI, MCE, and MPE share the same algorithm but in combination with different semirings. The theoretical foundation for the implementation is provided in the previous sections. In ASR, assume the two WFSTs P_{AM} (acoustic model) and P_{LM} (language model) from a recognition or rescoring pass. Typically, a weak unigram language model is used for discriminative training [Schlüter & Müller+ 99, Schlüter 00]. Table 3.5 exemplifies the different steps. The joint probability $P_{LM} \circ P_{AM}$ can be scaled by a factor $\gamma \in \mathbb{R}$ [Wessel & Macherey+ 98, Woodland & Povey 00]. The posterior WFST Q is computed over the criterion-specific semiring. This posterior WFST is then used for the accumulation of the discriminative statistics. In particular, MMI and MPE only differ in the choice of the semiring for the posterior calculation in Table 3.5. All remaining steps are identical.

Table 3.5: Comparison of MMI and MPE in our transducer-based implementation. WFST (P, A) over the expectation semiring has the edge weights $w_{(P,A)}(e) := (w_P(e), w_P(e)w_A(e))$. The accumulation is implemented by a depth first search (DFS).

	MPE	MMI
P	multiply$(P_{LM} \circ P_{AM}, \gamma)$	
Z	(P, A) (over expectation semiring)	P (over probability semiring)
Q	posterior$(Z)[v]$	posterior(Z)
Accumulation	For each edge e and for each time frame t: Accumulate feature x_t with weight $w_Q(e)$ for state s_t.	

3.7 Error Metrics

The error-based training criteria of the type

$$\mathcal{F}(\Lambda) = \sum_{r=1}^{R} E_{P_{\Lambda r}}[A_r] \tag{3.13}$$

are an important subclass of the unified training criterion in Equation (3.10). In this case, the training criterion represents some smooth approximation to the non-differentiable true empirical risk $\sum_{r=1}^{R} A_r$. For the efficient error-based training on lattices, the string errors need to be represented as a WFST with the same topology as the word lattice holding the joint probabilities. The word error rate is the conventional measure to evaluate speech recognition systems. Thus, the exact Levenshtein distance on word level is expected to perform best. The errors can be defined on different levels, leading to different training criteria like for example MCE (utterance), MWE (word), MPE and minimum phoneme frame error (MPFE) [Zheng & Stolcke 05b] (phoneme), *etc.* Some important metrics in the context of speech recognition are discussed in the next section.

3.7.1 Hamming distance

The Hamming distance is a metric between two strings of the same length. This metric counts the number of positions in which the corresponding symbols are different. Opposed to the Hamming distance, the Hamming accuracy is the number of matching positions, *e.g.* $A(w_1^N, v_1^N) := \sum_{n=1}^{N} \delta(w_n, v_n)$.

3.7.2 Edit distance between two strings

Let Σ be a finite alphabet of distinct symbols, and let $\Sigma^\star$ denote the set of all possible strings given Σ. The set of local edit operations is defined as the set $\mathcal{E} = \Sigma \times \Sigma \cup \Sigma \times \{\epsilon\} \cup \{\epsilon\} \times \Sigma$. Each local edit operation is assigned cost $c : \mathcal{E} \to \mathbb{R}$. Furthermore, an element $\pi \in \mathcal{E}^\star$ is called an *alignment* of the strings $V, W \in \Sigma^\star$ if $h(\pi) = (V, W)$ for the corresponding morphism

$h : \mathcal{E}^\star \rightarrow \Sigma^\star \times \Sigma^\star$. Then, the edit distance between these two strings is defined as

$$d(V, W) := \min_{\pi \in \mathcal{E}^\star : h(\pi) = (V,W)} \left\{ \sum_i c(\pi_i) \right\} \tag{3.14}$$

where π_i are the local edit operations of π. The Levenshtein distance is recovered if all local costs are set to unity except for matches which have zero cost [Levenshtein 66]. The Levenshtein distance on word level is typically used to assess speech recognition systems. The edit distance of two strings is solved efficiently by dynamic programming. The complexity of the resulting algorithm is $O(|V| \cdot |W|)$.

The definition in Equation (3.14) can be extended to the edit distance between two *sets* of strings, A_1 and A_2

$$d(A_1, A_2) := \min_{V \in A_1, W \in A_2} \{d(V, W)\}. \tag{3.15}$$

Setting A_1 to the correct word sequence(s) and A_2 to the hypotheses in the word lattice, the edit distance in Equation (3.15) calculates the graph error. The edit distance of two sets of strings can be calculated efficiently similarly to the two string case. The complexity of the resulting algorithm is $O(|E_1| \cdot |E_2|)$.

The edit distances can also be calculated with standard WFST algorithms. The *edit distance transducer* L is a WFST that defines the alignments $\mathcal{E}^\star$ with the costs, *i.e.,* each edge represents a local edit operation with the respective cost as the edge weight. The Levenshtein distance transducer is illustrated in Figure 3.4. The empty symbol ϵ is used to encode deletions and insertions. The two sets A_1, A_2 are represented as unweighted WFSTs, *i.e.,* WFSTs with all

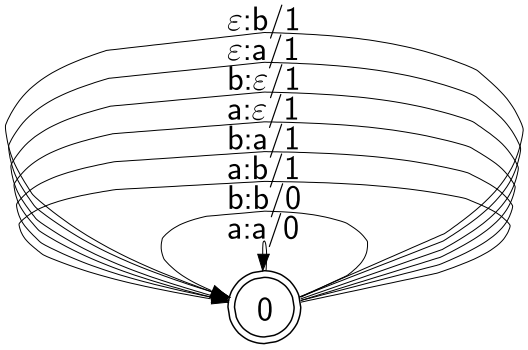


Figure 3.4: Levenshtein distance transducer for the alphabet $\Sigma = \{a, b\}$.

edge weights set to zero. Then, the alignments between A_1 and A_2 are extracted from L by composition. More precisely, the WFST

$$A_1 \circ L \circ A_2 \tag{3.16}$$

contains all alignments of A_1 and A_2 defined by the edit distance transducer L [Mohri 03]. The edit distance between A_1 and A_2, for example, is calculated efficiently by means of a single-source shortest-path algorithm such as best from FSA [Kanthak & Ney 04] using the tropical semiring, *i.e.,*

$$d(A_1, A_2) = \mathsf{best}(A_1 \circ L \circ A_2) \tag{3.17}$$

returns the alignment with the lowest cost [Mohri 03].

Note that this concept is general: any weighted transducer without negative cycles (*i.e.,* cycles with negative weight) might be substituted for the edit distance transducer. As a variant of the classical Levenshtein distance, for example, the weights of the edit distance transducer may be set to the values, estimated from a stochastic model for the edit distance [Ristad & Yianilos 98a].

3.7.3 Edit distances on WFSTs

The optimization of the error-based training criteria in Equation (3.13) requires the edit distance calculations between the reference and all competing hypotheses in the WFST. Hence, the goal is to find an algorithm that calculates all pairwise edit distances, avoiding duplicate calculations and storing the result in a compact way as far as possible. For our purposes, it is enough to have an algorithm that performs "efficiently" on the typical instances from ASR and not necessarily on the worst case scenario. The transducer-based approach appears interesting in this context because of its (usually) compact representation of combinatorially many sequences.

The problem under consideration is similar to the problem in Section 3.7.2. Instead of finding the shortest distance of any two strings in A_1 and A_2, however, all distances between the reference(s) A_1 and any string in A_2 (*e.g.* word sequences in the word lattice) are required. More formally, assuming two unweighted FSTs, A_1 and A_2, find a WFST with edge weights $w(e)$ such that the weight of any path π representing string W satisfies

$$w(\pi) \quad = \quad d(A_1, W). \tag{3.18}$$

This means that the edge weights are distributed over the WFST such that the accumulated edge weights provide the edit distance for each string in A_2 given the reference(s) in A_1. The weight of path π is obtained by summing up the corresponding edge weights. In general, the topology of A_2 needs to be modified to achieve this property. A transducer-based solution to this problem is presented next. The WFST algorithms used are summarized in Table 3.3.

Proposition 6. *Assume the edit-distance WFST L and two acyclic FSTs A_1, A_2, all over the* tropical *semiring. Then, the WFST*

$$\mathsf{determinize}(\mathsf{remove\text{-}epsilon}(\mathsf{project}_2(A_1 \circ L \circ A_2)))$$

is well-defined and satisfies $w(W) = d(A_1, W)$, $\forall W \in A_2$.

Proof. First, $A_1 \circ L \circ A_2$ is acyclic since A_2 is acyclic by assumption. According to [Allauzen & Mohri 03], any acyclic WFST has the twins property and, thus can be determinized, *i.e.,* the WFST is well-defined. Second, the determinization produces a deterministic WFST that is equivalent to the input WFST over the given, *i.e.,* the tropical semiring. A deterministic FST has the properties [Schützenberger 77]:

- a unique initial state;
- there exists at most one edge labeled with any label of the alphabet at each state.

This definition implies that any string in the deterministic WFST is unique. From these observations, the correctness of the algorithm follows. □

According to Table 3.3, the proposed algorithm has exponential complexity due to the determinization. Despite this exponential worst case complexity, a few optimizations can be done to make the algorithm efficient enough for practical purposes.

The edit distance transducer has a single state, but it has $|\Sigma|^2$ edges if $|\Sigma|$ is the alphabet size, see Figure 3.4. In ASR with large vocabularies this is prohibitive. For this reason, the vocabulary is restricted to the words occurring in A_1. In addition, an "out-of-vocabulary" word is introduced onto which all words of A_2 that do not appear in A_1, are mapped. Thereby, different word sequences may be mapped onto the same word sequence. For the training, however, all word sequences of the word lattice are required. Thus, the word sequences (and word boundaries) of the word lattice are recovered afterwards with an algorithm that performs similarly to the composition.

A simple optimization is to first minimize A_1 and A_2. This speeds up the algorithm in the context of discriminative training significantly. Several other optimizations are possible (*e.g.* pruning), which, however, do not guarantee the exactness of the algorithm in general.

3.7.4 Approximate accuracies on WFSTs

The previous investigations suggest that the calculation of the exact Levenshtein distances on a WFST has exponential complexity. Next, three approximations to the Levenshtein distance are discussed to avoid the exponential complexity. The approximations reduce the complexity by restricting or ignoring the edit distance alignment problem.

Beam-pruned Levenshtein distance. The calculation of the Levenshtein distance is basically a search problem over the alignments. Thus, reducing the search space will make the determinization in Lemma 6 more efficient. Levenshtein distances that are approximated by pruning are always an upper bound to the exact Levenshtein distance. Pruning with a limited beam, for instance, guarantees that the determinization can be performed in polynomial time.

Approximate word/phoneme accuracies. Another more pragmatic approach to approximate the Levenshtein distance was suggested in [Povey & Woodland 02, Povey 04]. This approximation is based on the notion of accuracy. The local accuracy operations are assigned the costs

$$c(\pi_i) \quad := \quad \begin{cases} 1 & \text{if } \pi_i = (w_i, w_i),\ w_i \in \Sigma \text{ (match)} \\ 0 & \text{if } \pi_i = (v_i, w_i),\ v_i, w_i \in \Sigma \wedge v_i \neq w_i \text{ (substitution)} \\ -1 & \text{if } \pi_i = (\epsilon, w_i),\ w_i \in \Sigma \text{ (insertion).} \end{cases} \tag{3.19}$$

The accuracy is defined similarly to the edit distance in Equation (3.14) and (3.15) where the min operation is replaced with the max operation. The accuracy and the edit distance are equivalent in the discriminative training due to the identity

$$A(V, W) \quad = \quad |V| - d(V, W). \tag{3.20}$$

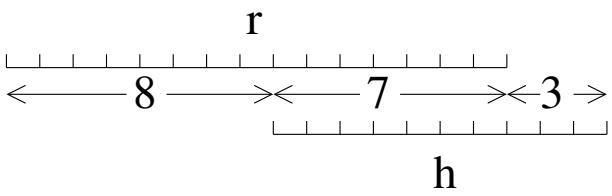


Figure 3.5: Illustration of temporal overlap, $o(r,h) = \frac{7}{15}$ in this example if h and r have the same label and zero otherwise.

The approximate accuracy assumes a time segmentation of the tokens. This decision avoids the Levenshtein alignment such that the local costs can be simply summed up to obtain the total path accuracy. The temporal overlap $o(h,r)$ of the reference r and the hypothesis h is the ratio between the number of frames shared by the reference and the hypothesis and the total number of reference frames if they have the same label, and zero otherwise. See Figure 3.5 for an example. The local approximate accuracy can be defined in terms of this temporal overlap

$$c(h,r) := \begin{cases} -1 + 2o(h,r) & \text{if } h \text{ and } r \text{ have same label} \\ -1 + o(h,r) & \text{otherwise,} \end{cases} \tag{3.21}$$

and the accuracy of hypothesis h then reads

$$c(h) := \max_r \{c(h,r)\} \tag{3.22}$$

[Povey 04]. Silence and noise hypotheses are discarded by setting the local accuracy $c(h,r)$ to zero if h represents silence or noise. This has proved to perform slightly better in practice than treating silence and noises like regular hypotheses.

Using word lattices with word boundaries, this approximation leads to an efficient algorithm because of the strictly local definition. The approximate accuracy can be defined on different token levels. Typical choices are the word (*cf.* MWE) and the phoneme level (*cf.* MPE).

Frame error. Finally, the frame error should be mentioned. This metric uses a state-based Hamming distance to avoid the alignment problem (Section 3.7.1). The labels are not necessarily defined on the state level [Wessel 02]. The frame error has been employed in discriminative training in slightly different variants [Zheng & Stolcke 05b, Povey & Kingsbury 07, Gibson 08].

3.8 Experimental Results

The discriminative framework described above is tested by investigating several basic issues in discriminative training, *e.g.* the choice of training criterion or optimization algorithm. The detailed descriptions of the different tasks and setups can be found in Appendix A. Unlike the systems of most other sites, RWTH Aachen University uses globally pooled variances, leading to highly competitive ML baseline systems consisting of a fairly high number of Gaussian densities. The software tools used for the experiments in this work are part of the

Table 3.6: Different training criteria, WER [%] on EPPS English.

Criterion	WER [%]		
	Dev06	Eval06	Eval07
ML	14.4	10.8	12.0
MMI	13.8	11.0	12.0
MCE	13.8	11.0	11.9
MPE	13.4	10.2	11.5

RWTH Aachen University speech recognition toolkit [Rybach & Gollan+ 09]. The software used in [Macherey 10] was the starting point for the development of these tools. The currently used MPE implementation is based on *word* and not *phoneme* lattices. The approximate phoneme accuracies are calculated as proposed in [Povey & Woodland 02] (Section 3.7.4). The (accumulated) phoneme accuracies are then represented in the original word lattice. This is in contrast to other MPE implementations, including [Macherey 10]. Note that this approach results in significantly reduced memory requirements because only the word and not phoneme lattices need to be stored.

3.8.1 Comparison of conventional training criteria

An in-depth experimental comparison of training criteria including ML, MMI, MCE, and MPE (see Table 3.4) can be found in [Macherey 10, Chapter 6]. In this thesis, a few additional comparative results are shown for the EPPS English task. Opposed to [Macherey 10] where MCE performed best, the error rates in Table 3.6 suggest that using our current settings, MPE with I-smoothing is the discriminative training criterion of choice. A similar tendency was observed on other tasks as well, see *e.g.* Table 5.6. Adding I-smoothing to MMI or MCE, leads to slightly more balanced results: $+0.1\%$ on the tuning corpus ('Dev06') and $-0.2\%/-0.1\%$ on the two test corpora ('Eval06'/'Eval07'). For this reason, MMI and MCE were not further pursued for LVCSR tasks.

3.8.2 Comparison of MWE with approximate and exact word errors

Using the exact Levenshtein distance in lieu of the approximate word accuracy in the MWE framework [Povey & Woodland 02], the quality of the approximation can be assessed [Heigold & Macherey+ 05]. This is done by comparing the performance of MWE with the approximate (Section 3.7.4) and exact (Section 3.7.3) word accuracies.

The calculation of the exact word accuracies on word lattices is based on the algorithm given in Section 3.7.3. To make the accumulation of the discriminative statistics efficient, the word lattices need to be modified such that the word accuracies can be incorporated into the lattices without losing the information used for the acoustic rescoring, *e.g.* the word boundaries. First, the tokens used to evaluate the word accuracies are not identical with the pronunciations stored in the word lattice. The corresponding mapping is accomplished by composing the lattices with a suitable transducer, *cf.* Paragraph "Composition" in Section 3.1.3. Then, the word lattice with the word accuracies is obtained by composing the original word lattice and the

Table 3.7: Word graph densities for the training lattices, before and after incorporating the Levenshtein distance. 4% of the edges are silence edges.

WSJ0+WSJ1	Word lattice	+Levenshtein distance
Average number of edges per spoken word	59	67
Average number of edges per time frame	31	35

Table 3.8: Word error rate (WER) on the North American Business (NAB) corpus for the approximate (MWE) and the exact (exactMWE) approach.

	WER [%]			
	NAB 20k		NAB 65k	
Corpus	Dev	Eval	Dev	Eval
ML	11.36	11.43	9.14	9.24
MWE	11.17	10.83	8.85	8.88
exactMWE	11.10	10.90	8.85	8.99

weighted transducer containing the word accuracies. As the composition is based on the state mapping (q_1, q_2) and $(q_1', q_2') \rightarrow ((q_1, q_2), (q_1', q_2'))$ the word boundaries (*e.g.* times) *etc.* can be recovered easily. It is important to avoid duplicate hypotheses (*i.e.,* identical word sequence *and* time alignment) in the resulting word lattice. Duplicate hypotheses in the lattice would change the summation space for the posterior probabilities entering the accumulation. To ensure this property, the WFST with the word accuracies needs to be determinized before the composition.

In general, the composition can split WFST states, increasing the size of the word lattices. The increase of the word lattices is small as shown in Table 3.7. In spite of the exponential worst case complexity of the algorithm in Proposition 6, this algorithm turned out to perform rather efficiently as long as the word lattices are not too dense and as long as the sentences are not too long, say fewer than 50 words in case of WSJ. The word error rates in Table 3.8 suggest that the approximate word accuracies are a sufficiently good approximation to the exact word accuracies.

3.8.3 Comparison of optimization algorithms

Numerical optimization is a crucial issue in discriminative training. The choice of the optimization algorithm can affect the performance in terms of convergence speed, memory requirements, and error rate. Conventionally, the extended Baum Welch (EBW) algorithm is used to optimize the discriminative training criteria for GHMMs in ASR. The convergence speed of EBW is controlled by the iteration constants. It can be proved that for GHMMs, finite iteration constants exist [Axelrod & Goel$^+$ 07], see also Section 6.2.2. In practice, several heuristics are employed to set the iteration constants such as to make EBW feasible. Typically, the Gaussian mixture weights are optimized using a different scheme. (Empirical) EBW appears in different variants. In this work, the version of EBW as proposed in [Macherey 10, Section 4.3] is used. For globally pooled variances, the iteration constants for EBW tend to be over-pessimistic (10-20 iterations until convergence), compared with the results for density-specific variances reported by other groups (<5 iterations until convergence). Different gradient-based

optimization algorithms like for example probabilistic gradient descent, L-BFGS, and Rprop [McDermott & Katagiri 05] have been rarely employed in this context, or compared with EBW [Gunawardana & Mahajan+ 05].

Here, the general purpose optimization algorithm Rprop [Riedmiller & Braun 93] is compared with the highly specialized EBW [Macherey 10]. Rprop has several advantages over EBW.

Generality. Unlike (empirical) EBW which is only applicable to GHMMs, Rprop is a general-purpose optimization algorithm for continuously differentiable training criteria, including GHMMs and log-linear models.

Memory requirement. Numerator and denominator statistics need to be stored separately for EBW because the determination of the iteration constants relies on this information. Rprop does not need to distinguish the contributions from the numerator and the denominator parts. Hence, the memory requirements for Rprop are approximately half of that for EBW. For large acoustic models (up to almost 1G), this leads to significantly reduced I/O which is typically a bottleneck in parallel computing.

Implementation. Rprop is a simple algorithm with a simple implementation. EBW is much more sophisticated and involves more heuristic parameters that may be tuned (although usually not done in practice).

Statistics canceling. It was shown that the canceling of any shared part of the numerator and denominator statistics on each frame may refine EBW [Povey & Kanevsky+ 08]. The gradient of the training criterion is the difference of numerator and denominator statistics. Hence, (explicit) cancellation is not required for gradient-based algorithms like for instance for Rprop.

Well-definedness. A more subtle problem with EBW arises from the ambiguity of the Gaussian model parameters in the discriminative formulation, see Section 4.3.4. In particular, the globally pooled variances are fully undetermined (the situation for specific variances is similar). Yet, EBW uses the variances to determine the iteration constants. An analogous argument applies to the mixture weights. This means that the initialization of the GHMM for discriminative training does have an impact on the convergence speed because of suboptimal iteration constants - besides the fact that only the global variances enter the iteration constants. Gradient-based optimization algorithms do not suffer from this problem because the gradient is perpendicular to equipotential hypersurfaces induced by these invariances.

Convergence. Under rather mild assumptions (*e.g.* the gradient of the training criterion must be Lipschitz-continuous), Rprop is guaranteed to converge to a local optimum [Anastasiadis & Magoulas+ 05]. No rigorous convergence proof for the empirical EBW as used in practice is known to the author. In particular, it is not known whether EBW prevents from convergence to a non-critical point due to too small step sizes.

Table 3.9 provides an experimental comparison of EBW and Rprop for completely different tasks. The results in this table suggest that Rprop tends to perform slightly better than EBW. The number of iterations until convergence is comparable for EBW and Rprop when using a conservative but "universal" initial step size for Rprop ($\approx 10 - 15$ iterations for Mandarin). Often, the convergence of Rprop can be made faster by choosing a larger initial step size, say

Table 3.9: EBW *vs.* Rprop, word error rate (WER) for different tasks. The ML baseline is added for comparison. M-MMI stands for the margin-based variant of MMI introduced in Chapter 5.

Task	Criterion	Optimization	WER [%]		
			Test		
SieTill (1 dns/mix)	ML	EM	3.8		
	M-MMI	EBW	2.7		
		Rprop	2.7		
SieTill (16 dns/mix)	ML	EM	2.0		
	M-MMI	EBW	1.9		
		Rprop	1.8		
SieTill (64 dns/mix)	ML	EM	1.8		
	M-MMI	EBW	1.7		
		Rprop	1.6		
			Dev06	Eval06	Eval07
EPPS En	ML	EM	14.4	10.8	12.0
	MPE	EBW	13.4	10.2	11.5
		Rprop	13.4	10.3	11.5
			Dev07	Eval06	Eval07
Mandarin Broadcasts	ML	EM	12.0	17.9	11.9
	MPE	EBW	11.0	17.0	11.2
		Rprop	10.8	16.5	11.1

5-10 iterations for Mandarin. The result in Table 4.10 suggests that convergence with second order features still is faster (< 5 iterations for Mandarin).

3.8.4 Generative *vs.* discriminative training (model complexity)

Few work has been done to study the theoretical behavior of ML and discriminative training criteria (*e.g.* MMI). According to [Ng & Jordan 02], two regimes can be distinguished. For little training data (relative to model complexity?), ML is expected to outperform MMI whereas MMI outperforms ML for much training data. [Nádas 83] showed that the asymptotic error rate for MMI is not worse than for ML. If the model assumptions are true, the two asymptotic error rates coincide. Figure 3.6 shows the correlation of the relative improvement due to the discriminative training with the model complexity. The experimental results were collected under different conditions to make the plot more universal. Although the conditions in [Ng & Jordan 02, Nádas 83] are not strictly satisfied, The expected tendency is observed, *i.e.,* the difference in the word error rate (WER) between MMI/MPE and ML increases with the number of observations per parameter.[1]

3.9 Summary

In this chapter, the basic definitions and ideas for discriminative training were introduced and discussed in a transducer-based formulation. The efficient calculation of the gradient of the training criterion is an issue in ASR because of the combinatorial number of possible word sequences that need to be considered in discriminative training. On the one hand, the proposed transducer-based framework provides an abstraction and generalization of the existing recursion formulae used for MMI and MPE. In particular, our approach unifies these two recursion formulae and generalizes the speech-specific recursion formulae to HCRFs. On the other hand, this work generalizes efficiently the idea of the network training in [Eisner 01, Li & Eisner 09] by supporting efficiently training criteria beyond MMI. This framework will facilitate the development of more refined training algorithms as it provides an efficient solution to the unified training criterion for string models [Macherey & Haferkamp+ 05]. The chapter concluded with comparative experimental results for the conventional discriminative training, *e.g.* comparison of different loss functions or optimization algorithms.

[1] Thanks to Christian Plahl for training the broadcast system with over 8M densities!

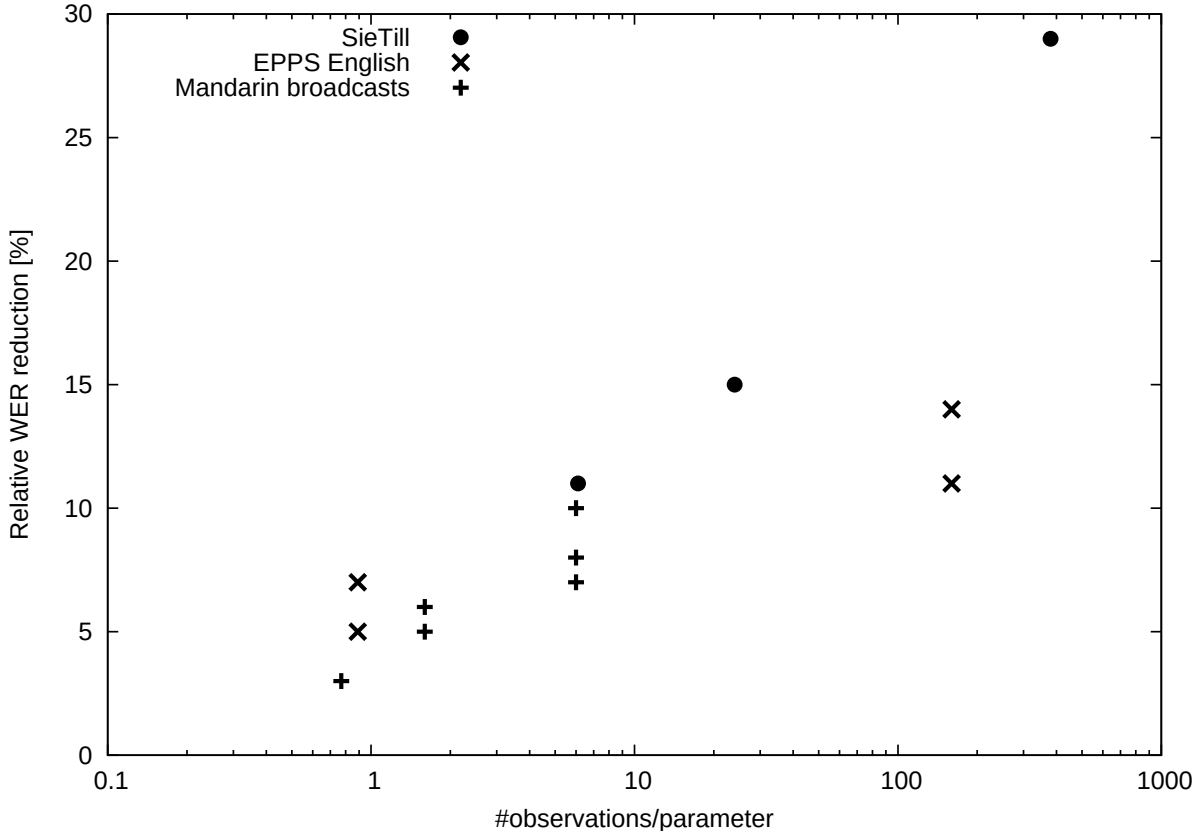


Figure 3.6: Relative reduction of word error rate (WER) over the number of observations per model parameter. Experimental results for different tasks using different features, different training criteria, and different number of densities.

Chapter 4

Equivalence Relations for Gaussian and Log-Linear HMMs

Conventional speech recognition systems are based on HMMs with Gaussian mixture models (GHMMs). Discriminative log-linear models are an alternative modeling approach and have been investigated recently in speech recognition. GHMMs are directed models with constraints, *e.g.* positivity of variances and normalization of conditional probabilities, while log-linear models do not use such constraints. This chapter compares the posterior form of typical generative models related to speech recognition with their log-linear model counterparts. The key result will be the derivation of the equivalence of these two different approaches under weak conditions. In particular, we study Gaussian mixture models, part-of-speech bigram tagging models and eventually, the GHMMs [Heigold & Schlüter+ 07, Heigold & Lehnen+ 08]. This result unifies two important but fundamentally different modeling paradigms in speech recognition on the functional level. Furthermore, this chapter will present comparative experimental results for various speech tasks of different complexity, including a digit string and large vocabulary continuous speech recognition tasks [Heigold & Wiesler+ 10].

4.1 Introduction

This chapter studies two important modeling paradigms in speech recognition: the generative models with prior and the log-linear discriminative models. In the traditional view, they are considered to be independent approaches that are motivated by fundamentally different points of view.

The posterior form of the *generative models* include the class prior. Typical of generative models is that they impose many constraints on the parameters, *e.g.* the positivity of the variances and the normalization of the conditional probabilities. The Gaussian model and the part-of-speech bigram tagging model are prototypical examples for single event and string input, respectively. The extension of these models to hidden variables includes the Gaussian mixture model and HMMs/GHMMs.

In contrast, *log-linear models* do not use any parameter constraints. The log-linear model [Anderson 82, Ng & Jordan 02, Saul & Lee 02] corresponds with the Gaus-

sian model. Linear-chain conditional random fields (CRFs) [Lafferty & McCallum+ 01, Sutton & McCallum 07] are the log-linear counterpart of Markov chains, *e.g.* the part-of-speech bigram tagging model. CRFs with hidden variables, termed hidden CRFs [Gunawardana & Mahajan+ 05, Hifny & Renals 09], are the analog of HMMs.

In this chapter, we shall use the following terminology and implicit assumptions. The term log-linear model refers both to models with log-linear parameterization, independent of the type of data, and to the specialization for single events. The specialization of the log-linear model for strings is called CRF. Here, CRF and linear-chain CRF are used interchangeably. Moreover, CRF stands for log-linear string models with first-order dependence assumptions and a specific choice of features. Note that the terms generative model, log-linear model, and CRF only define the type of parameterization. In particular, the parameterization does not imply a specific training criterion or optimization algorithm.

There has been a large number of papers that consider the relationship between generative and discriminative models. The common view in the literature is that the generative models are a subset of the respective log-linear counterpart because the constraints are relaxed in the log-linear parameterization [Anderson 82, Duda & Hart+ 01, Saul & Lee 02, Gunawardana & Mahajan+ 05, Sutton & McCallum 07]. In contrast, the transformation from the log-linear model into a generative model is less obvious because additional constraints need to be imposed on the model. For this reason, several authors speculate that log-linear models are more expressive than the posterior form of the associated generative model [Saul & Lee 02, Gunawardana & Mahajan+ 05]. At the same time, some of the authors claim the equivalence of these two approaches, but do not give a proof and do not address the question of how to handle the constraints of the generative models [Duda & Hart+ 01, Ng & Jordan 02, Sutton & McCallum 07].

In this chapter, we will study the equivalence of these two approaches in both directions. In particular, the novel contributions are:

- We will show that the log-linear models do not result in unique parameters and that the parameters are invariant under certain types of transformations.
- We will show under weak assumptions that the posterior form of the generative model *with* constraints is exactly equivalent to a log-linear model *without* any constraints. In other words, the generative model with constraints can always be converted into a log-linear discriminative model without any constraints, and vice versa.
- We will present experimental evidence for our theoretical findings.

The final goal of this chapter is to establish equivalence for typical GHMMs and linear-chain CRFs as used in speech recognition, including mixture models and scaling factors [Gunawardana & Mahajan+ 05, Hifny & Renals 09]. This will be accomplished in several steps. The derivation of the equivalence is based on the invariance of the log-linear models under certain transformations, which are studied in Section 4.3.4. Then, the Gaussian model and its log-linear counterpart are investigated in Section 4.5.2 to understand how the priors and the covariance matrices of the emission model of GHMMs can be transformed. Next, we move on to string models, starting with studying the part-of-speech bigram tagging model and its log-linear equivalent in Section 4.4.2. It illustrates how to transform the bigram

model parameters representing the prior (*cf.* m-gram language models). In Section 4.5.1, two approaches are discussed how to deal with the hidden variables originating from the HMM state sequences and the density indices of the mixture models. Restricted left-right HMM topologies are treated in Section 4.5.3 in the context of isolated word recognition. These preliminary results are combined in Section 4.5.4 to eventually derive the equivalence result for continuous speech recognition including word sequences of different length. A formalization and generalization of the previous results can be found in Section 4.6.

4.2 Related Work

In the terminology of [Ng & Jordan 02], equivalent generative and discriminative models are said to be a generative/discriminative pair. Only a very few such pairs have been mentioned in the literature. However, those papers are different from this work in various aspects. The authors discuss simple problems which are sub-problems of ours. Moreover, they only look at the one direction, claim two types of model to be equivalent without giving a proof, or they speculate that log-linear models are more expressive than their generative counterparts.

Also, comparative experimental results can be found in [Saul & Lee 02, Gunawardana & Mahajan$^+$ 05, Sha & Saul 07a], which are not always conclusive due to differences in the model, the training criterion, *etc.*

4.2.1 Single events: Gaussian *vs.* log-linear model

The log-linear and Gaussian models for single events, for example, have been thoroughly studied in the literature. As a matter of fact, [Anderson 82] addressed this problem first and showed that the posterior form of the Gaussian model is log-linear. However, he did not discuss how to impose the Gaussian model constraints when doing the transformation in the other direction. The authors in [Ng & Jordan 02] claim that these two models form a generative/discriminative pair, but do not give a proof. This result is supported indirectly by the analysis of the discriminant functions in [Duda & Hart$^+$ 01, pp.19], again without explicitly addressing the problem with the constraints. In contrast, [Saul & Lee 02] states clearly that the log-linear model is more expressive than the Gaussian counterpart.

4.2.2 Strings: HMM *vs.* linear-chain CRF

The situation for the more complex HMMs is similar. The authors in [Sutton & McCallum 07] claim that the transformation is possible, without giving a proof. Assuming a weighted finite-state transducer (WFST) with non-negative arc weights, weight pushing produces an equivalent stochastic WFST [Mohri 09, p.242]. This implies that the transformation is possible, at least under suitable boundary conditions. The class of WFSTs for which the algorithm terminates is not specified in this article. The detailed analysis in [Jaynes 03, pp.646] shows equivalence for the two approaches in the limit of infinitely long strings. For finite strings, the transition probabilities are non-stationary. Finally, [Gunawardana & Mahajan$^+$ 05] points out the problem

with the parameter constraints and concludes from this that GHMMs are a proper subset of the linear-chain CRFs.

4.3 Basic Concepts

This section introduces the notion of equivalence. Typical parameter constraints of generative models are then discussed. These constraints are the main source of difficulty in establishing equivalence relations. It will be shown that in general, different parameters can induce the same posterior. This ambiguity permits to impose the parameter constraints without changing the posterior model.

4.3.1 Posterior models

Assume posteriors $p : \{1, \ldots, C\} \times \mathbb{R}^D \to [0, 1]$, $(c, x) \mapsto p(c|x)$ subject to $\sum_c p(c|x) = 1$. Then, a *posterior model* is defined as a set of posteriors, $p_\Gamma := \{p_\Lambda(c|x) | \Lambda \in \Gamma\}$. We distinguish between direct/discriminative and indirect/generative posterior models in this chapter. Log-linear models are an example of a discriminative posterior model [Della Pietra & Della Pietra+ 97]

$$p_{\mathrm{CRF},\Gamma} := \left\{ p_{\mathrm{CRF},\Lambda}(c|x) = \frac{\exp\left(\sum_j \mu_j h_j(x,c)\right)}{\sum_{c'} \exp\left(\sum_j \mu_j h_h(x,c')\right)} \middle| \mu_j \in \mathbb{R} \right\}. \tag{4.1}$$

An alternative formulation of log-linear models is based on class-dependent model parameters and class-independent features

$$p_{\mathrm{CRF},\Gamma} := \left\{ p_{\mathrm{CRF},\Lambda}(c|x) = \frac{\exp\left(\sum_i \lambda_{ci} f_i(x)\right)}{\sum_{c'} \exp\left(\sum_i \lambda_{c'i} f_i(x)\right)} \middle| \lambda_{ci} \in \mathbb{R} \right\}. \tag{4.2}$$

In genral, the symbol j denotes some abstract index. It can be compound and also include the class index.

Note that the two definitions of log-linear models induce exactly the same class of models. This is shown as follows. First, assume μ_j, $h_j(x, c)$ in Equation (4.1), and define $\lambda_{c,\tilde{c}j} := \delta(\tilde{c}, c)\mu_j$ and $f_{\tilde{c}j}(x) := h_j(x, \tilde{c})$ for Equation (4.2). Note that now, the index i in Equation (4.2) denotes the index pair $(\tilde{c}, j)$. Then, the arguments of the exponential functions are identical

$$\sum_{\tilde{c},j} \lambda_{c,\tilde{c}j} f_{\tilde{c}j}(x) = \sum_{\tilde{c},j} \delta(\tilde{c}, c)\mu_j h_j(x, \tilde{c}) = \sum_j \mu_j h_j(x, c), \quad \forall x, c.$$

For the opposite direction, assume λ_{ci}, $f_i(x)$ in Equation (4.2), and define $\mu_{\tilde{c}i} := \lambda_{\tilde{c}i}$ and $h_{\tilde{c}i}(x, c) = \delta(\tilde{c}, c) f_i(x)$ for Equation (4.1), with the index pair $j = (\tilde{c}, i)$. Then, we have

$$\sum_{\tilde{c},i} \mu_{\tilde{c}i} h_{\tilde{c}i}(x, c) = \sum_{\tilde{c},i} \lambda_{\tilde{c}i} \delta(\tilde{c}, c) f_i(x) = \sum_i \lambda_{ci} f_i(x), \quad \forall x, c.$$

Hence, we can use the definition in Equation (4.2) without loss of generality.

Joint probabilities are defined as $p : \{1, \ldots, C\} \times \mathbb{R}^D \rightarrow [0, 1]$, $(c, x) \mapsto p(x, c)$ subject to $\sum_c \int dx p(x, c) = 1$. A generative model is a set of joint probabilities, $\{p_\theta(x, c) | \theta \in \Theta\}$. The posterior model induced by such a generative model is defined as

$$p_{\text{Gen},\Theta} := \left\{ \frac{p_\theta(x, c)}{\sum_{c'} p_\theta(x, c')} \middle| \theta \in \Theta \right\}.$$

4.3.2 Equivalence

We use the following notion of equivalence.

Definition 7 (Equivalence)**.** *The posterior model p_Γ and the posterior model $p'_{\Gamma'}$ are called* equivalent *if $p_\Gamma = p'_{\Gamma'}$.*

A consequence of this definition is that equivalent log-linear and generative posterior models are expected to perform equally for all *posterior-based* algorithms. For instance, equivalent log-linear and generative posterior models that are optimized with the same discriminative training criterion (*e.g.* MMI, MCE, MPE) lead to identical error rates in theory. This statement is true as long as *purely* posterior-based algorithms are used. The regularization/smoothing term may break the exact equivalence due to the direct dependency on the model parameters.

For the two posterior models $p_{\text{CRF},\Gamma}$ and $p_{\text{Gen},\Theta}$, the equivalence proof consists of two parts: showing that $p_{\text{Gen},\Theta} \subset p_{\text{CRF},\Gamma}$, and showing that $p_{\text{CRF},\Gamma} \subset p_{\text{Gen},\Theta}$. The first part of the proof (or the transformation from the generative to a log-linear model) is rather straightforward and well-known, see *e.g.* [Macherey & Ney 03, Gunawardana & Mahajan+ 05] for Gaussian models.[1] For this reason, we shall focus on the second part of the proof, the transformation of the log-linear into a generative model. Two types of proof for $p_{\text{CRF},\Gamma} \subset p_{\text{Gen},\Theta}$ can be found in this chapter. For simple models, a "guess" of the generative models is made and then verified to be a solution. This approach is not convenient for more complex models. They are rather proved by an iterative construction of the generative models, each step guaranteeing that the equivalence is preserved.

4.3.3 Parameter constraints

Unlike the unconstrained discriminative models (*e.g.* HCRFs), the generative models typically impose constraints and some structure on the parameters. A proper Gaussian model requires a positive-definite covariance matrix $\Sigma \in \mathbb{R}^{D \times D}$

$$\Sigma \succ 0 \qquad \text{(positive-definiteness)}. \tag{4.3}$$

Discrete-valued probabilities satisfy the normalization constraint

$$p(c) \geq 0, \sum_c p(c) = 1 \qquad \text{(normalization)}. \tag{4.4}$$

[1] However, it should be pointed out that this is not always possible, see [Bishop 06, p.393] for an example.

Additional restrictions on the structure are often made for conditional probabilities (*e.g.* Markov models)

$$p(c_n|c_1^{n-1}) \equiv p(c_n|c_{n-1}),\ \forall n > 1 \qquad \text{(dependence)} \tag{4.5}$$

$$p_m(c_{m+n}|c_{m+n-1}) \equiv p(c_n|c_{n-1}),\ \forall m \geq 0, \forall n > 1 \qquad \text{(stationarity)}. \tag{4.6}$$

The difficulty in establishing equivalence relations is to impose such constraints on the discriminative model without changing the posteriors. In the case where there are no (or only little) restrictions to the model, the transformation is rather straightforward. Such general results can be found *e.g.* in [Lauritzen & Dawid$^+$ 90].

4.3.4 Invariance transformations

The transformation of an unconstrained discriminative model (*e.g.* CRF) into an equivalent generative model is based on the observation that different $\Lambda, \Lambda' \in \Gamma$ can induce the same posterior, *i.e.*, $p_{CRF,\Lambda}(c|x) = p_{CRF,\Lambda'}(c|x),\ \forall c, x$. This leads to the definition of invariance transformations.

Definition 8 (Invariance transformation)**.** *An* invariance transformation *is a function* $f : \Gamma \to \Gamma,\ \Lambda \mapsto f(\Lambda)$ *such that* $p_{CRF,\Lambda}(c|x) = p_{CRF,f(\Lambda)}(c|x),\ \forall c, x,$ *and* $\Lambda \in \Gamma$.

For a log-linear model with second-order features derived from $x \in \mathbb{R}^D$ and the model parameters $\Lambda \in \{\{\Lambda_c \in \mathbb{R}^{D\times D}\}, \{\lambda_c \in \mathbb{R}^D\}, \{\alpha_c \in \mathbb{R}\}\} =: \Gamma$,

$$p_\Lambda(c|x) \quad = \quad \frac{\exp(x^\top \Lambda_c x + \lambda_c^\top x + \alpha_c)}{\sum_{c'} \exp(x^\top \Lambda_{c'} x + \lambda_{c'}^\top x + \alpha_{c'})}, \tag{4.7}$$

the invariance transformations are

$$\Lambda_c \mapsto \Lambda_c + \Delta\Lambda, \qquad \Delta\Lambda \in \mathbb{R}^{D\times D} \tag{4.8}$$

$$\lambda_c \mapsto \lambda_c + \Delta\lambda, \qquad \Delta\lambda \in \mathbb{R}^D \tag{4.9}$$

$$\alpha_c \mapsto \alpha_c + \Delta\alpha, \qquad \Delta\alpha \in \mathbb{R}. \tag{4.10}$$

The parameter offsets $\Delta\Lambda$, $\Delta\lambda$, and $\Delta\alpha$ add the same factors both in the numerator and denominator in Equation (4.7) that do not depend on the class index c and thus cancel. Clearly, the notion of invariance is more general than illustrated in Equation (4.8-4.10) where only "local" transformations are considered. In general, only the sum of all "local" transformations needs to be an invariance transformation. This shall be referred to as "passing of normalization constants."

The invariance transformations of Gaussian-based posteriors lead to a rather strange and counterintuitive behavior. The means of the Gaussian model, for instance, can be localized anywhere in parameter space as illustrated in Figure 4.1. This degeneracy of GMM-based posteriors was already pointed out in [Ristad & Yianilos 98b] in a different context.

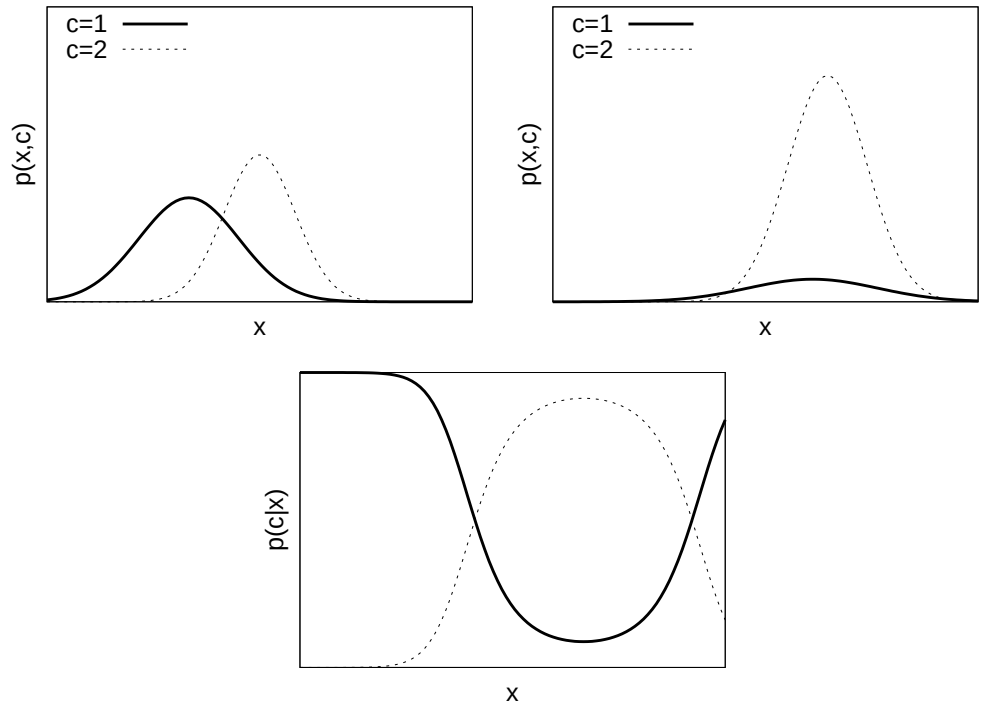


Figure 4.1: Illustration of invariance transformations for Gaussian-based posteriors: two Gaussian models with different parameters (mean, variance, and prior) can induce the same posterior by the Bayes rule.

The remainder of this chapter is organized as follows. The Gaussian mixture model and a simple tagging problem are first discussed in Section 4.4. These simple models illustrate how to handle Gaussian models with positive-definite covariances and conditional probabilities. The results are then used to show the equivalence of GHMMs and LHMMs in speech recognition (Section 4.5). The equivalence results are then formalized and generalized in Section 4.6. Finally, these theoretical results are experimentally verified in Sections 4.7 and 4.8 on different part-of-speech tagging and speech recognition tasks of completely different complexities.

4.4 Prototypical Equivalence Relations

The basic techniques used to establish the equivalence relations in this chapter are introduced on two simple example models.

4.4.1 Single Gaussian models

Denote a Gaussian density with density index l by

$$\mathcal{N}(x|\mu_c, \Sigma_c) \quad = \quad \frac{1}{|2\pi\Sigma_c|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(x-\mu_c)^\top \Sigma_c^{-1}(x-\mu_c)\right).$$

Table 4.1: Transformation from Gaussian into log-linear model parameters.

1.	Λ_c	$=$	$-\frac{1}{2}\Sigma_c^{-1}$
2.	λ_c	$=$	$\Sigma_c^{-1}\mu_c$
3.	α_c	$=$	$-\frac{1}{2}\left(\mu_c^\top\Sigma_c^{-1}\mu_c + \log\lvert 2\pi\Sigma_c\rvert\right) + \log p(c)$

Here, $\mu_c \in \mathbb{R}^D$ denotes the mean and $\Sigma_c \in \mathbb{R}^{D\times D}$ subject to Equation (4.3) stands for the covariance matrix of the Gaussian. The joint probability of the single Gaussian model includes the class prior $p(c) \in \mathbb{R}$ subject to Equation (4.4). It is defined as

$$p(x, c) \quad = \quad p(c)\mathcal{N}(x|\mu_c, \Sigma_c). \tag{4.11}$$

Using the Bayes rule

$$p(c|x) \quad = \quad \frac{p(x, c)}{\sum\limits_{c'} p(x, c')}, \tag{4.12}$$

the class posteriors induced by the single Gaussian model read

$$p_{\text{Gauss},\theta}(c|x) \overset{\text{Equation (4.11)}}{=} \frac{p(c)\mathcal{N}(x|\mu_c, \Sigma_c)}{\sum\limits_{c'} p(c')\mathcal{N}(x|\mu_{c'}, \Sigma_{c'})}. \tag{4.13}$$

The model parameters are $\theta \in \{\{\mu_c \in \mathbb{R}^D\}, \{\Sigma_c \in \mathbb{R}^{D\times D} | \Sigma_c \succ 0\}, \{p(c) \in \mathbb{R} | \sum\limits_c p(c) = 1\}\} =: \Theta$. The posterior model in Equation (4.13) can be represented as a log-linear model of the type

$$p_{\text{log-lin},\Lambda}(c|x) \quad = \quad \frac{\exp\left(x^\top\Lambda_c x + \lambda_c^\top x + \alpha_c\right)}{\sum\limits_{c'} \exp\left(x^\top\Lambda_{c'} x + \lambda_{c'}^\top x + \alpha_{c'}\right)}. \tag{4.14}$$

This was shown in [Anderson 82, Saul & Lee 02, Jebara 02, Macherey & Ney 03, Gunawardana & Mahajan$^+$ 05]. The log-linear parameters $\Lambda \in \{\{\Lambda_c \in \mathbb{R}^{D\times D}\}, \{\lambda_c \in \mathbb{R}^D\}, \{\alpha_c \in \mathbb{R}\}\} =: \Gamma$ can be determined from the Gaussian parameters Θ by comparing terms constant, linear, and quadratic in x. The resulting transformation rules are summarized in Table 4.1.

Assuming a log-linear model with parameters $\Lambda \in \Gamma$, an equivalent Gaussian model can be determined by solving the transformation rules in Table 4.1 for some Gaussian parameters θ. However, this approach does not define a proper Gaussian model in general, *i.e.*, $\theta \notin \Theta$. This is because the covariance matrix Σ_c is not guaranteed to be positive-definite (if the inverse is defined at all), and the priors $p(c)$ and mixture weights $p(l|c)$ do not need to be normalized. This observation could explain why some authors assume that the log-linear models in Equation (4.14) are "more expressive than their generative counterparts" [Saul & Lee 02, Gunawardana & Mahajan$^+$ 05, Sha & Saul 07a]. None of the existing work provides explicit transformation rules, see Section 4.2.1. Here, we derive the transformation rules by taking advantage of the ambiguity of the log-linear model parameters (see Section 4.3.4) to resolve this problem.

Table 4.2 summarizes the different steps required to transform a log-linear model into an equivalent (and proper) Gaussian model. First, observe that the matrix Λ_c is ambiguous. The

Table 4.2: Transformation from log-linear into Gaussian model parameters, '←' indicates an invariance transformation and "passing" is an abbreviation for "passing of normalization constant." See text for explanations.

1.	$\tilde{\Lambda}_c$	$\overset{\text{Equation (4.8)}}{\leftarrow}$	$\Lambda_c + \Delta\Lambda$
2.	Σ_c	$=$	$-\frac{1}{2}\tilde{\Lambda}_c^{-1}$
3.	μ_c	$=$	$\Sigma_c \lambda_c$
4.	$\tilde{\alpha}_c$	$\overset{\text{"passing"}}{\leftarrow}$	$\alpha_c + \frac{1}{2}\left(\mu_c^\top \Sigma_c^{-1} \mu_c + \log\lvert 2\pi\Sigma_c\rvert\right)$
5.	$p(c)$	$\overset{\text{Equation (4.10)}}{\leftarrow}$	$\exp\left(\tilde{\alpha}_c - \log\left(\sum_{c'} \exp(\tilde{\alpha}_{c'})\right)\right)$

invariance transformation in Equation (4.8) with a sufficiently negative-definite $\Delta\Lambda \in \mathbb{R}^{D\times D}$ (*i.e.,* the eigenvalues of $\Delta\Lambda$ are sufficiently negative) can be used to make Λ_c negative-definite. Hence, the covariance matrix Σ_c exists and is positive-definite (Step 2). The determination of the mean μ_c is straightforward. Next, the mixture weights are normalized. The normalization constant from the Gaussian density is incorporated into the prior parameter, $\tilde{\alpha}_c := \alpha_c + \frac{1}{2}\left(\mu_c^\top \Sigma_c^{-1} \mu_c + \log\lvert 2\pi\Sigma_c\rvert\right)$. The class prior is normalized by applying the invariance transformation in Equation (4.10) with $\Delta\alpha := -\log \sum_c \alpha_c$ (Step 5). The result also holds true for special cases such as diagonal or pooled covariance matrices.

This subsection can be summarized by the following lemma.

Lemma 9 (Equivalence (Gauss model))**.** *The posterior model $p_{log\text{-}lin,\Gamma}$ in Equation (4.14) and the posterior model $p_{Gauss,\Theta}$ induced by the generative model in Equation (4.13) are equivalent.*

The equivalence is proved by showing that (for example) $p_{\text{Gauss},\Theta} \subset p_{\text{log-lin},\Gamma}$ and $p_{\text{log-lin},\Gamma} \subset p_{\text{Gauss},\Theta}$. Derivations for the transformation rules in Table 4.1 can be found in the literature, *e.g.* [Saul & Lee 02, Jebara 02, Macherey & Ney 03, Gunawardana & Mahajan+ 05], proving $p_{\text{Gauss},\Theta} \subset p_{\text{log-lin},\Gamma}$. Thus, the proof focuses on the transformation from the log-linear model into a Gaussian model. In the above discussion, it was shown that a log-linear model can be transformed into a proper Gaussian model only applying invariance transformations. For this simple model, a direct proof fits on a page as well.

Proof. The idea of the proof consists of constructing a proper Gaussian model for each log-linear model. The Gaussian model parameters in Table 4.2 are well-defined in the sense of the constraints in Section 4.3.3 by construction. To show the equivalence of the original log-linear model and the resulting Gaussian model, we start with the Gaussian model and transform it into the numerator of the log-linear model up to a constant factor (*i.e.,* a factor that does not depend

on c). The shortcut $Z := \sum_c \exp(\alpha_c)$ is used. The indicated steps refer to Table 4.2.

$$
\begin{aligned}
p_{\text{Gauss},\Theta}(x, c) &\overset{\text{Step 5}}{=} \frac{1}{Z} \exp\left(\tilde{\alpha}_c\right) \mathcal{N}(x|\mu_c, \Sigma_c) \\
&\overset{\text{Step 4}}{=} \frac{1}{Z} \exp\left(\alpha_c\right) \exp\left(\frac{1}{2}\left(\mu_c^\top \Sigma_c^{-1} \mu_c + \log|2\pi\Sigma_c|\right.\right. \\
&\qquad \left.\left. -\mu_c^\top \Sigma_c^{-1} \mu_c - \log|2\pi\Sigma_c| + 2\mu_c^\top \Sigma_c^{-1} x - x^\top \Sigma_c^{-1} x\right)\right) \\
&= \frac{1}{Z} \exp\left(\alpha_c\right) \exp\left(\mu_c^\top \Sigma_c^{-1} x - \frac{1}{2} x^\top \Sigma_c^{-1} x\right) \\
&\overset{\text{Steps 2\&3}}{=} \frac{1}{Z} \exp\left(\alpha_c\right) \exp\left(\lambda_c^\top x - x^\top \tilde{\Lambda}_c x\right) \\
&\overset{\text{Step 1}}{=} \underbrace{\frac{\exp(x^\top \Delta\Lambda x)}{Z}}_{\text{constant factor}} \cdot \underbrace{\exp\left(\alpha_c + \lambda_c^\top x + x^\top \Lambda_c x\right)}_{\text{numerator of log-linear model in Equation (4.14)}}
\end{aligned}
$$

The first term in the last line is a constant w.r.t. class c and thus cancels in the posterior. □

The extension of this equivalence result to more general features is straightforward. For the general log-linear model in Equation (4.2), the feature vector x contains the kernel feature functions $f_i(x)$. In addition, the covariance matrices Σ_c and $-2\Delta\Lambda$ are set to the unity matrix and $\Lambda_c = 0$ in Tables 4.3 and 4.4. This implies that the equivalence holds for the general log-linear model in Equation (4.2) and the Gaussian model in Equation (4.22) under the weak assumption that the kernel feature function $f_0(x) = 1$ is included. The log-linear model in Equation (4.1) can be equally represented in the form in Equation (4.2) with restricted model parameters, $\lambda_c = (0, \ldots, 0, \lambda, 0, \ldots, 0)$ with the vector λ in the c-th position. This structure is preserved in the second steps of Tables 4.3 and 4.4. In particular, the number of degrees of freedom is the same in both models. Furthermore, binary and discrete features are a subset of the continuous features. This proves the equivalence of the general log-linear model in Equation (4.1) and the Gaussian model in Equation (4.22). In the remainder of this chapter, we will derive equivalence relations for structured string model classes.

Gaussian/log-linear models are local models. Conditional probabilities (*cf.* Markov models) shall be considered next. We start with a simple model for part-of-speech tagging, which will then be extended to speech recognition in Section 4.5.

4.4.2 Part-of-speech bigram tagging model

The construction of conditional probabilities from a log-linear CRF is illustrated by means of a simple, yet non-trivial model: part-of-speech tagging with a bigram model. Unlike speech recognition, the part-of-speech tagging (as considered here) assumes a one-to-one mapping from the words x_1^N (input) to the tags c_1^N (output), see Figure 4.2. The alignment problem is deferred until Section 4.5. For the time being, consider the joint probability

$$
p_{\text{Gen},\theta}(x_1^N, c_1^N) = \underbrace{p(\$|c_N) \prod_{n=1}^{N} p(c_n|c_{n-1})}_{\text{transition model}} \underbrace{\prod_{n=1}^{N} p(x_n|c_n)}_{\text{emission model}} \tag{4.15}
$$

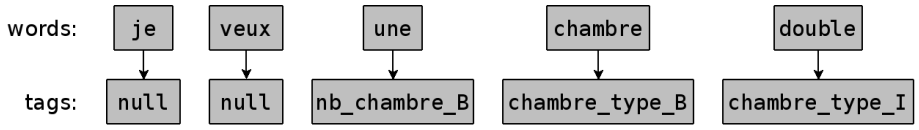


Figure 4.2: Example for part-of-speech tagging from the French Media corpus.

with the emission probabilities $p(x|c)$ and the bigram probabilities $p(c|c')$. The generative model parameters $\theta \in \{\{p(c|c') \in \mathbb{R}^+ | \sum_c p(c|c') = 1\}, \{p(x|c) \in \mathbb{R} | \sum_x p(x|c) = 1\}\} : \Theta$ are the look-up tables for the emission probabilities subject to Equation (4.4) and the bigram probabilities subject to Equations (4.4-4.6).

The linear-chain CRF with the same sufficient statistics reads

$$p_{\mathrm{CRF},\Lambda}(c_1^N|x_1^N) = \frac{1}{Z_\Lambda(x_1^N)} \underbrace{\exp(\alpha_{c_N\$}) \prod_{n=1}^{N} \exp(\alpha_{c_{n-1}c_n})}_{\text{transition model}} \underbrace{\prod_{n=1}^{N} \exp(\beta_{c_n x_n})}_{\text{emission model}} \tag{4.16}$$

with normalization constant $Z_\Lambda(x_1^N)$ (summation over all tag sequences) and the bigram and emission parameters as the model parameters $\Lambda \in \{\{\alpha_{c'c} \in \mathbb{R}\}, \{\beta_{cx} \in \mathbb{R}\}\} =: \Gamma$. In addition to the regular tags $c \in \Sigma$, we use the special tag \$ indicating the sentence end. Assume that this boundary tag is also part of the bigram model and that the sequences c_1^N start and end implicitly with this boundary tag, *i.e.*, $c_0 = c_{N+1} = \$$. This model serves as preparation for the transition and language models in speech recognition, which typically include such information (entry/exit states for HMMs, or sentence boundary symbol for language models), see Section 4.5.4.

Again, the goal is to transform the one model into the other. The transformation from the constrained generative Markov model into the unconstrained discriminative model is straightforward. To do so, set $\alpha_{c'c} := \log p(c|c')$ and $\beta_{cx} := \log p(x|c)$, similar to [Gunawardana & Mahajan$^+$ 05, Sutton & McCallum 07].

No concise and consistent statements on the transformation in the opposite direction can be found in the literature, see Section 4.2.2. Here, transformation rules are derived under the assumptions of non-negative irreducible transition matrices (see below) and a suitable boundary treatment (all tag strings start and end with the same boundary symbol). The solution is motivated by the solution for infinite strings in [Jaynes 03, p.646]. In contrast to that work, however, we provide a proof of existence, and due to the introduction of the boundary symbol, the solution also applies to finite strings. Opposed to [Mohri 09], our approach avoids problems with the convergence for cycles with weight greater than one, see Figure 4.3. A more general approach will be discussed in Section 4.6. The detailed calculations in [Jaynes 03, p.647] suggest that the equivalence does not hold true for sequences of finite length. In particular the transition probabilities are non-stationary, implying an explosion of the number of the parameters. The authors in [Gunawardana & Mahajan$^+$ 05] argue that the constraints of the generative models reduce the model flexibility compared with the unconstrained linear-chain CRF.

Here, transformation rules are derived under the assumptions of non-negative irreducible

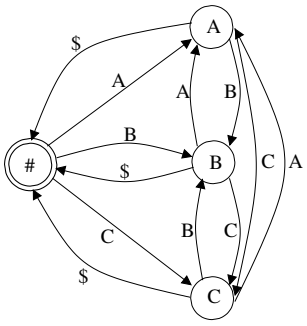


Figure 4.3: First-order Markov model (*e.g.* part-of-speech bigram model) represented as a WFST over the alphabet $\{\$, A, B, C\}$. The arcs describe the transitions $(c', c) \in \{\$, A, B, C\} \times \{\$, A, B, C\}$ with weight $\exp(\alpha_{c'c})$ (omitted for simplicity).

transition matrices (see below) and a suitable boundary treatment (all tag strings start and end with the same boundary symbol). The solution is motivated by the solution for infinite sequences in [Jaynes 03, p.646]. In contrast to that work, however, we provide a proof of existence and due to the introduction of the boundary symbol, the solution applies to finite sequences. A more general approach will be discussed in Section 4.6.

Emission model. The pseudo emission probabilities $\exp(\beta_{cx})$ can be normalized positionwise

$$p(x|c) = \frac{\exp(\beta_{cx})}{Z(c)}. \tag{4.17}$$

The normalization constant $Z(c) := \sum_x \exp(\beta_{cx})$ carries over to the bigram parameters, *i.e.*,

$$\alpha_{c'c} + \beta_{cx} = (\alpha_{c'c} + \log Z(c)) + (\beta_{cx} - \log Z(c)) \tag{4.18}$$

$$= \tilde{\alpha}_{c'c} + \tilde{\beta}_{cx} \tag{4.19}$$

with $\tilde{\alpha}_{c'c} := \alpha_{c'c} + \log Z(c)$ and $\tilde{\beta}_{cx} = \beta_{cx} - \log Z(c)$ such that the posterior remains unchanged. The normalization of the bigram probabilities is based on these modified pseudo probabilities, $\exp(\tilde{\alpha}_{c'c})$ and $\exp(\tilde{\beta}_{cx})$.

Transition model. The bigram probabilities can be constructed in a similar way as in [Jaynes 03]. To avoid lengthy calculations here (see Section 4.6 for a constructive proof), we state the solution and verify that this solution satisfies the properties in Equation (4.4-4.6). In contrast to [Jaynes 03], we do not only assume that a solution exists but also provide an existence proof. Furthermore, our result also applies to finite sequences and is not only valid in the limit of infinite sequences as in [Jaynes 03]. Here, the proof of the equivalence relation is based on the *Perron-Frobenius Theorem* for non-negative matrices [Rao & Rao 98, p.475].

Theorem 10 (Perron-Frobenius)**.** *Let $Q \in \mathbb{R}^{C \times C}$ be an irreducible matrix with only non-negative coefficients. Define q to be the maximum of the absolute values of the eigenvalues of Q. Then:*

1. $q > 0$.

2. *q is an eigenvalue of Q.*
3. *There exists an eigenvector of Q with only positive coefficients corresponding to the eigenvalue q.*
4. *The eigenvalue q is simple.*

Lemma 11 (Equivalence (Markov model))**.** *The posterior model $p_{CRF,\Gamma}$ in Equation (4.16) and the posterior model $p_{Gen,\Theta}$ induced by the generative model in Equation (4.15) are equivalent.*

Again, the proof consists of showing that $p_{\text{Gen},\Theta} \subset p_{\text{CRF},\Gamma}$ (without proof) and $p_{\text{CRF},\Gamma} \subset p_{\text{Gen},\Theta}$.

Proof. The result uses the matrix notation of the bigram probabilities. The transition matrix Q is defined to hold true the unnormalized bigram probabilities, $Q := [\exp(\tilde{\alpha}_{c'c})]_{c',c\in\Sigma\cup\{\$\}}$. Furthermore, v_c are the components of the right eigenvector of Q associated with the largest eigenvalue q. Define the bigram probabilities as

$$p(c|c') := \frac{Q_{c'c}v_c}{qv_{c'}} \tag{4.20}$$

First, the equivalence of the two posterior models can be verified by plugging the definitions for $p(x|c)$ in Equation (4.17) and $p(c|c')$ in Equation (4.20) into Equation (4.15)

$$\begin{aligned} p_{\text{Gen},\theta}(x_1^N, c_1^N) &= \exp(\tilde{\alpha}_{c_N\$}) \prod_{n=1}^{N} \exp(\tilde{\alpha}_{c_{n-1}c_n}) \prod_{n=1}^{N} \exp(\tilde{\beta}_{c_n x_n}) \\ &\overset{\text{Equation (4.20)}}{=} \frac{Q_{c_N\$}v_\$}{qv_{c_N}} \prod_{n=1}^{N} \frac{Q_{c_{n-1}c_n}v_{c_n}}{qv_{c_{n-1}}} \prod_{n=1}^{N} \exp(\tilde{\beta}_{c_n x_n}) \\ &= \underbrace{\frac{1}{q^{N+1}}}_{\text{constant factor}} \cdot \underbrace{\prod_{n=1}^{N+1} \frac{v_{c_n}}{v_{c_{n-1}}}}_{\text{telescope product}} \cdot Q_{c_N\$} \prod_{n=1}^{N} Q_{c_{n-1}c_n} \exp(\tilde{\beta}_{c_n x_n}). \end{aligned} \tag{4.21}$$

The constant factor $\frac{1}{q^{N+1}}$ cancels in the posterior induced by the Bayes rule in Equation (4.12). The telescope product is unity by our model assumption that $c_0 = c_{N+1} = \$$

$$\prod_{n=1}^{N+1} \frac{v_{c_n}}{v_{c_{n-1}}} = \frac{v_{c_1}}{v_\$}\frac{v_{c_2}}{v_{c_1}} \cdots \frac{v_{c_N}}{v_{N-1}}\frac{v_\$}{v_{c_N}} = 1.$$

The remaining part is transformed into

$$\begin{aligned} Q_{c_N\$} \prod_{n=1}^{N} Q_{c_{n-1}c_n} \exp(\tilde{\beta}_{c_n x_n}) &= \exp(\tilde{\alpha}_{c_N\$}) \prod_{n=1}^{N} \exp(\tilde{\alpha}_{c_{n-1}c_n}) \exp(\tilde{\beta}_{c_n x_n}) \\ &\overset{\text{Equations (4.18, 4.19)}}{=} \underbrace{Z(\$)}_{\text{constant factor}} \\ &\quad \cdot \exp(\alpha_{c_N\$}) \prod_{n=1}^{N} \exp(\alpha_{c_{n-1}c_n}) \exp(\beta_{c_n x_n}). \end{aligned}$$

In summary, the generative probability $p_{\text{Gen}}(x_1^N, c_1^N)$ is identical to the numerator of the CRF probability in Equation (4.16) up to the constant factor $Z(\$)$, which cancels in the posterior. Hence, equivalence holds true.

Second, we check that $p(c|c')$ in Equation (4.20) is well-defined and satisfies the properties in Equation (4.4-4.6). The properties in Equations (4.5-4.6) are satisfied by definition. All coefficients of the transition matrix Q are positive. Hence, the transition matrix Q is irreducible, *i.e.,* each state can be reached from any other state. According to the *Perron-Frobenius Theorem* (Theorem 10), the largest eigenvalue q of Q is positive and unique. Moreover, all coefficients v_c of the eigenvector corresponding with q are positive. Hence, the bigram probabilities in Equation (4.20) are non-singular (no division by zero) and positive. These quantities are normalized because v is an eigenvector of Q, *i.e.,* $\sum_c Q_{c'c} v_c = q v_{c'}$, $\forall c'$, which is equivalent to the normalization constraint in Equation (4.4). The solution is unique because all other eigenvectors must have at least one negative coefficient due to the orthogonality of the subspaces spanned by the eigenvectors with the same eigenvalue. □

Lemmata 12 and 11 are the key results used in the next section where the equivalence of GHMMs and LHMMs in speech recognition is proved.

4.5 Speech Recognition

The equivalence relation for GHMMs and LHMMs in speech recognition is proved step by step, starting with simple HMMs and then extending this result to LVCSR with an m-gram language model *etc.*

4.5.1 Hidden Variables

Conventional speech recognition systems are based on HMMs using Gaussian mixture models (GMMs). In particular, they include hidden variables as for example the density indices of the GMMs and the state sequences of HMMs.

Two approaches are commonly used in the literature to handle hidden variables in the log-linear framework. Similar to generative models, the log-linear framework is extended to incorporate hidden variables by marginalization. More formally, Equation (4.7) extends to $p(c|x) = \sum_h p(c, h|x)$ where h denotes the hidden variables and $p(c, h|x)$ is a log-linear model with the class pair (c, h) in Equation (4.7). Examples for this approach can be found in [Saul & Lee 02] (log-linear mixtures) and [Gunawardana & Mahajan$^+$ 05] (hidden CRFs). Alternatively, the log-linear model with hidden variables or hidden CRF is turned into a pure log-linear model or CRF by representing the true class (*e.g.* spoken sentence) by a single hidden variable (*e.g.* forced state alignment). This implies that the sum in the first approach is replaced by the maximum, $p(c|x) = \max_h\{p(c, h|x)\}$. This idea was pursued in [Lafferty & McCallum$^+$ 01, Sutton & McCallum 07, Sha & Saul 07a, Sha & Saul 07b, Hifny & Renals 09, Heigold & Rybach$^+$ 09]. More on this approach can be also found in Chapter 7.

Table 4.3: Transformation from GMM into LMM parameters.

1.	Λ_{cl}	$=$	$-\frac{1}{2}\Sigma_{cl}^{-1}$
2.	λ_{cl}	$=$	$\Sigma_{cl}^{-1}\mu_{cl}$
3.	α_{cl}	$=$	$-\frac{1}{2}\left(\mu_{cl}^\top\Sigma_{cl}^{-1}\mu_{cl} + \log\left\|2\pi\Sigma_{cl}\right\|\right) + \log p(c)$

Factors that do not depend on c, h can be extracted from the sum and the max. Hence, these factors cancel in the posterior as before. Thus, the extension of the equivalence results in Sections 4.4.1 and 4.4.2 to models with hidden variables is straightforward.

4.5.2 Gaussian mixture models (GMMs)

The Gaussian mixture model (GMM) is defined as the superposition of Gaussian densities with mixture weights $p(l|c) \in \mathbb{R}$ subject to Equation (4.4) (for all c)

$$p(x, c) \quad = \quad \sum_l p(l|c)\mathcal{N}(x|\mu_{cl}, \Sigma_{cl}). \tag{4.22}$$

The class posteriors include the priors $p(c) \in \mathbb{R}$ subject to Equation (4.4). Using the Bayes rule in Equation (4.12), they are defined as

$$p_{\text{GMM},\theta}(c|x) \quad \overset{\text{Equation (4.22)}}{=} \quad \frac{p(c)\sum_l p(l|c)\mathcal{N}(x|\mu_{cl}, \Sigma_{cl})}{\sum_{c'} p(c')\sum_l p(l|c')\mathcal{N}(x|\mu_{c'l}, \Sigma_{c'l})}. \tag{4.23}$$

The model parameters are $\theta \in \{\{\mu_{cl} \in \mathbb{R}^D\}, \{\Sigma_{cl} \in \mathbb{R}^{D\times D} | \Sigma_{cl} \succ 0\}, \{p(l|c) \in \mathbb{R} | \sum_l p(l|c) = 1\}, \{p(c) \in \mathbb{R} | \sum_c p(c) = 1\}\} =: \Theta$. The posterior model in Equation (4.23) can be represented as a log-linear model of the type

$$p_{\text{log-lin},\Lambda}(c|x) \quad = \quad \frac{\sum_l \exp\left(x^\top\Lambda_{cl}x + \lambda_{cl}^\top x + \alpha_{cl}\right)}{\sum_{c',l} \exp\left(x^\top\Lambda_{c'l}x + \lambda_{c'l}^\top x + \alpha_{c'l}\right)}. \tag{4.24}$$

This was shown in [Saul & Lee 02, Jebara 02, Gunawardana & Mahajan$^+$ 05]. Such a log-linear model shall be referred to as a log-linear mixture model (LMM). The log-linear parameters $\Lambda \in \{\{\Lambda_{cl} \in \mathbb{R}^{D\times D}\}, \{\lambda_{cl} \in \mathbb{R}^D\}, \{\alpha_{cl} \in \mathbb{R}\}\} =: \Gamma$ can be determined from the Gaussian parameters Θ by comparing terms constant, linear, and quadratic in x. The resulting transformation rules are summarized in Table 4.3. Keep in mind that according to the Bayes rule, the joint prior $p(c, l)$ is the product of the class prior $p(c)$ and the mixture weight $p(l|c)$, $p(c, l) = p(l|c)p(c)$.

Assuming an LMM with parameters $\Lambda \in \Gamma$, an equivalent GMM can be determined by solving the transformation rules in Table 4.3 for some Gaussian parameters θ. However, this approach does not define a proper GMM in general, *i.e.*, $\theta \notin \Theta$. This is because the covariance matrix Σ_{cl} is not guaranteed to be positive-definite (if the inverse is defined at all), and the priors $p(c)$ and mixture weights $p(l|c)$ do not need to be normalized. This observation could explain why some authors assume that the log-linear models in Equation (4.14) are "more

Table 4.4: Transformation of LMM into GMM parameters, '←' indicates an invariance transformation and "passing" is an abbreviation for "passing of normalization constant." See text for explanations.

1.	$\tilde{\Lambda}_{cl}$	$\overset{\text{Equation (4.8)}}{\leftarrow}$	$\Lambda_{cl} + \Delta\Lambda$
2.	Σ_{cl}	$=$	$-\frac{1}{2}\tilde{\Lambda}_{cl}^{-1}$
3.	μ_{cl}	$=$	$\Sigma_{cl}\lambda_{cl}$
4.	$\tilde{\alpha}_{cl}$	$\overset{\text{"passing"}}{\leftarrow}$	$\alpha_{cl} + \frac{1}{2}\left(\mu_{cl}^\top \Sigma_{cl}^{-1}\mu_{cl} + \log\lvert 2\pi\Sigma_{cl}\rvert\right)$
5.	α_c	$\overset{\text{"passing"}}{\leftarrow}$	$\log\left(\sum_l \exp(\tilde{\alpha}_{cl})\right)$
6.	$\log p(l\|c)$	$\overset{\text{"passing"}}{\leftarrow}$	$\tilde{\alpha}_{cl} - \alpha_c$
7.	$\log p(c)$	$\overset{\text{Equation (4.10)}}{\leftarrow}$	$\alpha_c - \log\left(\sum_{c'} \exp(\alpha_{c'})\right)$

expressive than their generative counterparts" [Saul & Lee 02, Gunawardana & Mahajan+ 05, Sha & Saul 07a]. None of the existing work provides explicit transformation rules, see Section 4.2.1. Here, we derive the transformation rules by taking advantage of the ambiguity of the log-linear model parameters (see Section 4.3.4) to resolve this problem.

Table 4.4 summarizes the different steps required to transform an LMM into an equivalent (and proper) GMM. First, observe that the matrix Λ_{cl} is ambiguous. The invariance transformation in Equation (4.8) with a sufficiently negative-definite $\Delta\Lambda \in \mathbb{R}^{D\times D}$ (*i.e.,* the eigenvalues of $\Delta\Lambda$ are sufficiently negative) can be used to make Λ_{cl} negative-definite. Hence, the covariance matrix Σ_{cl} exists and is positive-definite (Step 2). The determination of the mean μ_{cl} is straightforward. Next, the mixture weights are normalized. The normalization constant from the Gaussian is incorporated into the prior parameter, $\tilde{\alpha}_{cl} := \alpha_{cl} + \frac{1}{2}\left(\mu_{cl}^\top \Sigma_{cl}^{-1}\mu_{cl} + \log\lvert 2\pi\Sigma_{cl}\rvert\right)$. The mixture weights result from the such corrected parameters by normalization (Step 6). The additional normalization constant $\exp(\alpha_c)$ defined in Step 5 is passed to the prior. The priors can be normalized by applying the invariance transformation in Equation (4.10) with $\Delta\alpha := -\log\sum_c \alpha_c$ (Step 7).

This subsection can be summarized by the following lemma.

Lemma 12 (Equivalence (GMM))**.** *The posterior model $p_{LMM,\Gamma}$ in Equation (4.24) and the posterior model $p_{GMM,\Theta}$ induced by the generative model in Equation (4.23) are equivalent.*

The proof of this lemma directly follows from the transformation rules in Table 4.3 and Table 4.4.

4.5.3 GHMMs for isolated word recognition

Isolated word recognition is based on the probabilistic model in Section 4.4.2. Now, the input is the sequence of feature vectors $x_1^T \in \mathbb{R}^{T\cdot D}$ and the tag sequences are substituted with the state

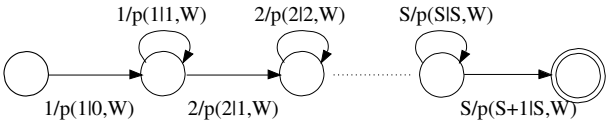


Figure 4.4: WFST representing the word-based transition model for isolated word recognition with loop and forward transitions, the edge labels $s/p \in \{1, \ldots, S, \$\} \times \mathbb{R}^+$ denote the HMM state and the transition weight (not normalized in general), respectively.

sequences s_1^T. In addition, W stands for the word.

$$p_{\text{W-LHMM},\Lambda}(W|x_1^T) = \frac{1}{Z_\Lambda(x_1^T)} \underbrace{\exp(\alpha_W)}_{\text{language model}} \sum_{s_1^T} \prod_{t=1}^{T} \underbrace{\exp(\alpha_{s_{t-1}s_tW})}_{\text{transition model}} \underbrace{\exp(\lambda_{s_tW}^\top x_t + \alpha_{s_tW})}_{\text{emission model}} \tag{4.25}$$

where $Z_\Lambda(x_1^T)$ denotes the normalization constant. The model parameters comprise $\Lambda \in \{\{\alpha_W \in \mathbb{R}\}, \{\alpha_{s'sW} \in \mathbb{R}\}, \{\alpha_{sW} \in \mathbb{R}\}, \{\lambda_{sW} \in \mathbb{R}^D\}\} =: \Gamma$. The generative analog with $\theta \in \{\{p(W) \in \mathbb{R}^+ | \sum_W p(W) = 1\}, \{p(s|s', W) \in \mathbb{R}^+ | \sum_s p(s|s', W) = 1\}, \{\mu_{sW} \in \mathbb{R}^D\}, \Sigma \in \mathbb{R}^{D\times D} | \Sigma \succ 0\} =: \Theta$ reads

$$p_{\text{W-GHMM},\theta}(x_1^T, W) = \underbrace{p(W)}_{\text{language model}} \sum_{s_1^T} \prod_{t=1}^{T} \underbrace{p(s_t|s_{t-1}, W)}_{\text{transition model}} \underbrace{\mathcal{N}(x_t|\mu_{s_tW}, \Sigma)}_{\text{emission model}}. \tag{4.26}$$

Lemma 13 (Equivalence (isolated words))**.** *The posterior model $p_{\text{W-LHMM},\Gamma}$ in Equation (4.25) and the posterior model $p_{\text{W-GHMM},\Theta}$ induced by the generative model in Equation (4.26) are equivalent.*

The proof of this lemma is based on the results in Section 4.4. Similar to above, the proof (to show $p_{\text{W-LHMM},\Gamma} \subset p_{\text{W-GHMM},\Theta}$) is step by step: the emission model is considered first, followed by the transition model, and then the language model is processed.

Emission model. The LMMs are transformed according to Table 4.4. The corrected transition parameters

$$\tilde{\alpha}_{s'sW} := \alpha_{s'sW} + \alpha_{sW} + \frac{1}{2}\mu_{sW}^\top \Sigma^{-1} \mu_{sW}. \tag{4.27}$$

will be used in the subsequent steps.

Word-based transition model. For this simple application, a word-based transition model is considered appropriate, as illustrated in Figure 4.4. The strict left-right topology of the transition probabilities leads to an upper triangular band transition matrix. In contrast to the bigram matrix Q in Section 4.4.2, this transition matrix is not strictly positive and is reducible (*i.e.,* a state cannot be reached by one of its subsequent states). Hence, the algorithm in Section 4.4.2 is not guaranteed to work. This, however, does not mean that the normalization is not possible. The normalization of the transition model is illustrated on the simple topology in Figure 4.4 only supporting loop and forward transitions. It is an example of the generalized framework introduced below (Section 4.6) that also covers more complex topologies.

Lemma 14 (Equivalence (loop/forward transitions))**.** *Assume the posterior model in Equation (4.25). Define the conditional probabilities of the generative model in Equation (4.26) as*

$$\begin{aligned} p(s|s,W) &:= \frac{\exp(\tilde{\alpha}_{ssW})}{z+\epsilon},\ s \in \{1,\dots,S\} \\ p(s+1|s,W) &:= 1 - p(s|s,W),\ s \in \{1,\dots,S-1\} \\ p(1|0,W) = p(S+1|S,W) &:= 1 \end{aligned}$$

with $z := \max_{s,W}\{\exp(\tilde{\alpha}_{ssW})\}$*, some* $\epsilon > 0$*, and* $\tilde{\alpha}_{ssW}$ *as defined in Equation (4.27). Then, the posterior model* $p_{W\text{-}LHMM,\Gamma}$ *in Equation (4.25) and the posterior model* $p_{W\text{-}GHMM,\Theta}$ *induced by the generative model in Equation (4.26) with the above defined transition probabilities are equivalent.*

Proof. The transition probabilities are plugged into the generative probability in Equation (4.26). This quantity is then transformed into the numerator in Equation (4.25) up to a constant factor (transition model only, emission model assumed to be normalized). Observe that each forward transition occurs exactly once and thus, the number of loops is $T-S-1$. The latter number is non-negative because $S \leq T-1$ in general.

$$\begin{aligned} p_{\text{W-GHMM},\Theta}(x_1^T, W) &\overset{\text{Equation (4.26)}}{=} p(W) \cdot \sum_{s_1^T} \prod_{t=1}^{T} p(s_t|s_{t-1},W)\mathcal{N}(x_t|\mu_{s_tW},\Sigma) \\ &\overset{\text{definition}}{=} p(W) \cdot \underbrace{\prod_{s=1}^{S+1} \frac{p(s|s-1,W)}{\exp(\tilde{\alpha}_{s-1sW})}}_{\text{correction factor (forwards)}} \underbrace{\left(\frac{1}{z+\epsilon}\right)^{T-S-1}}_{\text{correction factor (loops)}} \\ &\quad \cdot \sum_{s_1^T} \prod_{t=1}^{T} \exp(\tilde{\alpha}_{s_{t-1}s_tW})\mathcal{N}(x_t|\mu_{s_tW},\Sigma) \\ &= \underbrace{p(W) \prod_{s=1}^{S+1} \frac{p(s|s-1,W)}{\exp(\tilde{\alpha}_{s-1sW})}(z+\epsilon)}_{=:\exp(\tilde{\alpha}_W)} \cdot \underbrace{\left(\frac{1}{z+\epsilon}\right)^{T}}_{\text{constant factor}} \\ &\quad \cdot \sum_{s_1^T} \prod_{t=1}^{T} \exp(\tilde{\alpha}_{s_{t-1}s_tW})\mathcal{N}(x_t|\mu_{s_tW},\Sigma) \\ &= \left(\frac{1}{z+\epsilon}\right)^{T} \exp(\tilde{\alpha}_W) \cdot \sum_{s_1^T} \prod_{t=1}^{T} \exp(\tilde{\alpha}_{s_{t-1}s_tW})\mathcal{N}(x_t|\mu_{s_tW},\Sigma) \\ &\overset{\text{Equation (4.25)}}{=} \left(\frac{1}{z+\epsilon}\right)^{T} \cdot Z_\Lambda(x_1^T) p_\Lambda(W|x_1^T) \end{aligned}$$

The correction factors from the forward transitions and the word-dependent contribution of the correction factors from the loop transitions are put into the language model parameters. The word-independent contribution of the correction factors from the loop transitions cancels because it does not depend on the summation indices, *cf.* the invariance transformation in Equation (4.10). By definition of the parameters, the last line follows. □

The reader is referred to Section 4.6.4 for a constructive proof of this lemma.

Language model. Simple priors, *i.e.,* unstructured language models, are assumed for isolated word recognition. In this case, the normalization of the language model is similar to the normalization of the priors for GMMs. The normalization constant of the language model probabilities $p(W) := \frac{\exp(\tilde{\alpha}_W)}{\sum_V \exp(\tilde{\alpha}_V)}$ does not affect the posterior model because it is an invariance transformation of the type in Equation (4.10).

Finally, equivalence relations for HMMs in the context of continuous speech recognition are discussed in the next section.

4.5.4 GHMMs in continuous speech recognition

In continuous speech recognition, the label W in Equation (4.25) stands for a word sequence, $W = w_1^N$. In contrast to isolated word recognition, a structured language model (*e.g.* an m-gram language model) and a simplified transition model are assumed. Considering word sequences of variable length, an additional difficulty in Lemma 11, which assumes sequences of the same length, is introduced thereby.

Assuming a bigram language model for simplicity, the LHMM in Equation (4.25) is modified to

$$\begin{aligned} p_{\text{ASR-LHMM},\Lambda}(W|x_1^T) &= \frac{1}{Z_\Lambda(x_1^T)} \prod_{n=1}^{N} \underbrace{\exp(\alpha_{w_{n-1}w_n})}_{\text{language model}} \\ &\quad \cdot \sum_{s_1^T : w_1^N} \prod_{t=1}^{T} \underbrace{\exp(\alpha_{s_{t-1}s_t})}_{\text{transition model}} \underbrace{\exp(\lambda_{s_t}^\top x_t + \alpha_{s_t})}_{\text{emission model}} \end{aligned} \tag{4.28}$$

with the normalization constant $Z_\Lambda(x_1^T)$. The model parameters are $\Lambda \in \{\{\alpha_{vw} \in \mathbb{R}\}, \{\alpha_{s's} \in \mathbb{R}\}, \{\alpha_s \in \mathbb{R}\}, \{\lambda_s \in \mathbb{R}^D\}\} =: \Gamma$. Similarly, the generative model is

$$p_{\text{ASR-GHMM},\theta}(x_1^T, W) = \prod_{n=1}^{N} \underbrace{p(w_n|w_{n-1})}_{\text{language model}} \sum_{s_1^T : w_1^N} \prod_{t=1}^{T} \underbrace{p(s_t|s_{t-1})}_{\text{transition model}} \underbrace{\mathcal{N}(x_t|\mu_{s_t}, \Sigma)}_{\text{emission model}}. \tag{4.29}$$

The model parameters are $\theta \in \{\{p(w|v) \in \mathbb{R}^+ | \sum_w p(w|v) = 1\}, \{p(s|s')| \sum_s p(s|s') = 1\}, \{\mu_s \in \mathbb{R}^D\}, \Sigma \in \mathbb{R}^{D\times D} | \Sigma \succ 0\} =: \Theta$.

Lemma 15 (Equivalence (continuous speech))**.** *The posterior model $p_{\text{ASR-LHMM},\Gamma}$ in Equation (4.28) and the posterior model $p_{\text{ASR-GHMM},\Theta}$ induced by the generative model in Equation (4.29) are equivalent.*

The proof of this lemma is along the same lines as above (*e.g.* Section 4.5.3), *i.e.,* the submodels are normalized step by step.

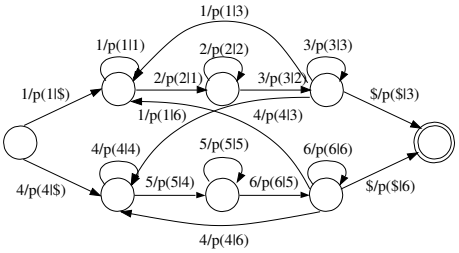


Figure 4.5: WFST representing a phoneme-based transition model for continuous speech recognition with with loop and forward transitions, the edge labels $s/p \in \{1, \ldots, 6, \$\} \times \mathbb{R}$ denote the HMM state and the transition weight (not necessarily normalized as implied by the symbol p), respectively. Keep in mind that $\$ \rightarrow 1/4$ and $3/6 \rightarrow \$$ implement the entry and exit transitions.

Emission model. Similar to Section 4.5.3, the normalization of the Gaussians results in corrected transition parameters, $\tilde{\alpha}_{s's} := \alpha_{s's} + \alpha_s + \frac{1}{2}\mu_s^\top \Sigma^{-1} \mu_s$. In continuous speech recognition, some state tying (*e.g.* CART) is typically used. It is clear from the dependencies of the emission and transition parameters that the passing of the normalization constant is not conflicting as long as the state tying for the transition model is not coarser than the state tying used for the emission model.

Phoneme-based transition model. For continuous speech recognition, a phoneme-based transition model is typically used. The WFST in Figure 4.5 models the word HMMs. The additional normalization constant $Z(\$) \equiv Z$ (see Sections 4.5.3 and 4.6) is propagated to the language model, $\tilde{\alpha}_{vw} := \alpha_{vw} - \log Z$. The invariance transformation

$$\alpha_{s's} \mapsto \alpha_{s's} + \Delta\alpha, \ \Delta\alpha \in \mathbb{R} \tag{4.30}$$

implies that the constant Z is not unique. This ambiguity shall be used in the next paragraph to normalize the language model.

***M*-gram language model.** The bigram model from Section 4.4.2 and illustrated in Figure 4.3 is used. Define the transition matrix $Q := [\exp(\tilde{\alpha}_{vw})]$. Let ν_w be the coefficients of the eigenvector of Q associated with the greatest eigenvalue q. The bigram probabilities $p(w|v) = \frac{Q_{vw}\nu_w}{q\nu_v}$ are the solution if $q = 1$. Otherwise, the factor $\frac{1}{q^{N+1}}$ in Equation (4.21) causes problems because it does not cancel. Two solutions are presented to avoid these problems. First, the language model can be extended to include the length N, $p(N, w_1^N) = p(N)p(w_1^N)$ with $p(N) \propto \exp(\alpha N)$. Then, the additional factors are put into this prior on N. The second solution does not modify the language model. The next lemma proves that q is a surjective continuous function of $\Delta\alpha$ in Equation (4.30) with image $(0, \infty)$. Then, according to the *Intermediate Value Theorem* in [Walter 99, Band 1, p.123], some $\Delta\alpha$ exists such that $q(\Delta\alpha) = 1$.

Lemma 16 (Continuity)**.** *Assume the set* $A := \{\Delta\alpha \in \mathbb{R} : \exp(s, s) < 1, \ \forall s\}$. *Then,* $q : A \rightarrow (0, \infty), \ \Delta\alpha \mapsto q(\Delta\alpha)$ *defines a surjective continuous function of* $\Delta\alpha$, cf. *Theorem 10.*

Proof. The function q is the concatenation of surjective continuous functions and thus, is also continuous. To show this, consider the functions:

i) $Z : A \to (0,\infty),\ \Delta\alpha \mapsto \sum_{s_1^T} \prod_{t=1}^{T} \exp(\tilde{\alpha}_{s_{t-1}s_t} + \Delta\alpha)$ is continuous as long as the infinite sum converges, *i.e.,* $\Delta\alpha \in A$. The function is surjective because the extremal points $Z \overset{\Delta\alpha\to-\infty}{\to} 0$ and $Z \overset{\Delta\alpha\to\sup\{A\}}{\to} \infty$ are a subset of the image and thus, the complete interval by the *Intermediate Value Theorem*.

ii) $Q_{vw} : (0,\infty) \to (0,\infty),\ Z \mapsto Q_{vw} = Z \cdot \exp(\tilde{\alpha}_{vw})$ is continuous, and surjective because Z is merely scaled by the positive constant $\exp(\tilde{\alpha}_{vw})$.

iii) Consider the function $f : \mathbb{R}^{S\times S} \times \mathbb{R} \to (0,\infty),\ (Q,q) \mapsto \det(Q - q\mathcal{I})$. Then, the implicit function $f(Q,q) = 0$ defines the eigenvalues q of Q. Choose q to be the greatest eigenvalue of Q. Under this assumption, the Jacobian $\frac{\partial f(Q,q)}{\partial q}$ does not vanish because otherwise the multiplicity of the greatest eigenvalue would be greater than one. This would contradict the statement of the *Perron-Frobenius Theorem* [Rao & Rao 98, p.475]. Hence, the *Implicit Function Theorem* in [Walter 99, Band 2, p.114] applies, *i.e., q* is locally continuous. By a finite coverage, q is continuous on the complete domain.

This function q is surjective because $q(-\infty) = 0$ and $q(\infty) = \infty$ are a subset of the image and thus, the complete interval by the *Intermediate Value Theorem*. □

After this manipulation, the algorithm from Section 4.4.2 applies to the transformation matrix Q induced by $\tilde{\alpha}_{vw} = \alpha_{vw} + \log Z(\$)$.

So far, we have derived the solution for bigram probabilities. This result can be extended to m-gram probabilities *etc.* In general, the transition matrix in Section 4.4.2 describes the non-negative transition probabilities between two states, c and c', which encode the dependency of the conditional probabilities. If the transition matrix $Q = [Q_{c'c}]$ is irreducible, then according to the extension of the *Perron-Frobenius Theorem* in [Rao & Rao 98], Lemma 11 applies.

In the case of m-gram models, the dependency (also known as history) consist of the previous $m-1$ words. For a vocabulary of size C, this results in an approximately $C^{m-1} \times C^{m-1}$ transition matrix. In particular, higher-order m-gram models can be tackled in the same way as bigram models. This is in contrast to the belief in [Jaynes 03] that higher-order m-gram models require tensors of rank more than two which would go beyond the standard matrix formalism.

Typical ASR systems involve several heuristics and approximations. The next subsection shows to what extent they are compatible with the equivalence relations derived so far.

4.5.5 Heuristics & approximations

The submodels in Equations (4.26,4.29) are typically scaled, *e.g.* the scaling of the language model $p(w|v) \to p(w|v)^A$ in Equation (4.29). Unlike the generative formulation (*i.e.,* ML), these additional scaling factors do not add flexibility to the model in the discriminative formulation. This can be seen by combining these scaling factors with the log-linear model parameters, *e.g.* the language model parameter α_{vw} is replaced by $A \cdot \alpha_{vw}$. Strictly speaking, these scaling factors are redundant in the discriminative framework, *i.e.,* they do not need to be tuned or justified. In practice, they might have indirect impact on the results due to the spurious local optima

of conventional training criteria (*cf.* HCRFs). The redundancy of the scaling factors has a couple of unexpected effects, which are discussed on the example for part-of-speech tagging in Section 4.4.2 to keep the notation simple.

First, the scaled generative model can be replaced with an equivalent generative model *without* scaling factors such that the two induced posterior models are the same. A different interpretation of this effect is that the ML training criterion is suboptimal, and the scaling factors can compensate for this deficiency to some degree. More refined training criteria (*e.g.* MMI) will hopefully be closer to the optimal solution.

Lemma 17 (Scaled *vs.* unscaled model)**.** *The scaled generative model in Equation (4.15)*

$$p_{Gen,\theta AB}(x_1^N, c_1^N) \quad = \quad p(\$|c_N)^A \prod_{n=1}^{N} p(c_n|c_{n-1})^A p(x_n|c_n)^B \tag{4.31}$$

with scaling factors $A, B \in \mathbb{R}$, and the unscaled generative model

$$\tilde{p}_{Gen,\theta}(x_1^N, c_1^N) \quad = \quad \tilde{p}(\$|c_N) \prod_{n=1}^{N} \tilde{p}(c_n|c_{n-1}) \tilde{p}(x_n|c_n). \tag{4.32}$$

induce equivalent posterior models.

Proof. The proof is similar to the proof of Lemma 11. Define the emission and bigram probabilities of the unscaled generative model as

$$\tilde{p}(x|c) := \frac{p(x|c)^B}{\sum_{x'} p(x'|c)^B} \qquad \tilde{p}(c|c') := \frac{Q_{c'c} v_c}{q v_{c'}}$$

with the transition matrix $Q := [p(c|c')^A \sum_x p(x|c)^B]$, the greatest eigenvalue q of Q, and v_c the components of the eigenvector associated with q. These generative probabilities are well-defined and can be checked easily. The equivalence of the two posterior models is verified by plugging the definitions for the emission and bigram probabilities $\tilde{p}(x|c)$ and $\tilde{p}(c|c')$ into the unscaled generative model, and showing that it is identical to the scaled generative model up to a constant factor

$$\begin{aligned}
\tilde{p}_{\mathrm{Gen},\theta}(x_1^N, c_1^N) \quad &= \quad \tilde{p}(\$|c_N) \prod_{n=1}^{N} \tilde{p}(c_n|c_{n-1}) \cdot \tilde{p}(x_n|c_n) \\
&= \quad \frac{Q_{c_N \$} v_\$}{q v_{c_N}} \prod_{n=1}^{N} \frac{Q_{c_{n-1} c_n} v_{c_n}}{q v_{c_{n-1}}} \cdot \frac{p(x_n|c_n)^B}{\sum_x p(x|c_n)^B} \\
&= \quad \frac{\sum_x p(x|\$)^B}{q^{N+1}} \underbrace{\prod_{n=1}^{N+1} \frac{v_{c_n}}{v_{c_{n-1}}}}_{\text{telescope product}} \cdot p(\$|c_N)^A \prod_{n=1}^{N} p(c_n|c_{n-1})^A p(x_n|c_n)^B \\
&= \quad \underbrace{\frac{\sum_x p(x|\$)^B}{q^{N+1}}}_{\text{constant factor}} \cdot p_{\mathrm{Gen},\theta AB}(x_1^N, c_1^N).
\end{aligned}$$

The telescope product over $\frac{v_c}{v_{c'}}$ is 1 by our model assumption that $c_0 = c_{N+1} = \$$. The constant factor cancels in the posterior induced by the Bayes rule in Equation (4.12). Hence, equivalence holds true. □

Second, the scaling factors can be restored when transforming a log-linear model into the generative model. Again, this is illustrated for the example in Section 4.4.2.

Lemma 18 (Restoring scaling)**.** *Assume the scaled generative model in Equation (4.31) with*

$$p(x|c) := \left(\frac{\exp(\beta_{cx})^{\frac{1}{B}}}{\sum_{x'}\exp(\beta_{cx'})^{\frac{1}{B}}}\right) \qquad p(c|c') := \frac{Q_{c'c}v_c}{qv_{c'}}.$$

Here, $Q := [\exp(\alpha_{c'c} + B\log\sum_x \exp(\beta_{cx})^{\frac{1}{B}})^{\frac{1}{A}}]$ *denotes the transition matrix,* q *the greatest eigenvalue of* Q*, and* v_c *the components of the eigenvector associated with* q*. Then, the posterior in Equation (4.16) and the posterior induced by the scaled generative probability defined above are identical.*

Proof. The emission and bigram models of the scaled generative model are well-defined and can be checked easily. To show that the two posteriors are identical, the definitions for $p(x|c)$ and $p(c|c')$ are plugged into Equation (4.31)

$$\begin{aligned} p_{\mathrm{Gen},\theta AB}(x_1^N, c_1^N) &= p(\$|c_N)^A \prod_{n=1}^{N} p(c_n|c_{n-1})^A \cdot \frac{\exp(\beta_{c_n x_n})}{\left(\sum_x \exp(\beta_{c_n x})^{\frac{1}{B}}\right)^B} \\ &= \underbrace{\frac{\left(\sum_x \exp(\beta_{\$x})^{\frac{1}{B}}\right)^B}{q^{A(N+1)}}}_{\text{constant factor}} \underbrace{\prod_{n=1}^{N+1} \frac{v_{c_n}^A}{v_{c_{n-1}}^A}}_{\text{telescope product}} \cdot \exp(\alpha_{c_N\$}) \prod_{n=1}^{N} \exp(\alpha_{c_{n-1}c_n} + \beta_{c_n x_n}). \end{aligned}$$

The constant factor cancels in the posterior induced by the Bayes rule in Equation (4.12). The telescope product is 1 by our model assumption that $c_0 = c_{N+1} = \$$. The remaining term is identical to the numerator in Equation (4.16). □

Finally, the maximum rather than the exact sum is used on different levels in speech recognition. The sum can be replaced by the maximum in the above derivations without changing the equivalence relations. This is possible because, like for the sum, (positive) constant factors can be moved outside the maximum. Also, the normalization constant for the posteriors is typically approximated (*e.g.* word lattices to approximate the summation space). The equivalence relations do not fail in this approximation because ratios are considered for which the true normalization constant cancels.

Next, the techniques introduced so far are formalized to derive a general transformation algorithm.

4.6 Generalization

The equivalence of undirected discriminative models (*cf.* Markov random fields) and directed generative models (*cf.* Bayesian networks) is formalized in this section. In particular, conditions for log-linear models are formulated that are sufficient to transform a log-linear model into an equivalent generative model. The construction of the equivalent generative model is based on the ideas in [Jaynes 03] and [Mohri 09, p.242]. The above equivalence relations (*e.g.* Section 4.5.4) are non-trivial examples.

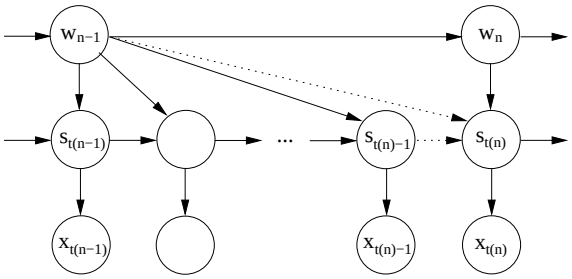


Figure 4.6: Dependency network for continuous speech recognition and bigram language model, the dotted arrows show the dependency added by across word modeling.

4.6.1 Definitions

Observed and unobserved random variables are distinguished, $x \in X$ and $c \in C$, respectively. Sequences of these variables are denoted by $x_1^M \in \mathcal{X}$ and $c_1^N \in C$. In general, the sequences can be of different length, and $\mathcal{X}$ and C do not need to be the complete set of all possible sequences, $\mathcal{X} \subset \cup_M X^M$ and $C \subset \cup_N C^N$. The results in this section are restricted to sequences of finite length to avoid technical complications with infinite sums. Infinite sequences can be regarded as the limit of finite sequences. This assumes that the limits exist which is not considered an issue because the quantities of interest are ratios of infinite sums as will become clear below. To simplify the notation, $n \in X \cup C$ stands for a variable either from X or C, and $n_1^T \in \mathcal{X} \cup C$ is a sequence of variables n such that n_1^T without the variables from X is an element of C, and vice versa. Example (part-of-speech tagging in Section 4.4.2): $n_1^{T=2N} = x_1, c_1, x_2, c_2, \ldots, x_N, c_N$. A few more definitions are needed for the next subsection [Lauritzen 96].

Definition 19 (Parents of node)**.** *Assume the graph $\mathcal{N} = (V, E)$. The* parents *of a node $n \in V$ is the set of nodes that have a link to node n, $Par(n) = \{n' \in V | (n', n) \in E\}$.*

Definition 20 (Dependency network)**.** *A* dependency network *of a distribution is a graph $\mathcal{N} = (V, E)$ with $V = X \cup C$. The set of links is defined to be the intersection of all sets of links such that $n \in V$ is conditionally independent of $V \backslash n$ given $Par(n)$.*

If the dependency network $\mathcal{N}$ is a directed acyclic graph (DAG), a topological ordering exists such that

$$\tilde{p}(n_1^T) = \prod_{t=1}^{T} \tilde{p}(n_t | Par(n_t)) \tag{4.33}$$

with $\tilde{p}(n|Par(n)) \geq 0$, but not necessarily normalized as indicated by $\tilde{}$. This definition reminds of *Bayesian networks*. In the example of Figure 4.6, we set: $X = \mathbb{R}^D$, $C = V \cup \{1, \ldots, S\}$ (V: vocabulary), and $\mathcal{X} \cup C$ is restricted to state sequences s_1^T that represent a valid word sequence.

Definition 21 (Future)**.** *Assume a partial (start) sequence n_1^t. The set of partial (end) sequences given the past n_1^t, $\mathcal{F}(n_1^t, t) = \{n_{t+1}^T | n_1^T \in \mathcal{X} \cup C\}$ is called the future of n_1^t.*

Note that the future $\mathcal{F}(n_1^t, t)$ typically does not depend on the complete sequence n_1^t nor the length of the sequence, but rather only on a few variables, *e.g.* only on $Par(n_{t+1})$ for all $n_1^T \in \mathcal{X} \cup C$. Now, we are in the position to formulate the sufficient conditions for the log-linear models.

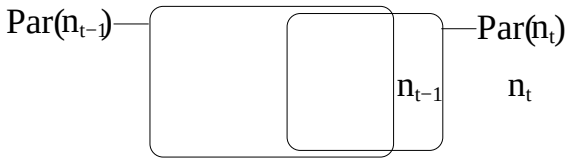


Figure 4.7: Illustration of second condition (nesting of variables).

4.6.2 General transformation of log-linear into generative models: Sufficient conditions

A set of sufficient conditions for the log-linear models are introduced and discussed. A constructive proof can be found in the next subsection.

Lemma 22 (Sufficient conditions)**.** *Assume a log-linear model $p_\Lambda(c_1^N|x_1^M)$ with feature functions $f_i(x_1^M, c_1^N)$, $x_1^M \in \mathcal{X}$ and $c_1^N \in C$. For each i, choose a pseudo probability $\tilde{p}(n|Par(n)) := \exp(f_i(x_1^M, c_1^N))$, $n \in X \cup C$, $Par(n) \subset X \cup C$, $f_i(x_1^M, c_1^N) \equiv f_i(n, Par(n))$ such that:*

1. *The dependency network $\mathcal{N}$ induced by $\tilde{p}(\cdot|\cdot)$ is a DAG with topological ordering n_1^T and $\tilde{p}(n_1^T) = \prod_{t=1}^{T} \tilde{p}(n_t|Par(n_t))$.*
2. *$\forall n_t \in \mathcal{N}$: $Par(n_t) \subseteq \{n_{t-1}\} \cup Par(n_{t-1})$.*
3. *$Par(n_{t+1}) = Par(\tilde{n}_{\tilde{t}+1}) \Rightarrow \mathcal{F}(n_1^t, t) = \mathcal{F}(\tilde{n}_1^{\tilde{t}}, \tilde{t})$, $\forall n_1^t, \tilde{n}_1^{\tilde{t}} \in \mathcal{X} \cup C$.*

Then, probabilities $p(n|Par(n))$ exist such that

$$\frac{\prod_{t=1}^{T} p(n_t|Par(n_t))}{\sum_{\tilde{n}_1^{\tilde{T}} \in \{x_1^M\} \cup C} \prod_{t=1}^{T} p(\tilde{n}_t|Par(\tilde{n}_t))} \equiv p_\Lambda(c_1^N|x_1^M) \; (\forall c_1^N \in C, x_1^M \in X),$$

i.e., *the generative model induced by $p(n|Par(n))$ and the log-linear model $p_\Lambda(c_1^N|x_1^M)$ are equivalent in the sense of Definition 7.*

A few comments are due. The first condition allows for a decomposition according to the Bayes rule. The resulting model thus is in agreement with a fundamental property of probability distributions. According to the second condition, the random variables need to be nested as illustrated in Figure 4.7. This condition guarantees that $p(n|Par(n))$ can be properly normalized without changing $p(c_1^N|x_1^M)$ by passing the local normalization constants from one position t to the next lower without breaking the independence assumptions. Example (part-of-speech tagging model in Section 4.4.2): the normalization constant of the emission scores only depends on the current tag such that it can be propagated to the bigram parameter as illustrated in Equation (4.19). The third condition is required to make the conditional probabilities position independent, *i.e.,* stationary. This is achieved by assuming that the future only depends on the parents of the variable under consideration, $\mathcal{F}(n_1^t, t) \equiv \mathcal{F}(Par(n_{t+1}))$, or the state index in case of finite state automata.

These issues will become more clear in the next subsection where we provide the general procedure of generating the generative model from a log-linear model satisfying these conditions.

4.6.3 Construction of generative models from discriminative models

The construction of the generative models is based on the invariance transformations introduced in Section 4.3.4, *i.e.,* the conditional probabilities are normalized locally and the resulting additional normalization constants are then passed to a lower (*i.e.,* not yet processed) position. The key quantities are the sums of the pseudo probabilities over all valid sequences sharing the past (*cf.* marginalization and backward probabilities in particular)

$$Z(n_1^t, t) \quad = \quad \sum_{n_{t+1}^T \in \mathcal{F}(n_1^t, t)} \tilde{p}(n_1^T). \tag{4.34}$$

For the empty sequence $n_1^0 = \epsilon$, this quantity provides the normalization constant. For the full sequence n_1^T, this quantity is equal to the pseudo probability $\tilde{p}(n_1^T)$. The next lemma shows how to construct a generative model from a given log-linear model.

Lemma 23 (Construction)**.** *Define the functions* $f(n_1^t, t) := \frac{Z(n_1^t, t)}{Z(n_1^{t-1}, t-1)}$. *Under the assumptions of Lemma 22,*

1. $f(n_1^t, t) \equiv p(n_t|Par(n_t))$, i.e., $f(n_1^t, t)$ *satisfies the properties of conditional probabilities in Equation (4.4-4.6).*
2. *The posterior model induced by the generative model determined by* $p(n|Par(n))$ *and the log-linear model* $p_\Lambda(c_1^N|x_1^M)$ *are equivalent.*

Proof. The proof of this lemma is similar to the proof in Section 4.4.2 concerning conditional probabilities: 1) check the properties of $p(n|Par(n))$ and 2) verify that $\prod_{t=1}^T p(n_t|Par(n_t)) \propto \tilde{p}(n_1^T)$ where the proportionality constant does not depend on c_1^N (equivalence).

1. The auxiliary quantity $f(n_1^t, t)$ is non-negative by definition and normalized because $\sum_{n_t} f(n_1^t, t) \overset{\text{Equation (4.34)}}{=} \frac{\sum_{n_t} Z(n_1^t, t)}{Z(n_1^{t-1}, t-1)} = \frac{Z(n_1^{t-1}, t-1)}{Z(n_1^{t-1}, t-1)} = 1$. Hence, it defines a proper probability distribution $p_t(n_t|n_1^{t-1}) \equiv f(n_1^t, t)$. Dependence and stationarity properties follow from the conditions in Lemma 22

$$\begin{aligned}
f(n_1^t, t) &\overset{\text{def.}}{=} \frac{Z(n_1^t, t)}{Z(n_{t-1}, t-1)} \\
&\overset{\text{Equation (4.34) \& Condition 1}}{=} \frac{\tilde{p}(n_t|Par(n_t)) \sum_{n_{t+1}^T \in \mathcal{F}(n_1^t, t)} \prod_{\tau=t+1}^T \tilde{p}(n_\tau|Par(n_\tau))}{\sum_{n_t^T \in \mathcal{F}(n_1^{t-1}, t-1)} \prod_{\tau=t}^T \tilde{p}(n_\tau|Par(n_\tau))} \\
&\equiv f(n_1^t \cap \cup_{\tau=t}^T Par(n_\tau), t) \\
&\overset{\text{Condition 2}}{\equiv} f(\{n_t\} \cup Par(n_t), t) \\
&\overset{\text{Condition 3}}{\equiv} f(\{n_t\} \cup Par(n_t)).
\end{aligned}$$

In summary, $f(n_1^t, t) \equiv f(n_t, Par(n_t)) \equiv p(n_t|Par(n_t))$.

2. The equivalence holds true because

$$\prod_{t=1}^{T} \tilde{p}(n_t|Par(n_t)) \overset{\text{Equation (4.34)}}{=:} Z(n_1^T, T)$$

$$\overset{\text{telescope product}}{=} Z(\epsilon, 0) \cdot \prod_{t=1}^{T} \frac{Z(n_1^t, t)}{Z(n_1^{t-1}, t-1)}$$

$$\overset{\text{first item of proof}}{=} Z(\epsilon, 0) \cdot \prod_{t=1}^{T} p(n_t|Par(n_t)).$$

□

Corollary 24 (Construction)**.** *Lemma 22 extends to models with hidden variables h of the type $p(c) = \sum_h p(c, h)$ or $p(c) = \max_h\{p(c, h)\}$.*

The above discussion implies that sums over all variable sequences need to be calculated. The calculation, however, can be made more local and efficient by processing the nodes in the dependency network in reverse topological order and correcting the parameters (quantities with ˜), if necessary. The example calculations in this chapter are in this vein.

4.6.4 Examples

A few examples are given to illustrate the theoretical results of this section.

Local context-dependency. Consider the part-of-speech tagging model with log-linear model parameters $\alpha_{c_{n-1}c_n}$ and $\beta_{x_{n-1}x_nx_{n+1}c_n}$. This model refines the model from Section 4.4.2 by adding dependency regarding x. The choice of the generative models $p(c_n|c_{n-1})$ and $p(x_n|x_{n-1}, x_{n+1}, c_n)$ lead to the violation of the first condition in Lemma 22. Alternatively, assume the generative models $p(c_n|c_{n-1})$ and $p(x_{n+1}|x_{n-1}, x_n, c_n)$ to satisfy the first condition. This ansatz, however, violates the second condition. This suggests that it is not possible to find an equivalent generative model with the same structure. Nevertheless, we can define windowed features $X_n = (x_{n-1}, x_n, x_{n+1})$ (common trick in speech recognition to take account of local context-dependency) and use them together with the simple tagging model introduced in Section 4.4.2. Clearly, this log-linear model is identical to the refined part-of-speech tagging model under consideration and thus, a generative model exists that induces an equivalent posterior model.

Maximum entropy Markov model (MEMM). According to the Bayes rule and independence assumptions, $p(c_1^N|X)$ can be decomposed into $\prod_{n=1}^{N} p(c_n|c_{n-1}, X)$, leading to MEMMs. The most general associated log-linear model uses feature functions of the type $f(c', c, X)$. In this general situation, the properties of Lemma 22 are all satisfied and thus, the MEMM/CRF pair is equivalent. This equivalence result does not contradict the "label bias" problem [Bottou 91, Lafferty & McCallum+ 01]. Typically, a subset of X rather than the complete X is used. This might be one of the reasons why CRFs outperform MEMMs in practice.

Word-based transition model. Consider word-based transition probabilities mentioned in Section 4.5.3. The strict left-right topology of the transition probabilities leads to an upper triangular band transition matrix such that the algorithm from Section 4.4.2 is not applicable. For this reason, we employ the general approach of this section. W.l.o.g. the loop transitions (the only cycles in the WFST) are assumed to have costs less than 1. This guarantees convergence in the marginalization step (summation over the state sequences).

If only loop and forward transitions with pseudo probabilities $\exp(\alpha_{s's|W})$ with $|s' - s| \leq 1$ are allowed as shown in Figure 4.4, then the transition probabilities can be calculated explicitly from backward scores defined in Equation (3.2)

$$\Psi(s, W) \quad := \sum_{s_t^T : s_t = s, s_T = S+1} \prod_{\tau=t+1}^{T} p(s_\tau | s_{\tau-1}, W).$$

The sum is over all state sequences s_t^T starting with the state, $s_t = s$, and ending with the final state, $s_T = S + 1$. The recursion formula for these quantities reads

$$\begin{aligned} \Psi(S+1, W) &= \exp(\alpha_{SS+1W}) \\ \Psi(s, W) &= \frac{\exp(\alpha_{ss+1W})\Psi(s+1, W)}{1 - \exp(\alpha_{ssW})} \\ \Psi(0, W) &= \exp(\alpha_{S+11W}) \cdot \Psi(1, W) \end{aligned}$$

for $s = S - 1, \ldots, 1$ and for all W. The factor $\frac{1}{1-\exp(\alpha_{ssW})}$ arises from the infinite sum accounting for the contributions of the loop transitions (*cf.* geometric series). These backward scores and the constants in Equation (4.34) are related as follows

$$Z(s_1^t, W, t) = \prod_{\tau=0}^{t} \exp(\alpha_{s_{\tau-1}\tau W}) \cdot \Psi(s_t, W).$$

Applying Lemma 23 using these partial sums, results in the transition probabilities

$$\begin{aligned} p(s|s, W) &= \exp(\alpha_{ssW}) & p(s+1|s, W) &= 1 - \exp(\alpha_{ssW}) \\ p(1|S+1, W) &= 1 & p(S+1|S, W) &= 1. \end{aligned}$$

The transition probabilities do not depend on $\exp(\alpha_{ss+1W})$ because the contribution of the forward transitions are the same for all state sequences and can be integrated in the language model, $\tilde{\alpha}_W = \alpha_W + \log Z(W)$. The same approach can be used for more complex topologies (*e.g.* including skips). In general, however, no analytical solution exists.

The across word modeling in combination with word-based transition probabilities is more tricky than for phoneme-based transition probabilities. Figure 4.6 suggests that the proposed algorithm fails due to the link between the final state s_{t_n-1} of the previous word w_{n-1} and the first state s_{t_n} of the current word w_n. This additional link avoids that the corresponding normalization constant can be distributed over the preceding CRF parameters as before, *i.e.,* the second condition of Lemma 22 is violated. This, however, is not critical in speech recognition because the last state of the previous word cannot be skipped by assumption and thus, is a function of the other variables, $s_{t_n-1} \equiv S(w_{n-1}, w_n, s_{t_n})$. Hence, $\tilde{\alpha}_{vw} = \alpha_{vw} + \log Z(v, w)$ where $Z(v, w)$ is the HMM normalization constant. Such across word models would require at least a bigram language model because the normalization of the transition parameters introduces this dependency.

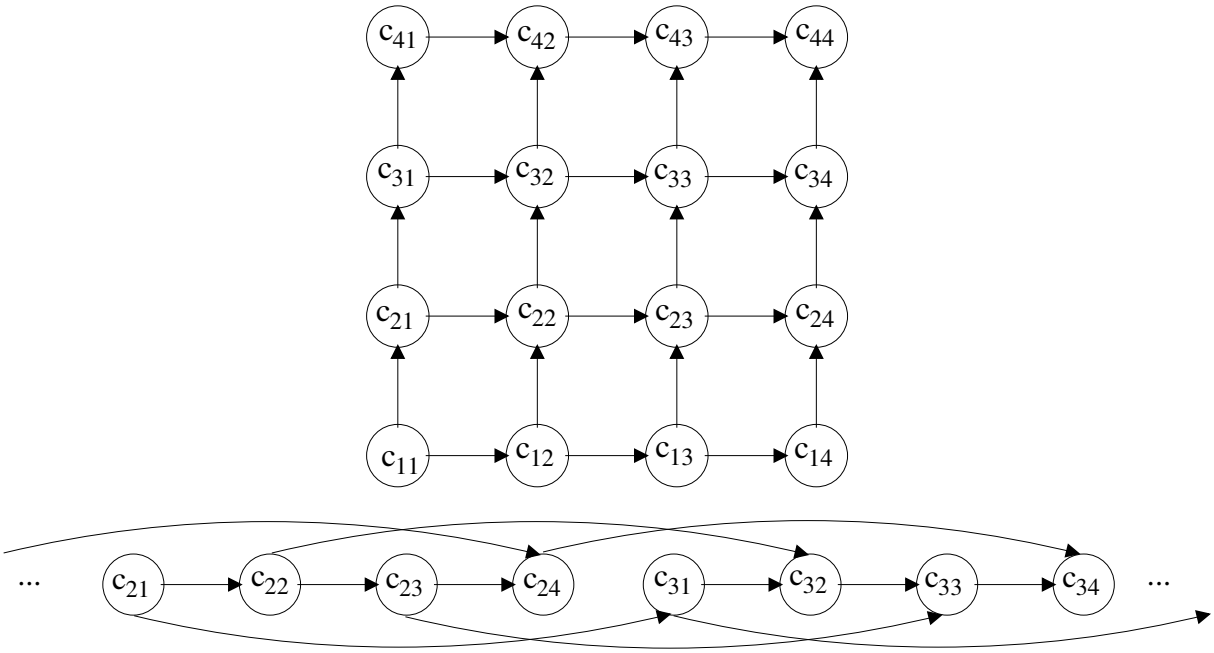


Figure 4.8: Dependency network for a 2-dimensional Markov model with nearest neighbors dependencies only, 2-dimensional (top) *vs.* 1-dimensional (bottom) representation.

Probabilistic weighted finite-state transducer (WFST). A word-based transition model (see the last paragraph) extends to arbitrary probabilistic WFSTs, interpreted either as a directed generative model or as a discriminative undirected model. Interestingly, the transformation from the undirected to the directed model can be performed by the weight pushing algorithm, see Section 3.1.3 [Mohri 09]. Weight pushing is one of the normalization steps used to check if two WFST instances are equivalent. Loosely related work for generative Bayesian networks can also be found in [Dupont & Denis$^+$ 05].

2-dimensional Markov model. Does the equivalence result for 1-dimensional Markov models extend to 2-dimensional Markov models? Figure 4.8 depicts the dependency network of a 2-dimensional Markov model with nearest neighbor dependencies only. Consider the topological ordering $n_1^T = c_{11}, c_{12}, \ldots, c_{21}, c_{22}, \ldots$ (row by row). Then, the second condition in Lemma 22 is violated

$$Par(c_{ij}) = \{c_{i-1j}, c_{ij-1}\} \quad \not\subseteq \quad \{c_{i-1j-1}, c_{ij-2}, c_{ij-1}, \} = \{c_{ij-1}\} \cup Par(c_{ij-1})$$

with $n_{t-1} = c_{ij-1}$, $n_t = c_{ij}$. As shown in Figure 4.8, the 2-dimensional Markov model can be represented as a 1-dimensional m-gram model with gaps such that Lemma 22 is fulfilled and the generalized approach can be applied.

4.7 Experimental Verification of Equivalence Relation

In this section, we check the correctness of the theoretical results experimentally. Different testing scenarios are reasonable.

Table 4.5: Concept error rate (CER) for different setups on the French Media evaluation set (*not* used directly for verification of equivalence).

Setup	Baseline	+$	+window+spelling
CER [%]	14.6	14.7	11.5

Indirect approach. An equivalent CRF/generative pair can be optimized separately and then, the performance of the two classifiers is compared [Macherey & Ney 03, Gunawardana & Mahajan+ 05]. Section 4.8 provides such comparisons for different speech tasks. For the complex tasks under consideration, it is difficult to control all parameters in practice, and the two classifiers typically lead to slightly different results. This might be due to numerical issues, local optima *etc.* For this reason, a more direct approach is preferred first.

Direct approach. A CRF is estimated, the resulting CRF is transformed into an equivalent generative model, and then it is shown that this generative model produces the same posteriors and decisions as the original associated CRF.

We start with the direct approach in this section.[2] The indirect approach is deferred until the next section. For the experiment, we used the CRF in Equation (4.16) that serves as a prototype for conditional probabilities. With this choice, the computational complexity can be kept low while avoiding artificial data.

Semantic part-of-speech tagging is a comparatively straightforward application of CRFs [Hahn & Lehnen+ 08]. It is usually defined as the extraction of a sequence of tags out of a given word sequence. A tag represents the smallest unit of meaning that is relevant for a specific task. A tag may contain various information, *e.g.* the attribute name or the corresponding value. An example from the French Media corpus is given in Figure 4.2, see [Devillers & Maynard+ 04] and Appendix A.2.1.

The experiments were carried out on the French Media corpus, see Appendix A.2.1. An attribute name is tagged for every source word to get a one-to-one alignment and use the suffixes "start_" and "_end" to indicate the start and end of a tag. The feature functions of the CRF use lexical features considering the current word only and transition features similar to a tag bigram model as in Equation (4.16). This CRF is estimated on the training part of the Media corpus. The resulting CRF is transformed according to the rules in Equation (4.17) and Lemma 11 into an equivalent generative model as given in Equation (4.15). The tagging of the training corpus using this generative model leads to exactly the same number of errors as using the original CRF, 9.3% concept error rate. The (differences of the) logarithmic probabilities of both models are illustrated in Figure 4.9. They can be considered identical within the numerical precision ($\approx 1 \pm 10^{-4}$) as the large peak at zero in Figure 4.9 clearly shows.

Table 4.5 provides a few additional error rates on the French Media task to give the interested reader an idea of the relative importance of the different feature functions. Like for speech recognition, the effect of the additional boundary symbol $ is marginal. Our best standard setup described in Appendix A.2 uses lexical features that not only consider the current word and spelling features in addition. As already mentioned, the corpus does not fully comply with the Media evaluation guidelines but fits well for a comparison of the systems.

[2] Thanks to Patrick Lehnen for the substantial contributions to this paragraph.

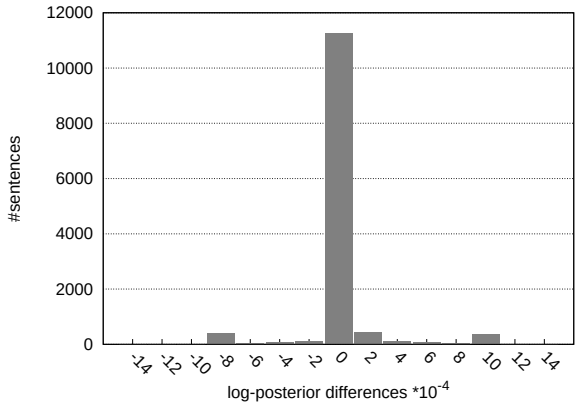


Figure 4.9: Distribution of log-posterior differences, zero difference means that the two log-posteriors are identical.

Table 4.6: Corpora and setups, BN (broadcast news), BC (broadcast conversation).

Identifier (description)	Audio data [h]	#States/#Dns Features/Setup
SieTill (German digit strings)	11.3 (Train) 11.4 (Test)	430/430-27k 25 LDA(MFCC)
EPPS En (English Parliament plenary sessions)	90 (Train) 3.2 (Dev06) 3.2 (Eval06)/2.9 (Eval07)	4,500/830k 45 LDA(MFCC+voicing) +VTLN+SAT/CMLLR
BNBC Cn (Mandarin BN & BC)	1,500 (Train) 2.6 (Dev07) 2.2 (Eval06)/2.9 (Eval07)	4,500/1,200k 45 SAT/CMLLR(PLP+voicing 3 tones+32 NN)+VTLN

4.8 Experimental Comparison of GHMMs and LHMMs

This section presents experimental results for the indirect approach, as discussed in the last section. Comparisons are provided for different speech recognition tasks, ranging from a simple digit string recognition task to large vocabulary continuous speech recognition (LVCSR) tasks, trained on up to 1,500h audio data, see Table 4.6 for an overview on the different tasks. Due to the equivalence relations of Gaussian and log-linear models, simply two different parameterizations of the same acoustic model are compared.

4.8.1 German digit strings

The recognition system is based on gender-dependent whole-word HMMs. 430 HMM states are used in total. The vocabulary consists of the German digits. The front-end consists of conventional cepstral features without derivatives. Temporal context is included by an LDA applied to a window of 5 consecutive frames, projecting the feature vector to 25 dimensions, see Appendix A.1.1. The corpus statistics is summarized in Table 4.6. The ML baseline system uses Gaussian mixtures with globally pooled variances to model the HMM states. These models are

Table 4.7: Word error rates (WER) for SieTill test corpus. The models differ in the number of densities per mixture, #Dns/Mix.

Model-#Dns/Mix	Criterion	Optimization	WER [%]
GHMM-1	ML	EM	3.8
	M-MMI	EBW	2.7
		Rprop	2.7
LHMM-1			2.7
GHMM-16	ML	EM	2.0
	M-MMI	EBW	1.9
		Rprop	1.8
LHMM-16			1.7
GHMM-64	ML	EM	1.8
	M-MMI	EBW	1.7
		Rprop	1.6
LHMM-64			1.6

Table 4.8: Word error rates (WER) for EPPS En test corpora.

Model	Criterion	Optimization	WER [%]		
			Dev06	Eval06	Eval07
GHMM	ML	EM	14.4	10.8	12.0
	MPE	EBW	13.4	10.2	11.5
		Rprop	13.4	10.3	11.5
LHMM			13.6	10.2	11.5

refined by discriminative training using M-MMI (see Chapter 5). The optimization was carried out with EBW or Rprop (GHMMs) and Rprop (LHMMs). The results are shown in Table 4.7. The observed differences between GHMMs and LHMMs are statistically insignificant.

4.8.2 English Parliament plenary sessions (EPPS)

This task contains recordings from the European Parliament plenary sessions (EPPS). The setup and corpus statistics are described in detail in Appendix A.1.3. A summary of this information can be found in Table 4.6. The acoustic front end comprises MFCC features augmented by a voicing feature. Nine consecutive frames are concatenated and the resulting vector is projected to 45 dimensions by means of an LDA. The MFCC features are warped using a fast variant of the vocal tract length normalization (VTLN). On top of this, speaker adaptive training (SAT) is applied. The triphones are clustered using CART, resulting in 4,501 generalized triphone states. For recognition, a lexicon with 50k entries in combination with a 4-gram language model is used. The ML baseline system uses Gaussian mixtures with globally pooled variances. These models are reestimated using MPE. Again, the GHMMs were optimized using EBW or Rprop while the log-linear models used Rprop for optimization. See Table 4.8 for the comparison of GHMMs and LHMMs for the EPPS task. The observed differences are not significant.

Table 4.9: Word error rates (WER) for BNBC Cn test corpora.

Model	Criterion	Optimization	WER [%]		
			Dev07	Eval06	Eval07
GHMM	ML	EM	12.0	17.9	11.9
	MPE	EBW	11.0	17.0	11.2
		Rprop	10.8	16.5	11.1
LHMM			10.8	16.2	10.8

Table 4.10: Globally pooled (first-order features) *vs.* density-specific diagonal covariance matrices (first- and diagonal second-order features) in the log-linear framework. Word error rates (WER) for BNBC Cn test corpora.

Model	Criterion	Features	WER [%]		
			Dev07	Eval06	Eval07
GHMM	ML	first	12.0	17.9	11.9
LHMM	MPE	first	10.8	16.2	10.8
		+diagonal second	10.8	16.2	10.8

4.8.3 Mandarin broadcasts

The second LVCSR task consists of Mandarin broadcast news and conversations (BN/BC). The experiments are based on the setup described in Appendix A.1.4. The corpus statistics of the system are shown in Table 4.6. The BNBC Cn system uses PLP features. Nine consecutive frames are concatenated and projected to 45 dimensions by means of an LDA. These base features are augmented with three tone and 32 neural network (NN) based posterior features. The features are adapted using VTLN and SAT. The lexicon with 60k entries and the 4-gram language model from [Hoffmeister & Plahl+ 07] are used for recognition. The results for this setup are shown in Table 4.9. Again, the differences are not considered to be significant.

In contrast to GHMMs, the transition from globally pooled to density-specific (diagonal) covariance matrices is straightforward in the log-linear framework. To emulate density-specific diagonal covariance matrices, the feature vector $(x_1, \ldots, x_D)$ is replaced with the augmented vector $(x_1, \ldots, x_D, x_1^2, \ldots, x_D^2)$ (*i.e.,* first- and diagonal second-order features instead of first-order features only). The experiment in Table 4.10 uses an ML optimized GHMM with globally pooled variances for the initialization of the discriminative training. The second-order features are only added for the discriminative training. No improvement over the system with first-order features has been observed, although convergence was reached after significantly fewer iterations (4 *vs.* 12 iterations).

4.8.4 Discussion

Tables 4.7, 4.8, and 4.9 show comparative results for GHMMs and LHMMs for the three speech tasks summarized in Table 4.6. These tasks are of completely different complexity. In all these cases, the equivalence is not perfect. This is because, as usual, only the acoustic model is trained while the transition and language models are kept fixed. Thus, the LHMMs refine the

unigram model parameters implicitly. We expect that this effect is covered by the full m-gram language model (*e.g.* $m = 4$). Furthermore, different regularization terms are used: I-smoothing (GHMMs) *vs.* centered 2 -regularization (LHMMs). It is not obvious how to eliminate this mismatch as the type of regularization is rather model-specific. In case of pooled variances, the choice of the parameterization is not considered to be an issue. In consequence, there are some differences in performance between these two types of model, see Table VII. Overall, however, no consistent or significant differences are observed. This result is in contrast to [6] which reports on statistically significant differences between GHMMs and LHMMs for phone classification. To the best of our knowledge, the equivalence is not broken in the setup in [6] as all model parameters are optimized jointly and no regularization is used. The most likely reasons for this different outcome may be the usage of density-specific variances (in contrast to global variances in our case) and local optima.

4.9 Summary

Conventional GHMMs and LHMMs ("Gaussian-like" log-linear HMMs) derive from fundamentally different paradigms in statistical pattern recognition. In spite of this, these two models were shown to be equivalent on the functional level. This result might appear surprising and counterintuitive because the parameter constraints and directed dependencies of the GHMM do not reduce the model flexibility of the fully unconstrained respective LHMM. This is possible because the parameters of GHMMs are ambiguous in the discriminative formulation. This ambiguity of the parameters also makes the interpretation of GHMMs in the discriminative formulation tricky (*e.g.* delocalization of means). The equivalence relations for GHMMs and LHMMs, however, do not guarantee identical performance of GHMMs and LHMMs in practice. For this reason, an extensive experimental comparison of GHMMs and LHMMs was done. Potential differences may originate from numerical issues (*e.g.* inversion of covariance matrices for GHMMs), local optima (non-convex objective function for HCRFs), or different optimization criteria (*e.g.* different regularization/smoothing terms). In general, it is essential to consider the complete optimization problem and not only parts of it (*e.g.* not only the acoustic model) to establish the exact equivalence relations for GHMMs and LHMMs. The careful analysis of GHMMs and LHMMs in this chapter helps to better understand why the conceptually more refined LHMMs do not outperform the conventional GHMMs, and to detect potential sources for improved acoustic modeling. Nevertheless, we consider the log-linear framework attractive for the flexible and intuitive incorporation of additional knowledge sources and dependencies. Last but not least, the convexity of the optimization problem of pure log-linear models might be a real advantage in practice (Chapter 7).

Chapter 5

Integration of Margin Concept into Standard Training

Typical training criteria for string recognition like for example minimum phone error (MPE) and maximum mutual information (MMI) in speech recognition are based on a (regularized) loss function. In contrast, large margin classifiers - the *de-facto* standard in machine learning - maximize the separation margin. An additional loss term penalizes misclassified samples. This paper shows how typical training criteria like for example MPE or MMI can be extended to incorporate the margin concept, and that such modified training criteria are smooth approximations to support vector machines with the respective loss function. The proposed approach takes advantage of the generalization bounds of large margin classifiers while keeping the efficient framework for conventional discriminative training in Chapter 3. This allows us to evaluate *directly* the utility of the margin term for string recognition. Experimental results are presented using the proposed modified training criteria for different tasks from speech recognition (including large vocabulary continuous speech recognition tasks trained on up to 1,500h audio data) [Heigold & Deselaers+ 08b, Heigold & Schlüter+ 09, Heigold & Dreuw+ 10], part-of-speech tagging, [Hahn & Lehnen+ 09] and handwriting recognition [Dreuw & Heigold+ 09].

A similar approach can be found in [Povey & Kanevsky+ 08, Saon & Povey 08]. The work in this chapter was developed independently at the same time. In addition to margin-based MMI, the present work includes margin-based MPE and other conventional training criteria as well.

5.1 Introduction

The estimation of parameters on a limited amount of data constitutes one of the fundamental problems in pattern recognition. On the one hand, we seek a solution that approximates the data well. On the other hand, the solution should generalize well to unseen data. Thus, the estimate will be the tradeoff between these two competing objectives in general.

The first aspect of the parameter estimation problem has been carefully investigated in speech recognition for many years, resulting in a wealth of penalty-like training criteria. These conventional training criteria were unified in [Macherey & Haferkamp+ 05, He & Deng+ 08].

Table 5.1: Relative importance of loss and margin term under different training conditions. The two extremes are dominated by the loss (left-hand side) or the margin (right-hand side).

Loss	*vs.*	Margin
infinite data	↔	sparse data
many training errors	↔	few training errors

Some of these training criteria include a regularization term like for example a non-uniform prior over the model parameters (*cf.* maximum *a posteriori* estimation), or an explicit ℓ_2-regularization.

Large margin training is a relatively new concept to pattern recognition. It was introduced to control the model complexity and the generalization ability. The objective of large margin training is the separation of the data with maximal margin (confidence). This approach is motivated by the theoretical generalization bounds derived in statistical learning theory [Vapnik 95]. Depending on the training conditions, we expect different relative importance of the margin and the loss term, as illustrated in Table 5.1.

5.1.1 Statistical learning theory

Assume a model with free parameters and a finite number of observations. The goal of machine learning consists of finding "optimal" model parameters with good generalization ability. An interesting result from information theory is the PAC bound on the empirical risk [Vapnik 95]. The Vapnik-Chervonenkis (VC) dimension plays an important role in the derived inequality and is a direct measure for the generalization ability. This bound is general in the sense that it neither depends on the underlying probability distribution nor on the specific loss function. Furthermore, the bound implies that in general, the consideration of the empirical risk alone is suboptimal [Vapnik 95]. Assuming that the features are in a sphere, the VC dimension of gap-tolerant classifiers is bounded above by an expression that is inversely proportional to the margin [Jebara 02]. These results are the theoretical foundation for large margin classifiers. The goal of this chapter is to assess the utility of the margin concept for string recognition, in particular for large vocabulary continuous speech recognition (LVCSR).

5.1.2 Motivation

The goal of this work is to study the potential of the margin concept for string recognition in practice. The focus shall be on large vocabulary continuous speech recognition (LVCSR). More explicitly, our objectives for such an investigation are:

1. Direct evaluation of the utility of the margin term. Ideally, we can turn on/off the margin term in the optimization problem. In particular, we want to avoid effects arising from different loss functions, optimization algorithms, model parameterizations, convergence speed *etc.* Unfortunately, but similar to most other approaches, we cannot exclude spurious local optima.

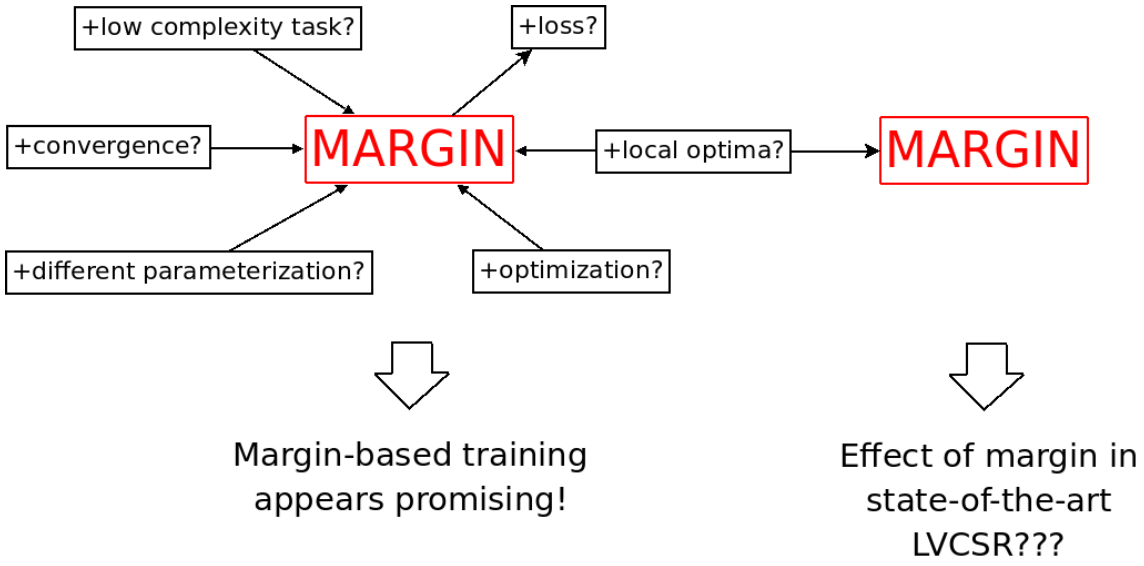


Figure 5.1: Left: existing approaches to large margin optimization in ASR. Besides the margin term, many other parameters and components are changed such that it is difficult to isolate the effect of the margin. Right: our objective to evaluate the utility of the margin term.

2. Evaluation on state-of-the-art systems. Ideally, we improve directly over the best discriminative system, *e.g.* conventional (*i.e.,* without margin) MPE for LVCSR.

3. Showing a clear relationship of conventional training criteria to existing large margin classifiers.

In our opinion, existing approaches to margin-based training implement insufficiently these objectives [Yin & Jiang 07, Sha & Saul 07a, Jiang & Li 07, Li & Yan+ 07, Yu & Deng+ 08, Saon & Povey 08]. To the best of the author's knowledge, no consistent evaluation of the margin term has been done for string recognition so far. The current situation may be summarized as in Figure 5.1.

5.1.3 Related work & our approach

Large margin classifiers, with the support vector machine (SVM) [Vapnik 95, Altun & Tsochantaridis+ 03, Taskar & Guestrin+ 03] as the most prominent example, have been used successfully for many applications in pattern recognition. The direct application of SVMs in speech recognition, however, has not been successful so far. A reason for this failure might be that SVMs are not flexible enough to deal with the speech-specific problems. They include the choice of the loss function (*e.g.* MPE appears to be the training criterion of choice in LVCSR, see Section 3.8.1), the immense amount of data to train state-of-the-art LVCSR systems, and the combinatorial number of valid word sequences. Stimulated by the success of SVMs, several margin-based training algorithms have been proposed in speech recognition that fit the speech-specific requirements in a better way, *e.g.* [Yin & Jiang 07,

Sha & Saul 07a, Jiang & Li 07, Li & Yan+ 07, Yu & Deng+ 08, Saon & Povey 08]. Although the reported results for these approaches look very promising, the existing approaches are limited concerning the scalability (*e.g.* LVCSR) or the choice of the training criterion. In addition, it is often difficult to draw clear conclusions on the utility of the margin term from the reported experiments. This is due to the fact that margin-based training criteria are compared with conventional training criteria using different loss functions, different optimization algorithms, different model parameterizations, or not taking into account potential differences in convergence speed.

In this work, conventional training criteria (*e.g.* MPE) are modified to incorporate a margin term. Such modified training criteria for log-linear models are shown to be a smooth approximation to the optimization problem of SVMs with a suitable loss function (Section 5.4.1). Thus, our approach combines the advantages of conventional training criteria (the efficient algorithms from Chapter 3) and of large margin classifiers (the generalization bounds). Similar ideas can be found in [Zhang & Jin+ 03] where a multiclass SVM suggested in [Weston & Watkins 99] with the hinge loss function is approximated by modified logistic regression. Recognition results were presented for the recognition of single symbols. To the best of our knowledge, modified logistic regression is computationally unfeasible for string recognition because of the pairwise treatment of the correct and the exponential number of competing word sequences. To avoid this exponential complexity, the formulation of the hidden Markov SVM proposed in [Altun & Tsochantaridis+ 03] is used. Using the smoothed sentence error of minimum classification error (MCE) in combination with $N-$best lists and *without* regularization, the margin-based MCE criterion proposed in [Yu & Deng+ 08] is recovered as a special instance of our approach. The authors in [Povey & Kanevsky+ 08] proposed an improved MMI criterion motivated by the boosting technique. It can be shown that this training criterion is identical to our margin-based MMI, apart from some technical details concerning the optimization algorithm. Similarly, [McDermott & Nakamura 08] defined boosted MPE, which is identical to our margin-based MPE. This is the only approach found in the literature that can be interpreted as a margin-based MPE training.

The remainder of this chapter is organized as follows. Section 5.2 introduces the modifications that are required to incorporate a margin term into conventional training criteria. The task-specific details are discussed in Section 5.3. The formal relationship of the proposed modified training criteria with large margin classifiers is shown in Section 5.4. Related approaches are discussed in Section 5.5. Comparative experimental results for the different tasks are presented in Section 5.6. The chapter concludes with the summary in Section 5.7.

5.2 Incorporation of Margin Term

The training criteria are introduced next. Assume the joint probability $p_\Lambda(X, W)$ of the features X and the symbol string W. The exact meaning of X and W depends on the task, and will be discussed in Section 5.3. In general, the joint probability does not need to be normalized as in case of the conditional random fields (CRFs) discussed below. The model parameters are indicated by Λ. The training set consists of $r = 1, \ldots, R$ labeled sentences, $(X_r, W_r)_{r=1,\ldots,R}$.

According to Bayes rule, the joint probability $p_\Lambda(X, W)$ induces the posterior

$$p_{\Lambda,\gamma}(W|X) = \frac{p_\Lambda(X, W)^\gamma}{\sum_V p_\Lambda(X, V)^\gamma}. \tag{5.1}$$

The likelihoods are scaled with some factor $\gamma \in \mathbb{R}^+$. This is a common trick in speech recognition to scale them to the "real" posteriors. Analogously, the *margin-posterior* can be introduced as

$$p_{\Lambda,\gamma\rho}(W|X) = \frac{[p_\Lambda(X, W)\exp(\rho A(W, W))]^\gamma}{\sum_V [p_\Lambda(X, V)\exp(\rho A(V, W))]^\gamma}. \tag{5.2}$$

Compared with the posterior in Equation (5.1), the margin-posterior includes the margin term $\exp(\rho A(V, W))$. It is based on the string accuracy $A(V, W)$ between the two strings V, W. The accuracy counts the number of matching positions of V, W and will be approximated for efficiency reasons. In general, the accuracy is scaled with some $\rho \in \mathbb{R}^+$. From the perspective of boosting, this term weights up the likelihoods of the competing hypotheses compared with the correct hypothesis [Povey & Kanevsky+ 08]. On the contrary, the discussion in Section 5.4 will show that this term can be interpreted equally as a margin term.

5.2.1 Maximum mutual information (MMI)

The MMI training criterion is defined as

$$\mathcal{F}_\gamma^{(\mathrm{MMI})}(\Lambda) = C \log p(\Lambda) + \sum_{r=1}^{R} \log p_{\Lambda,\gamma}(W_r|X_r). \tag{5.3}$$

This formulation of MMI includes a prior over the model parameters, $\log p(\Lambda)$, also known as regularization. For example, the ℓ_2-regularization (*i.e.,* Gaussian prior with zero mean) is typically used for log-linear models. The regularization constant $C \in \mathbb{R}^+$ is used to balance the regularization term and the loss term including the log-posteriors.

Conventional MMI is based on the true posteriors in Equation (5.1). Using the margin-posterior in Equation (5.2) instead, leads to modified/margin-based MMI (M-MMI)

$$\mathcal{F}_{\gamma\rho}^{(\text{M-MMI})}(\Lambda) = C \log p(\Lambda) + \sum_{r=1}^{R} \log p_{\Lambda,\gamma\rho}(W_r|X_r). \tag{5.4}$$

M-MMI includes a margin term through the margin-posterior. The loss functions for MMI and M-MMI are compared with the hinge loss function in Figure 5.2.[1] The example is given for a binary classification problem with single observations (*i.e.,* no symbol strings). The loss function is plotted against the log-ratio of the posterior of the correct class W_r to the posterior of the competing class $\bar{W}_r$ (*cf.* distance in Equation (5.13))

$$d := \log\left(\frac{p_{\Lambda,1}(X_r, W_r)}{p_{\Lambda,1}(X_r, \bar{W}_r)}\right) \tag{5.5}$$

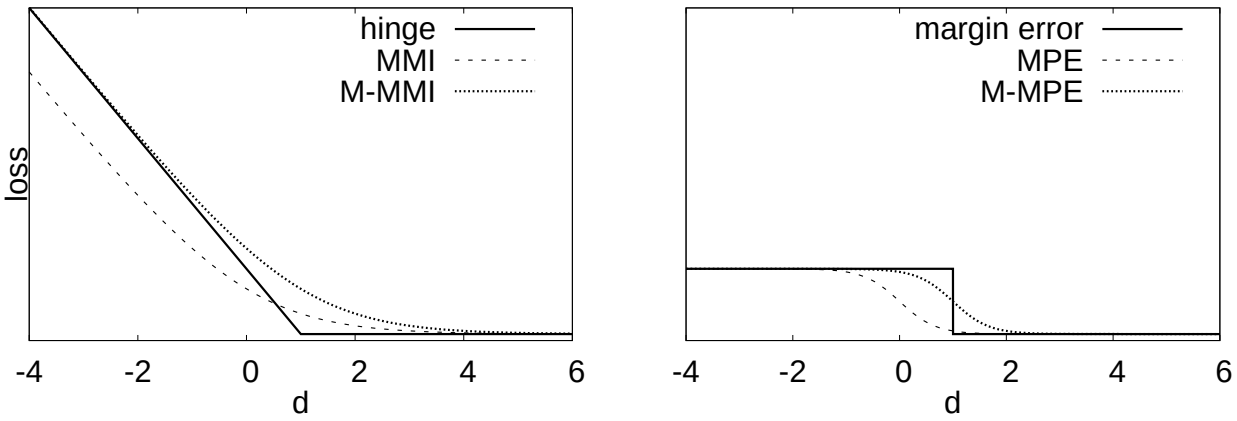


Figure 5.2: Comparison of loss functions for a binary classification problem with d as defined in Equation (5.5). Left: comparison of MMI and M-MMI loss functions with the hinge loss function. Right: comparison of MPE and M-MPE loss functions with the margin error. Note that the margin term shifts the loss function such that the inflection point is at $d = 1$ and not $d = 0$.

for $\gamma = 3, \rho = 1, A(V, W) = \delta(V, W)$. MMI and M-MMI differ by an offset $d = 1$, and M-MMI is a smooth approximation to the hinge loss function.

The logarithm $\log u$ diverges for $u = 0$. Hence, the MMI training criterion is sensitive to outliers, see Section 5.2.4. To avoid the divergence of the logarithm, the identity $\log u = \lim_{\kappa \to 0} \frac{u^\kappa - 1}{\kappa}$ is used to approximate the logarithm. This power approximation leads to the training criterion POW

$$\mathcal{F}_\gamma^{(\mathrm{POW})}(\Lambda) = C \log p(\Lambda) + \sum_{r=1}^{R} \frac{p_{\Lambda,\gamma}(W_r|X_r)^\kappa - 1}{\kappa}. \tag{5.6}$$

In contrast to MMI, POW is bounded below for fixed $\kappa > 0$. For this reason, POW is expected to perform more robustly than MMI. Combining this power approximation and the margin-posterior in Equation (5.2), results in modified/margin-based POW (M-POW)

$$\mathcal{F}_{\gamma\rho}^{(\text{M-POW})}(\Lambda) = C \log p(\Lambda) + \sum_{r=1}^{R} \frac{p_{\Lambda,\gamma\rho}(W_r|X_r)^\kappa - 1}{\kappa}. \tag{5.7}$$

This modification can be made in error-based training criteria in an analogous way.

5.2.2 Minimum phone error (MPE)

Probably, MPE is the training criterion of choice in LVCSR [Povey 04]. It is defined as the (regularized) posterior risk based on the error function $E(V, W)$ like for example the approximate

[1] A similar figure can be found in [Hastie & Tibshirani+ 01, p.380] for the hinge and MMI loss. Interestingly, the squared-error loss is qualitatively similar to the MPE loss, if the loss is plotted against the distance d.

phoneme error [Povey 04]

$$\mathcal{F}_{\gamma}^{(\text{MPE})}(\Lambda) = C \log p(\Lambda) + \sum_{r=1}^{R} \sum_{W} E(W, W_r) p_{\Lambda,\gamma}(W|X_r). \tag{5.8}$$

Again, replacing the scaled posterior $p_{\Lambda,\gamma}(W|X)$ in Equation (5.8) with the margin-posterior in Equation (5.2), leads to the associated modified/margin-based MPE (M-MPE)

$$\mathcal{F}_{\gamma\rho}^{(\text{M-MPE})}(\Lambda) = C \log p(\Lambda) + \sum_{r=1}^{R} \sum_{W} E(W, W_r) p_{\Lambda,\gamma\rho}(W|X_r). \tag{5.9}$$

Keep in mind that due to the relation $E(V, W) = |W| - A(V, W)$ where $|W|$ denotes the number of symbols in the reference string, the error $E(V, W)$ and the accuracy $A(V, W)$ can be equally used in Equations (5.8) and (5.9). The accuracy for MPE and for the margin term do not need to be the same quantity.

The loss functions for MPE and M-MPE are compared in Figure 5.2. The illustration is given for a binary classification problem with single observations for $E(V, W) = 1 - \delta(V, W), A(V, W) = \delta(V, W), \gamma = 1, \rho = 1$ (see also Section 5.2.1). M-MPE is a horizontally shifted version of MPE, and M-MPE approximates the margin error. Note the similarity of MPE, POW, and MCE in this simple situation.

Finally, other instances of the unified training criterion in Section 3.3 (*e.g.* MCE) can be modified in an analogous way to incorporate a margin term.

5.2.3 Unified training criterion

The standard unified training criterion introduced in Section 3.3 is based on the joint probabilities. In case of speech recognition, these are the combined acoustic and language model scores. The margin introduced above can be incorporated into the unified training criterion by multiplying the joint probabilities P with the margin term $M := \exp(-\mathsf{multiply}(A, \rho))$. It is straightforward to extend the transducer-based implementation from Section 3.6 to incorporate this additional margin term. Table 5.2 compares MMI and MPE with their modified variants. The WFST P is defined as in Table 3.5. The WFST Z is defined on the modified WFST P', if necessary. The accumulation of the discriminative statistics is then done according to Table 3.5. Important about our transducer-based implementation is that the standard training criteria and the associated modified training criteria only differ by the additional composition of the probabilistic WFST P with the margin WFST M. Thus, the reader is referred to Chapter 3 for algorithmic and implementation details.

The clear distinction between the model, the training criterion, and the optimization algorithm throughout this work makes the proposed approach to margin-based training rather flexible. For instance, the model could also be represented by a neural network where the margin term is added to the correct output, before the soft-max function.

Table 5.2: Comparison of MMI/MPE with M-MMI/M-MPE in our transducer-based implementation. WFST (P, A) over the expectation semiring has the edge weights $w_{(P,A)}(e) := (w_P(e), w_P(e)w_A(e))$. The accumulation is implemented by a depth first search (DFS).

	MPE	M-MPE	M-MMI	MMI
P'	P	$P \circ M$		P
Z	(P', A)		P'	
Q	posterior$(Z)[v]$		posterior(Z)	
Accumulation	For each edge e and for each time frame t: Accumulate feature x_t with weight $w_Q(e)$ for state s_t.			

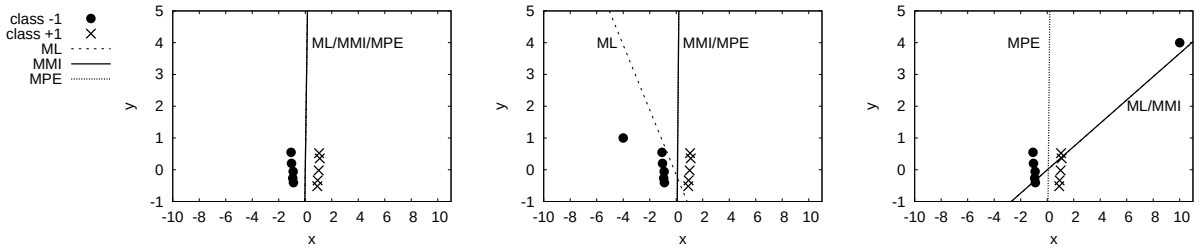


Figure 5.3: Robustness of outliers for different loss functions. Left: clean data, all decision boundaries coincide. Center: clean data plus observation at (-4.0,1.0) such that there is a mismatch between the data and the model, ML decision boundary is affected, MMI/MPE decision boundaries remain unchanged. Right: clean data plus outlier at (10.0,4.0) such that the data is no longer linearly separable, only MPE gives the optimal decision boundary.

5.2.4 Robustness of training criteria

In our opinion, the hinge/MMI loss function has two drawbacks in pattern recognition. First, this loss function differs from the loss function that is used to evaluate the recognition system eventually, typically the recognition error. This implies that margin-based training provides some guarantee regarding the generalization for the hinge/MMI loss function, but *not* for the recognition error. It is not clear how these two quantities are related in general. Second, the hinge/MMI loss function penalizes incorrectly classified symbols (approximately) with their distance from the decision boundary. In contrast, the MPE loss function is bounded as shown in Figure 5.2. This qualitative difference may affect the robustness of the respective estimator. Figure 3.3 illustrates the same issue from another point of view, *i.e.,* plots the accumulation weight over the posterior. Removing observations with low posterior from training as in [Li & Yan+ 07] has a similar effect as an error-based training criterion. In the sense of [Huber 81], robustness means the sensitivity of the estimator to outliers, incorrect model assumptions *etc.* The MMI loss function leads to an estimator that is not (optimally) robust against outliers (*e.g.* erroneous transcriptions, wrong model assumptions) because a single observation can dominate the training criterion. This issue is illustrated on the simple toy example in Figure 5.3. It is assumed that either class is modeled by a single Gaussian. The covariance matrix is shared by the two models. This model assumption leads to a linear decision boundary. In the case of clean data (*i.e.,* matching data), the decision boundaries of the different

training criteria coincide (Figure 5.3, left plot). The Cramer-Rao lower bound guarantees that the lowest variance estimate of the model parameters will be obtained with ML. Thus, if the model is correct, ML is preferred over MMI. If the data and the model do not match, MMI may outperform maximum likelihood (ML) (Figure 5.3, center plot) [Nádas & Nahamoo+ 88]. Moreover, MPE tends to be less sensitive to outliers than MMI (Figure 5.3, right plot). These observations are in agreement with the findings in [Hampel 86].

5.2.5 Optimization of margin-based training criteria

The modified training criteria in this section can be optimized within the transducer-based framework in Chapter 3. The required changes are discussed in Section 5.2.3.

The regularization constant C, the approximation level γ, and the margin scale ρ are chosen beforehand and then kept fixed during the complete optimization. The regularization constant C and the margin scale ρ are not completely independent of each other. Thus, keeping the regularization constant C fixed and tuning the margin scale ρ leads to similar results as keeping the margin scale ρ fixed and tuning the regularization constant C, as long as the scores are in a reasonable numerical range. The latter approach is chosen if the model is optimized from scratch (part-of-speech tagging). In all other cases (speech and handwriting recognition), the margin scale is tuned as well to guide the non-convex optimization in a better way.

In general, the training criteria in speech recognition are non-convex such that the numerical optimization might get stuck in spurious local optima. Convex optimization problems for HMMs have been proposed. These approaches have in common that they are based on the hinge loss and ignore the alignment problem in the sense that the true HMM state sequence is assumed to be known (Chapter 7). For the time being and as it is typical of all state-of-the-art speech recognition systems, the issue of spurious local optima is ignored. Alternatively, the problem with local optima may be alleviated by stochastic annealing techniques where the approximation level acts as the temperature. This would be similar to the iterative optimization strategy suggested by [Zhang & Jin+ 03].

5.3 Tasks

Section 5.2 introduced the training criteria on a rather abstract level. This section discusses the task-specific details of the training criteria, consisting of four major components. First, the probabilistic model (*e.g.* represented by the joint probability $p_\Lambda(X, W)$) parameterizes the decision boundaries by Λ. Second, the regularization log $p(\Lambda)$ restricts the model parameters Λ. In the absence of regularization, the margin can be made arbitrarily large by scaling the model parameters appropriately. In this case, the optimization problem would not be well-defined like for example in [Sha & Saul 07a, Li & Yan+ 07, Yu & Deng+ 08, Saon & Povey 08]. Third, the loss function is used to penalize incorrectly classified observations (see Figure 5.2) and finally, the margin term which is determined by the accuracy $A(V, W)$.

5.3.1 Speech recognition

In speech recognition, the feature $X = x_1^T = x_1, \ldots, x_T$ stands for the sequence of feature vectors $x_t \in \mathbb{R}^D$ and W denotes the word sequence. The joint probability $p_\Lambda(x_1^T, W)$ (not necessarily normalized) is decomposed into the language model $p(W)$ and the acoustic model $p_\Lambda(x_1^T|W)$ by the Bayes rule. To account for different speech rates, the acoustic model is represented by HMMs with state sequences s_1^T

$$p_\Lambda(x_1^T|W) = \sum_{s_1^T} \prod_{t=1}^{T} p_\Lambda(x_t|s_t, W) p(s_t|s_{t-1}, W). \tag{5.10}$$

The probabilities $p_\Lambda(x|s, W)$ and $p(s|s', W)$ are termed the emission and transition model, respectively. The dependence on Λ indicates that only the emission model is optimized while the transition and language models are kept fixed. Conventionally, the emission probabilities are represented by Gaussian mixture models (GHMMs). Alternatively, log-linear (mixture) models (LHMMs) can be used for the emission probabilities (Chapter 4). I-smoothing [Povey 04] is used for the MMI/MPE training of GHMMs while the ℓ_2-regularization is used for the optimization of LHMMs. I-smoothing can be interpreted as a prior in the Gaussian parameter space [Povey 04], and is comparable to the centered ℓ_2-regularization for HCRFs [Li 07]. The centered ℓ_2-regularization includes the simple ℓ_2-regularization as a special case

$$\underbrace{J_0^{-1}\|\lambda\|^2}_{\text{simple } \ell_2} + \underbrace{J_1^{-1}\|\lambda - \lambda_0\|^2}_{\text{centered } \ell_2} = J^{-1}\|\lambda - \lambda_0'\|^2 + \text{const}(\lambda)$$

with $J^{-1} := J_0^{-1} + J_1^{-1}$ and $\lambda_0' := \frac{1}{1+\frac{J_1}{J_0}}\lambda_0$. In speech recognition, word lattices restricting the search space are used to make the summation over all competing hypotheses (sums over W in Section 5.2) efficient. The exact accuracy on phoneme or word level cannot be computed efficiently due to the Levenshtein alignments in general, although feasible under certain conditions as shown in Section 3.8.2. Thus, the approximate phoneme/word accuracy known from MPE/MWE [Povey 04] is used for the margin instead. With this choice of accuracy, the margin term can be represented as an additional layer in the common word lattices such that efficient training is possible, *cf.* Section 5.2.3.

5.3.2 Part-of-speech tagging

Here, part-of-speech tagging refers to the process of extracting the smallest units of meaning out of a given input sentence. Formally speaking, part-of-speech tagging transforms a sequence of words $X = x_1^N = x_1, \ldots, x_N$ into a sequence of concepts $W = c_1^N = c_1, \ldots, c_N$. A concept may contain various pieces of information, *e.g.* the attribute name. An example from the French Media corpus [Devillers & Maynard$^+$ 04] is given in Figure 5.4. The alignment between words x_1^N and concepts c_1^N is assumed to be known for training. Moreover, the considered concept strings are all of the same length such that the simple Hamming accuracy between two concept strings can be used for the margin [Taskar & Guestrin$^+$ 03], see Section 3.7.1.

In this thesis, conditional random fields (CRFs) are used to implement part-of-speech tagging. CRFs are a graphical framework to build discriminative models [Lafferty & McCallum$^+$ 01]. The feature functions $f_i(x_1^N, c_1^N)$, each associated with the

Table 5.3: Overview on modified training criteria used in this work, *i.e.,* for speech recognition of digit strings, LVCSR, part-of-speech tagging, and handwriting recognition.

Task	Margin	Model	Regularization	Loss
Speech (digit strings)	approx. word accuracy	LHMM	ℓ_2	MMI
Speech (LVCSR)	approx. phone accuracy	GHMM	I-smoothing	MPE/MMI
Tagging	Hamming accuracy	CRF	ℓ_2	MMI/POW
Handwriting	approx. word accuracy	GHMM	I-smoothing	MMI

model parameter $\lambda_i \in \mathbb{R}$ fully specify a CRF in the log-linear parameterization

$$p_\Lambda(c_1^N|x_1^N) = \frac{1}{Z_\Lambda(x_1^N)} \exp\left(\sum_i \lambda_i f_i(x_1^N, c_1^N)\right). \tag{5.11}$$

The normalization constant $Z_\Lambda(x_1^N)$ is the sum over all concept strings c_1^N. The model parameters Λ comprise the vector $\lambda = (\lambda_1, \lambda_2, \dots)$. The feature functions are gathered in the vector $f(x_1^N, c_1^N) = (f_1(x_1^N, c_1^N), f_2(x_1^N, c_1^N), \dots)$. For the training criteria in Section 5.2, the pseudo joint probability is defined as $p_\Lambda(x_1^N, c_1^N) := \exp(\sum_i \lambda_i f_i(x_1^N, c_1^N))$. For this choice of model, MMI in Equation (5.3) is a convex optimization problem. This property carries over to M-MMI. For POW/M-POW, however, this is not true.

The CRFs used for the experiments in Section 5.6.2 include the following feature functions: lexical features considering the nearest neighbors, bigram concept features, and word part features (capitalization features, prefix and suffix features).

5.3.3 Handwriting recognition

The recognition of isolated handwritten words shall be considered in the same framework. To reduce the two-dimensional to a one-dimensional problem, the two-dimensional representation of the image is turned into a string representation $X = X_1 \dots X_T$ where X_t is a fixed-length array assigned to each column in the image, see Section 5.6.3 for further details. The word W is represented by a character string. The HMM in Equation (5.10) is used directly (*i.e.,* without a language model) with the states describing the characters of word W and a left-to-right topology [Dreuw & Heigold+ 09]. Similar to speech recognition, the approximate word accuracy is used for the margin.

The different modified training criteria used in the next section are summarized in Table 5.3.

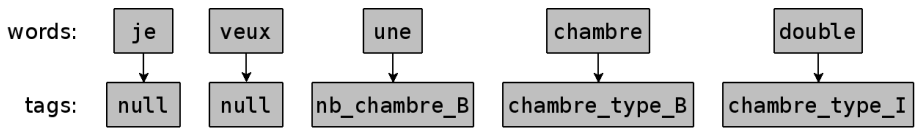


Figure 5.4: Example for part-of-speech tagging from the French Media corpus.

5.4 M-MMI/M-MPE as Smooth Approximations to SVMs

The modified training criteria for log-linear models (*e.g.* CRFs) in Section 5.2 are closely related to SVMs, which shall serve as an example for large margin classifiers. Observing that Gaussian and log-linear models are equivalent (Chapter 4), this relationship is valid for Gaussian models as well.

5.4.1 Support vector machines (SVMs)

We use the definition of SVMs in [Altun & Tsochantaridis+ 03] because it fits our purpose best. The notation is chosen in order to highlight the similarities of SVMs to the training criteria in Section 5.2.

A classification problem with classes W and features X is considered. For training, R labeled training samples $(X_r, W_r)_{r=1,\dots,R}$ are available. Similar to CRFs (Section 5.3.2), assume feature functions $f(X,W) := (f_1(X,W), f_2(X,W), \dots)$ associated with the model parameters $\Lambda = \{\lambda\}$ with $\lambda := (\lambda_1, \lambda_2, \dots)$. Then, according to [Altun & Tsochantaridis+ 03], the optimization problem of SVMs can be formulated as

$$\hat{\lambda} \quad = \quad \operatorname*{argmax}_{\lambda} \left\{ -\frac{C}{2}\|\lambda\|^2 - \sum_{r=1}^{R} l(W_r, d_r; \rho) \right\}. \tag{5.12}$$

For SVMs, the distance vector d_r has the components

$$d_{rW} \quad := \quad \lambda^{\top}(f(X_r, W_r) - f(X_r, W)). \tag{5.13}$$

The empirical constant $C > 0$ is used to balance the ℓ_2-regularization $\log p(\lambda) = -\frac{1}{2}\|\lambda\|^2$, and the loss term. In the context of SVMs, the loss function is typically set to the hinge loss. The multiclass hinge loss function is defined as

$$l^{(hinge)}(W_r, d_r; \rho) \quad := \quad \max_{W \neq W_r} \{\max\{-d_{rW} + \rho(A(W_r, W_r) - A(W, W_r)), 0\}\}. \tag{5.14}$$

In this formulation, ρ is kept fixed but is used for consistency with the formulation of M-MMI/M-MPE. The model parameters λ_i are scaled to adjust the effective margin. This formulation reduces effectively the multiclass problem to a binary classification problem ("correct" *vs.* "recognized" class).

Due to the definition of the loss function and in contrast to [Altun & Tsochantaridis+ 03], this formulation of SVM does not require the introduction of slack variables and side conditions. The generally non-smooth optimization problem will be smoothed for the gradient-based optimization. This definition allows for the efficient calculation of the sum over the competing symbol strings, *e.g.* the exponential number of word sequences in speech recognition.

The hinge loss in Equation (5.14) is the typical loss function used in combination with large margin classifiers and leads to a convex optimization problem. In pattern recognition, however, the margin error is expected to be more appropriate

$$l^{(error)}(W_r, d_r; \rho) \quad := \quad E(\hat{W}_{\rho}(X_r), W_r). \tag{5.15}$$

Here, $\hat{W}_\rho$ stands for the symbol string that yields the minimum margin-distance, $\hat{W}_\rho(d_r) := \text{argmin}_W\{d_{rW}+\rho A(W, W_r)\}$. $E(V, W)$ denotes some error measure for the string pair (V, W). In the simplest case, this loss function counts the number of misclassified sentences, $1-\delta(\hat{W}_\rho(d_r), W_r)$ (*cf.* MCE). For string recognition (*e.g.* speech recognition), a string-based error measure is probably more adequate, *e.g.* the word or phoneme error.

The extension of the 1-0 margin for single symbols to the string accuracy $A(V, W)$ for two symbol strings V, W is reasonable [Taskar & Guestrin$^+$ 03, Sha & Saul 07a] because the margin is proportional to the number of correct symbols in the string. Moreover, it guarantees consistency of Equation (5.12) with the standard SVM for single independent and identically-distributed symbols.

In contrast to SVMs, the optimization problem in Equation (5.12) is non-differentiable and highly non-convex in general. For this reason, smooth approximations to SVMs are discussed next.

5.4.2 Smooth approximations to SVM

In this section we show that M-MMI is a smooth approximation to the SVM with the hinge loss function. Similarly, we show that M-MPE is a smooth approximation to the SVM with the margin phoneme error. Technically speaking, the original loss function l of the SVM is replaced by a smooth loss function l_γ such that $l_\gamma \overset{\gamma\to\infty}{\to} l$ in some sense, without breaking the large margin nature of the original large margin classifier. The parameter $\gamma \in \mathbb{R}^+$ controls the smoothness of the approximation.

From Equation (5.4), the M-MMI loss function is the soft-max approximation to the hinge loss function

$$l_\gamma^{(\text{M-MMI})}(W_r, d_r; \rho) \quad := \quad -\frac{1}{\gamma}\log p_{\Lambda,\gamma\rho}(W_r|X_r). \tag{5.16}$$

See Figure 5.2 for a comparison of the hinge, MMI, and M-MMI loss functions.

Lemma 25 (M-MMI/hinge)**.** $l_\gamma^{(M\text{-}MMI)} \overset{\gamma\to\infty}{\to} l^{(hinge)}$ *(pointwise convergence).*

The proof is similar to the proof in [Zhang & Jin$^+$ 03].

Proof. Define $\Delta A(W, W_r) := A(W_r, W_r - A(W, W_r))$.

$$
\begin{aligned}
l_\gamma^{(\text{M-MMI})}(W_r, d_r; \rho) &\overset{\text{Equation (5.16)}}{=} -\frac{1}{\gamma} \log p_{\Lambda,\gamma\rho}(W_r|X_r) \\
&\overset{\text{Equation (5.2)}}{=} -\frac{1}{\gamma} \log \left(\frac{\exp(\gamma(\lambda^\top f(X_r, W_r) - \rho A(W_r, W_r)))}{\sum_W \exp(\gamma(\lambda^\top f(X_r, W) - \rho A(W, W_r)))} \right) \\
&= -\frac{1}{\gamma} \log \left(\frac{1}{\sum_W \exp(\gamma(\lambda^\top (f(X_r, W) - f(X_r, W_r)) + \rho \Delta A(W, W_r)))} \right) \\
&\overset{\text{Equation (5.13)}}{=} \frac{1}{\gamma} \log \left(1 + \sum_{W \neq W_r} \exp(\gamma(-d_{rW} + \rho \Delta A(W, W_r))) \right) \\
&\overset{\gamma\to\infty}{\to} \begin{cases} \max_{W \neq W_r} \{-d_{rW} + \rho \Delta A(W, W_r)\} & \text{if } \exists W \neq W_r : d_{rW} < \rho \Delta A(W, W_r) \\ 0 & \text{otherwise.} \end{cases} \\
&= \max_{W \neq W_r} \{\max\{-d_{rW} + \rho \Delta A(W, W_r), 0\}\} \\
&\overset{\text{Equation (5.14)}}{=:} l^{(\text{hinge})}(W_r, d_r; \rho).
\end{aligned}
$$

□

M-MPE in Equation (5.9) implies a (weighted) *margin error* $E(V, W)$ (*e.g.* phoneme error) combined with a *weighted margin*

$$
l_\gamma^{(\text{M-MPE})}(W_r, d_r; \rho) := \sum_W E(W, W_r) p_{\Lambda,\gamma\rho}(W|X_r). \tag{5.17}
$$

Again, the distance in Equation (5.13) is only used implicitly in this definition. Keep in mind the subtle difference to the work in [Taskar & Guestrin+ 03] and [Sha & Saul 07a] where a *weighted margin* together with the *hinge/MMI loss* function was used instead. Figure 5.2 depicts the differences between the MPE and M-MPE loss functions, and the margin error.

Lemma 26 (M-MPE/error)**.** $l_\gamma^{(M\text{-}MPE)} \overset{\gamma\to\infty}{\to} l^{(error)}$ *(almost sure convergence).*

Proof. The margin-posteriors in Equation (5.2) converge almost surely (*i.e.,* everywhere except for points on the decision boundary where the loss function is not continuous) to a Kronecker delta. Again, the shortcut $\Delta A(W, W_r) := A(W_r, W_r) - A(W, W_r)$.

$$
\begin{aligned}
p_{\Lambda,\gamma\rho}(W|X_r) &\overset{\text{Equation (5.2)}}{=} \frac{\exp\left(\gamma\left(\lambda^\top f(X_r, W) - \rho A(W, W_r)\right)\right)}{\sum_V \exp\left(\gamma\left(\lambda^\top f(X_r, V) - \rho A(V, W_r)\right)\right)} \\
&= \frac{\exp\left(\gamma\left(\lambda^\top (f(X_r, W) - f(X_r, W_r)) - \rho A(W, W_r)\right)\right)}{\sum_V \exp\left(\gamma\left(\lambda^\top (f(X_r, V) - f(X_r, W_r)) - \rho A(V, W_r)\right)\right)} \\
&\overset{\text{Equation (5.13)}}{=} \frac{\exp\left(\gamma\left(-d_{rW} - \rho A(W, W_r)\right)\right)}{\sum_V \exp\left(\gamma\left(-d_{rV} - \rho A(V, W_r)\right)\right)} \\
&\overset{\gamma\to\infty}{\to} \delta(W, \hat{W}_\rho(X_r))
\end{aligned}
$$

where $\hat{W}_\rho(X_r) := \text{argmax}_W\{p_{\Lambda,\gamma\rho}(W|X_r)\}$ denotes the symbol string that attains the maximum margin-posterior. The last line follows from the limit $\lim_{n\to\infty} \sqrt[n]{\sum_i a_i^n} = \max_i\{a_i\}$ for $a_i \geq 0$ [Walter 99, Band 1, p.78]. Hence, only a single term contributes to the sum in Equation (5.17) such that the loss function $l_\gamma^{(\text{M-MPE})}$ converges to the loss function $l^{(\text{error})}$ in Equation (5.15).

$$
\begin{aligned}
l_\gamma^{(\text{M-MPE})}(W_r, d_r; \rho) &\overset{\text{Equation (5.17)}}{=} \sum_W E(W, W_r) p_{\Lambda,\gamma\rho}(W|X_r) \\
&\overset{\gamma\to\infty}{\to} \sum_W E(W, W_r)\delta(W, \hat{W}_\rho(X_r)) \\
&= E(\hat{W}(X_r), W_r) \\
&\overset{\text{Equation (5.15)}}{=:} l^{(\text{error})}(W_r, d_r; \rho).
\end{aligned}
$$

□

Finally, it should be pointed out that the same ideas also apply to other loss functions, *e.g.* the smoothed sentence error used for MCE.

5.5 Related Approaches

A few related approaches are briefly discussed to make the proposed margin-based framework more clear.

5.5.1 M-MPE *vs.* MPE

Observe that formally, M-MPE is similar to conventional MPE. Indeed, M-MPE gives some new insight into several heuristics typically used for conventional discriminative training.

- Scaling of posteriors [Wessel & Macherey+ 98, Woodland & Povey 00]. The smoothing parameter γ corresponds with the scaling factor for the posteriors.

- Weak language model [Schlüter 00]. The margin term weakens the prior (*i.e.,* language model). Hence, the weak language model can be considered an approximation of the margin term. We believe that the frame-based approach proposed to improve the confusability in training [Povey & Woodland 99] is another attempt to approximate the margin concept by replacing the true FB probabilities by the global relative frequencies [Heigold & Schlüter+ 07].

- I-smoothing [Povey & Woodland 02]. I-smoothing is a special type of MAP estimation. The parameter prior is centered at a reasonable initial acoustic model (*e.g.* ML model). In other words, I-smoothing can be considered a refined regularization term like the centered ℓ_2-regularization $\|\Lambda - \Lambda_0\|^2$ for log-linear models [Li 07].

5.5.2 M-MMI *vs.* boosted MMI (BMMI)

Boosting techniques were incorporated into conventional MMI, leading to boosted MMI (BMMI) [Povey & Kanevsky+ 08]. For GHMMs, it can be shown that BMMI and M-MMI are equivalent training criteria. In practice, BMMI differs from M-MMI by the choice of the acoustic model for I-smoothing and the optimization algorithm (highly tuned EBW *vs.* Rprop). Similarly, [McDermott & Nakamura 08] introduced boosted MPE (BMPE) which can be shown to be equivalent to our M-MPE.

5.5.3 M-MPE *vs.* integrated MPE (iMPE)

Margin-based training like for example M-MPE typically uses a single margin value. iMPE extends this idea by considering an interval of margin values [McDermott & Watanabe+ 09]. This generalization permits to show a clear relationship between MPE and MMI-based training criteria [McDermott & Watanabe+ 09].

5.5.4 Modified error-based *vs.* minimum Bayes risk (MBR) training

MBR training (*e.g.* MPE) has become popular in ASR for its effectiveness. This type of training criteria is motivated by the MBR decoding principle and minimizes the *expected* risk per segment r, $\sum_W p(W|X_r)E(W,W_r)$ by adjusting the model parameters [Kaiser & Horvat+ 02, Doumpiotis & Byrne 05, Gibson & Hain 06]. In contrast to MBR training, M-MPE (without margin term for simplicity, but using the interpretation of Lemma 26) approximates the loss function as

$$\sum_W \frac{p(X_r,W)^\gamma}{\sum\limits_{W'} p(X_r,W')^\gamma} E(W,W_r) \quad \stackrel{\gamma\to\infty}{\to} \quad E(\hat{W}(X_r),W_r) \tag{5.18}$$

In summary, the latter approach provides an (approximately) consistent estimator for the empirical risk while for MBR training, the estimator is not consistent. This is a subtle but important difference between the MBR training methodology and the methodology of modified training criteria. In practice, however, the two approaches are identical up to the margin term.

Assuming model-free discriminant functions $p(W|X)$, it can be shown that the global optimum of the MBR training coincides with the global optimum of the exact overall risk. This result extends to models which include the optimum decision boundary and allow $p(W|X) \in [0,1]$ (*i.e.,* no regularization).

5.5.5 Risk-based training *vs.* MBR decoding

MPE is an example of a risk-based training criterion. This type of training criteria optimizes directly the decision boundaries regarding some (smoothed) recognition error. Strictly speaking, $p(W|X)$ does not represent true probabilities but rather parameterize the set of discriminant functions. In general, the quantity $p(W|X)$ does not converge to the true probability. The

decision rule $\hat{W} := \text{argmax}_W\{p(W|X)\}$ is expected to be optimal in this framework.[2] Opposed to this approach, MBR decoding assumes that the (true) posteriors $p(W|X)$ are known. Under this assumption and for a predefined risk matrix $E(W, W')$, the decision rule $\hat{W} := \text{argmax}_W\{\sum_{W'} R(W, W')p(W'|X)\}$ is optimal, *i.e.,* the expected risk is minimal. In practice, this approach requires the estimation of probability densities in high dimensional feature spaces using probabilistic training criteria, *e.g.* ML or MMI. This approach might be suboptimal because of the indirect optimization of the expected risk.

5.6 Experimental Results

The modified training criteria introduced in Section 5.2 allow the direct evaluation of the utility of the margin term for string recognition (our objectives in Section 5.1.2). Experimental results are provided for three different speech tasks (digit strings, European Parliament plenary speech, broadcasts), two part-of-speech tagging tasks (French Media, Polish), and a handwriting task (IFN/ENIT). The training criterion (MMI or MPE) is determined on the training conditions. If the system makes no or only a few training errors, the margin term dominates and the loss term has no or only little impact. In this case, MMI is chosen for convenience. Otherwise, MPE is used. This rationale is consistent with the standard choice of the conventional training criteria. The statistical significance of the differences in the error rates are calculated with the bootstrap approach described in [Bisani & Ney 04]. The variety of tasks considered here allow the systematic evaluation of the margin term under completely different conditions. This will help to improve the understanding of the utility of the margin term in practice.

5.6.1 Speech recognition

The digit string recognition task uses LHMMs of different complexity while the LVCSR systems are based on GHMMs with globally pooled variances. This allows us to produce rather good ML baseline models consisting of a fairly high number of densities, *cf.* Table 5.4. The ML baselines are used to initialize the discriminative training, both for GHMMs and LHMMs. The language model scale (if necessary), the best training iteration, and the optimal margin parameter ρ are all tuned on the training or development data. All test data are reserved for the final evaluation of the acoustic models. The optimization is done with Rprop except for the European Parliament plenary speech task which is optimized with EBW. Unless otherwise stated, the scaling factor is set to the inverse of the language model scale and the margin scale is set such that $\gamma\rho = 0.5$.

German digit strings. M-MMI is first applied to the SieTill task consisting of spoken digit strings (Appendix A.1.1). The recognition system is based on gender-dependent whole-word HMMs. For each gender, 214 distinct HMM states plus one for silence are used. The vocabulary consists of the eleven German digits (including the pronunciation variant 'zwo'). The observation vectors consist of 12 MFCC features without temporal derivatives. The

[2] Keep in mind that the probabilistic constraints on the discriminant functions do not restrict the set of decision boundaries $d(W, X)$ because $p(W|X) := \frac{\sigma(d(W,X))}{\sum_{W'} \sigma(d(W,X))}$ for some sigmoid function $\sigma : \mathbb{R} \mapsto \mathbb{R}^+$.

Table 5.4: Corpus statistics and acoustic setups for speech recognition tasks.

Speech task	Train [h] Test [h]	#States/#Densities Features
SieTill	11 11 (Test)	430/27k 25 LDA(MFCC)
EPPS En	92 2.9 (Eval07)	4,500/830k 45 LDA(MFCC+voicing)+VTLN+SAT/CMLLR
BNBC Cn 230h	230 2.2 (Eval06)	4,500/1,100k 45 LDA(MFCC)+3 tones+VTLN+SAT/CMLLR
BNBC Cn 1500h	1,500 2.2 (Eval06)	4,500/1,200k 45 SAT/CMLLR(PLP+voicing+3 tones+32 NN)+VTLN

gender-independent linear discriminant analysis (LDA) is applied to five consecutive frames and projects the resulting feature vector to 25 dimensions. The gender-dependent acoustic models are trained jointly as described in Chapter 7. The corpus statistics are summarized in Table 5.4. This simple task suffers severely from overfitting. The training error tends to zero after only a few training iterations. This observation implies that the loss term vanishes (*i.e.,* the choice of the loss function is irrelevant) and thus, the margin term will dominate. For this reason, only results for MMI and M-MMI are shown for this simple task with $\gamma\rho = 1, \gamma = 25^{-1}$ and ℓ_2-regularization. Figure 5.5 compares the progress of the error rate with the training iteration for different variants of MMI for an LHMM with 16 densities per mixture. In this case, the margin term is better able than the regularization term to prevent the training from overfitting.

In Table 5.5, LHMMs of different complexity are investigated, including a log-linear model with a single density per mixture but with augmented features (n-th order features up to $n = 3$). This approach is similar to an SVM with a polynomial kernel. For this simple task, these higher-order features and the use of mixtures are equally good at modeling the non-linearities in the decision boundaries.

European Parliament plenary sessions (EPPS). This task contains recordings from the European Parliament plenary sessions (EPPS). The acoustic front end comprises MFCC features augmented by a voicing feature. Nine consecutive frames are concatenated and the resulting vector is projected to 45 dimensions by means of LDA. The MFCC features are warped using a fast variant of vocal tract length normalization (VTLN). On top of this, speaker adaptive training (SAT) and constrained MLLR (CMLLR) are applied. The triphones are clustered using CART, resulting in 4,501 generalized triphone states. The HMM states are modeled by Gaussian mixtures with globally pooled variances. The ML baseline system is made up of over 800k densities. For recognition, a lexicon with 50k entries in combination with a 4-gram language model is used (Appendix A.1.3). A summary of the information is provided in Table 5.4. First, some basic issues such as the choice of the margin and the correlation of the margin term with the weak unigram language model are investigated, see Table 5.6. The experiments with language models of different order appear to support our hypotheses that the margin term compensates for the weak language model. The results imply that the approximate phoneme accuracy is a reasonable choice for the margin. For this reason, the remaining experiments in Table 5.7 use the approximate phoneme accuracy. The comparison of MPE and M-MPE is shown in Table 5.7. The margin term only leads to small improvements for this task.

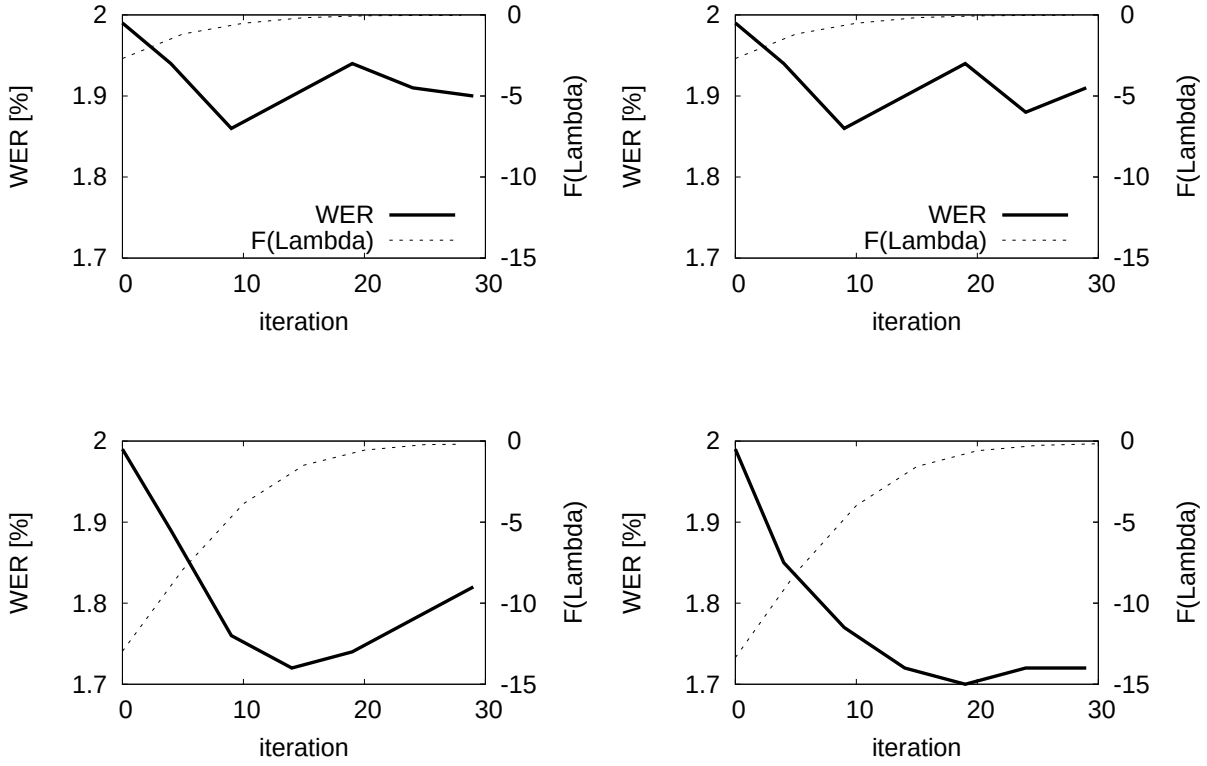


Figure 5.5: Effect of regularization and margin: progress of objective function $\mathcal{F}(\Lambda)$ on the SieTill training corpus, and word error rate (WER) on the SieTill test corpus. Upper left: MMI without regularization. Upper right: MMI with regularization. Lower left: M-MMI without regularization. Lower right: M-MMI.

Table 5.5: Word error rate (WER) for SieTill test corpus. The first two systems are LHMMs with the given number of densities per mixture ('Dns/Mix'), the last system is a single density log-linear model with all zeroth-, first-, second-, and third-order features, *i.e.,* 'feature order'=third.

Dns/Mix	Feature order	Criterion	WER [%]
1	first	ML	3.8
		MMI	2.9
		M-MMI	2.7
16		ML	2.0
		MMI	1.9
		M-MMI	1.7
64		ML	1.8
		MMI	1.8
		M-MMI	1.6
1	third	Frame	1.8
		MMI	1.7
		M-MMI	1.5

Table 5.6: Word error rates (WER) for EPPS English corpus, M-MPE with different margins and different language models for training.

LM (in training)	Margin	WER [%]		
		Dev06	Eval06	Eval07
1g	none	13.4	10.1	11.5
	word	13.4	10.2	11.3
	phone	13.3	10.2	11.3
2g	none	13.3	10.3	11.6
	word	13.2	10.2	11.3
	phone	13.2	10.2	11.3

Table 5.7: Word error rate (WER) for EPPS English (Eval07) and BNBC Mandarin (Eval06).

	WER [%]		
	EPPS En	BNBC Cn	
Criterion	90h	230h	1500h
ML	12.0	21.9	17.9
MMI		20.8	
M-MMI		20.6	
MPE	11.5	20.6	16.5
M-MPE	11.3	20.3	16.3

Mandarin broadcasts. The second LVCSR task consists of Mandarin broadcast news and conversations. The experiments are based on the same setup as described in Appendix 4.8.3. The corpus statistics of the two systems under consideration are shown in Table 5.4. The BNBC Cn 230h system uses MFCC features. Nine consecutive frames are concatenated and projected to 45 dimensions by means of LDA. A tonal feature with first and second derivatives is added. The MFCC features are warped using a fast variant of VTLN. On top of this, SAT/CMLLR is applied. The BNBC Cn 1500h system uses PLP features augmented with a voicing feature. Nine consecutive frames are concatenated. Tonal features and neural network (NN) based posterior features are added and projected to 45 dimensions by means of SAT/CMLLR. The PLP features are warped using a fast variant of VTLN. The lexicon has 60k entries. A 4-gram language model is used for recognition (Appendix 4.8.3). The results for the two different setups are shown in Table 5.7. Similar to EPPS En, small but consistent improvements are observed if adding the margin term.

The experiments in Table 5.7 suggest that MPE and M-MMI perform equally. The margin term, however, does not compensate the original difference between MMI and MPE in this experiment. M-MPE uses about twice as many training iterations as M-MMI until convergence.

The improvements for the digit string recognition task in Table 5.5 are significant at the level $\alpha = 0.1\%$. M-MPE performs significantly better than MPE for EPPS En and BNBC Cn 230h ($\alpha = 1\%$) while the difference for BNBC Cn 1500h is not significant ($\alpha = 10\%$), see Table 5.7. These results are in agreement with our expectations from Table 5.1.

5.6.2 Part-of-speech tagging

The task of part-of-speech tagging is described in Section 5.3.2. The well-known concept error rate (CER) [Hahn & Lehnen$^+$ 09] is used as the evaluation criterion of the CRFs. Experimental results for two different languages are given to compare the performance of MMI and POW with their respective modified variants, M-MMI and M-POW (Section 5.2). All CRFs are optimized from scratch. The margin scale ρ and the approximation level γ are both set to unity; only the regularization constant C and the parameter κ are tuned. The feature functions are selected depending on the language. All tuning is done on the Dev corpora. A detailed description of the setups can be found in Appendix A.2.[3]

[3]Thanks to Stefan Hahn and Patrick Lehnen for providing the baseline systems and assisting me with the experiments.

Table 5.8: Corpus statistics for part-of-speech tagging corpora. The vocabulary counts refer to the number of concepts or words observed in the corpus and covered by the vocabulary.

Corpus		Data #Sentences	Vocabulary #Words	Vocabulary #Concepts
French	Train	12,908	2,210	99
	Dev	1,259	838	66
	Eva	3,005	1,276	78
Polish	Train	8,341	4,081	195
	Dev	2,053	2,028	157
	Eva	2,081	2,057	159

Table 5.9: Concept error rate (CER) for part-of-speech tagging, French Media (Eva) and Polish (Eva).

Criterion	CER [%] French	CER [%] Polish
MMI	11.5	22.6
M-MMI	10.6	21.5
POW	11.3	22.5
M-POW	10.7	21.2

French Media. The French Media corpus covers the domain of the reservation of hotel rooms and tourist information and the incorporated concepts have been designed to match this task. The reader is referred to Tables A.2 and 5.8 for the corpus statistics. The results are summarized in Table 5.9. The optimal regularization constants C are 2^{-3} and 2^{-2} for MMI/POW and M-MMI/M-POW, respectively. The optimal parameter κ of the power approximation to the logarithm in Equation (5.6) is 0.01, both for POW and M-POW.

Polish. The data for the Polish corpus have been collected at the Warsaw Transportation call-center [Marasek & Gubrynowicz 08]. Tables A.2 and 5.8 suggest that the Polish task is more difficult than the French task because there are less training data and more concepts. The results for the Polish corpus are shown in Table 5.9. The optimal regularization constants C are 2^{-6} and 2^{-2} for the original and the modified training criteria, respectively. The optimal parameter κ of the power approximation to the logarithm is 0.1 for POW and 0.0001 for M-POW. The non-convexity of the training criteria based on the power approximation (see Corollary 43) does not seem to be an issue here. Like for the French Media corpus and similar to MPE/M-MPE mentioned above, POW/M-POW tend to converge more slowly than MMI/M-MMI.

The margin term helps significantly both for MMI and POW in Table 5.9 ($\alpha = 0.1\%$). The differences between MMI/M-MMI and POW/M-POW are not significant in general.

Table 5.10: Corpus statistics for handwriting (sub-)corpora, a, b, c, d, and e are the different folds.

Corpus		#Observations [k] Towns	Frames
IFN/ENIT	a	6.5	452
	b	6.7	459
	c	6.5	452
	d	6.7	451
	e	6.0	404

5.6.3 Handwriting recognition

Finally, the margin concept is applied to a handwriting recognition task (Section 5.3.3). The experiments are conducted on the IFN/ENIT database, see Appendix A.3.2.[4]

IFN/ENIT. This database contains Arabic handwriting. The database is divided into four training folds with an additional fold for testing [Märgner & Abed 07]. The current database version contains a total of 32k Arabic words handwritten by about 1,000 writers. A character-based lexicon is used to represent the town names. It comprises 937 Tunisian town names. Here, we follow the same evaluation protocol as for the ICDAR 2005 and 2007 competition [Dreuw & Heigold$^+$ 09]. The corpus statistics for the different folds can be found in Table 5.10. Without any preprocessing of the input images, simple intensity-based image features X_t are extracted by moving a sliding window over the image. These features are augmented by their spatial derivatives in horizontal direction $\Delta = X_t - X_{t-1}$. In order to incorporate temporal and spatial contexts into the features, seven consecutive features are concatenated in a sliding window, which are then projected to a 30-dimensional feature vector X_t by means of a PCA transformation. The character-based model includes 36k Gaussian densities with globally pooled variances. Model length estimation is included to account for character dependent model lengths [Dreuw & Heigold$^+$ 09]. Similar to the digit string recognition task above, the training word error rate is very low such that the generalization is an issue and the choice of the loss function is not important. This is why the experiments were only done for MMI. As expected, the discriminative training ($\alpha = 0.01\%$) benefits significantly from the margin term, see Table 5.11. The settings from the digit string recognition task in Section 5.6.1 were used for the discriminative training.

5.7 Conclusion

An approach was discussed how to modify existing training criteria for speech recognition like for example MMI and MPE to include a margin term. These modified training criteria (*e.g.* M-MPE and M-MMI) were shown to be closely related to existing large margin classifiers (*e.g.*

[4]Thanks to Philippe Dreuw for providing the baseline system and assisting me with the experiments.

Table 5.11: Word error rate (WER) for handwriting recognition corpora (IFN/ENIT). The corpus identifier 'Train-Test' (*e.g.* 'abcd-e') indicates the folds used for training and testing, respectively.

	WER [%]				
Criterion	abc-d	abd-c	acd-b	bcd-a	abcd-e
ML	7.8	8.7	7.8	8.7	16.8
MMI	7.4	8.2	7.6	8.4	16.4
M-MMI	6.1	6.8	6.1	7.0	15.4

SVMs) with the respective loss function. This approach allows for the direct evaluation of the utility of the margin term for string recognition. As expected, the benefit from the additional margin term depends clearly on the training conditions. For simple tasks like for example the recognition of spoken digit strings, overfitting is an issue and thus, the use of the margin term leads to nice reductions in the error rates. For more complex tasks like for example LVCSR, the additional margin term is clearly less important, although consistent improvements were observed. Less than 25% of the total discriminative improvement is typically due to the margin, compared with the best state-of-the-art systems. Reasons for this outcome might be that, due to the huge amount of training data, the loss term dominates in LVCSR, and that the margin concept is already well approximated by several heuristics like for example the use of a weak language model in conventional discriminative training.

Chapter 6

Optimization with Growth Transformations

Numerical optimization is an important component in parameter estimation. Efficient optimization algorithms like for example (empirical variants of) extended Baum Welch (EBW) and Rprop have been successfully used in practice to optimize the different training criteria in speech recognition. Most of these algorithms converge to a critical point of the training criterion, *i.e.,* points with a vanishing gradient. Growth transformations are a class of optimization algorithms which in addition guarantee to increase the training criterion in each iteration and which are parameter-free (*e.g.* no learning rates need to be tuned). The art of constructing growth transformations consists of reducing the original optimization problem to a simpler problem with the required properties. Well-known examples are expectation-maximization (EM) for the generative training of GHMMs, EBW for the discriminative training of GHMMs, or generalized iterative scaling (GIS) for the MMI training of conditional random fields (CRFs). This chapter introduces two novel growth transformations for the conventional training criteria (*e.g.* MMI, MCE, MPE). The one leads to EBW-like update rules for GHMMs with *constructive* finite iteration constants. The other generalizes standard GIS to HCRFs and other conventional training criteria [Heigold & Deselaers+ 08a]. The GIS-like algorithm for MMI from incomplete data proposed in [Riezler 98, Riezler & Kuhn+ 00] for natural language processing is very similar to our extension of GIS. We became aware of this work only after presenting our algorithm. Compared with [Riezler 98, Riezler & Kuhn+ 00], this work introduces a more general result including MPE, for instance and tests the algorithm on significantly larger datasets in combination with continuous-valued features.

6.1 Overview

Several growth transformations have been proposed in the literature. Here, we focus on growth transformations for (discriminative) Gaussian and log-linear models.

Most algorithms used for the optimization of GHMMs in speech recognition are based on extended Baum Welch (EBW) [Normandin & Morgera 91, Gopalakrishnan & Kanevsky+ 91, Gunawardana 01, Kanevsky 04]. The so-called iteration constants control the conver-

gence of EBW. The existence of finite iteration constants have been proved [Kanevsky 04, Axelrod & Goel+ 07] but in practice, the iteration constants are determined upon a few heuristics [Povey 04, Axelrod & Goel+ 07, Macherey 10]. Reverse Jensen inequality introduced in [Jebara 02] leads to update rules similar to the EBW update rules. This approach was tested in speech recognition assuming many approximations [Afify 05].

Log-linear models are traditionally optimized using generalized iterative scaling (GIS). Among others, this optimization algorithm cannot deal with hidden variables. A few approaches have been proposed to solve this problem. The problem can be solved by decomposing the problem into simpler subproblems. The overall optimization is then performed by alternating optimization of the subproblems. Typical examples of this methodology are generalized EM (GEM) [Wang & Schuurmans+ 02] and the extension of GIS proposed in [Saul & Lee 02]. The growth transformation derived in Section 6.4 avoids such indirections and optimizes directly the objective function using a single auxiliary function.

An extension of GIS similar to ours was proposed in natural language processing [Riezler 98, Riezler & Kuhn+ 00, Wang & Schuurmans+ 02], *i.e.,* a variant of GIS for optimizing HCRFs using MMI. We became aware of this work only after presenting our algorithm in [Heigold & Deselaers+ 08a]. Compared with [Riezler 98, Riezler & Kuhn+ 00, Wang & Schuurmans+ 02], this work introduces a more general result including MPE, for instance, uses continuous- and not discrete-valued features, and tests the algorithm on comparably large data sets.

The above mentioned growth transformations for the Gaussian models may be applied to log-linear models. Applying these functions to log-linear models, however, results in purely linear equations which might be problematic. The use of some regularization, for example, avoids this problem, but then other problems occur. As an example, the reverse Jensen inequality requires non-vanishing second derivatives of the argument of the exponential. This assumption is not fulfilled by log-linear models. Finally, a subset of the log-linear models is equivalent to the Gaussian models (Chapter 4). Thus, after transforming the log-linear model into an equivalent Gaussian model, the growth transformations for Gaussian models can be employed in the usual way. This approach has the disadvantage that in general, the model parameters and thus the iteration constants (see end of Section 6.3 for a concrete example) are ambiguous. Hence, the efficiency of these algorithms relies heavily on the initial choice of the parameters. In addition, to calculate efficiently the iteration constants for complex problems in speech recognition, several approximations have been made, *e.g.* [Afify 05].

6.2 Growth Transformations

Based on previous work [Gopalakrishnan & Kanevsky+ 91, Kanevsky 04], growth transformations are defined and discussed. Slightly generalized versions of EM and GIS are revisited in the context of auxiliary functions [Della Pietra & Della Pietra+ 97, Povey 04, Bishop 06] to illustrate the concept and to prepare for the derivation of G-GIS in Sections 6.3 and 6.4.

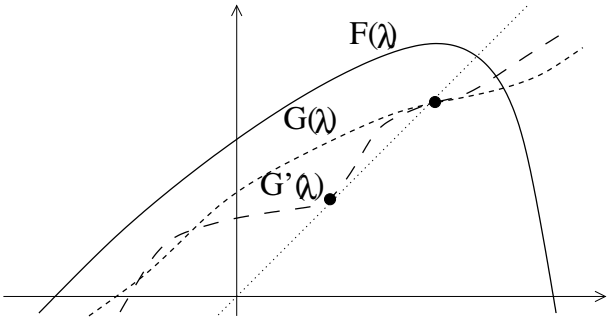


Figure 6.1: Illustration of growth transformation. Potential fixed points lie on the dotted line, the black points indicate the fixed points of the parameter transformations $\mathcal{G}$ and $\mathcal{G}'$. $\mathcal{G}$ and $\mathcal{G}'$ both increase the training criterion $\mathcal{F}$ in each step but unlike $\mathcal{G}, \mathcal{G}'$ is not guaranteed to converge to a critical point of $\mathcal{F}$.

6.2.1 Definition & properties

Assume a training criterion, also known as objective function, $\mathcal{F} : \Gamma \to \mathbb{R},\ \Lambda \mapsto \mathcal{F}(\Lambda)$ to be maximized. A growth transformation maps the current parameters $\Lambda' \in \Gamma$ to new parameters $\Lambda \in \Gamma$ such that the objective function $\mathcal{F}$ increases.

Definition 27 (Growth transformation)**.** *A* growth transformation *of F is defined to be a continuous function $\mathcal{G} : \Gamma \to \Gamma,\ \Lambda \mapsto \mathcal{G}(\Lambda)$ such that $\mathcal{F}(\mathcal{G}(\Lambda)) > \mathcal{F}(\Lambda)$ for all $\Lambda \neq \mathcal{G}(\Lambda)$. Moreover, a fixed point of $\mathcal{G}$, $\mathcal{G}(\Lambda) = \Lambda$, implies a critical point of $\mathcal{F}$, $\nabla\mathcal{F}(\Lambda) = 0$.*

The growth transformation induces the sequence $\{\Lambda^{(k)} = \mathcal{G}(\Lambda^{(k-1)})\}_{k=1}^{\infty}$. It is initialized with some $\Lambda^{(0)} \in \Gamma$. If the objective function $\mathcal{F}$ is bounded above, this sequence converges [Walter 99, Band 1, p.65]. The limit $\Lambda^{(\infty)}$ is a fixed point of the growth transformation and thus, a critical point of $\mathcal{F}$ by definition. Note that without the extra condition on the fixed points, which is in contrast to [Gopalakrishnan & Kanevsky$^+$ 91, Kanevsky 04], the sequence is not guaranteed to converge to a critical point of $\mathcal{F}$, see Figure 6.1.

The following lemma taken from [Gunawardana 01] shall serve as a simple example of a growth transformation.

Lemma 28 (Rational)**.** *Assume an objective function $\mathcal{F}(\Lambda) := \frac{P(\Lambda)}{Q(\Lambda)}$ based on the two positive functions $P(\Lambda)$ and $Q(\Lambda)$. Then, $\mathcal{G}(\Lambda) := \operatorname{argmax}_{\Lambda'}\{P(\Lambda') - \mathcal{F}(\Lambda)Q(\Lambda') + \Delta\}$ defines a growth transformation of $\mathcal{F}$ for any* iteration constant $\Delta \in \mathbb{R}$.

Two objective functions that differ in a strictly monotone function (*e.g.* log) have the same growth transformations. Combining this observation with Lemma 28 applied to the two positive functions $P(\Lambda) := p_\Lambda(x, c)$ and $Q(\Lambda) := p_\Lambda(x)$, provides a growth transformation for the MMI training criterion, $\mathcal{F}(\Lambda) := \log p_\Lambda(c_1^N|x_1^N)$. Here, the evidence $p_\Lambda(x_1^N)$ is obtained by marginalization of the joint probability, $p_\Lambda(x_1^N) = \sum_{c_1^N} p_\Lambda(x_1^N, c_1^N)$. The class posterior is then determined by the Bayes rule. Hence, this objective function is in the rational form. More general objective functions will be discussed below.

6.2.2 Armijo's approach

Probably the most general and simplest growth transformation traces back to Armijo [Armijo 66]. He showed that for training criteria with Lipschitz continuous first derivatives, global and non-vanishing step sizes exist. This is a simple example to illustrate the notion of growth transformations. Moreover, the study of this approach gives some idea under which conditions growth transformations exist, what the effect of the parameterization is, and why more sophisticated growth transformations are needed in practice.

Lemma 29 (Armijo)**.** *Assume a Lipschitz continuously differentiable objective function* $\mathcal{F} : \Gamma \to \mathbb{R},\ \Lambda \mapsto \mathcal{F}(\Lambda)$. *The Lipschitz continuity of the first derivative implies* $\|\nabla\mathcal{F}(\Lambda) - \nabla\mathcal{F}(\Lambda')\| \leq L\|\Lambda - \Lambda'\|$ *for all* $\Lambda, \Lambda' \in \Gamma$ *where* $L > 0$ *is the Lipschitz constant. Then,* $\mathcal{G}(\Lambda) := \Lambda + \frac{1}{L}\nabla\mathcal{F}(\Lambda)$ *is a growth transformation of* $\mathcal{F}$.

Proof. The set of critical points of $\mathcal{F}$ is identical to the set of zeroes of $\nabla\mathcal{F}$. Hence, $\mathcal{G}$ is a growth transformation of $\mathcal{F}$ if the smallest $p \geq 0$ such that $\nabla\mathcal{F}(\Lambda + \frac{p}{L}\nabla\mathcal{F}(\Lambda)) = 0$ is greater than 1, *i.e.,* $\|\mathcal{G}(\Lambda) - \Lambda\| \leq \|\Lambda_0 - \Lambda\|$ with $\Lambda_0 := \Lambda + \frac{p}{L}\nabla\mathcal{F}(\Lambda)$. This inequality is a direct consequence of the Lipschitz continuity of $\nabla\mathcal{F}$

$$\|\mathcal{G}(\Lambda) - \Lambda\| \overset{\text{definition of } \mathcal{G}}{=} \frac{\|\nabla\mathcal{F}(\Lambda)\|}{L} \overset{\nabla\mathcal{F}(\Lambda_0)=0}{=} \frac{\|\nabla\mathcal{F}(\Lambda_0) - \nabla\mathcal{F}(\Lambda)\|}{L} \overset{\text{Lipschitz continuity}}{\leq} \|\Lambda_0 - \Lambda\|.$$

□

In general, the Lipschitz constant L depends on the training criterion and the class of functions under consideration. The Lipschitz constant may also depend on the training data. These dependencies do not affect the feasibility of the approach as long as a reasonable estimate of the Lipschitz constant can be determined in a preprocessing step (see examples below).

Many training criteria and classes of functions satisfy the Lipschitz condition in Lemma 29. Examples of practical relevance include log-linear models and Gaussian models with floored variances and if restricted to some compact set of model parameters. HMMs also induce Lipschitz continuous training criteria as long as not the limit of infinite training data is considered.

To get an idea of the efficiency of the growth transformation, an explicit Lipschitz constant is needed. Here, we derive an explicit upper bound of the Lipschitz constant for the class of log-linear models and the MMI training criterion for log-linear models

$$\mathcal{F}(\Lambda) := \sum_{n=1}^{N} \log p_\Lambda(c_n|x_n) = \sum_{n=1}^{N} \log\left(\frac{\exp(\sum_i \lambda_i f_i(x_n, c_n))}{\sum_c \exp(\sum_i \lambda_i f_i(x_n, c))}\right). \tag{6.1}$$

Keep in mind that this has not been possible for EBW (a different type of growth transformation), for which only the existence of sufficiently large iteration constants have been proved [Kanevsky 04, Axelrod & Goel+ 07]. For continuously differentiable functions, the Lipschitz constant coincides with the maximum absolute slope, *i.e.,* $L = \max_\Lambda\{\|\nabla^2\mathcal{F}(\Lambda)\|\}$ where $\nabla^2\mathcal{F}(\Lambda)$ denotes the Jacobian matrix of $\nabla\mathcal{F}$.[1] The spectral norm of the Jacobian matrix

[1] See `http://en.wikipedia.org/wiki/Lipschitz_continuity`.

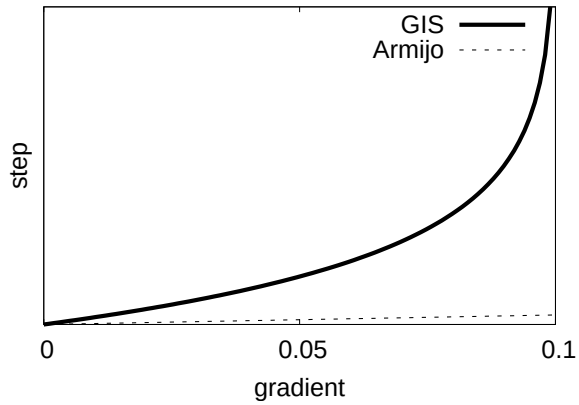


Figure 6.2: Parameter update over gradient for Armijo's approach and GIS for a typical real task, see text for more details.

for the log-linear model in Equation 6.1 can be bounded above by

$$\|\nabla^2 \mathcal{F}(\Lambda)\| \leq \sum_{n=1}^{N} \sum_{c} p_\Lambda(c|x_n)(1 - p_\Lambda(c|x_n))\|f(x_n, c)\|^2.$$

Assuming bounded features $\|f(x,c)\|^2 \leq R^2$ for all x, c, the expression reduces to $\|\nabla^2 \mathcal{F}(\Lambda)\| \leq NR^2$. This bound also holds for log-linear models with hidden variables. Hence, $L = NR^2$ induces a growth transformation according to Lemma 29. Both the Lipschitz constant L and the gradient $\nabla\mathcal{F}$ scale linearly with the number of observations N such that the step size does not explicitly depend on N. The resulting step sizes are compared with the step sizes generated by GIS, see Figure 6.2. Armijo's step sizes turn out to be overly pessimistic compared with GIS, see Section 6.2.4 for further details. This result clearly motivates the investigation of more refined growth transformations for log-linear models with hidden variables.

A similar result can be derived for Gaussian models, which gives some insight into the effect of the choice of parameterization (Gaussian *vs.* log-linear). For simplicity, consider single Gaussians $\mathcal{N}(x|\mu_c, 1)$ with unit variance. Assume that the means μ_c and features x are bounded such that the first derivative of the MMI training criterion is Lipschitz continuous with some constant L. Then, find an iteration constant E such that the EBW update [Kanevsky 04] is smaller than the update by Lemma 29, $\|\Delta\mu_c(E)\| \leq \frac{1}{L}\|\frac{\partial\mathcal{F}}{\partial\mu_c}\|$. This inequality implies $E \geq L + N$ [Schlüter & Macherey$^+$ 01]. Under the additional condition that $\|x_n - \mu_c\| \leq 2R$, the Lipschitz constant can be shown to be $L = 2N(4R^2 + 1)$. Again, the explicit dependency of L and E on N can be avoided by dividing the training criterion by N. Unlike for log-linear models, the Lipschitz constant is finite only for bounded model parameters and explicitly depends on μ_c (which are ambiguous according to [Heigold & Schlüter$^+$ 07, Heigold & Lehnen$^+$ 08]). These observations make the log-linear parameterization a more promising candidate for growth transformations.

The goal of this paper is not to find a finite iteration constant for EBW and Gaussian models but rather to derive a growth transformation (of different type) for general log-linear models with hidden variables. This example only serves for illustration and motivation purpose.

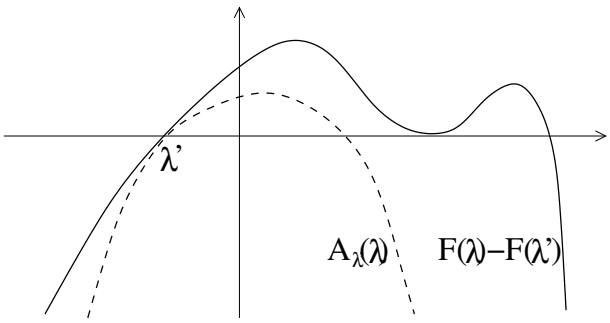


Figure 6.3: Illustration of auxiliary function. The auxiliary function $\mathcal{A}_{\Lambda'}(\Lambda)$ is a lower bound of the training criterion and has tangential contact at Λ' with the difference of the training criterion $\mathcal{F}(\Lambda) - \mathcal{F}(\Lambda')$.

6.2.3 Auxiliary functions

Auxiliary functions [Della Pietra & Della Pietra+ 97, Povey 04], also known as lower bounds [Bishop 06, pp.450], are a useful concept for the construction of growth transformations. According to [Della Pietra & Della Pietra+ 97, Povey 04], an auxiliary function is defined in terms of the old (current estimate) and the new (to be estimated) parameters Λ' and Λ, respectively.

Definition 30 (Auxiliary function)**.** *Assume an objective function $\mathcal{F} : \Gamma \to \mathbb{R},\ \Lambda \mapsto \mathcal{F}(\Lambda)$ to be maximized. An* auxiliary function *(in the strong sense) of the objective function $\mathcal{F}$ at Λ' is a continuously differentiable function $\mathcal{A}_{\Lambda'} : \Gamma \to \mathbb{R},\ \Lambda \mapsto \mathcal{A}_{\Lambda'}(\Lambda)$ that satisfies the inequality $\mathcal{F}(\Lambda) - \mathcal{F}(\Lambda') \geq \mathcal{A}_{\Lambda'}(\Lambda)$. Equality must hold true for $\Lambda = \Lambda'$.*

Here, we consider the absolute value $\mathcal{A}_{\Lambda'}(\Lambda)$ with the extra condition $\mathcal{A}_{\Lambda'} = 0$ instead of the equivalent formulation of the difference $\mathcal{A}_{\Lambda'}(\Lambda) - \mathcal{A}_{\Lambda'}(\Lambda')$ without constraints [Povey 04]. With the additional assumption on the differentiability of the auxiliary function to avoid pathological cases, the property $\nabla\mathcal{A}_{\Lambda'}(\Lambda') = \nabla\mathcal{F}(\Lambda')$ directly follows. Hence, our definition is also consistent with the definition in [Della Pietra & Della Pietra+ 97].

Each auxiliary function $\mathcal{A}$ induces a growth transformation by $\mathcal{G}(\Lambda) = \operatorname{argmax}_{\Lambda'}\{\mathcal{A}_{\Lambda}(\Lambda')\}$. Thus, the auxiliary functions inherit the properties of the growth transformations. Namely, these are the guaranteed increase of the objective function in each iteration and under mild assumptions, the convergence to a critical point of the objective function, similar to [Wu 83].

The goal of an auxiliary function is to break down the potentially difficult optimization problem into simpler subproblems that can be tackled more easily. For example (GIS), the auxiliary function decouples the parameters and an analytical solution exists.

The next lemma is used to generate new auxiliary functions by combining (simpler) existing auxiliary functions.

Lemma 31 (Transitivity)**.** *Let $\mathcal{B}_{\Lambda'}$ be an auxiliary function of $\mathcal{F}$ and let $\mathcal{A}_{\Lambda'}$ be an auxiliary function of $\mathcal{B}_{\Lambda'}$. Then, $\mathcal{A}_{\Lambda'}$ is also an auxiliary function of $\mathcal{F}$.*

A trivial example for this lemma are additive objective functions $\mathcal{F} = \mathcal{F}_1 + \mathcal{F}_2$ as used below for GIS. Assume auxiliary functions $\mathcal{A}_{1,\Lambda'}$ and $\mathcal{A}_{2,\Lambda'}$ for $\mathcal{F}_1$ and $\mathcal{F}_2$ at Λ', respectively. Setting $\mathcal{B}_{\Lambda'} := \mathcal{A}_{1,\Lambda'} + \mathcal{F}_2$ in the above lemma, $\mathcal{A}_{\Lambda'} := \mathcal{A}_{1,\Lambda'} + \mathcal{A}_{2,\Lambda'}$ is an auxiliary function of $\mathcal{F}$ at Λ'.

The following inequality [Cover & Thomas 91] shall turn out to be useful for deriving auxiliary functions below.

Lemma 32 (Jensen)**.** *If $E[\cdot]$ is the expectation of a random variable X and $f : \mathbb{R} \to \mathbb{R},\ x \mapsto f(x)$ a strictly convex function, then $f(E[X]) \leq E[f(X)]$ with equality if $X = const.$*

A similar inequality is valid for concave functions (*e.g.* log). Next, an example bound derived from this inequality is given.

Lemma 33 (Decomposition)**.** *Assume a measure μ and positive $f_\Lambda(x)$ such that the decomposition of the objective function $\mathcal{F}(\Lambda) = \log\left(\int f_\Lambda(x)\, d\mu(x)\right)$ exists. Then,*

$$\mathcal{A}_{\Lambda'}(\Lambda) \;:=\; \int \frac{f_{\Lambda'}(x)}{\int f_{\Lambda'}(x)\, d\mu(x)} \log\left(\frac{f_\Lambda(x)}{f_{\Lambda'}(x)}\right) d\mu(x).$$

is an auxiliary function of $\mathcal{F}$ at Λ'.

Proof. Basically, the same inequality as for the proof of expectation-maximization (EM) [Dempster & Laird$^+$ 77] is used:

$$\begin{aligned}
\mathcal{F}(\Lambda) - \mathcal{F}(\Lambda') &\overset{\text{assumption}}{=} \log\left(\frac{\int f_\Lambda(x)\, d\mu(x)}{\int f_{\Lambda'}(x)\, d\mu(x)}\right) \\
&\overset{\text{extension by } f_{\Lambda'}(x)}{=} \log\left(\int \frac{f_{\Lambda'}(x)}{\int f_{\Lambda'}(x)\, d\mu(x)} \frac{f_\Lambda(x)}{f_{\Lambda'}(x)}\, d\mu(x)\right) \\
&\overset{\text{Lemma 32}}{\geq} \int \frac{f_{\Lambda'}(x)}{\int f_{\Lambda'}(x)\, d\mu(x)} \log\left(\frac{f_\Lambda(x)}{f_{\Lambda'}(x)}\right) d\mu(x) \\
&=: \mathcal{A}_{\Lambda'}(\Lambda).
\end{aligned}$$

Equality holds true for $\Lambda = \Lambda'$. □

The auxiliary function of the previous lemma can be simplified, leading to the growth transformation suggested in [Gunawardana 01].

Corollary 34 (Decomposition)**.** *The function*

$$\mathcal{A}'_{\Lambda'}(\Lambda) \;:=\; \int f_{\Lambda'}(x) \log\left(\frac{f_\Lambda(x)}{f_{\Lambda'}(x)}\right) d\mu(x)$$

induces the same growth transformation as the auxiliary function $\mathcal{A}_{\Lambda'}$ from Lemma 33.

Proof. $\mathcal{A}_{\Lambda'}$ and $\mathcal{A}'_{\Lambda'}$ induce the same growth transformation if the optimum Λ is the same. Indeed, this condition is fulfilled because $\mathcal{A}'_{\Lambda'}(\Lambda) = \int f_{\Lambda'}(x)\, d\mu(x) \cdot \mathcal{A}_{\Lambda'}(\Lambda)$, *i.e.*, $\mathcal{A}_{\Lambda'}$ and $\mathcal{A}'_{\Lambda'}$ only differ in a factor that does not depend on Λ. Hence, $\operatorname{argmax}_\Lambda\{\mathcal{A}_{\Lambda'}(\Lambda)\} = \operatorname{argmax}_\Lambda\{\mathcal{A}'_{\Lambda'}(\Lambda)\}$. □

Expectation-maximization (EM). The EM algorithm [Dempster & Laird+ 77] can be formulated as a corollary of Lemma 33. Consider the objective function

$$\mathcal{F}(\Lambda) \quad := \quad \sum_{n=1}^{N} \log\left(\sum_{c} a_n(c)\tilde{p}_\Lambda(x_n, c)\right) \tag{6.2}$$

where $a_n(c)$ are non-negative weights and $\tilde{p}_\Lambda(x_n, c)$ stands for non-negative but not necessarily normalized scores. Standard EM used for the ML training of generative models is recovered for the true joint probabilities $p_\Lambda(x, c)$ [Bishop 06, pp.439]. In case of mixture models, the index c denotes the mixture/density index pair. The weights $a_n(c)$ filter out all densities of a mixture which represents the class. The auxiliary function is defined in terms of the *generalized* numerator posteriors

$$q_{a\Lambda}(c|x_n) \quad := \quad \frac{a_n(c)\tilde{p}_\Lambda(x_n, c)}{\sum_{c'} a_n(c')\tilde{p}_\Lambda(x_n, c')}. \tag{6.3}$$

Corollary 35 (EM)**.** *Assume the objective function $\mathcal{F}$ in Equation (6.2). Then,*

$$\mathcal{A}_{\Lambda'}(\Lambda) \quad := \quad \sum_{n=1}^{N}\sum_{c} q_{a\Lambda'}(c|x_n)\log\left(\frac{\tilde{p}_\Lambda(x_n, c)}{\tilde{p}_{\Lambda'}(x_n, c)}\right)$$

is an auxiliary function of $\mathcal{F}$ at Λ' where $q_{a\Lambda}(c|x)$ denotes the generalized numerator posterior in Equation (6.3).

Generalized iterative scaling (GIS). Like EM, GIS is based on the concept of growth transformations. First, an auxiliary function for the (partial) objective function

$$\mathcal{F}(\Lambda) \quad := \quad -\sum_{n=1}^{N} \log\left(\sum_{c} b_n(c)\exp\left(\sum_{i} \lambda_i f_i(x_n, c)\right)\right) \tag{6.4}$$

with $\Lambda := \{\lambda_i\}$ is provided. The result is stated in terms of the generalized numerator posteriors defined in Equation (6.3) and the similarly defined *generalized* denominator posteriors

$$p_{b\Lambda}(c|x_n) \quad := \quad \frac{b_n(c)\tilde{p}_\Lambda(x_n, c)}{\sum_{c'} b_n(c')\tilde{p}_\Lambda(x_n, c')}. \tag{6.5}$$

Like $a_n(c)$ for the numerator posteriors, $b_n(c)$ denote some non-negative weights.

Lemma 36 (GIS)**.** *Assume the objective function $\mathcal{F}$ from Equation (6.4) subject to the constraints $f_i(x_n, c) \geq 0$ for all i, n, c, and $\sum_i f_i(x_n, c) \equiv F$ for all n, c. Then,*

$$\mathcal{A}_{\Lambda'}(\Lambda) \quad := \quad N - \sum_{n=1}^{N}\sum_{c} p_{b\Lambda'}(c|x_n)\sum_{i}\frac{f_i(x_n, c)}{F}\exp\left(F(\lambda_i - \lambda_i')\right)$$

is an auxiliary function of $\mathcal{F}$ at Λ'. The generalized posteriors from Equation (6.5) with $\tilde{p}_\Lambda(x, c) := \exp\left(\sum_i \lambda_i f_i(x, c)\right)$ are used in this equation.

The assumptions on the feature functions are not restrictive. Without changing the posteriors, any set of feature functions can be transformed into a set of positive feature functions and augmented with a dummy feature $F - \sum_i f_i(x, c)$ such as to satisfy the sum constraint, see the invariance transformations in Section 4.3.4.

Proof. Basically, the same inequalities as for the proof of GIS [Darroch & Ratcliff 72] are used:

$$\begin{aligned}
\mathcal{F}(\Lambda) - \mathcal{F}(\Lambda') &\overset{\text{Equations (6.4),(6.5)}}{=} -\sum_{n=1}^{N} \log\left(\sum_c p_{b\Lambda'}(c|x_n) \exp\left(\sum_i (\lambda_i - \lambda_i') f_i(x_n, c)\right)\right) \\
&\overset{\log x \le x-1}{\geq} N - \sum_{n=1}^{N} \sum_c p_{b\Lambda'}(c|x_n) \exp\left(\sum_i F(\lambda_i - \lambda_i') \frac{f_i(x_n, c)}{F}\right) \\
&\overset{\text{Lemma 32}}{\geq} N - \sum_{n=1}^{N} \sum_c p_{b\Lambda'}(c|x_n) \sum_i \frac{f_i(x_n, c)}{F} \exp\left(F(\lambda_i - \lambda_i')\right) \\
&=: \mathcal{A}_{\Lambda'}(\Lambda).
\end{aligned}$$

Equality holds true for $\Lambda = \Lambda'$. □

Corollary 37 (GIS)**.** *The function*

$$\mathcal{A}_{\Lambda'}(\Lambda) := \sum_{n=1}^{N} \sum_i \lambda_i f_i(x_n, c_n) + N - \sum_{n=1}^{N} \sum_c p_{\Lambda'}(c|x_n) \sum_i \frac{f_i(x_n, c)}{F} \exp\left(F(\lambda_i - \lambda_i')\right) \quad (6.6)$$

is an auxiliary function of the objective function

$$\mathcal{F}(\Lambda) := \sum_{n=1}^{N} \log p_\Lambda(c_n|x_n) = \sum_{n=1}^{N} \log\left(\frac{\exp(\sum_i \lambda_i f_i(x_n, c_n))}{\sum_c \exp(\sum_i \lambda_i f_i(x_n, c))}\right)$$

at Λ'. The (true) posterior is denoted by $p_\Lambda(c|x) \equiv p_{1\Lambda}(c|x)$.

Proof. Decompose the objective function into the numerator and denominator part, $\mathcal{F}(\Lambda) = \mathcal{F}^{(\text{num})}(\Lambda) + \mathcal{F}^{(\text{den})}(\Lambda)$ with $\mathcal{F}^{(\text{num})}(\Lambda) := \sum_{n=1}^{N} \sum_i \lambda_i f_i(x_n, c_n)$ and $\mathcal{F}^{(\text{den})}(\Lambda) := -\log\left(\sum_c \exp\left(\sum_i \lambda_i f_i(x_n, c)\right)\right)$. Apply Lemma 36 to the denominator part with $b_n(c) = 1$. Then, an auxiliary function of the complete objective function follows from Lemma 31. □

The (unique) zero of the gradient of this auxiliary function determines the GIS update rules for Λ. In terms of the sufficient statistics

$$N_i := \sum_{n=1}^{N} \delta(c, c_n) f_i(x_n, c) \quad Q_i(\Lambda') := \sum_{n=1}^{N} \sum_c p_{\Lambda'}(c|x_n) f_i(x_n, c) \quad F = \max_{n,c}\left\{\sum_i f_i(x_n, c)\right\}, \quad (6.7)$$

the update rule reads

$$\lambda_i = \lambda_i' + \frac{1}{F} \log\left(\frac{N_i}{Q_i(\Lambda')}\right). \quad (6.8)$$

Extended Baum Welch (EBW) for discrete distributions. This paragraph summarizes the results in [Gunawardana 01, He & Deng+ 06]. For simplicity, simple (*i.e.,* no mixtures) discrete distributions are considered here. The extension to mixtures and HMMs is straightforward and can be found in [Gunawardana 01, He & Deng+ 06]. In the next lemma, an objective function in the rational form is considered [Kanevsky 04, He & Deng+ 06], representing a subset of the unified training criterion introduced in Section 3.3. The conventional training criteria including MMI, MCE, and MWE/MPE are covered by this objective function [He & Deng+ 08].

Lemma 38 (EBW (discrete))**.** *Assume the objective function*

$$\mathcal{F}(\Lambda) \quad := \quad \frac{\sum\limits_{c_1^N} a(c_1^N) p_\Lambda(x_1^N, c_1^N)}{\sum\limits_{c_1^N} b(c_1^N) p_\Lambda(x_1^N, c_1^N)}$$

with non-negative weights $a(c_1^N)$ and $b(c_1^N)$. Then, the function

$$\mathcal{A}_{\Lambda'}(\Lambda) \quad := \quad \sum_{c_1^N} \sum_{y_1^N} p_{\Lambda'}(y_1^N, c_1^N) \Big(\delta(y_1^N, x_1^N) a(c_1^N) - \delta(y_1^N, x_1^N) \mathcal{F}(\Lambda') b(c_1^N) + d(c_1^N) \Big) \log p_\Lambda(y_1^N, c_1^N)$$

induces a growth transformation of $\mathcal{F}$ for sufficiently large $d(c_1^N)$.

Proof. According to Lemma 28, it suffices to find a function $\mathcal{A}$ that induces a growth transformation of $\mathcal{H}_{\Lambda'}(\Lambda) = \log\left(P(\Lambda) - \mathcal{F}(\Lambda')Q(\Lambda) + \Delta\right)$ with $P(\Lambda) := \sum_{c_1^N} q(c_1^N) p_\Lambda(x_1^N, c_1^N)$ and $Q(\Lambda) := \sum_{c_1^N} p(c_1^N) p_\Lambda(x_1^N, c_1^N)$. For this purpose, the function $\mathcal{H}$ is decomposed as follows

$$\mathcal{H}_{\Lambda'}(\Lambda) \quad = \quad \log \left(\sum_{c_1^N} \sum_{y_1^N} p_\Lambda(y_1^N, c_1^N) \Big(\delta(y_1^N, x_1^N) a(c_1^N) - \delta(y_1^N, x_1^N) \mathcal{F}(\Lambda') b(c_1^N) + d(c_1^N) \Big) \right)$$

with the iteration constant

$$\Delta \quad := \quad \sum_{c_1^N} d(c_1^N). \tag{6.9}$$

Setting the iteration constants $d(c_1^N)$ such that

$$\delta(y_1^N, x_1^N) q(c_1^N) - \delta(y_1^N, x_1^N) \mathcal{F}(\Lambda') p(c_1^N) + d(c_1^N) \quad \geq \quad 0, \tag{6.10}$$

Corollary 34 applies such that

$$\mathcal{A}_{\Lambda'}(\Lambda) \quad = \quad \sum_{c_1^N} \sum_{y_1^N} p_{\Lambda'}(y_1^N, c_1^N) \Big(\delta(y_1^N, x_1^N) q(c_1^N) - \delta(y_1^N, x_1^N) \mathcal{F}(\Lambda') p(c_1^N) + d(c_1^N) \Big) \log p_\Lambda(y_1^N, c_1^N).$$

This concludes the proof. □

This last function $\mathcal{A}$ is optimized by setting the gradient to zero and solving the resulting equations for Λ. This leads to the well-known EBW update rules [Normandin & Morgera 91, Schlüter 00, Kaiser & Horvat+ 00, Gunawardana 01, He & Deng+ 06, Macherey 10]. It should be pointed out that this approach only provides finite iteration constants Δ for discrete-valued variables. In particular, the lemma fails for Gaussian models because the kernel function $\delta(\cdot, \cdot)$ in Equation (6.10) and thus, the iteration constant Δ in Equation (6.9) becomes infinity. To overcome the problem with infinite iteration constants for Gaussian models, a different kernel function is chosen in the next section to decompose the objective function.

6.2.4 Armijo's approach *vs.* GIS

One of the great challenges about the design of growth transformations is to find step sizes that are not overly pessimistic. Figure 6.2 illustrates this issue by comparing the step sizes from two different growth transformations (Armijo's approach *vs.* GIS). The example is shown for the USPS setup described in Section A.3.1 where $D = 513$, $F = 138$, and $N_i = 0.1$ for a typical component i. Thus, the Lipschitz constant in Lemma 29 is $L/N = R^2 \approx 100$. The gradient of the training criterion in Equation (6.1) can be expressed as the difference of the numerator and denominator accumulation statistics in Equation (6.7).

6.3 Extended Baum Welch (EBW) for GHMMs

This section extends the EBW result for discrete distributions proposed in [Gunawardana 01, He & Deng+ 06] and introduced in Section 6.2.3, to Gaussian models. More precisely, the emission probabilities $p_\Lambda(x|c) = \mathcal{N}(x|\mu_c, \Sigma_1)$ are represented by single Gaussians with mean $\mu_c \in \mathbb{R}^D$, a globally pooled covariance matrix $\Sigma_1 \in \mathbb{R}^{D\times D}$, and $\Lambda = \{\{\mu_c\}, \Sigma_1\}$. Again, the discussion is restricted to single Gaussians; the extension to GMMs and GHMMs is straightforward.

6.3.1 Assumption

The training criterion $\mathcal{F}$ for GHMMs is Lipschitz continuous except for an ϵ-neighborhood around vanishing variances. Lemma 29 (including the discussion) suggests that finite iteration constants can be only derived if zero variances are excluded. For this reason, the variances are bounded below by some $\Sigma_0 \in \mathbb{R}^{D\times D}$, *i.e.,* $\Sigma := \Sigma_0 + \Sigma_1$. This assumption permits to write the emission probabilities as the convolution of two Gaussians [Weisstein 09]

$$p_\Lambda(x|c) \quad := \mathcal{N}(x|\mu, \Sigma_0 + \Sigma_1) \quad = \int \mathcal{N}(y|\mu, \Sigma_1)\mathcal{N}(y|x, \Sigma_0)\, dy. \tag{6.11}$$

6.3.2 Decomposition

The result in Equation (6.11) is used to decompose the objective function in Lemma 38, *i.e.,* the kernel function is exchanged to avoid the pathological Dirac delta. Clearly, the decomposition is not unique because it depends on the choice of Σ_0. For $\Sigma_0 \to 0$, this decomposition and the composition suggested by [Gunawardana 01] coincide because the box functions and the Gaussians both approximate the Dirac delta.

Lemma 39 (EBW (Gauss))**.** *Assume the objective function*

$$\mathcal{F}(\Lambda) \quad := \quad \frac{\sum_{c_1^N} a(c_1^N)\mathcal{N}(x_1^N|\mu_{c_1}, \ldots, \mu_{c_N}, \Sigma_1)}{\sum_{c_1^N} b(c_1^N)\mathcal{N}(x_1^N|\mu_{c_1}, \ldots, \mu_{c_N}, \Sigma_1)}$$

with non-negative weights $a(c_1^N), b(c_1^N) \in \mathbb{R}^+$. *Then, the function*

$$\mathcal{A}_{\Lambda'}(\Lambda) := \sum_{c_1^N} \int \mathcal{N}(y_1^N|\mu'_{c_1},\ldots,\mu'_{c_N},\Sigma'_1)\left(\mathcal{N}(y_1^N|x_1^N,\Sigma_0)p(c_1^N)\left(a(c_1^N)-\mathcal{F}(\Lambda')b(c_1^N)\right)+d(c_1^N)\right) \cdot \log \mathcal{N}(y_1^N|\mu_{c_1},\ldots,\mu_{c_N},\Sigma_1)\, dy_1^N$$

induces a growth transformation of $\mathcal{F}$ *for sufficiently large iteration constants* $d(c_1^N) \in \mathbb{R}$.

The proof of this lemma is similar to the proof of Lemma 38 and thus, is omitted. It can be shown that the local iteration constants

$$\begin{aligned} d(c_1^N) &> \max_{y_1^N}\left\{-\mathcal{N}(y_1^N|x_1^N,\Sigma_0)p(c_1^N)\left(a(c_1^N)-\mathcal{F}(\Lambda')b(c_1^N)\right)\right\} \\ &= -\frac{1}{|2\pi\Sigma_0|^{\frac{N}{2}}}p(c_1^N)\left(a(c_1^N)-\mathcal{F}(\Lambda')b(c_1^N)\right) \end{aligned} \tag{6.12}$$

are "sufficiently large."

6.3.3 Update rules

The EBW update rules are determined by setting the gradient of the function $\mathcal{A}$ in Lemma 39 to zero and solving the equations for Λ. The solution is unique because $\mathcal{A}$ is the superposition of log-Gaussians with exclusively positive weights by construction. After some algebraic manipulations similar to [Gunawardana 01], we get the EBW reestimation formulae

$$\mu_c = \frac{\sum_{n=1}^N z_n \sum_{c_1^N:c_n=c} p_{b\Lambda'}(c_1^N|x_1^N)\left(\frac{a(c_1^N)}{\mathcal{F}(\Lambda')}-b(c_1^N)\right)+\Delta_c\mu'_c}{\sum_{n=1}^N \sum_{c_1^N:c_n=c} p_{b\Lambda'}(c_1^N|x_1^N)\left(\frac{a(c_1^N)}{\mathcal{F}(\Lambda')}-b(c_1^N)\right)+\Delta_c} \tag{6.13}$$

$$\Sigma_{1c} = \frac{\sum_{n=1}^N z_n z_n^\top \sum_{c_1^N:c_n=c} p_{b\Lambda'}(c_1^N|x_1^N)\left(\frac{a(c_1^N)}{\mathcal{F}(\Lambda')}-b(c_1^N)\right)+\Delta_c\mu'_c\mu'^\top_c}{\sum_{n=1}^N \sum_{c_1^N:c_n=c} p_{b\Lambda'}(c_1^N|x_1^N)\left(\frac{a(c_1^N)}{\mathcal{F}(\Lambda')}-b(c_1^N)\right)+\Delta_c} - \mu_c\mu_c^\top \tag{6.14}$$

$$+\frac{\sum_{n=1}^N \sum_{c_1^N:c_n=c} p_{b\Lambda'}(c_1^N|x_1^N)\left(\frac{a(c_1^N)}{\mathcal{F}(\Lambda')}-b(c_1^N)\right)}{\sum_{n=1}^N \sum_{c_1^N:c_n=c} p_{b\Lambda'}(c_1^N|x_1^N)\left(\frac{a(c_1^N)}{\mathcal{F}(\Lambda')}-b(c_1^N)\right)+\Delta_c}\left(\Sigma_{0c}^{-1}+\Sigma'^{-1}_{1c}\right)^{-1} \tag{6.15}$$

$$\Sigma_c = \Sigma_0 + \Sigma_{1c}. \tag{6.16}$$

These formulae are based on the generalized denominator posteriors in Equation (6.5). The class-specific iteration constants are defined as $\Delta_c := \sum_{n=1}^N \sum_{c_1^N:c_n=c} \frac{d(c_1^N)}{p(x_1^N)_{\Lambda'}\mathcal{F}(\Lambda')}$. The features z are the original features x smoothed with the mean μ'_c from the previous iteration

$$z := Px + (\mathcal{I}-P)\mu \tag{6.17}$$

where $P := \Sigma_1(\Sigma_0 + \Sigma_1)^{-1}$. The use of z implicitly reduces the convergence speed, the larger Σ_0 is. In contrast, the iteration constants in Equation (6.12) are the larger the smaller the smoothing (*i.e.,* Σ_0) is. Hence, the optimum Σ_0 will be a tradeoff between these two terms. This is similar to the update rules derived from the reverse Jensen inequality [Jebara 02], and is different from the conventional EBW update rules. The covariance matrices Σ_{1c} are positive-definite by construction of the iteration constants. The covariance matrix Σ is floored with Σ_0 by definition. In case of mixtures, similar update rules can be derived for the mixture weights. Again, the updated mixture weights are positive by construction of the iteration constants. This implies that the empirical iteration constants determined by imposing the positivity constraints of the variances and mixture weights are necessary but not sufficient conditions for the increase of the objective function. In particular, several heuristic constraints are replaced by a single and more restrictive constraint.

The above mentioned update rules are in the form used for HMMs. In case of i.i.d. observations, these update rules simplify considerably. For MMI, for instance, the update rules for the means then read

$$\mu_c = \frac{\sum_{n=1}^{N} z_n \left(\delta(c, c_n) - p_{b\Lambda'}(c|x_n)\right) + \Delta_c \mu'_c}{\sum_{n=1}^{N} \left(\delta(c, c_n) - p_{b\Lambda'}(c|x_n)\right) + \Delta_c}. \tag{6.18}$$

In case of GHMMs, the sums in the update rules can be identified with n-th order statistics. See Chapter 3 for the efficient calculation of these quantities. Last but not least, these update rules directly extend to the margin concept from Chapter 5 because the margin term only modifies the prior (*e.g.* the language model).

Keep in mind that the iteration constants depend on Σ_0. Using the invariance transformations in Section 4.3.4, Σ_0 can be made arbitrarily large for any posterior and thus, the iteration constants become arbitrarily large. This observation implies that a reasonable initial estimate for the variances is required for the optimal convergence speed. To the best of the author's knowledge, this ambiguity is also an issue for the algorithm based on the reverse Jensen inequality [Jebara 02]. GIS applied to the equivalent log-linear model does not suffer from this problem.

6.4 Generalized Iterative Scaling (GIS) for HCRFs

CRFs are often estimated using an entropy-based criterion in combination with GIS, or variants thereof. Like other algorithms based on growth transformations, GIS offers the immediate advantages that it is locally convergent, completely parameter free, and guarantees an improvement of the training criterion in each step. GIS, however, is limited in two aspects. GIS cannot be applied if the model incorporates hidden variables (*e.g.* HCRFs), and it can only be used for the MMI training criterion. In this section, the GIS algorithm from Section 6.2.3 is extended to resolve these two limitations. In particular, the new approach applies to HCRFs optimized with MMI or MPE. The proposed GIS-like method shares the above-mentioned theoretical properties of GIS.

6.4.1 Generalized objective function

Many problems of practical importance like for example HCRFs do not match the simple objective function in Corollary 37. The objective functions often involve hidden variables in some sense, requiring a more general formulation. Using prior-like (but not necessarily normalized) and sample-dependent weights $a_n(c), b_n(c) \geq 0$, the objective function

$$\mathcal{F}^{(\text{hidden})}(\Lambda) = \sum_{n=1}^{N} \log \left(\frac{\sum_c a_n(c) \exp\left(\sum_i \lambda_i f_i(x_n, c)\right)}{\sum_c b_n(c) \exp\left(\sum_i \lambda_i f_i(x_n, c)\right)} \right) \tag{6.19}$$

shall be considered. In fact, this objective function is equivalent to the objective function used in Lemma 39. The parameters to be estimated are denoted by $\Lambda = \{\lambda_i \in \mathbb{R}\}$. The major difference between the objective functions in Corollary 37 and Equation (6.19) is the (weighted) sum over the classes in the numerator. Equation (6.19) reduces to the conventional training criterion for log-linear models in Corollary 37 for $a_n(c) = \delta(c, c_n)$ and $b_n(c) = 1$. In this case, the sum in the numerator consists of a single summand and standard GIS can be applied. More complex examples are discussed in Section 6.4.3.

In the next subsection we propose an auxiliary function for this generalized criterion. For this purpose, it is convenient to rewrite the criterion as the sum of two objective functions $\mathcal{F}^{(\text{hidden})}(\Lambda) = \mathcal{F}^{(\text{num})}(\Lambda) + \mathcal{F}^{(\text{den})}(\Lambda)$ with

$$\mathcal{F}^{(\text{num})}(\Lambda) := \sum_{n=1}^{N} \log \left(\sum_c a_n(c) \exp\left(\sum_i \lambda_i f_i(x_n, c)\right) \right). \tag{6.20}$$

The objective function $\mathcal{F}^{(\text{den})}(\Lambda)$ is obtained from Equation (6.20) by replacing $a_n(c)$ with $b_n(c)$.

6.4.2 Generalized auxiliary function

In this section, we derive an auxiliary function for the generalized objective function in Equation (6.19). The definition and basic examples of auxiliary functions were given in Section 6.2.3. The desired auxiliary function is constructed by decomposing the problem into well-known subproblems and then combining these partial auxiliary functions to a complete auxiliary function of $\mathcal{F}^{(\text{hidden})}$ in Equation (6.19).

In Section 6.2.3, two separate auxiliary functions for the numerator and the denominator objective functions were provided. The combination of these auxiliary functions leads to an auxiliary function of the complete objective function.

Lemma 40 (G-GIS)**.** *Assume the objective function $\mathcal{F}^{(\text{hidden})}$ in Equation (6.19) with feature functions $f_i(x, c)$ subject to the assumptions in Lemma 36. Define the partial auxiliary functions*

$$\mathcal{A}^{(EM)}_{\Lambda'}(\Lambda) := \sum_{n=1}^{N} \sum_c q_{a\Lambda'}(c|x_n) \log \left(\frac{p_\Lambda(x_n, c)}{p_{\Lambda'}(x_n, c)} \right)$$

$$\mathcal{A}^{(GIS)}_{\Lambda'}(\Lambda) := N - \sum_{n=1}^{N} \sum_c p_{b\Lambda'}(c|x_n) \sum_i \frac{f_i(x_n, c)}{F} \exp\left(F(\lambda_i - \lambda'_i)\right).$$

Then, $\mathcal{A}^{(hidden)}_{\Lambda'} := \mathcal{A}^{(EM)}_{\Lambda'} + \mathcal{A}^{(GIS)}_{\Lambda'}$ *is an auxiliary function of* $\mathcal{F}^{(hidden)}$ *at* Λ'.

Proof. From Corollary 35 with $\tilde{p}_\Lambda(x,c) := \exp(\sum_i \lambda_i f_i(x,c))$ follows that $\mathcal{A}^{(\mathrm{EM})}_{\Lambda'}$ is an auxiliary function of $\mathcal{F}^{(\mathrm{num})}$ in Equation (6.20) at Λ'. Similarly, Lemma 36 shows that $\mathcal{A}^{(\mathrm{GIS})}_{\Lambda'}$ is an auxiliary function of $\mathcal{F}^{(\mathrm{den})}$ at Λ'. From the additivity of the objective function $\mathcal{F}^{(\mathrm{hidden})} = \mathcal{F}^{(\mathrm{num})} + \mathcal{F}^{(\mathrm{den})}$ follows that $\mathcal{A}_{\Lambda'} := \mathcal{A}^{(\mathrm{EM})}_{\Lambda'} + \mathcal{A}^{(\mathrm{GIS})}_{\Lambda'}$ is an auxiliary function of $\mathcal{F}^{(\mathrm{hidden})}$, see comment on Lemma 31. □

Setting the first derivatives of the auxiliary function $\mathcal{A}^{(\mathrm{hidden})}_{\Lambda'}(\Lambda)$ to zero and solving the equations for $\Delta\lambda_i := \lambda_i - \lambda'_i$ provides the update rules for the generalized objective function. With generalized definitions for the sufficient statistics

$$N_{ai}(\Lambda') := \sum_{n=1}^{N} \sum_{c} q_{a\Lambda'}(c|x_n) f_i(x_n, c) \qquad Q_{bi}(\Lambda') := \sum_{n=1}^{N} \sum_{c} p_{b\Lambda'}(c|x_n) f_i(x_n, c)$$
$$F = \max_{n,c} \left\{ \sum_i f_i(x_n, c) \right\}, \tag{6.21}$$

the gradients read

$$\frac{\partial \mathcal{A}^{(\mathrm{hidden})}_{\Lambda'}(\Lambda)}{\partial(\Delta\lambda_i)} = \frac{\partial \mathcal{A}^{(\mathrm{EM})}_{\Lambda'}(\Lambda)}{\partial(\Delta\lambda_i)} + \frac{\partial \mathcal{A}^{(\mathrm{GIS})}_{\Lambda'}(\Lambda)}{\partial(\Delta\lambda_i)} \overset{\text{see Lemma 40}}{=} N_{ai}(\Lambda') - Q_{bi}(\Lambda') \exp(F\Delta\lambda_i). \tag{6.22}$$

The update rules have the same structure as for standard GIS in Equation (6.8)

$$\Delta\lambda_i = \frac{1}{F} \log\left(\frac{N_{ai}(\Lambda')}{Q_{bi}(\Lambda')}\right). \tag{6.23}$$

Compared with Equation (6.7), Equation (6.21) uses the generalized numerator posteriors in Equation (6.3) which simplifies to $\delta(c, c_n)$ for standard GIS, and the generalized denominator posteriors in Equation (6.5) instead of the true posteriors.

6.4.3 Examples

There are several examples of practical interest which reduce to the generalized training criterion in Equation (6.19). The examples are based on feature functions of the type $f_{c'd}(x,c) = \delta(c,c')f_d(x)$ with the kernel feature functions $f_d : \mathbb{R}^D \to \mathbb{R}^+ : x \mapsto f_d(x)$ $(d = 1, \ldots, I)$. With this choice of feature functions, the sufficient statistics in Equation (6.21) simplify to

$$N_{a,cd}(\lambda') = \sum_{n=1}^{N} q_{a\lambda'}(c|x_n) f_d(x_n)$$
$$Q_{b,cd}(\lambda') = \sum_{n=1}^{N} p_{b\lambda'}(c|x_n) f_d(x_n)$$
$$F = \max_{n} \left\{ \sum_d f_d(x_n) \right\}. \tag{6.24}$$

See Chapter 4 for the definition of the log-linear models.

Log-linear mixtures (LMMs) & MMI. A log-linear mixture model (LMM) is a log-linear model of the type

$$p_\Lambda(s|x) = \frac{1}{Z_\Lambda(x)} \cdot \sum_l \exp\left(\sum_d \lambda_{sld} f_d(x)\right) \tag{6.25}$$

with the model parameters $\Lambda = \{\lambda_{sl} \in \mathbb{R}^I\}$. The normalization constant $Z_\Lambda(x)$ is computed over all mixture/component index pairs (s, l). MMI for LMMs can be embedded in the generalized training criterion in Equation (6.19) by the following interpretation of the symbols: n denotes the observation index, $c = (s, l)$, $a_n(s, l) = \delta(s, s_n)\delta(l \in s_n)$ (filter out the components of the correct mixture s_n), and $b_n(s, l) = 1$. This choice of parameters models the class posteriors $p_\Lambda(s_n|x_n)$. Recall that the mixture weights are represented by the kernel feature function $f_{sl}(x) = 1$. This unified treatment of the LMM parameters avoids the indirection proposed in [Saul & Lee 02].

Log-linear HMMs (LHMMs) & MMI. Log-linear HMMs (LHMMs) are linear-chain HCRFs [Gunawardana & Mahajan+ 05]. They can be considered a specialization of the LMMs in the last paragraph for strings

$$p_\Lambda(W|x_1^T) = \frac{1}{Z_\Lambda(x_1^T)} \cdot \sum_{s_1^T \in W} \prod_{t=1}^{T} \exp\left(\alpha_{s_{t-1}s_t} + \sum_d \lambda_{s_t d} f_d(x_t)\right) \tag{6.26}$$

with the state sequences s_1^T and the correct hypothesis W. The normalization is computed over all competing index pairs (V, s_1^T). The LHMM parameters are $\Lambda = \{\{\lambda_{sl} \in \mathbb{R}^I\}, \{\alpha_{s's} \in \mathbb{R}\}\}$. MMI for LHMMs is an instance of the generalized training criterion in Equation (6.19) when interpreting n as the sentence index r and setting $a_r(W, s_1^{T_r}) = \delta(W, W_r)\delta(s_1^{T_r} \in W_r)$, $b_r(W, s_1^{T_r}) = 1$. The transition from HMMs with log-linear models to HMMs with LMMs is realized by augmenting the HMM state s by the component index l. Additional scaling factors (*e.g.* the language model scale in case of continuous speech recognition) can be absorbed by the LHMM parameters. Hence, G-GIS also applies in this situation. Plugging the definitions into Equation (6.21), leads to the constant

$$F = \max_r \left\{\sum_{t=1}^{T_r} \sum_i f_i(x_t)\right\}. \tag{6.27}$$

In contrast to LMMs, the constant F is defined on the sentence level for LHMMs. Hence, the convergence of G-GIS for LHMMs will be very slow. For this reason, we discuss the hybrid approach next to break the definition of F down to the frame level.

LHMMs & frame-based MMI using context priors. In the hybrid approach, the HMM state posteriors are estimated with a suitable static classifier, *e.g.* neural networks (NNs) [Robinson & Hochberg+ 96, Stadermann 06] or support vector machines (SVMs) [Ganapathisraju 02]. Here, a log-linear model is employed to represent the posteriors $p_\Lambda(s|x)$. MMI is used to estimate the log-linear model

$$\mathcal{F}^{(\text{frame})}(\Lambda) = \sum_{t=1}^{T} \log\left(\frac{\sum_s a_t(s) \exp\left(\sum_d \lambda_{sd} f_d(x_t)\right)}{\sum_s b_t(s) \exp\left(\sum_d \lambda_{sd} f_d(x_t)\right)}\right). \tag{6.28}$$

Traditionally, the numerator and denominator weights are set to $a_t(s) = \delta(s, s_t)p(s_t)$ and $b_t(s) = p(s)$. The state priors $p(s)$ are the relative frequencies. Standard GIS applies in this conventional situation. Unlike MMI, frame-based MMI sets the constant F on the frame level, resulting in considerably faster convergence of GIS. This is possibly at the expense of a suboptimal training criterion because all context (*e.g.* language and transition models) and structural (*e.g.* restriction to valid state sequences) information is ignored.

Comparing this frame-based training criterion with MMI

$$\mathcal{F}^{(\mathrm{MMI})}(\Lambda) = \log\left(\frac{\sum\limits_{s_1^T \in W}\prod\limits_{t=1}^{T} p(s_t|s_{t-1})\exp\left(\sum_d \lambda_{s_t d} f_d(x_t)\right)}{\sum\limits_{V}\sum\limits_{s_1^T \in V}\prod\limits_{t=1}^{T} p(s_t|s_{t-1})\exp\left(\sum_d \lambda_{s_t d} f_d(x_t)\right)}\right),$$

refined priors can be derived [Heigold & Schlüter+ 07]. Frame-based MMI and (sentence-based) MMI differ in the choice of the classes to be discriminated (HMM states *vs.* HMM state sequences) and thus the summation space. MMI can be rewritten on the frame level

$$\mathcal{F}^{(\mathrm{MMI})}(\Lambda) = \frac{1}{T}\sum_{t=1}^{T}\log\left(\frac{\sum\limits_{s} p_{\Lambda t}(s|x_1^T \backslash x_t, W)\exp\left(\sum_d \lambda_{s_t d} f_d(x_t)\right)}{\sum\limits_{s} p_{\Lambda t}(s|x_1^T \backslash x_t)\exp\left(\sum_d \lambda_{s_t d} f_d(x_t)\right)}\right). \tag{6.29}$$

This frame-based formulation of MMI is based on the FB probabilities in Section 3.1.3

$$\begin{aligned} p_{\Lambda t}(s|x_1^T \backslash x_t, V) &= \sum_{s_1^T \in V: s_t = s}\prod_{\tau \neq t} p(s_\tau|s_{\tau-1})\exp\left(\sum_d \lambda_{s_\tau d} f_d(x_\tau)\right) \\ p_{\Lambda t}(s|x_1^T \backslash x_t) &= \sum_V p_{\Lambda t}(s|x_1^T \backslash x_t, V). \end{aligned} \tag{6.30}$$

As usual, the forward/backward probabilities are calculated efficiently with the forward/backward algorithm, see *e.g.* [Schlüter & Macherey+ 01]. If the dependency of the forward/backward probabilities on Λ is dropped, MMI in Equation (6.29) defines a frame-based MMI in Equation (6.28) with $a_t(s) = p(s|x_1^T \backslash x_t, W)$ and $b_t(s) = p(s|x_1^T \backslash x_t)$. In this case, the forward/backward probabilities are called *context priors*. They are computed on the initial model and then kept fixed during a number of training iterations. The training criterion is referred to as *frame-based MMI using context priors*. The context priors offer a principled way to consider some context and to smooth over competing states while keeping the advantages of the frame-based approach. This training criterion is an instance of the generalized training criterion in Equation (6.19) and thus, can be optimized with G-GIS.

LHMMs & minimum phone error (MPE). Minimum phone error (MPE) [Povey 04] was introduced in large vocabulary continuous speech recognition for the efficient error-based training, see Equation (3.13). Assume the string accuracy $A(V, W)$ between hypothesis V and the correct hypothesis W. MPE is defined as the expected accuracy

$$\mathcal{F}^{(\mathrm{MPE})}(\Lambda) = \sum_V p_\Lambda(V|x_1^T)A(V, W)$$

where $p_\Lambda(V|x_1^T)$ denotes an LHMM (see Section 6.4.3). Note that adding a constant to the accuracy (*e.g.* $A(V,W) - \min_{V',W'}\{A(V',W')\} \geq 0$) does not change the gradient of $F^{(\mathrm{MPE})}$. Thus, we can assume non-negative accuracies without loss of generality. To bring Equation (5.8) into the general form in Equation (6.19), the training criterion is defined on word sequences running over the complete corpus rather than over single sentences, *i.e.,* n is obsolete. Furthermore, set $c = (V, s_1^T)$, $a(V, s_1^T) = A(V,W)$, and $b(V, s_1^T) = 1$. Then, the MPE training criterion conforms with the generalized training criterion and thus, can be optimized with G-GIS. The constant F in Equation (6.24) for MPE and MMI coincide because the denominator is the same.

6.4.4 Refinements

Refinements that are compatible with the extension of GIS (Section 6.4.2) are discussed next, *e.g.* regularization and margin-based training.

Regularization. An additive regularization term based on the p-norm

$$\mathcal{R}_{pC}(\Lambda, \Lambda_0) = -\frac{C}{p}\sum_i |\lambda_i - \lambda_{0i}|^p \tag{6.31}$$

can be incorporated into G-GIS. For $p = 2$, *i.e.,* the Euclidean norm, the regularization term corresponds with a Gaussian prior with parameters $C \in \mathbb{R}^+$ (scaling) and $\Lambda_0 = \{\lambda_{0i} \in \mathbb{R}\}$ (centers) [Chen & Rosenfeld 99]. The gradient of the regularization term can be written as

$$\frac{\partial \mathcal{R}_{2C}}{\partial(\Delta\lambda_i)}(\Lambda, \Lambda_0) = -C(\lambda_i - \lambda_{0i}) = -C(\Delta\lambda_i + \lambda_i' - \lambda_{0i}). \tag{6.32}$$

The optimum updates $\Delta\lambda_i = \lambda_i - \lambda_i'$ are the zeroes of the gradient of the auxiliary function in Lemma 40 including the gradient of the regularization term in Equation (6.32)

$$N_i(\Lambda') - Q_i(\Lambda')\exp(F\Delta\lambda_i) - C(\Delta\lambda_i + \lambda_i' - \lambda_{0i}) = 0.$$

In contrast to the auxiliary function without a regularization term, the zero needs to be determined by an iterative procedure, *e.g.* Newton's method. The solution is unique because the expression is the derivative of a convex function. For $p \neq 2$, similar equations can be derived because the parameters are decoupled. Regularization with a Gaussian prior is comparable with I-smoothing used in discriminative training of GHMMs (Section 5.3.1).

Margin. The modified training criteria from Chapter 5 can be optimized with G-GIS. To do so, the weights $a_n(c), b_n(c) \in \mathbb{R}^+$ in Equation (6.19) are scaled with the margin term

$$\begin{aligned} a_n(c) &\leftarrow a_n(c)\exp(-\rho A(c, c_n)) \\ b_n(c) &\leftarrow b_n(c)\exp(-\rho A(c, c_n)) \end{aligned}$$

while the other steps remain unchanged.

Improved iterative scaling (IIS). The idea of improved iterative scaling (IIS) [Berger & Della Pietra+ 96, Della Pietra & Della Pietra+ 97] is compatible with G-GIS. Here, this approach is not pursued further to keep the algorithm as direct and simple as possible.

LHMMs & MMI. Assume that G-GIS is used to optimize LHMMs with MMI. From Equation (6.27) follows that the constant F scales with the number of time frames. This property of G-GIS is undesirable because it makes the step sizes overly pessimistic. The approximation of the context priors introduced in Section 6.4.3 avoids this effect. This section addresses the question under which conditions it is possible to relax this strict approximation to combine the advantages of frame-based and sentence-based MMI. Assuming that the context priors vary slowly from one iteration to the next, a slightly modified optimization algorithm can be derived.

Let $\mathcal{F}^{(\star,\text{num})}$, $\mathcal{F}^{(\star,\text{den})}$, and $\mathcal{F}^{(\star)}$ denote the training criteria in Equations (6.20),(6.4),(6.19), respectively, for the setting in Section 6.4.3 (MMI for LHMMs) or the setting in Section 6.4.3 (frame-based MMI for LHMMs). Using the auxiliary parameters $\alpha, \beta, \gamma \in \mathbb{R}^+$, the following utility training criteria are defined

$$\begin{aligned} \mathcal{F}_{\alpha}^{(\star,\text{num})}(\Lambda) &= \mathcal{F}^{(\star,\text{num})}(\Lambda) + \alpha\|\Lambda - \Lambda'\|_1 \\ \mathcal{F}_{\beta}^{(\star,\text{den})}(\Lambda) &= \mathcal{F}^{(\star,\text{den})}(\Lambda) + \beta\|\Lambda - \Lambda'\|_1 \\ \mathcal{F}_{\alpha,\beta,\gamma}^{(\star)}(\Lambda) &= \mathcal{F}_{\alpha}^{(\star,\text{num})}(\Lambda) + \mathcal{F}_{\beta}^{(\star,\text{den})}(\Lambda) - \gamma\|\Lambda - \Lambda'\|_1. \end{aligned} \tag{6.33}$$

The terms based on the ℓ_1-norm denoted by $\|\Lambda - \Lambda'\|_1$ in the last equation can be arbitrarily distributed over the different positions, *e.g.*

$$\mathcal{F}_{\alpha,\beta,\alpha+\beta}^{(\text{MMI})}(\Lambda) = \mathcal{F}_{0,0,0}^{(\text{MMI})}(\Lambda). \tag{6.34}$$

Using these definitions, we prove the following lemma.

Lemma 41 (LHMMs&MMI)**.** *Assume some compact domain Λ. The log-context priors in Equation (6.30) are Lipschitz continuous in the variable Λ with the Lipschitz constant $\epsilon > 0$. Then, $\mathcal{F}_{0,0,2\epsilon/T}^{(frame)}$ is a lower bound of $\mathcal{F}_{0,0,0}^{(MMI)} \equiv \mathcal{F}^{(MMI)}$ with contact in Λ'.*

The restriction of the model parameters to some compact Λ guarantees that the context priors are strictly positive. Thus, the log-context priors are well-defined and Lipschitz continuous. In practice, the use of some regularization (*e.g.* ℓ_2-regularization), assures that this condition is satisfied without explicitly restricting the space of the model parameters. In general, the Lipschitz constant ϵ depends on T. Equation (6.34) allows us to introduce some correction term to bound the context priors.

Proof. Using the identity in Equation (6.34) with $\alpha = \beta = \epsilon/T$, it suffices to prove that

- $\mathcal{F}_{0}^{(\text{frame,num})}$ is a lower bound of $\mathcal{F}_{\epsilon/T}^{(\text{MMI,num})}$ with contact in Λ';
- $\mathcal{F}_{0}^{(\text{frame,den})}$ is a lower bound of $\mathcal{F}_{\epsilon/T}^{(\text{MMI,den})}$ with contact in Λ'.

From these two auxiliary functions, the correctness of the lemma follows directly. To bound $\mathcal{F}^{(\mathrm{MMI,den})}_{\epsilon/T}$ in Equation (6.33), the frame-based representation of MMI in Equation (6.29) is used with $b_t(s) = p_{\Lambda t}(s|x_1^T \backslash x_t)$ and $b'_t(s) = p_{\Lambda' t}(s|x_1^T \backslash x_t)$

$$
\begin{aligned}
& T \cdot (\mathcal{F}^{(\mathrm{MMI,den})}_{\epsilon/T}(\Lambda) - \mathcal{F}^{(\mathrm{MMI,den})}_{\epsilon/T}(\Lambda')) \\
=& - \sum_{t=1}^{T} \log \left(\frac{\sum_s b_t(s) \exp(\lambda_s^\top f(x_t)) \exp(-\epsilon \|\Lambda - \Lambda'\|_1)}{\sum_s b'_t(s) \exp(\lambda_s'^\top f(x_t))} \right) \\
=& - \sum_{t=1}^{T} \log \left(\frac{\sum_s b'_t(s) \exp(\lambda_s^\top f(x_t)) \cdot \frac{b_t(s)}{b'_t(s)} \exp(-\epsilon \|\Lambda - \Lambda'\|_1)}{\sum_s b'_t(s) \exp(\lambda_s'^\top f(x_t))} \right) \\
\geq& - \sum_{t=1}^{T} \log \left(\frac{\sum_s b'_t(s) \exp(\lambda_s^\top f(x_t))}{\sum_s b'_t(s) \exp(\lambda_s'^\top f(x_t))} \right) \\
=& T \cdot (\mathcal{F}^{(\mathrm{frame,den})}_{0}(\Lambda) - \mathcal{F}^{(\mathrm{frame,den})}_{0}(\Lambda')).
\end{aligned}
$$

The ℓ_1-norm vanishes in the denominator because $\Lambda = \Lambda'$. Extending the terms in the numerator by $b'_t(s)$, we obtain the second identity. The lower bound follows from the inequality $\frac{b_t(s)}{b'_t(s)} \exp(-\epsilon \|\Lambda - \Lambda'\|_1) \leq 1$, which is trivial for $b_t(s) \leq b'_t(s)$. Otherwise, the inequality is a direct consequence of the Lipschitz continuity of the log-context priors

$$
\begin{aligned}
\log \frac{b_t(s)}{b'_t(s)} - \epsilon \|\Lambda - \Lambda'\|_1 &= \log b_t(s) - \log b'_t(s) - \epsilon \|\Lambda - \Lambda'\|_1 \\
&\leq \epsilon \|\Lambda - \Lambda'\|_1 - \epsilon \|\Lambda - \Lambda'\|_1 = 0.
\end{aligned}
$$

Furthermore, equality holds for $\Lambda = \Lambda'$. The bound for the numerator $\mathcal{F}^{(\mathrm{MMI,num})}$ can be derived similarly. □

Define $\mathcal{A}^{(\mathrm{frame})}_{\Lambda'}$ to be the auxiliary function in Lemma 40 for the setting in Section 6.4.3 using context priors. Then, $\mathcal{A}^{(\mathrm{frame})}_{\Lambda'}(\Lambda) - \epsilon/T \|\Lambda - \Lambda'\|_1$ using context priors is a lower bound of $\mathcal{F}^{(\mathrm{MMI})}$ with contact in Λ' for sufficiently large ϵ. The advantage of this approach is that $F = \max_t \{\sum_i f_i(x_t)\}$ is determined on the frame rather than the segment level, resulting in tighter bounds and thus faster convergence. The disadvantage of this approach is that the bound is not an auxiliary function as defined in Section 6.2.3 because the ℓ_1-norm is not continuously differentiable. Thus, a fixed point does not necessarily imply a critical point of the training criterion. Apart from the different definition of F, the update rules remain unchanged. In case of independent frames, the context priors are constant, *i.e.*, $\epsilon = 0$, and the original formulae for G-GIS are recovered. The additional term is negligible if $\epsilon/T \ll \frac{1}{\Delta\lambda}$ where $\Delta\lambda$ is the typical step size.

6.4.5 Convergence rate

The G-GIS update rules in Equation (6.23) lead to the growth transformation

$$
\mathcal{G}^{(\mathrm{G\text{-}GIS})} : \Gamma \to \Gamma, \ \Lambda \mapsto \Lambda + \frac{1}{F} (\log N_a(\Lambda) - \log Q_b(\Lambda))
$$

where we use the vector notation with the addition and logarithm defined componentwisely. The Taylor expansion of the growth transformation around the critical point $\Lambda^{(\infty)}$ leads to

$$\mathcal{G}^{(\text{G-GIS})}(\Lambda) \approx \Lambda^{(\infty)} + M^{(\text{G-GIS})}(\Lambda^{(\infty)})(\Lambda - \Lambda^{(\infty)}) \tag{6.35}$$

with the convergence rate matrix $M^{(\text{G-GIS})}(\Lambda^{(\infty)}) = \mathcal{I} + \frac{1}{F}\text{diag}(N_a(\Lambda^{(\infty)}))^{-1}\left(\nabla N_a(\Lambda^{(\infty)}) - \nabla Q_b(\Lambda^{(\infty)})\right)$. Here, we used $N_a(\Lambda^{(\infty)}) = Q_a(\Lambda^{(\infty)})$ in a critical point $\Lambda^{(\infty)}$ of the training criterion to simplify the expression. The Hessian matrix of $\mathcal{F}^{(\text{G-GIS})}$ in Equation (6.19) is denoted by $\nabla N_a(\Lambda^{(\infty)}) - \nabla Q_b(\Lambda^{(\infty)})$. This approximation leads to the inequality

$$\|\Lambda - \Lambda^{(\infty)}\|_\infty \leq \|M^{(\text{G-GIS})}(\Lambda^{(\infty)})\|_\infty \cdot \|\tilde{\Lambda} - \Lambda^{(\infty)}\|_\infty, \tag{6.36}$$

which describes the convergence rate of G-GIS. Similar ideas for GIS were presented in [Salakhutdinov & Roweis+ 03]. From the maximum norm follows that the convergence rate depends on the fundamental eigenvalue (*i.e.,* the maximum absolute value of the eigenvalues) of the convergence rate matrix. This result is known as Ostrowski's theorem [Ostrowski 60].

Like most other optimization algorithms, the convergence rate of G-GIS is linear. Moreover, the matrices $\nabla N_a(\Lambda)$ and $\nabla Q_b(\Lambda)$ are positive semidefinite and the Hessian matrix $\nabla N_a(\Lambda) - \nabla Q_b(\Lambda)$ is negative semidefinite around a local maximum. For GIS, the matrix $\nabla N_a(\Lambda)$ vanishes. This implies that G-GIS converges more slowly than GIS. For example, approximating the sum in the numerator in Equation (6.19) by the maximum summand will speed up the convergence.

6.4.6 Experimental Results

The proposed algorithm (G-GIS) is applied to the well-known United States Postal Service (USPS) database containing handwritten digits[2] and to the German digit string speech recognition task SieTill. The presented experiments go beyond standard GIS because of the density indices or HMM state sequences, see Section 6.4.3 for further details. G-GIS is compared with Rprop [Riedmiller & Braun 93], QProp [Fahlman 88], or EBW [Macherey & Schlüter+ 04] (if suitable). Since all these optimization algorithms make use of exactly the same statistics of the data, estimation time per iteration is comparable. Thus, a comparable number of iterations implies comparable computation time.

Handwritten digits (USPS). The well-known USPS handwritten digit database consists of isolated and normalized images of handwritten digits taken from US mail envelopes scaled to 16 x 16 pixels. The database contains a separate training and test set, with 7,291 and 2,007 images, respectively[3]. One disadvantage of the USPS corpus is that no development test set exists, resulting in the possible underestimation of error rates for all of the reported results. Note that this disadvantage holds for almost all data sets available for image object recognition. The US Postal Service task is still one of the most widely used reference data sets for handwritten character recognition and allows fast experiments due to its small size. The test set contains a large amount of image variability and is considered to be a "hard" recognition task. Good

[2]Thanks to Thomas Deselaers for providing the baseline systems and assisting me with the experiments.

[3]Data available from `ftp://ftp.kyb.tuebingen.mpg.de/pub/bs`.

error rates are in the range of 2-3% and use advanced modeling techniques, *e.g.* deformation models [Keysers & Deselaers+ 07].

LMMs & MMI. Here, we use log-linear mixture models (LMMs) with 16 components for each digit in combination with the gray-scale features augmented with Sobel-based derivatives, amounting to a total of 512 features. The model is optimized using MMI with ℓ_2-regularization. Comparative results are shown in Figure 6.4 for different optimization algorithms (G-GIS, Rprop, QProp) and for two different initialization points (from scratch, ML estimate of associated Gaussian mixture model (GMM) [Heigold & Schlüter+ 07]). The convergence speed

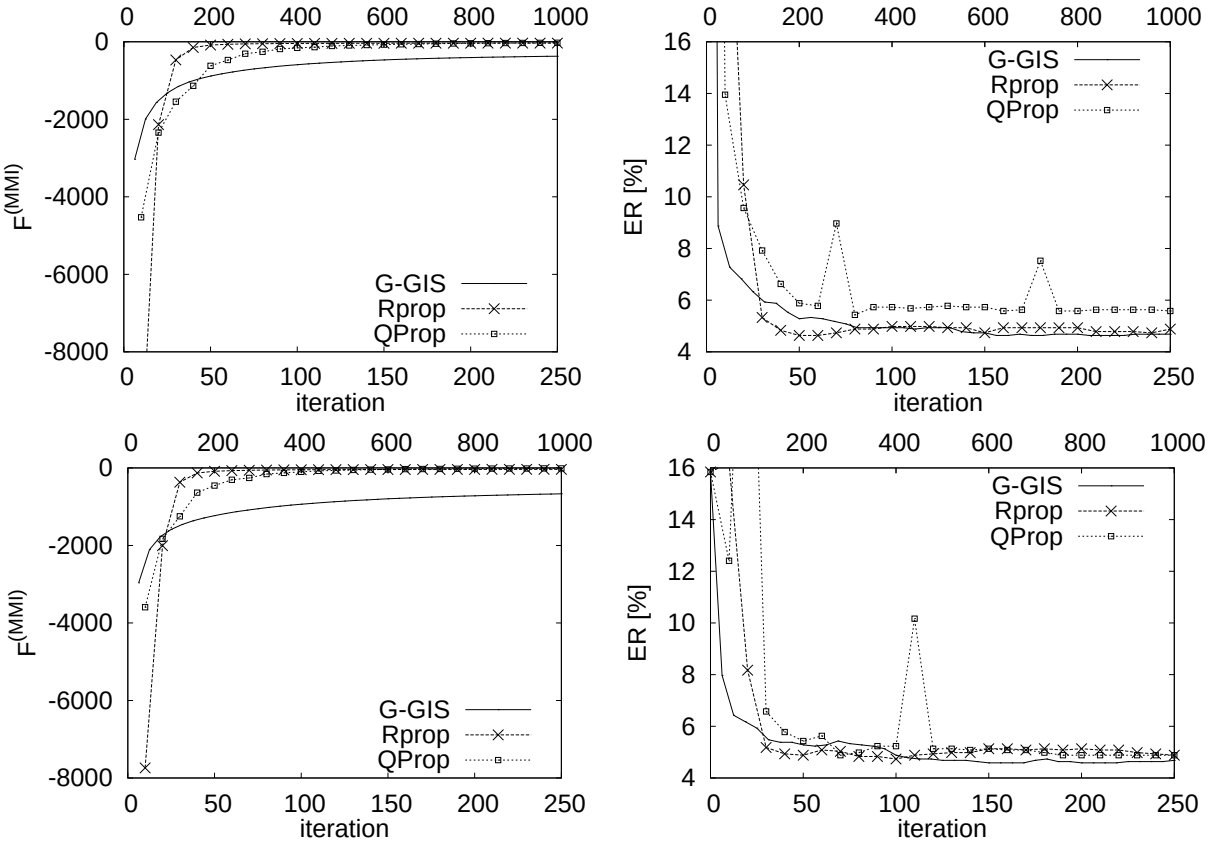


Figure 6.4: Comparison of different optimization algorithms (G-GIS, Rprop, QProp) for log-linear mixture models using MMI on USPS task. Upper: initialization from scratch. Lower: initialization with GMMs. Left: evolution of $\mathcal{F}^{(\mathrm{MMI})}$ on training corpus. Right: evolution of error rate (ER) on test corpus. Note the different scaling of the x-axis for G-GIS (upper axis) and QProp/Rprop (lower axis).

(and thus the computation time) for G-GIS, Rprop, and QProp is comparable, although G-GIS tends to be slower than Rprop and QProp. This is not surprising because G-GIS is derived for the worst case scenario. Furthermore, G-GIS achieves the same test error rates as Rprop and QProp, see Table 6.1. This was to be expected because the optimization problem is the same and is only solved differently. For this simple example, the initialization of the model does not seem to be an issue.

Table 6.1: Error rates (ER) on USPS test corpus for different optimization algorithms and initialization.

	ER [%]	
optimization	from scratch	from Gauss
Rprop	4.9	4.9
QProp	5.6	4.9
G-GIS	4.7	4.7

Spoken digit strings (SieTill). The SieTill task consists of spoken digit strings [Eisele & Haeb-Umbach+ 96]. The recognition system is based on gender-dependent whole-word HMMs with 430 distinct states in total. The vocabulary consists of the German digits, including a pronunciation variant. The feature vectors consist of twelve cepstral features without temporal derivatives. They are included by the linear discriminant analysis (LDA), which is applied to five consecutive frames and projects the resulting feature vector to 25 dimensions. Both training and test corpus consist of about 5.5h audio data/21k spoken digits per gender.

LHMMs & frame-based MMI using context priors. The ML baseline system uses single Gaussians with globally pooled variances. The progress of conventional MMI training using the de-facto standard EBW [Macherey & Schlüter+ 04] is shown in Figure 6.5 for comparison. This is the typical performance of EBW we observe for GHMMs using globally pooled variances [Heigold & Wiesler+ 10]. First, the convergence is relatively slow, in particular compared with systems using untied variances [Macherey & Schlüter+ 04]. Second, the empirical iteration constants lead to well-defined GHMMs [Macherey & Schlüter+ 04, Povey 04, Axelrod & Goel+ 07] but not necessarily to an improvement of the training criterion. The latter is only guaranteed for sufficiently large iteration constants [Kanevsky 04, Axelrod & Goel+ 07]. Both [Kanevsky 04] and [Axelrod & Goel+ 07] do not make an explicit statement on what "sufficiently large" means in this context. More severely, it is not clear if empirical EBW converges to a critical point or if it converges at all. EBW may find better local optima than other gradient-based optimization algorithms in general. This does not seem to be the case in this example. Similarly, it has not been proved EBW finds better local optima than other gradient-based optimization algorithms in general. The Gaussian ML baseline model is used to initialize the log-linear model with first order features [Gunawardana & Mahajan+ 05] for further training with frame-based MMI using context priors. The posterior of the spoken digit string is obtained by marginalization over the HMM state sequences. Hence, standard GIS does not apply. The context priors are computed on word lattices and are recomputed after a certain number of training iterations (period). In Figure 6.5, G-GIS is compared with QProp and EBW. The word error rate (WER) for frame-based MMI (without context priors) is 3.1%. In contrast to EBW, G-GIS converges monotonically and smoothly to the same word error rate (Table 6.2). However, this appears to be at the expense of a considerably slower convergence: the computation time of G-GIS is 1000 times larger than for EBW. This is probably due to the fact that the MFCC features are basically unbounded, which drives the constant F in Equation (6.27) to huge values even in the frame-based approach. This can be avoided by a suitable choice of the features as discussed next.

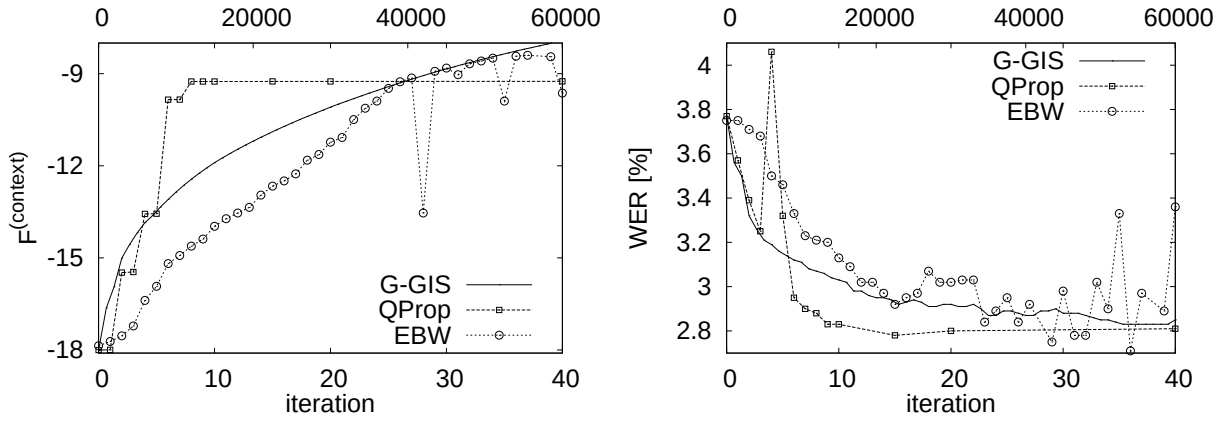


Figure 6.5: Comparison of different optimization algorithms (G-GIS, QProp, EBW) for log-linear models with frame-based MMI using context priors on male portion of SieTill, period=250 (G-GIS), 2 (QProp), 1 (EBW, *i.e.,* conventional MMI), see text for explanation. Left: evolution of $\mathcal{F}^{(\text{frame})}$ on training corpus. Right: evolution of word error rate (WER) on test corpus. Note the different scaling of the x-axis for G-GIS (upper axis) and QProp (lower axis).

Table 6.2: Word error rates (WER) on SieTill test corpus for different optimization algorithms. Keep in mind that the error rates for the system using MFCCs and the system using cluster features are not directly comparable. The latter is a stand-alone log-linear system and thus, EBW cannot be used. The result for frame-based MMI (without context priors) is included for comparison.

	WER [%]	
optimization	MFCCs	clusters
EBW	2.8	N/A
QProp/Rprop	2.8	2.2
G-GIS	2.8	2.2
QProp (frame-based MMI)	3.1	

LHMMs & M-MMI. Consider the marginal likelihood $p(x)$. Here, this quantity is approximated by a GMM, $p(x) = \sum_l p(l)\mathcal{N}(x|\mu_l, \Sigma)$. The priors $p(l)$, the clusters μ_l, and the pooled covariance matrix Σ are estimated in a preprocessing step. Then, the cluster features are defined as $f_l(x) = p(l)p(x|l)/\sum_{l'} p(l')p(x|l')$. Temporal context can be taken into account by a sliding window. See [Abdel-Haleem 06, Wiesler & Nußbaum-Thom+ 09] for further details on this type of features. The clustering features appear unusual in the view of GHMMs but may be more promising in log-linear modeling [Abdel-Haleem 06, Wiesler & Nußbaum-Thom+ 09]. The cluster features have the advantage of being bounded and to sum up to one, *cf.* the constant F. Thus, a higher convergence speed for G-GIS is expected. The gender-dependent LHMMs are jointly optimized with M-MMI (see Chapter 5 and Section 6.4.4). No approximations like for example the use of (pruned) word lattices or replacing the sum by the maximum are employed. The optimization problem is non-convex. This is why the LHMMs are initialized with some (suboptimal) frame-based MMI estimate. The training is then continued with M-MMI using ℓ_2-regularization centered around this initial model. The regularization-like term $\|\Lambda - \Lambda'\|_1$ in Lemma 41 is ignored. This drastic step may be justified by the rather pessimistic step sizes

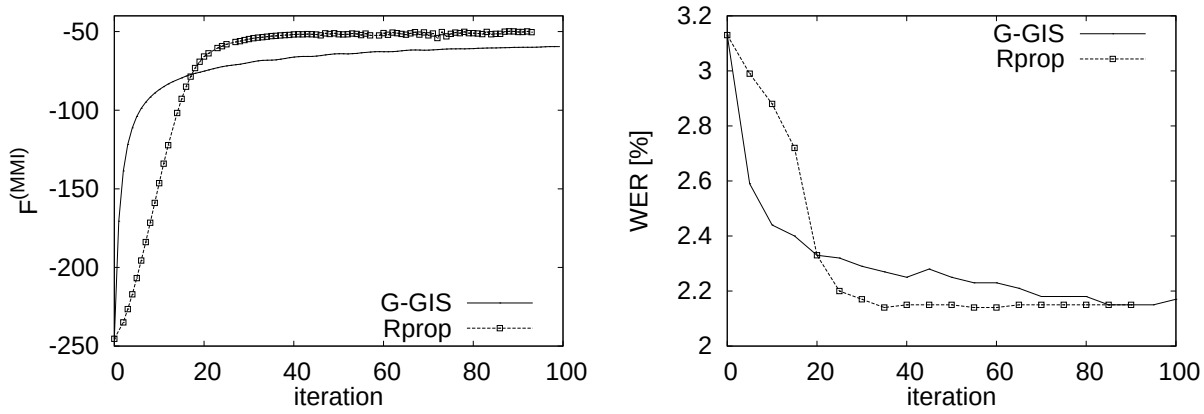


Figure 6.6: Comparison of different optimization algorithms (G-GIS, Rprop) for LHMMs using (exact) MMI on complete SieTill task. Left: evolution of $\mathcal{F}^{(\mathrm{MMI})}$ on training corpus. Right: evolution of word error rate (WER) on test corpus.

in Figure 6.6 and by the assumption of a weak transition model. Figure 6.6 suggests that this choice of features and assumptions can speed up G-GIS considerably. The word error rates for G-GIS and Rprop are the same, see Table 6.2.

6.5 Summary

We proposed two novel growth transformations in this chapter. First, the well-known GIS algorithm was extended to deal with hidden variables. This extension does not only apply to the MMI estimation of HCRFs but it also allows for the training of HCRFs using more refined training criteria, *e.g.* MPE in ASR. Hence, this generalized GIS can be considered the analog for log-linear discriminative models of EM used for generative models. The experimental results confirmed the theoretical properties of the proposed optimization algorithm. Moreover, the algorithm compared well with standard gradient-based optimization algorithms. The error rates are consistent with those of conventionally optimized systems. Second, a new growth transformation for Gaussian models was derived. The resulting update rules resemble the EBW update rules. In contrast to other types proof, our approach provides explicit finite iteration constants for Gaussian models.

Chapter 7

Convex Optimization using Log-Linear HMMs

Conventional discriminative training has proved to significantly improve maximum likelihood (ML) optimized acoustic models. To make it work well in practice, however, it involves much engineering work including the choice of the initialization to avoid spurious local optima, the tuning of parameters such as the scaling factors and the time distortion penalties, and many heuristics (*e.g.* the splitting of densities) and approximations (*e.g.* the word lattices). This makes it difficult not only to reproduce experiments but strictly speaking also to compare different algorithms due to spurious local optima. For this reason, a fool-proof training algorithm would be attractive. This chapter studies convex optimization based on the log-linear parameterization in speech recognition. Convex optimization techniques have the additional property that the global optimum is accessible. This topic has rarely been studied in the context of discriminative training in speech recognition [Abdel-Haleem 06, Sha & Saul 07a]. Experimental results are presented for a digit string recognition task [Heigold & Rybach+ 09] to investigate the feasibility and utility of this concept. Also, first results are shown for the European Parliament plenary speeches task.

7.1 Introduction

First, the notion of a "fool-proof" training algorithm is described in detail. A description of the convex training criteria[1] is given in Section 7.2. Finally, the practical issues, to be checked in Section 7.3, are discussed.

7.1.1 Properties of fool-proof training

A "fool-proof" training algorithm is assumed to have the following properties:

- Well-definedness of global optimum.

[1] Mathematically speaking, the training criteria are concave. Here, we will use the notion of convexity and concavity interchangeably.

- Uniqueness of global optimum. This implies that the result is independent of the initialization.
- Accessibility of global optimum in finite time. This issue addresses the question if the global optimum can be efficiently found in practice (*cf.* convex optimization).
- Well-posedness of optimization problem.
- Joint optimization of all model parameters. Some simplified training criteria (*e.g.* frame-based MMI) do not have this property.
- Small mismatch between training criterion and evaluation measure.

Such a training algorithm should be able to optimize all model parameters from scratch, without any tuning *etc.* and independent of the optimization algorithm and its parameters. Of course, the definition of a convex training criterion is not unique. Similar work can be found in literature. Examples include [Kuo & Gao 06] (focus on the choice of feature functions), [Abdel-Haleem 06] (no comparable training from scratch), and [Sha & Saul 07a] (GHMMs, no training from scratch). Here, the training criterion is based on M-MMI introduced in Chapter 5. The implementation reuses the transducer-based discriminative framework from Chapter 3.

7.1.2 Assumptions for convex optimization in speech recognition

The optimization of the entropy (MMI) for log-linear conditional random fields (CRFs) leads to a convex optimization problem [Lafferty & McCallum+ 01, Sutton & McCallum 07]. Hidden CRFs (HCRFs) are CRFs that allow for hidden variables in addition [Gunawardana & Mahajan+ 05]. They are closely related to Gaussian HMMs, see Chapter 4. The optimization problem for HCRFs is no longer convex. Besides the log-linear parameterization of the model and MMI as the training criterion, a few more assumptions need to be made to combine the advantages of CRFs (convexity) and HCRFs (model structure). First, assume that the HMM state alignment is known before the training and kept fixed during the training. Second, augmented features are used (*cf.* kernel trick) to avoid mixtures. In fact, the mixtures could be treated in a similar way as the HMM state alignment [Abdel-Haleem 06, Sha & Saul 07a]. This approach has not been implemented to limit the number of approximations and to avoid problems with the initialization of the density indices if starting from scratch. Under these assumptions, convex training criteria are derived, see Section 7.2.[2]

7.1.3 Practical issues to be checked

The definition of a "fool-proof" training algorithm as discussed in Section 7.1.1 allows for the optimization of all model parameters from scratch in a principled way. Section 7.3 checks how well the theoretical concepts carry over to practice. Special attention will be paid to the following issues.

[2] Although GHMMs and LHMMs are equivalent as shown in Chapter 4, the use of GHMMs [Sha & Saul 07a] increases considerably the complexity of the optimization algorithm due to the parameter constraints of GHMMs.

Sensitivity to HMM state alignment? The key assumption is that the HMM state alignment is known and kept fixed during the training. Yet, the oracle alignment is not available in practice, and needs to be estimated. In general, the model used to generate the alignment is not related to the model to be estimated. This is the case if for example the discriminative model cannot be initialized with a corresponding GHMM as *e.g.* in [Abdel-Haleem 06, Sha & Saul 07a]. Thus, how sensitive is the performance to this (initial) alignment?

Correlation of training criterion and recognition error? The overall goal is the discrimination of different word sequences. To derive convex training criteria, it is assumed that the correct word sequence can be represented appropriately by a single HMM state sequence. This implies the discrimination of HMM state sequences belonging to the same word sequence. Also, the convex training criteria in Section 7.2 are all based on MMI. The loss function of MMI is not directly related to the recognition error, see Section 5.2.4. Thus, does the convex training criterion define a reasonable optimum?

Dependency on model initialization? According to the theory, the performance of the models should be independent of the initialization. Is this true in practice as well, or do the training algorithms suffer from numerical stability problems?

The goal of this chapter is not so much to find improved features for log-linear models [Abdel-Haleem 06, Kuo & Gao 06]. It rather focuses on the investigation of the utility and feasibility of convex training criteria using log-linear models in speech recognition. The experiments are performed on a simple, yet competitive model, which allows for a thorough experimental investigation of the above issues.

7.2 Convex Optimization in Speech Recognition

This section starts with the definition of the gender-specific models considered in this chapter. Convex training criteria defined on the frame and sentence level are then discussed.

7.2.1 Gender-specific log-linear models

Here, simple linear-chain CRFs (*cf.* Section 4.5.4 without language model) are considered [Gunawardana & Mahajan$^+$ 05, Abdel-Haleem 06]. The model includes gender-dependent emission features and gender-independent transition features. Features to represent the language model are not used because the focus is on the recognition of digit strings.

In the following, $x \in \mathbb{R}^D$ denotes a feature vector, s is an HMM state, and $g \in \{\male, \female\}$ stands for the gender. For convenience, the (pseudo) emission model

$$\Phi_{\Lambda_{em}}(x, s, g) \quad = \quad \exp\left(\alpha_{sg} + \lambda_{sg}^{\top} x\right)$$

(zeroth- and first-order features only) or

$$\Phi_{\Lambda_{em}}(x, s, g) \quad = \quad \exp\left(\alpha_{sg} + \lambda_{sg}^{\top} x + x^{\top} \lambda_{sg} x\right)$$

(zeroth-, first-, and all second-order features), and the (pseudo) transition model

$$\Psi_{\Lambda_{tdp}}(s', s) \quad = \quad \exp(\alpha_{s's})$$

are introduced. The model parameters are $\Lambda = \Lambda_{em} \cup \Lambda_{tdp}$ with $\Lambda_{em} := \{\{\alpha_{sg}\}, \{\lambda_{sg}\}\}$ or $\Lambda_{em} := \{\{\alpha_{sg}\}, \{\lambda_{sg}\}, \{\lambda_{sg}\}\}$, and $\Lambda_{tdp} := \{\alpha_{s's}\}$. The HMM state sequence s_1^T is assumed to define the digit string uniquely such that the dependency on the digits can be dropped. This approach can be extended to more sophisticated features, *e.g.* [Kuo & Gao 06, Abdel-Haleem 06, Wiesler & Nußbaum-Thom+ 09]. This definition of the model leads to the decision rule

$$\hat{s}_1^T \quad = \quad \arg\max_{s_1^T} \left\{ \max_{g} \left\{ \prod_{t=1}^{T} \Psi_{\Lambda_{tdp}}(s_{t-1}, s_t) \Phi_{\Lambda_{em}}(x_t, s_t, g) \right\} \right\}. \tag{7.1}$$

As already mentioned, the best state sequence $\hat{s}_1^T$ is assumed to uniquely define the recognized digit string.

Before discussing the different training criteria for these models, the general problem of estimating gender-specific models in the discriminative framework is addressed.

7.2.2 Discriminative training of gender-specific models

The decision rule in Equation (7.1) requires that the scores of both genders are comparable. For ML, this is not an issue because the optimization problem decouples into the two gender-dependent optimization problems, *i.e.,* the gender-specific models can be optimized separately

$$\begin{aligned} \operatorname*{argmax}_{\Lambda_{♂}, \Lambda_{♀}} \left\{ \prod_{r=1}^{R} p_{\Lambda_{♂}, \Lambda_{♀}}(x_1^{T_r} | s_1^{T_r}, g_r) \right\} \quad = \quad & \operatorname*{argmax}_{\Lambda_{♂}} \left\{ \prod_{r: g_r = ♂} p_{\Lambda_{♂}}(x_1^{T_r} | s_1^{T_r}, ♂) \right\} \\ & \cdot \operatorname*{argmax}_{\Lambda_{♀}} \left\{ \prod_{r: g_r = ♀} p_{\Lambda_{♀}}(x_1^{T_r} | s_1^{T_r}, ♀) \right\} \end{aligned}$$

for segments $r = 1, \ldots, R$. If the models are optimized in the discriminative framework independently, the scores are no longer guaranteed to be comparable due to the invariance transformations in Section 4.3.4

$$\begin{aligned} \operatorname*{argmax}_{\Lambda_{♂}, \Lambda_{♀}} \left\{ \prod_{r=1}^{R} p_{\Lambda_{♂}, \Lambda_{♀}}(s_1^{T_r}, g_r | x_1^{T_r}) \right\} \quad \neq \quad & \operatorname*{argmax}_{\Lambda_{♂}} \left\{ \prod_{r: g_r = ♂} p_{\Lambda_{♂}}(s_1^{T_r} | x_1^{T_r}, g_r) \right\} \\ & \cdot \operatorname*{argmax}_{\Lambda_{♀}} \left\{ \prod_{r: g_r = ♀} p_{\Lambda_{♀}}(s_1^{T_r} | x_1^{T_r}, g_r) \right\}. \end{aligned}$$

For this reason, the two gender-specific models need to be optimized jointly. This is in contrast to previous work [Macherey 10]. There, this issue is not considered critical because the discriminative training was initialized with ML optimized GHMMs.

The complexity of the combined training algorithm is roughly four times larger per iteration than for the isolated training of the gender-specific models. This increase in complexity arises

from the increased amount of training data (factor of two) and from the augmented summation space (another factor of two). In addition, the convergence rate is expressed in terms of some metric on the parameter space [Nocedal & Wright 99, pp. 28]. Thus, the convergence rate is the slower the more parameters are considered for the optimization.

7.2.3 Refinements to maximum mutual information (MMI)

Several refinements to MMI are considered here. First, ℓ_2-regularization is used

$$\mathcal{R}_C(\Lambda) \quad := \quad -\frac{C_{em}}{2}\sum_{s,g}\lambda_{sg}^2 - \frac{C_{tdp}}{2}\sum_{s',s}\alpha_{s's}^2.$$

Furthermore, the posteriors can be scaled by some $\gamma \in \mathbb{R}^+$, and a margin term scaled with some $\rho \in \mathbb{R}^+$ can be incorporated into standard MMI. These modifications are implemented by substituting the original scores

$$\prod_{t=1}^{T}\Psi_{\Lambda_{tdp}}(s_{t-1}, s_t)\Phi_{\Lambda_{em}}(x_t, s_t, g_t)$$

by the scaled margin-scores

$$\left(\exp(-\rho A(s_1^T, \hat{s}_1^T))\prod_{t=1}^{T}\Psi_{\Lambda_{tdp}}(s_{t-1}, s_t)\Phi_{\Lambda_{em}}(x_t, s_t, g_t)\right)^{\gamma}.$$

Here, $A(\cdot,\cdot)$ denotes some accuracy between two strings, *e.g.* the Hamming accuracy in Section 3.7.1. The resulting variant of MMI is called modified/margin-based MMI (M-MMI). See Chapter 5 for further details.

Now, we are in the position to define the different variants of M-MMI used in Section 7.3.

7.2.4 Sentence-based M-MMI

We start with the non-convex lattice-based M-MMI training criterion (Chapter 3) and then, derive a convex formulation from this training criterion.

Lattice-based M-MMI. Conventional lattice-based M-MMI training uses word lattices D to approximate the normalization constant for the string posterior. In addition, the maximum approximation is assumed such that each hypothesis in the word lattice uniquely defines an HMM state sequence. An exemplary word lattice is shown in Figure 7.1. The numerator lattice N is the set of HMM state sequences representing the correct word sequence. These assumptions lead to the M-MMI training criterion

$$\mathcal{F}^{(\text{lattice})}(\Lambda) \quad = \tag{7.2}$$

$$\sum_{r=1}^{R}\log\left(\frac{\sum\limits_{s_1^{T_r}\in N_r}\prod\limits_{t=1}^{T_r}\Psi_{\Lambda_{tdp}}(s_{t-1}, s_t)\Phi_{\Lambda_{em}}(x_t, s_t, g_r)\exp(-\rho\delta(s_t \in N_{rt}))}{\sum\limits_{g\in\{\male,\female\}}\sum\limits_{s_1^{T_r}\in D_r}\prod\limits_{t=1}^{T_r}\Psi_{\Lambda_{tdp}}(s_{t-1}, s_t)\Phi_{\Lambda_{em}}(x_t, s_t, g)\exp(-\rho\delta(s_t \in N_{rt}))}\right) + \mathcal{R}_C(\Lambda).$$

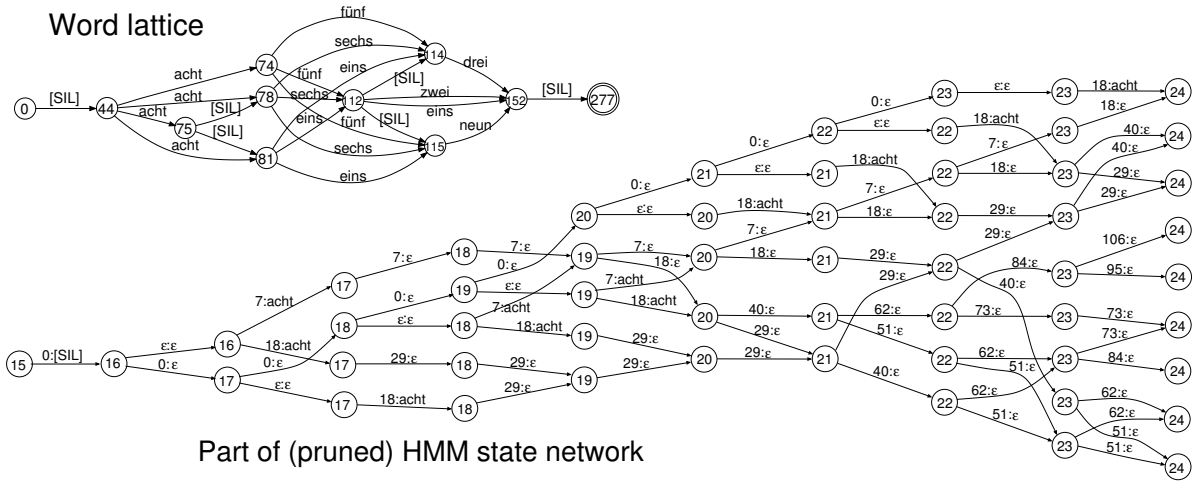


Figure 7.1: Word lattice D to approximate the summation space (left) *vs.* full summation space S (right).

The margin term fires if and only if the HMM state is in the numerator lattice at the time frame t, *i.e.*, $s_t \in N_{rt}$. The posterior is normalized over all HMM state sequences s_1^T in the denominator lattice and genders g. The transition parameters can be estimated in this framework. This, however, was not done in this work, but they were tuned manually as usual.

This choice of the posterior results in a *non-convex* training criterion, both for GHMMs and HCRFs. This is due to the sum in the numerator of the string posterior, and the incomplete sum for the normalization constant in combination with realigning the hypotheses.

"Fool-proof" M-MMI (convex formulation). This conventional training criterion can be made convex, *i.e.,* the HCRF is cast into a CRF, similar to [Abdel-Haleem 06, Sha & Saul 07a]. This is achieved by replacing the normalization constant by the sum over the complete set S of HMM state sequences, and by using a single HMM state sequence $\hat{s}_1^T$ representing the correct hypothesis string in the numerator. The HMM state sequence $\hat{s}_1^T$ is determined by some existing acoustic model, or initialized from the linear segmentation.

$$\mathcal{F}^{(\text{fool-proof})}(\Lambda) = \sum_{r=1}^{R} \log \left(\frac{\prod_{t=1}^{T_r} \Psi_{\Lambda_{tdp}}(\hat{s}_{t-1}, \hat{s}_t) \Phi_{\Lambda_{em}}(x_t, \hat{s}_t, g_r) \exp(-\rho \delta(\hat{s}_t, \hat{s}_t))}{\sum_{g \in \{\male, \female\}} \sum_{s_1^{T_r} \in S_r} \prod_{t=1}^{T_r} \Psi_{\Lambda_{tdp}}(s_{t-1}, s_t) \Phi_{\Lambda_{em}}(x_t, s_t, g) \exp(-\rho \delta(s_t, \hat{s}_t))} \right) + \mathcal{R}_C(\Lambda). \tag{7.3}$$

This training criterion is referred to as "fool-proof" M-MMI because it possesses all properties from Section 7.1.1.

This training criterion was implemented in our transducer-based discriminative framework (Chapter 3). A weighted finite-state transducer represents the complete set of valid HMM state sequences, which can be of different length.[3] The edge weights are set to the transition scores. The emission scores are stored in another transducer, having a WFST state for each time frame

[3]Thanks to David Rybach for providing the HMM state networks for the fool-proof MMI training.

and having an edge in each WFST state for each HMM state. The denominator lattice is then obtained by composition of these two transducers. The margin transducer is treated in the same way, if necessary. The resulting transducer is similar to the network used for transducer-based search. For the training, however, duplicate hypotheses need to be avoided (log *vs.* tropical semiring). An essential difference from the lattice-based formulation is that the "fool-proof" training criterion discriminates between HMM state sequences even if they represent the same word sequence.

7.2.5 Frame-based M-MMI

Due to the summation over all HMM state sequences, the approach in Equation (7.3) is only feasible for small tasks (*e.g.* digit strings). For larger tasks, we adopt the hybrid approach to optimize the emission parameters in Equation (4.28). Here, log-linear models instead of neural networks [Robinson & Hochberg+ 96] or support vector machines [Ganapathisraju 02] are taken as the static classifiers. All other parameters cannot be optimized in this approach. This simplification considerably speeds up the training. Similar to "fool-proof" MMI, the best HMM state sequence $\hat{s}_1^T$ is assumed to be known and kept fixed during the training. The symbol posterior includes the HMM state prior $p(s)$ (*e.g.* relative frequencies)

$$\mathcal{F}^{(\text{frame})}(\Lambda) = \sum_{r=1}^{R} \sum_{t=1}^{T_r} w(\hat{s}_t, g_r) \log \left(\frac{p(\hat{s}_t) \Phi_{\Lambda_{em}}(x_t, \hat{s}_t, g_r) \exp(-\rho \delta(\hat{s}_t, \hat{s}_t))}{\sum_{g \in \{♂,♀\}} \sum_{s} p(s) \Phi_{\Lambda_{em}}(x_t, s, g) \exp(-\rho \delta(s, \hat{s}_t))} \right) + \mathcal{R}_C(\Lambda). \tag{7.4}$$

This frame-based training criterion is convex but not "fool-proof" in the sense of Section 7.1.1, *cf.* the last two properties.

The experiments suggest that it is essential to down-weight silence/noise frames for accumulation, see the weights $w(s, g) \in \mathbb{R}^+$ in Equation (7.4). This is probably due to the high silence portion. In practice, setting the total silence weight to the average weight of all other states turned out to be a good (initial) choice. In fact, the parameters are not defined uniquely, see Section 4.3.4. For this reason, one of the states (*e.g.* the silence state) does not need to be explicitly estimated and can be arbitrarily set. This statement is only exact without regularization. Our experience is that the parameters (*e.g.* the time distortion penalties, language model scale) need to be badly retuned after the training.

Table 7.1 provides an overview of the different variants of MMI and their properties.

7.3 Model Training: Experimental Results

Different aspects (*cf.* Section 7.1.3) of the training criteria discussed in the previous section are studied on the German digit string recognition task SieTill, see A.1.1. The ultimate goal is the optimization of all model parameters from scratch (Table 7.3).

Table 7.1: Comparison of different variants of MMI and their properties.

Property	Frame-based MMI	Lattice-based M-MMI	Fool-proof M-MMI
Well-definedness	✓	✓	✓
Uniqueness	✓	✓	✓
Accessibility	✓	local optima	✓
Well-posedness	to be checked	to be checked	to be checked
Model parameters	emission	all	all
Small mismatch	to be checked	to be checked	to be checked

Table 7.2: Comparison of MMI-based training criteria for SieTill test corpus, simple setup (first-order features, transition parameters tuned manually), initialization with corresponding ML optimized GHMM.

Model	Criterion	Convex	WER [%]
GHMM	ML	no	3.8
	lattice-based M-MMI	no	2.7
log-linear model/HCRF/CRF	frame-based MMI	yes	3.0
	frame-based M-MMI		3.0
	lattice-based MMI	no	2.9
	lattice-based M-MMI		2.7
	fool-proof MMI	yes	3.1
	fool-proof M-MMI		2.5

7.3.1 Effect of margin term

Preliminary studies were performed on a very simple setup to check several basic issues, *e.g.* the choice of the training criterion. We used the gender-specific model in Equation (7.2.1) with only first-order features. The transition parameters were kept fixed during the training, unless otherwise stated. The model was initialized with the associated GHMM to speed up the training. The results are summarized in Table 7.2. Unlike fool-proof MMI where the margin term appears to be essential for this setup, frame-based MMI does not benefit from the margin term.

To check the estimation of the transition features, the transition features were estimated from scratch, using the system optimized with M-MMI. The resulting error rate does not differ significantly from that in Table 7.2, *i.e.,* the optimization works but the manually tuned values are already pretty close to the optimum.

These preliminary results suggest that convex optimization may help. It is essential to define a suitable training criterion to achieve good results. Here, the convex training criterion defined on the sentence level including a margin term performed best.

7.3.2 Dependency on model initialization

Next, it is checked if the convex training criteria produce the same word error rate for different model initializations. We used the model in Equation (7.2.1) with first- and second-order

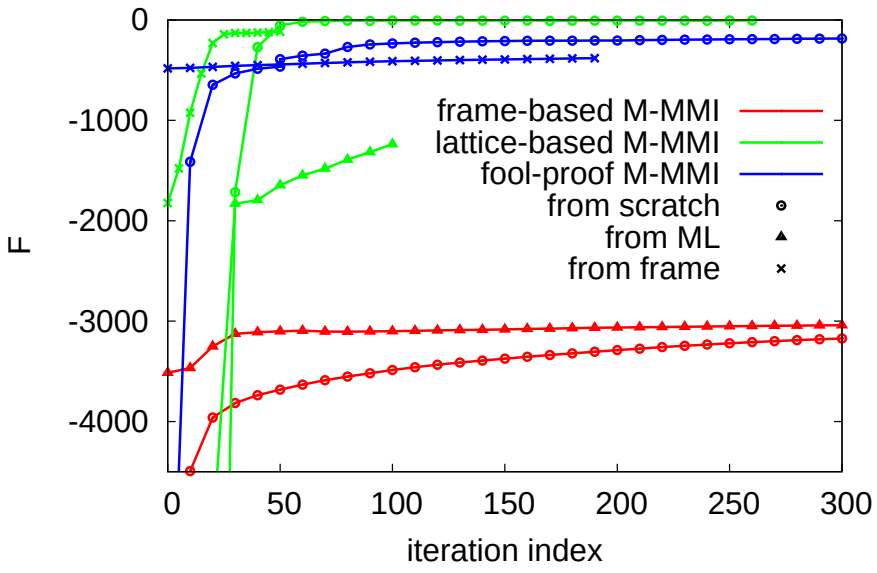


Figure 7.2: Progress of training criterion $\mathcal{F}$ *vs.* training iteration index for SieTill training corpus. Note that the lattice-based training criteria are scaled up by a factor of 1000.

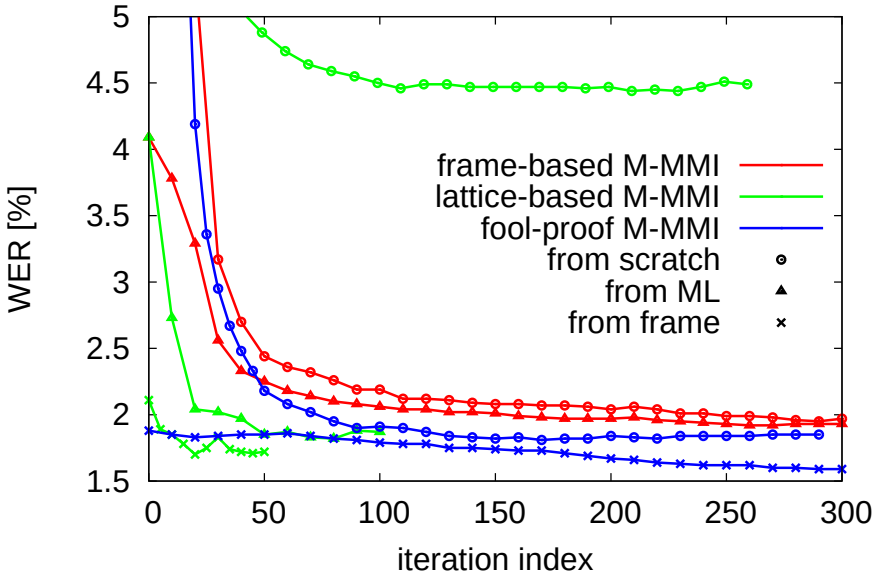


Figure 7.3: Progress of word error rate (WER [%]) *vs.* training iteration index for SieTill test corpus.

features, and the (global) transition features (for fool-proof MMI only). The single HMM state sequence representing the correct hypothesis is determined by some reasonable GHMM with a single globally pooled diagonal covariance matrix. The convergence behavior of the different training criteria is shown in Figures 7.2 and 7.3. The corresponding word error rates after convergence can be found in Table 7.3. In case of ML initialization ('ML'), the ML optimized

Table 7.3: Impact of model initialization on word error rate (WER) for SieTill test corpus. The model includes first- and second-order features. In case of fool-proof MMI, the transition parameters are also optimized.

Model	Criterion	Initialization	Convex	WER [%]
GHMM	ML	from scratch	no	1.8
(64 dns/mix)	lattice-based M-MMI	ML	no	1.6
log-linear model/	frame-based M-MMI	from scratch	yes	1.9
HCRF/CRF		ML	yes	1.9
	lattice-based M-MMI	from scratch	no	4.5
		ML	no	2.0
		frame-based M-MMI	no	1.8
	fool-proof M-MMI	from scratch	yes	1.8
		frame-based M-MMI	yes	1.5

Table 7.4: Frame-based MMI model training from scratch for different initial alignments with realignment, first- and second-order features.

Model	Criterion	Initial alignment	#Realign.	WER [%]
GHMM	ML	linear segmentation	16	1.8
(64 dns/mix)	lattice-based M-MMI	ML	30	1.6
log-linear model	frame-based MMI	linear segmentation	5	1.9
		ML (1 dns/mix)	1	1.9
		ML (16 dns/mix)	2	1.9

GHMM baseline with globally pooled variances served as initialization.

The experiments in Table 7.3 suggest that fool-proof MMI can reliably estimate all model parameters from scratch. Moreover, the performance of the model is competitive with our best GHMM (64 densities/mixture, notably having over four times more model parameters than the log-linear model with first- and second-order features).

7.3.3 Correlation of training criterion and word error rate

Figures 7.2 and 7.3 suggest that the training criterion and the word error rate are sufficiently correlated for the task under consideration. A larger value of the training criterion (only comparable within the same training criterion), however, does not necessarily imply a lower word error rate, see lattice-based or fool-proof MMI.

7.3.4 Sensitivity to initial alignment & realignment

So far, we have assumed that a good initial alignment is known for training. Table 7.4 investigates the sensitivity of the word error rate to the initial alignment and realignments. Keep in mind that if allowing for realignments, the training criterion is no longer guaranteed to converge to the global optimum.

Table 7.5: Comparison of frame-based MMI (from scratch) and fool-proof MMI (initialized with frame-based MMI) for different window sizes, first- and second-order features.

Window size	WER [%]	
	frame-based MMI	+fool-proof MMI
5	1.9	1.5
11	1.5	1.4

Table 7.6: Effect of higher-order features for SieTill test corpus, frame-based MMI (convex) *vs.* lattice-based MMI (non-convex).

Feature order			#Parameters	WER [%]	
0th/1st	2nd	3rd	[k]	frame-based MMI	lattice-based M-MMI
✓			11	3.0	2.7
✓	diagonal		22	2.7	2.2
✓	full		151	1.9	1.8
✓	full	✓	1,409	1.8	1.5
				ML	lattice-based M-MMI
Gaussian HMM			715	1.8	1.6

7.3.5 Increased temporal context

Temporal context can be taken into account by a sliding window before the LDA. Can we improve on the above word error rates by increasing the window size while keeping the outgoing feature dimension fixed? Table 7.5 summarizes the results for frame-based and fool-proof MMI. The results suggest that frame-based and fool-proof MMI perform equally if a sufficient temporal context is considered. Otherwise fool-proof MMI appears to better compensate for the insufficient acoustic modeling. One might speculate if this is the same effect as studied in [Nádas & Nahamoo+ 88] for ML and MMI.

7.3.6 Feasibility and utility of higher-order features

The effect of higher-order features is studied on the SieTill and EPPS tasks.

German digit strings. The same setup as above and described in Appendix A.1.1 is used. Table 7.6 shows the effect of higher-order features (up to degree three) for frame-based MMI and conventional lattice-based MMI. The tuning of the regularization constant C in Equation (7.2) is illustrated in Figure 7.4.

EPPS English. This task contains recordings from the European Parliament plenary sessions (EPPS). In contrast to SieTill, EPPS is a LVCSR task based on phoneme models represented by 3x2-states HMMs. The RWTH setup from the TC-STAR evaluation campaign 2006 is used [Lööf & Bisani+ 06b, Lööf & Bisani+ 06a]. The acoustic front end comprises MFCC features augmented by a voicing feature. Nine consecutive frames are concatenated and the resulting

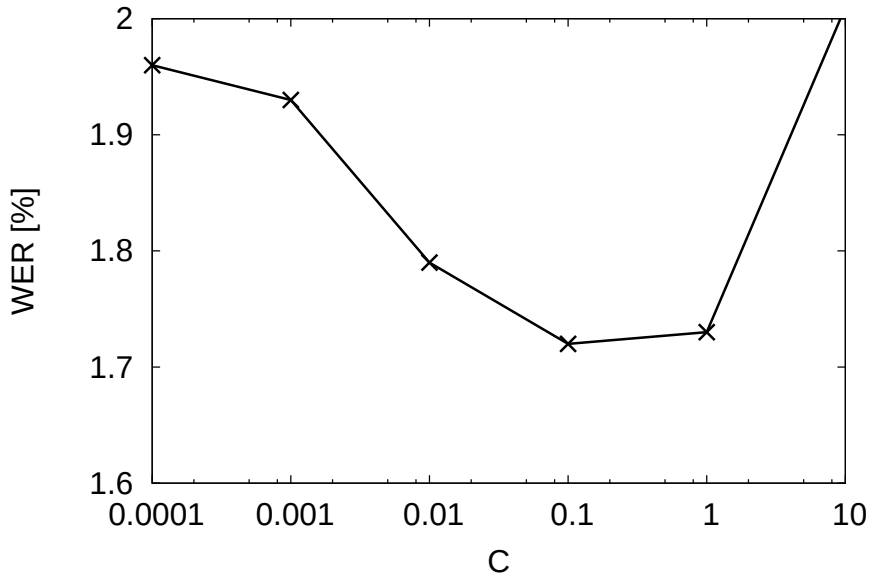


Figure 7.4: Word error rate (WER, [%]) *vs.* regularization constant C for SieTill test corpus, first- and second-order features, 50 Rprop training iterations with lattice-based M-MMI initialized with frame-based MMI.

vector is projected to 45 dimensions by means of an LDA. The MFCC features are warped using a fast variant of the vocal tract length normalization (VTLN). The triphones are clustered using CART, resulting in 4,501 generalized triphone states. For recognition, a lexicon with 50k entries in combination with a 4-gram language model is used. The ML baseline system uses Gaussian mixtures with globally pooled variances. The corpus statistics is described in detail in Appendix A.1.3. For frame-based MMI, the weights $w(s)$ in Equation (7.4) were set to zero for silence and noise states to avoid annoying tuning of these parameters. According to the discussion in Section 7.2.5, this is not expected to restrict the model too much. The frame-based training of the acoustic model with only first-order features was initialized with the ML optimized GHMM using single densities. Adding higher-order features was done step by step. The sentence-based training was initialized with the corresponding frame-based MMI optimized acoustic model. The results are summarized in Table 7.7

Higher-order features beyond degree three are not feasible. For larger tasks like for example EPPS, already the use of third-order features leads to unacceptably high training times while the second-order features are limited regarding WER. Additional (sparse) features were considered in [Wiesler & Nußbaum-Thom$^+$ 09].

7.4 Linear Feature Transforms in Log-Linear Framework

Linear discriminant analysis (LDA) [Häb-Umbach & Ney 92, Kumar & Andreou 98] has been established as an important means for dimension reduction and decorrelation in speech

Table 7.7: Word error rate (WER) on EPPS English test corpora, frame-based training with higher-order features of different degree.

Model	Criterion	Feature order: zeroth/first	Feature order: second	#Prm. [k]	WER [%]: Dev06	WER [%]: Eval06
log-linear model	frame-based MMI	✓		207	26.1	22.0
		✓	diagonal	410	24.9	20.5
		✓	full	4,866	20.8	16.8
HCRF	lattice-based M-MMI	✓	full	4,866	20.2	16.4
GHMM	ML	first+mixtures		207	29.2	24.7
				6,477	18.9	16.1
				39,915	16.6	13.7

recognition. The major points of criticism of LDA are that the estimation is performed in a separate preprocessing step, and that it uses an *ad hoc* training criterion that is not directly related to the word error rate. This section introduces a new discriminative training method for the estimation of (projecting) linear feature transforms [Tahir & Heigold+ 09]. More precisely, the problem is formulated in the log-linear framework such that the convex training criteria in Section 7.2 can be used for optimization. The proposed approach is compared with LDA on the digit string recognition task, both for ML and MMI optimized acoustic models. Related work for ML optimized GHMMs can be found in [Omar & Hasegawa-Johnson 03].

7.4.1 Log-linear representation of linear feature transforms

Assume that the linear feature transform is represented by the transformation matrix $A = [a_{dd'}] \in \mathbb{R}^{d \times D}$. Then, the feature vector $x \in \mathbb{R}^D$ transforms into the feature vector $y \in \mathbb{R}^d$ via $y = Ax$. These transformed features can be plugged into the frame-based training criterion in Equation (7.4)

$$\begin{aligned} \mathcal{F}^{(\text{frame})}(\Lambda, A) &= \sum_{r=1}^{R} \sum_{t=1}^{T_r} w(\hat{s}_t, g_r) \log \left(\frac{p(\hat{s}_t) \exp(\alpha_{\hat{s}_t g_r} + \lambda_{\hat{s}_t g_r}^\top A x_t)}{\sum_{g \in \{\male, \female\}} \sum_s p(s) \exp(\alpha_{sg} + \lambda_{sg}^\top A x_t)} \right) \\ &= \sum_{r=1}^{R} \sum_{t=1}^{T_r} w(\hat{s}_t, g_r) \log \left(\frac{p(\hat{s}_t) \exp(\alpha_{\hat{s}_t g_r} + \sum_{d,d'} a_{dd'} \lambda_{\hat{s}_t g_r, d} x_{td'})}{\sum_{g \in \{\male, \female\}} \sum_s p(s) \exp(\alpha_{sg} + \sum_{d,d'} a_{dd'} \lambda_{sgd} x_{td'})} \right). \end{aligned} \tag{7.5}$$

The regularization and margin terms have been ignored for the sake of simplicity. The d-th component of the feature vector x and the model vector λ_{sg} are denoted by x_d and λ_{sgd}. If the model parameters $\Lambda = \{\{\lambda_{sgd}\}, \{\alpha_{sg}\}, \{\alpha_{s's}\}\}$ are kept constant, then Equation (7.5) defines a log-linear model in the matrix coefficients $\{a_{dd'}\}$ associated with the abstract features $\lambda_{sgd} x_{d'}$ (and vice versa). A similar model can be derived for the other training criteria in Section 7.2, *e.g.*

fool-proof MMI in Equation (7.3)

$$\mathcal{F}^{(\text{fool-proof})}(\Lambda, A) = \sum_{r=1}^{R} \log \left(\frac{\prod_{t=1}^{T_r} \exp(\alpha_{\hat{s}_{t-1}\hat{s}_t} + \alpha_{\hat{s}_t g_r} + \lambda_{\hat{s}_t g_r}^{\top} A x_t)}{\sum_{g \in \{\male, \female\}} \sum_{s_1^{T_r} \in S_r} \prod_{t=1}^{T_r} \exp(\alpha_{s_{t-1}s_t} + \alpha_{s_t g_r} + \lambda_{s_t g_r}^{\top} A x_t)} \right). \tag{7.6}$$

This approach can be extended to mixtures, see Chapter 4. Unless the optimum density of a mixture is chosen in the numerator and kept fixed, the resulting training criteria are no longer convex. In fact, this is the same idea as used for the HMM state sequences.

7.4.2 Optimization

The model parameters Λ and the feature transform A can be optimized jointly as indicated by the arguments of the training criterion $\mathcal{F}$ in Equations (7.5) and (7.6). The training criterion is convex if either Λ or A is kept constant. This suggests an alternating optimization strategy, *i.e.,* optimize first A for fixed Λ, then optimize Λ for fixed A, *etc.*

The training criterion $\mathcal{F}(\Lambda, A)$ is not convex in all parameters (Λ, A). This shall be illustrated by means of a simplified model. Assume the model parameters $\Lambda = \{\lambda_1, \ldots, \lambda_S\} \in \mathbb{R}^{Sd}$. Then, the joint training criterion is equivalent to the training criterion of the unprojected log-linear model restricted to the subset

$$\Gamma \quad := \quad \left\{ \Lambda \in \mathbb{R}^{SD} | \lambda_s = \sum_{i=1}^{d} a_{si} v_i \text{ for all } s,\ a_{si} \in \mathbb{R}, v_i \in \mathbb{R}^D \right\} \subset \mathbb{R}^{SD}.$$

The training criterion would be convex if the restriction Γ were a convex set, *i.e.,*

$$\forall \Lambda, \Lambda' \in \Gamma \quad \Rightarrow \quad p\Lambda + (1-p)\Lambda' \in \Gamma,\ \forall p \in [0,1],$$

or equivalently

$$p \sum_{i=1}^{d} a_{si} v_i + (1-p) \sum_{i=1}^{d} a'_{si} v'_i \quad = \quad \sum_{i=1}^{d} a''_{si} v''_i,\ \forall p \in [0,1] \tag{7.7}$$

for all s and suitable $a''_{si} \in \mathbb{R}, v''_i \in \mathbb{R}^D$. This condition is not true in general. Figure 7.5 shows an example for a model with $C = 3, D = 2, d = 1, p = 0.5$. The subset Γ is non-convex because $\lambda''(1), \lambda''(2), \lambda''(3)$ are not in a linear subspace of $\mathbb{R}^2$ (*i.e.,* on a line) as required in Equation (7.7).

7.4.3 Experimental results

The proposed log-linear framework for estimating linear feature transforms is compared with standard LDA. Note that a different feature extraction (size of sliding window for LDA increased

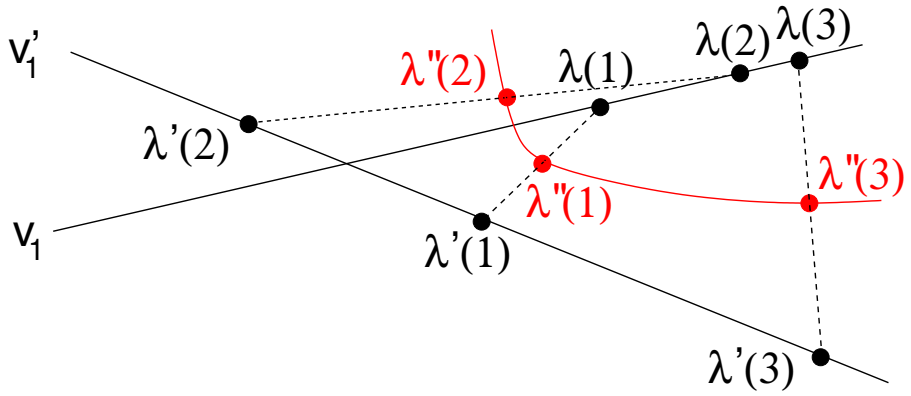


Figure 7.5: Example for non-convex subset Γ.

Table 7.8: Comparison feature transform in log-linear framework with LDA for SieTill test corpus.

	Feature transform	Acoustic model	WER [%] frame-based	lattice-based mod.
1 dns/mix	LDA	ML	3.5	
		MMI	2.7	2.5
	MMI	ML	3.5	2.8
		MMI	2.7	2.4
16 dns/mix	LDA	ML	1.9	
		MMI	1.6	1.5
	MMI	ML	1.9	1.8
		MMI	1.6	1.5

from 5 to 11) is used for the next experiments such that the results cannot be directly compared with the above mentioned results.[4]

The proposed approach is compared with standard LDA, both for ML and MMI optimized acoustic models. The matrix A consists of a projection and a rotation in the feature space. Strictly speaking, the latter is not used for the dimension reduction although it can have a substantial impact on ML optimized Gaussian models with diagonal covariance matrices. In the discriminative setting, the rotation is redundant because it can be implicitly represented by the model parameters λ_{sg}. Thus, a potential improvement is only due to the projection, *i.e.,* finding a better affine feature subspace. Table 7.8 shows the results for the non-alternating optimization approach. The alignment for frame-based MMI was taken from a conventional ML optimized single Gaussians system with globally pooled variances. The matrix A was optimized from scratch for frame-based MMI while lattice-based MMI was initialized with the LDA matrix. An example of the alternating optimization is given in Figure 7.6.

[4]Thanks to Muhammad Ali Tahir for running the experiments.

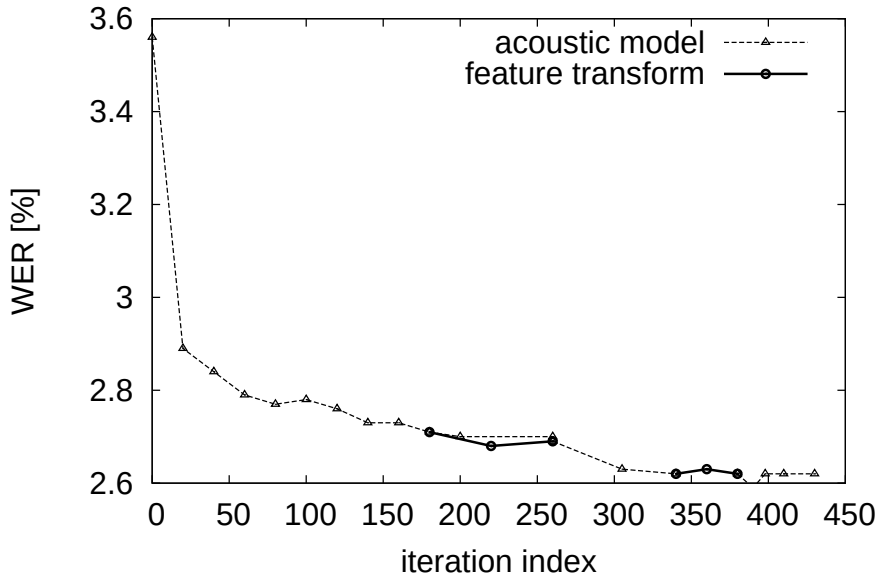


Figure 7.6: Alternating optimization: progress of word error rate (WER, [%]) *vs.* iteration index for SieTill test corpus.

7.4.4 Discussion

A new estimation algorithm for (projecting) linear feature transforms was introduced. It can be used in a preprocessing step similar to LDA or directly on the best model like fMPE [Povey & Kingsbury+ 05]. The results on a German digit string recognition task suggest that the proposed training algorithm works but does not achieve better word error rates than standard LDA.

This experimental finding needs to be confirmed on large vocabulary continuous speech recognition tasks. LDA fails [Katz & Meier+ 02, Schlüter & Zolnay+ 06] if applied to large, linearly dependent feature spaces. How does the proposed algorithm behave in this degenerate situation? Finally, this log-linear framework can and will be applied to other feature transforms, *e.g.* fMPE [Povey & Kingsbury+ 05] or speaker adaptive training [Lööf & Schlüter+ 07].

7.5 Limitations of Convex Optimization using Log-Linear Models

A couple of limitations of the above mentioned approach to convex optimization in speech recognition are discussed next. First, an obvious deficiency of the above shown approach to convex optimization in speech recognition is that the HMM state alignment is assumed to be known and kept fixed during the training. To avoid this restriction, the HMM structure might be incorporated into the feature functions of the CRF. Second, another shortcoming of the discussed approach might be that it is based on MMI, which might be not the training criterion of choice (see Section 5.2.4). In particular, the approach cannot be extended to error-based

training criteria for log-linear models. This follows directly from the next lemma.

Lemma 42. *Assume a strictly convex function* $\mathcal{F} : \mathbb{R}^D \to \mathbb{R},\ \Lambda \mapsto \mathcal{F}(\Lambda)$ *with* $\mathcal{F} \in C^1$. *Then, the function* $\mathcal{F}$ *is not bounded above.*

Proof. Due to the convexity of the function $\mathcal{F}$, the first-order approximation of $\mathcal{F}$ around any point $\Lambda_0 \in \mathbb{R}^D$ is a global underestimator [Boyd & Vandenberghe 04, pp.69], *i.e.,*

$$\mathcal{F}(\Lambda_0) + \nabla\mathcal{F}(\Lambda_0)^\top(\Lambda - \Lambda_0) \quad \leq \quad \mathcal{F}(\Lambda),\ \forall\Lambda \in \mathbb{R}^D.$$

The strict convexity of $\mathcal{F}$ guarantees that there is some $\Lambda_0 \in \mathbb{R}^D$ such that the gradient is non-zero, *i.e.,* $\nabla\mathcal{F}(\Lambda_0) \neq 0$. This implies that the underestimator is not bounded above and thus, $\mathcal{F}$ is also not bounded above. □

Error-based training criteria refer to training criteria that are bounded above and below, *e.g.* MPE but not MMI. Recall that the log-linear model parameters are unconstrained, *i.e.,* $\Lambda \in \Gamma = \mathbb{R}^D$ for some D.

Corollary 43. *No error-based training criterion* $\mathcal{F} \in C^1$ *exists for log-linear models.*[5]

Furthermore, convexity cannot be achieved by warping the training criterion in a suitable way, $g(\mathcal{F})$. This is because the warping g can only add zeroes to the gradient, as can be seen directly by applying the chain rule. These results do not imply that convex error-based training criteria do not exist. However, such a training criterion cannot be based on the log-linear parameterization. According to [Ben-David & Simon 01], error-based training *including* a margin can be solved efficiently (*i.e.,* is not NP-hard).

7.6 Summary

Convex optimization using log-linear HMMs was investigated for a digit string recognition task. Convex optimization problems both, on the sentence and the frame level, were defined. They showed good performance and stable convergence at the same time. Assuming some (good) initial state alignment, the training criterion defined on the sentence level was used to estimate successfully all model parameters from scratch. Our observation is that a carefully optimized but relatively simple setup can achieve good performance, comparable with conventional training criteria and state-of-the-art Gaussian HMMs. This might be a good starting point for adding more sophisticated features (*e.g.* higher-order features, posterior features) to refine the acoustic model. Of course, this is only a first step towards convex optimization in speech recognition. More effort needs to be spent on the incorporation of these ideas into large vocabulary speech recognition. Frame-based MMI, although tending to perform slightly worse than sentence-based MMI in general, offers a quick and robust way to setup a discriminative (baseline) model, similar to ML in case of generative models.

[5] Thanks to Simon Wiesler for the technical elaboration of this proof.

Chapter 8

Scientific Contributions

The aim of this work was to investigate log-linear techniques for string recognition. The focus was thereby on refined acoustic models in automatic speech recognition (ASR). A log-linear modeling framework for speech recognition was developed. It resulted in the following contributions that cover different aspects of a training algorithm:

Equivalence relations for Gaussian and log-linear HMMs in ASR. Conventional speech recognition systems are based on Gaussian HMMs. These are constrained models because of the Gaussian parameter constraints, the local normalization constraints of HMMs, the directed dependencies *etc.* Do these constraints reduce the flexibility of the Gaussian HMMs compared with the unconstrained log-linear HMMs? This thesis established equivalence relations for Gaussian and log-linear HMMs.
The simpler and more direct parameterization of log-linear models may be more suitable for numerical optimization and may simplify the optimization algorithms. A comprehensive experimental comparison of Gaussian and log-linear HMMs was presented in this thesis, including LVCSR tasks trained on up to 1,500h audio data. The experimental results are in a good agreement with our theoretical expectations.

Margin-based training for LVCSR (M-MMI/M-MPE). Large margin classifiers are well-studied training algorithms in statistical machine learning. This thesis presented a unifying framework to incorporate a margin term into the conventional training criteria (*e.g.* MPE), allowing for efficient margin-based training in LVCSR. The resulting training criteria for HCRFs were shown to be closely related to support vector machines (SVMs) using an appropriate loss function.
The proposed modified training criteria were used to directly evaluate the utility of the margin concept for string recognition, including examples from ASR, part-of-speech tagging, and handwriting recognition. The benefit from the additional margin term clearly depends on the training conditions. For simple tasks like for example spoken digit string recognition or handwriting recognition, overfitting is an issue, and more than $\gtrsim$75% of the total discriminative improvement is typically due to the margin term. For more complex tasks like for example LVCSR, the additional margin term is less important. Less than 25% of the total discriminative

improvement is typically due to the margin term, compared with the best state-of-the-art systems.

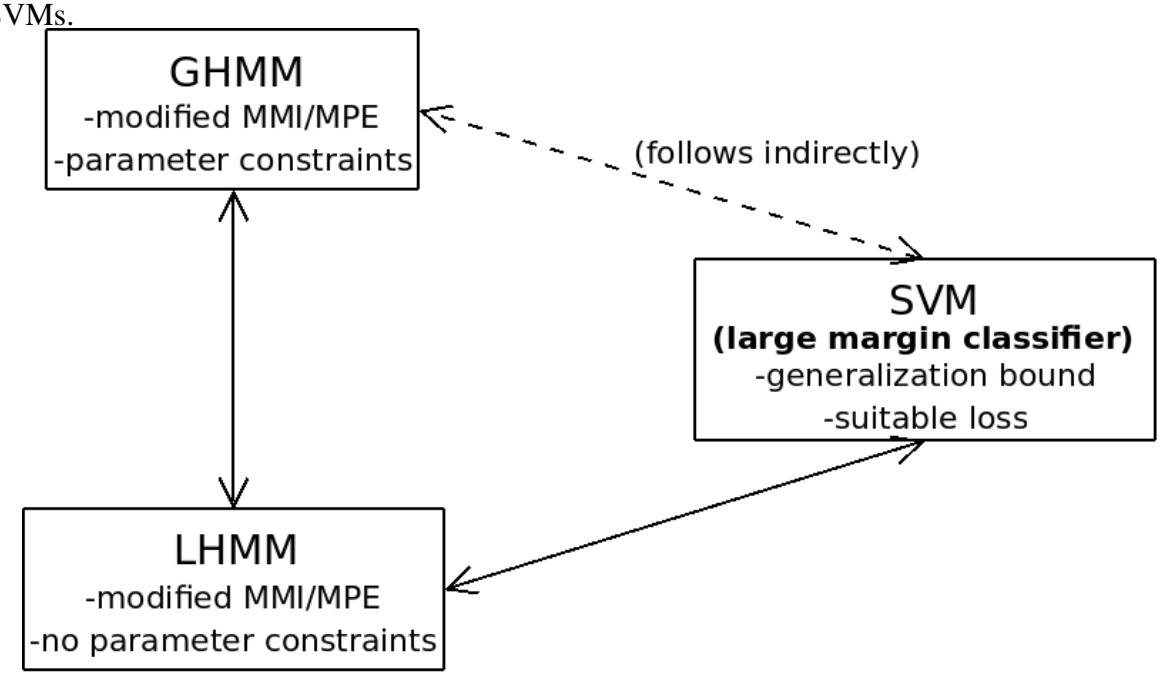
Optimization with growth transformations (hiddenGIS). The numerical optimization of the training criteria is an issue. Compared with standard optimization algorithms, growth transformations have the advantage to increase the training criterion in each iteration, to converge and to be parameter-free. An example of this concept is the generalized iterative scaling (GIS) that is used to optimize conventional CRFs using MMI. GIS does *not* apply to many extensions and variants of CRFs considered within this work. This thesis proposed an extension of GIS to hidden variables (*e.g.* HCRFs) and different training criteria (*e.g.* MPE).
The effectiveness of the proposed optimization algorithm was tested on an optical character recognition (OCR) and a digit string speech recognition task. The extension of GIS performs equally well as conventional gradient-based optimization algorithms (*e.g.* Rprop) in terms of the error rate. The experimental results suggest that the convergence is reasonably fast for bounded feature functions (OCR task) whereas it can be rather slow in case of basically unbounded feature functions (*e.g.* MFCC features in speech recognition).

Full model training from scratch using convex optimization. Conventional speech recognition systems optimize and tune the different components in several independent steps. The discriminative training, for example, is performed in a postprocessing step. In addition, the training criteria are non-convex and thus, cannot guarantee to reach the global optimum. This approach is considered suboptimal because the outcome depends on the initialization and requires much engineering work.
This thesis investigated the potential of convex training criteria with preferably no parameters to be tuned. Such training criteria would allow the model training of all model parameters from scratch in a principled way. On a digit string recognition task, competitive error rates were achieved.

A transducer-based discriminative framework. A general and flexible implementation speeds up the development of refined training algorithms (*e.g.* margin-based training) across different tasks (ASR, part-of-speech tagging, handwriting recognition). This was realized by a transducer-based approach in this work.

Gradient-based optimization of unified training criterion with expectation semiring. The unified training criterion facilitates the comparison and implementation of different training criteria. To the author's best knowledge, no efficient solution is known to compute the gradient of the unified training criterion. The gradient can be written in terms of the abstract covariance of two acyclic WFSTs. This thesis proposed an algorithm to calculate efficiently this quantity, using the standard forward/backward algorithm in combination with the expectation semiring. Besides this application, the proposed algorithm will be a nice feature of any probabilistic WFST library.
Our transducer-based discriminative framework is based on this novel algorithm. The flexible implementation facilitated greatly the development and testing of refined training criteria, *e.g.* the margin-based training criteria mentioned above.

Figure 8.1: Unified view of Gaussian HMMs (GHMMs), log-linear HMMs (LHMMs), and SVMs.



Minimum word error training with exact word error (exactMWE). In ASR, the exact word error is usually considered to be the optimum loss function for training. For efficiency reasons, error-based training criteria like for example minimum word error (MWE) use an approximate metric or N-best lists instead. How good are these approximations?
In this thesis, a transducer-based approach was chosen to compute the exact word errors on word lattices. The quality of the approximate word error was then assessed by performing MWE using the exact word errors. The experiments on a small LVCSR task suggest that the approximation to the word error in MWE is sufficiently good in the context of error-based training.

In summary, the theoretical contributions in Chapter 4 and Chapter 5 lead to a unified view of three major technologies in pattern recognition: Gaussian HMMs, log-linear HMMs, and SVMs as illustrated in Figure 8.1. This unification is not concerned with practical questions like for example how to optimize the training criteria. The investigation on convex optimization (Chapter 7) and the extensions of GIS (Chapter 6) address two issues on numerical optimization of practical relevance. The proposed transducer-based framework (Chapter 3) is sufficiently flexible and efficient to evaluate the proposed concepts and algorithms on a variety of string recognition tasks.

Carrying the above ideas to the extreme, would allow for margin-based training in speech recognition with a parameter-free optimization algorithm that guarantees to increase the training criterion in each iteration (see Section 6.4.6). In case of MMI, it converges to the global optimum.

Chapter 9

Outlook

Log-linear techniques are an emerging field in speech recognition. The present thesis can touch only a few issues. Further work needs to be done to promote log-linear techniques in speech recognition and to take full advantage of this framework. Potential questions that remain open and may serve as starting point for future research include (without any claim of being complete):

Choice of feature functions. Log-linear models are feature-based, *i.e.,* all the information comes in through the feature functions. Hence, the choice of feature functions is essential in the log-linear modeling approach. Simple features derived from the conventional features in speech recognition (*e.g.* MFCC features) were used within this work. In the future, the development of more refined features will be along two main directions [Yu & Deng+ 09, Wiesler & Nußbaum-Thom+ 09]. On the one hand, more refined generic "kernel" features need to be found. On the other hand, more flexible features modeling additional dependencies and knowledge sources need to be investigated to overcome the limitations of conventional HMM-based acoustic models [Ma & Lee 07, Heigold & Li+ 09, Zweig & Nguyen 09]. The example in Figure 9.1 suggests that human language processing may go beyond the conventional sequential modeling approach.

Fool-proof training algorithm. The outcome of existing training algorithms of HMMs depends on the initialization and many heuristics. A few convex optimization problems for HMMs have been proposed so far. All of them, however, ignore the alignment problem, *i.e.,* assume that the alignment is known beforehand and kept fixed. Can we overcome this limitation by using log-linear models and suitably defined feature functions [Schölkopf & Tsuda+ 04]?

Unsupervised discriminative training. This thesis studied the optimization of direct models in the supervised mode. In many applications (*e.g.* speaker adaptation), the unsupervised (re-) estimation of the models is required. Refined training algorithms need to be developed to make the log-linear approach competitive in practical applications, see *e.g.* [Sindhwani & Keerthi 06, Li 07].

Is hmuan lnagauge proecssnig baesd on seuqenital mdoels and m-grmas? Epmiiracl stduies sgugset taht huamn lnagugae porecssnig deos not rley on m-garms.

Figure 9.1: Is the sequential modeling approach using m-gram statistics appropriate for natural language processing?

Feature transforms. The integrated estimation of linear feature transforms was investigated in this thesis. This approach might be extended to non-linear feature transforms using the kernel trick [Schölkopf & Smola+ 99] or neural networks, for instance.

Appendix A

Corpora and Systems

This annex summarizes the information about the different corpora and systems used in this work. The corpora and systems are separated by task.

A.1 Speech Recognition

Experiments were carried out on a variety of different speech recognition corpora and systems. In contrast to most other state-of-the-art speech recognition systems found in the literature, we use a globally pooled diagonal variance matrix. This allows us to produce rather good ML baselines consisting of a fairly high number of densities. The systems are evaluated on independent test corpora for which manually transcribed reference transcriptions are available. We adopt the common word error rate (see Section 3.7 for the definition of the Levenshtein distance) for the evaluation of the speech systems.

A.1.1 Continuous digit strings

The experiments for continuous digit string recognition reported in this work have been performed on the SieTill corpus [Eisele & Haeb-Umbach+ 96] for continuously spoken German digits recorded over the telephone line from adult speakers. The vocabulary comprises the ten German digits plus the pronunciation variant 'zwo' for 'zwei'. The statistics on the corpora are summarized in Table 6.2. Male and female speakers are represented equally.

The recognition system is based on a one-pass decoder design. Details on the baseline acoustic modeling are summarized in the following. In previous work, the two gender-dependent models were optimized independently. This simplification is exact for ML models. In the discriminative framework, however, this can lead to problems in recognition because the scores from the two genders are not guaranteed to be comparable. For this reason, the two gender- independent acoustic models are optimized together (Section 7.2.2).

Acoustic modeling: SieTill corpus.

- telephone line recorded German digits;

Table A.1: Statistics for speech corpora.

Corpus		#Sentences	#Words [k]	Audio data [h]	Silence portion [%]
SieTill	Train	12,948	43	11.3	55
	Test	13,114	43	11.4	
NAB-20k/ NAB-60k	Train	37,474	642	81.4	26
	Dev	310	7	0.8	18
	Eval	316	8	0.9	18
EPPS En	Train	67,000	660	91.6	30
	Dev06	726	29	3.2	
	Eval06	742	30	3.2	
	Eval07	644	27	2.9	
BNBC Cn	Train 230h	206,000	2,200	230	13
	Train 1500h	1,300,000	15,500	1,534	
	Dev07	1,700	45	2.6	
	Eval06	1,300	37	2.2	
	Eval07	1,600	42	2.9	

- 11 whole word HMMs;
- per gender 214 states plus 1 for silence, no state tying;
- HMM segments with 2 identical emission distributions;
- Gaussian mixture densities;
- pooled diagonal covariances;
- 12 MFCC features;
- LDA on 5 adjacent input frames ($5 \times 12 = 60$ input features), which are reduced to 25 output features.

A.1.2 Read speech

In this work, American English read speech is investigated on the Wall Street Journal (WSJ) corpora. The WSJ corpora are composed of business journal texts, which are read by American journalists [Pallett & Fiscus$^+$ 93, Pallett & Fiscus$^+$ 95, Kubala 95] and recorded under clean conditions. The WSJ data has been collected by the National Institute of Standards and Technology (NIST) under the Advanced Research Projects Agency (ARPA) human technology research program.

The WSJ0 training corpus consists of approximately 15 hours of speech. In addition, the November '94 NAB training corpus consists of the 84 speakers of the above WSJ0 corpus plus the 200 additional speakers of the WSJ1 corpus, leading to a total of approximately 81

hours of speech. Recognition systems are available for vocabularies of 20k and 65k words, for which the evaluation is performed on the NAB November '94 H1 development test corpus. The development corpus is composed of approximately 49 minutes of speech from 20 speakers. The corpus statistics is given in Table A.1. The out-of-vocabulary (OOV) rate is 2.6% (NAB 20k) and 0.7% (NAB 65k) on the combined Dev/Eval corpus.

Recognition system: Nov. '94 NAB.

- vocabularies:
  - 19,978 words plus 2,434 pronunciation variants (NAB 20k),
  - 64,736 words plus 5,234 pronunciation variants (NAB 65k);
- trigram language model with perplexity (PP):
  - $PP_{Dev} = 125$, $PP_{Eval} = 137$ (NAB 20k),
  - $PP_{Dev} = 146$, $PP_{Eval} = 144$ (NAB 65k);
- 3×2-states HMMs;
- 7,000 decision tree-based triphone states plus one silence state;
- across-word acoustic model;
- mixtures with a total of 412k Gaussian densities;
- one pooled diagonal covariance;
- 16 MFCC features plus first temporal derivatives and second derivative of the energy;
- LDA on 3 adjacent input frames ($3 \times 33 = 99$ input features), which are reduced to 33 output features.

A.1.3 European Parliament plenary speech (EPPS)

This task contains recordings from the European Parliament plenary sessions (EPPS). Again, the corpus statistics can be found in Table A.1. The training is only done on the transcribed data. The Dev06, Eval06, and Eval07 corpus are made up of 41, 41, and 50 different politicians and interpreters, respectively. The experiments are evaluated on these corpora via the NIST scoring toolkit[1]. The lexicon is derived from the British English example pronunciation dictionary (BEEP). Using this dictionary, statistical grapheme-to-phoneme conversion models [Bisani & Ney 03] are trained and used to produce pronunciations for words not covered by the original lexicon [Lööf & Bisani+ 06b, Lööf & Bisani+ 06a, Lööf & Gollan+ 07].

[1] http://www.nist.gov/speech/tools/

Recognition system: EPPS En.

- vocabulary: 52k words;
- 4-gram language model: $PP_{Dev06} = 96$, $PP_{Eval06} = 106$, $PP_{Eval07} = 110$;
- 3×2-states HMMs;
- across-word acoustic model;
- 4,501 mixtures with a total of 830k Gaussian densities;
- 16 MFCC features + 1 voicing feature;
- warped with fast variant of VTLN;
- LDA on 9 adjacent input frames ($9 \times 17 = 153$ input features) which are reduced to 45 output coefficients;
- SAT/CMLLR.

A.1.4 Mandarin broadcasts

1,534h of broadcast news (BN) and broadcast conversations (BC) of speech data collected by LDC are used for the training. The corpus includes data from the Hub4 and TDT4 corpora and from the first three years of the GALE project (releases P1R1-4, P2R1-2, P3R1). For the development cycle of the system, a 230h subset of the corpus has been created. The subset contains the HUB4 corpus (30h), 100h of BN and 100h of BC from the four releases of the first year of the GALE project. Table A.1 offers detailed statistics for the corpora used. For the final systems we use the GALE 2007 development corpus (Dev07) for tuning and the GALE 2006 (Eval06) and GALE 2007 evaluation corpus (Eval07) for testing. The three corpora used are manually segmented and provided by LDC. In addition, the training transcripts were preprocessed by UW-SRI as described in [Venkataraman & Stolcke+ 04]. The NIST scoring toolkit is used for evaluation. The RWTH Mandarin LVCSR system follows a common approach for Mandarin LVCSR systems and uses word-based toneme pronunciation models [Plahl & Hoffmeister+ 08, Plahl & Hoffmeister+ 09]. The language model used in this work was kindly provided by UW and SRI. It is the pruned 4-gram language model used in the GALE 2007 summer evaluation [Hoffmeister & Plahl+ 07, Plahl & Hoffmeister+ 08, Plahl & Hoffmeister+ 09].

Recognition system: BNBC Cn 230h.

- vocabulary: 60k words;
- 4-gram language model ($PP_{Dev07} = 367$, $PP_{Eval06} = 636$);
- 3×1-states HMMs;

- across-word acoustic model;
- 4,501 mixtures with a total of 1,100k Gaussian densities;
- 16 MFCC features;
- warped with fast variant of VTLN;
- LDA on 9 adjacent input frames ($16 \times 9 = 144$ input features), which are reduced to 45 output coefficients;
- 1 tone feature including first and second derivatives;
- SAT/CMLLR.

Recognition system: BNBC Cn 1500h.

- vocabulary: 60k words;
- 4-gram language model ($PP_{Dev07} = 367$, $PP_{Eval06} = 636$);
- 3×1-states HMMs;
- across-word acoustic model;
- 4,501 mixtures with a total of 1,200k Gaussian densities;
- 16 PLP features + 1 voicing feature;
- warped with fast variant of VTLN;
- window over 9 consecutive frames;
- plus 1 tone feature including first and second derivatives;
- augmented with 32 neural network (NN) features [Hwang & Peng+ 07, Chen & Zhu+ 04, Hermansky & Ellis+ 00b];
- dimension reduction of input features ($(16+1)\times 9+3+32 = 188$) to 80 output coefficients by means of SAT/CMLLR.

A.2 Part-of-Speech Tagging

Part-of-speech tagging is the process of extracting the smallest units of meaning (concepts) out of a given input sentence. Adopting the approach in [Ramshaw & Marcus 95], part-of-speech tagging transforms a sequence of words $x_1^N = x_1, \ldots, x_N$ into a sequence of tags $c_1^N = c_1, \ldots, c_N$. The task of part-of-speech tagging is illustrated in Figure A.1. In this work, CRFs are used to implement part-of-speech tagging. The best models include lexical features considering the two nearest neighbors, bigram tagging features, capitalization features, prefix and suffix features

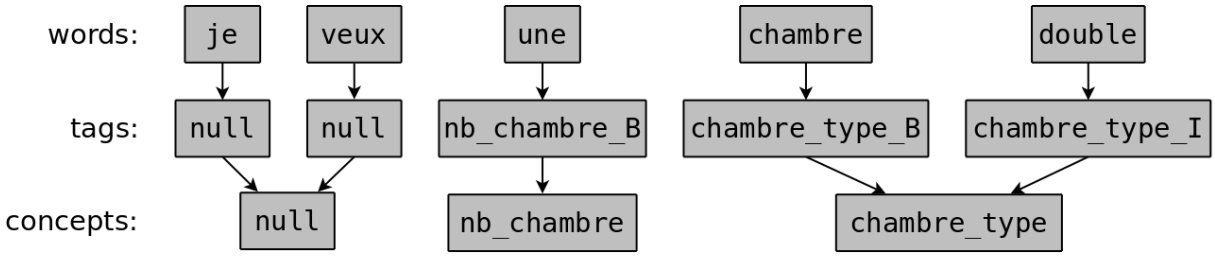


Figure A.1: The task of part-of-speech tagging.

of length four, and a sentence end feature. The experiments are evaluated on the respective development and test sets for the three corpora via the NIST scoring toolkit. As error criterion we use the well-known concept error rate (CER), which is defined as the ratio of the sum of deleted, inserted and confused conepts (not tags, see Figure A.1) and the total number of concepts in all reference strings. Substitutions, deletions and insertions are calculated using a Levenshtein-alignment between a hypothesis and a given reference concept string. NULL tokens are deleted from hypothesis and reference transcription before scoring.

A.2.1 French Media

The so-called Media corpus is a state-of-the-art corpus especially designed for the evaluation of spoken language understanding systems [Devillers & Maynard+ 04]. It covers the domain of the reservation of hotel rooms and tourist information and the incorporated concepts have been designed to match this task. There is *e.g.* a concept for a hotel name or a room type. The corpus is divided into three parts: a training set, a development set and an evaluation set. Within this corpus, modes and specifiers are also manually annotated. The experiments carried out in this thesis can be directly compared with the so-called relaxed-simplified condition within the Media/Evalda project. Here, some specifiers are dropped and thus, the resulting data is not as sparse. The corpus statistics is given in Table A.2. The best model comprises 1.7M feature functions [Hahn & Lehnen+ 08, Hahn & Lehnen+ 09].

Tagging system: French Media.

- vocabulary: 2,210 words and 99 concepts;
- lexical features, bigram concept features, word part features (capitalization, suffixes).

A.2.2 Polish

The data for the Polish corpus has been collected at the Warsaw Transportation call-center [Marasek & Gubrynowicz 08]. Also as part of the LUNA project, the manual annotation of these human-human dialogs has been performed [Mykowiecka & Marasek+ 07]. This corpus covers the domain of transportation information like *e.g.* transportation routes, itinerary, stops, or fare reductions. Three subsets have been created using the available data

Table A.2: Statistics for part-of-speech tagging corpora. The vocabulary counts refer to the number of concepts or words observed in the corpus and covered by the vocabulary.

Corpus		Data					Vocabulary	
			#Tokens		#NULL tokens			
		#Sent.	Words	Concepts	Words	Concepts	#Words	#Concepts
French	Train	12,908	94,466	43,078	32,580	11,442	2,210	99
	Dev	1,259	10,849	4,705	4,157	1,372	838	66
	Eva	3,005	25,606	11,383	9,040	2,999	1,276	78
Polish	Train	8,341	53,418	28,157	21,973	9,811	4,081	195
	Dev	2,053	13,405	7,160	5,680	2,384	2,028	157
	Eva	2,081	13,806	7,490	5,743	2,486	2,057	159

subsets. It is the first SLU database for Polish. The corpus statistics is summarized in Table A.2.

Tagging system: Polish.

- vocabulary: 4,081 words and 195 concepts;
- lexical features, bigram concept features, word part features (capitalization, prefixes, suffixes).

A.3 Handwriting Recognition

At some points in this work, complex algorithms are tested on "simple" image recognition tasks rather than on the more complex speech recognition tasks.

A.3.1 Isolated digits

The well-known United States Postal Service (USPS) handwritten digit database consists of isolated and normalized images of handwritten digits taken from US mail envelopes scaled down to 16x16 pixels. The database contains a separate training and test set with 7,291 and 2,007 images, respectively.[2] One disadvantage of the USPS corpus is that no development test set exists, resulting in the possible underestimation of error rates for all of the reported results. Note that this disadvantage holds true for almost all data sets available for image object recognition. The US Postal Service task is still one of the most widely used reference data sets for handwritten character recognition and allows fast experiments due to its small size. The test set contains a large amount of image variability and is considered to be a "hard" recognition task. Good error rates are in the range of 2-3% and use advanced modeling techniques, *e.g.*

[2]Data available from `ftp://ftp.kyb.tuebingen.mpg.de/pub/bs`

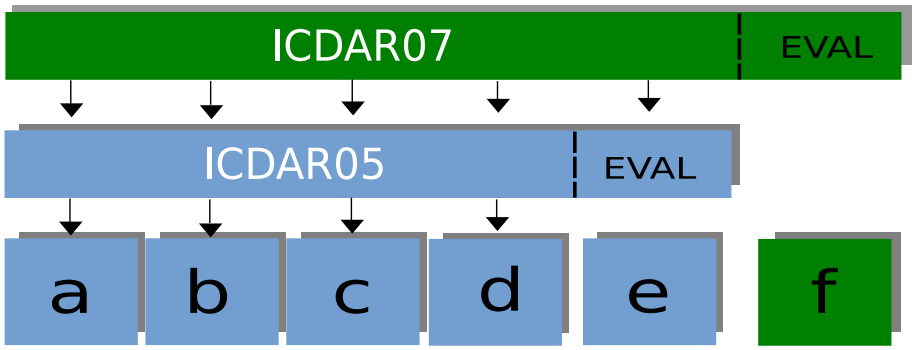


Figure A.2: IFN/ENIT corpora splits used in 2005 and 2007.

deformation models [Keysers & Deselaers+ 07].

Recognition system: USPS.

- vocabulary: 10 digits;
- gray-scale features augmented with Sobel-based derivatives, amounting to 512 features;
- 10 GMMs, each with 16 densities.

A.3.2 Isolated town names

The IFN/ENIT database [Pechwitz & Maddouri+ 02] contains Arabic handwriting. The database is divided into four training folds with an additional fold for testing [Märgner & Pechwitz+ 05]. The current database version (v2.0p1e) contains a total of 32,492 Arabic words handwritten by more than 1,000 writers, and has a vocabulary size of 937 Tunisian town names. Additionally, the submitted systems to the ICDAR 2007 competition [Märgner & Abed 07] are trained on all datasets of the IFN/ENIT database and evaluated for known datasets. Here, we follow the same evaluation protocol as for the ICDAR 2005 and 2007 competition (see Figure A.2). The corpus statistics for the different folds can be found in Table A.3.

Without any preprocessing of the input images, simple appearance-based image slice features X_t are extracted at every time step $t = 1, \ldots, T$. These features are augmented by their spatial derivatives in horizontal direction $\Delta_t = X_t - X_{t-1}$. In order to incorporate temporal and spatial context into the features, 7 consecutive features in a sliding window are concatenated, which are then reduced by a PCA transformation matrix to a feature vector x_t. A character-based lexicon is used to represent the town names [Dreuw & Heigold+ 09].

Recognition system: IFN/ENIT.

- vocabulary: 937 town names;
- appearance-based image slice features augmented with first spatial derivatives;

Table A.3: Statistics for handwriting corpora, a, b, c, d, and e are the different folds of the IFN/ENIT database.

Corpus		#Observations	
USPS		#Digits	
	train	7,291	
	test	2,007	
IFN/ENIT		#Towns	#Frames
	a	6,537	451,860
	b	6,710	459,446
	c	6,477	451,524
	d	6,735	451,466
	e	6,033	404,489

- PCA on 7 adjacent slices, projected down to 30 dimensions;
- 121 characters ("monophones") to represent town names, including silence;
- 361 HMM states modeled by 36k Gaussian densities with globally pooled variances,
- model length estimation (MLE) for character-dependent model lengths as proposed in [Dreuw & Jonas+ 08, Dreuw & Rybach+ 09].

Appendix B

Symbols and Acronyms

In this appendix, all relevant mathematical symbols and acronyms which are used in this thesis are defined for convenience. Detailed explanations are given in the corresponding chapters.

B.1 Mathematical Symbols

α_c	log-linear model parameter associated with feature function $f_\gamma(x, c) = \delta(c, \gamma)$
$A(\cdot, \cdot)$	accuracy between two strings, *e.g.* phoneme accuracy
$A(\cdot, \cdot)$	auxiliary function
c	class c
c_{sl}	mixture weight of a Gaussian distribution in a Gaussian mixture model (GMM) where s is the state and l is the index of the Gaussian distribution
$const(x)$	function that does not depend on variable x
D	dimension of acoustic feature vector
$\delta(i, j)$	Kronecker delta, equals one for $i = j$, and zero otherwise
det, $\lvert \cdot \rvert$	determinant of a matrix
$E(\cdot, \cdot)$	error between two strings, *e.g.* word error
f	smoothing function in unified training criterion
$f_i(x, c)$	feature function $f_i : \mathbb{R}^D \times \mathbb{Z} \rightarrow \mathbb{R}$ in log-linear models
$\mathcal{G}(\cdot)$	growth transformation
$\mathcal{I}$	unity matrix
l	density index of a mixture

$l(\cdot)$	loss function
λ_i	model parameter associated with feature function f_i in a log-linear model
λ	vector of log-linear model parameters, $\lambda := (\lambda_1, \dots)$
Λ	set of model parameters, *e.g.* $\Lambda = \{\lambda\}$ in case of log-linear models
μ	mean of a Gaussian distribution
$N,\ N_r,\ M$	number of words in a speech segment
$\mathcal{N}(x\|\mu;\Sigma)$	Gaussian distribution with mean vector μ and with covariance matrix Σ
$p(s_t\|s_{t-1}, W)$	first-order transition probability given the spoken word sequence W (transition model)
$p(s)$	HMM state prior
$p_t(s_t = s\|X, W)$	posterior of HMM state s at time frame t given acoustic observation vectors X and word sequence W (FB probability)
$p_t(s_t = s\|X\backslash x_t, W)$	$p_t(s_t = s\|X\backslash x_t, W) := p_t(s_t = s\|X, W)/p(x_t\|s_t)$ (context prior)
$p(w\|h)$	language model probability of a word w given the history h
$p(W\|X)$	posterior for the spoken word sequence W given the acoustic observations X
$p(W),\ p(w_1^N)$	prior for a word sequence (language model)
$p(X),\ p(x_1^T)$	probability for the acoustic observations (evidence)
$p(X\|W),\ p_\Lambda(X\|W)$	probability for the acoustic observations X given the word sequence W (acoustic model)
$p(X, S\|W)$	joint probability for the acoustic observations x_1^T and sequence of Hidden Markov Model states given the word sequence w_1^N
r	index of speech segment
R	number of speech segments
s	HMM state
$s_1^T,\ S$	HMM state sequence
σ^2	variance of a Gaussian distribution
Σ	covariance matrix of a Gaussian distribution
$t,\ \tau$	time frame index
T	number of time frames in a segment

$\top$	transpose of a vector or matrix
tr, **trace**	trace of a matrix
$w,\ v$	word indices
$w_1^N,\ v_1^M,\ W,\ V$	word sequence, *e.g.* $w_1^N = w_1 w_2 \ldots w_N$
$x_1^T,\ X$	sequence of acoustic observation vectors, *e.g.* $x_1^T = x_1, x_2, \ldots, x_T$
x	feature vector
x_t	feature vector at time frame t
$\mathcal{F}(\Lambda)$	objective function $\mathcal{F} : \mathbb{R}^{\lvert\Lambda\rvert} \mapsto \mathbb{R}$ (training criterion), to be maximized
$\circ$	composition of two WFSTs

B.2 Acronyms

ASR	**A**utomatic **S**peech **R**ecognition
CART	**C**lassification **A**nd **R**egression **T**ree
CER	**C**oncept **E**rror **R**ate
CMLLR	**C**onstrained **M**aximum **L**ikelihood **L**inear **R**egression
CRF	**C**onditional **R**andom **F**ield
EBW	**E**xtended **B**aum **Welch**
EM	**E**xpectation **M**aximization
EPPS	**E**uropean **P**arliament **P**lenary **S**essions
FB	**F**orward **B**ackward
FST	**F**inite **S**tate **T**ransducer
GALE	**G**lobal **A**utonomous **L**anguage **E**xploitation
GD	**G**radient **D**escent
GIS	**G**eneralized **I**terative **S**caling
GHMM	**G**aussian **HMM**
GMM	**G**aussian **M**ixture **M**odel
HCRF	**H**idden **CRF**
HMM	**H**idden **M**arkov **M**odel
LDA	**L**inear **D**iscriminant **A**nalysis
LHMM	**L**og-linear **HMM**
LM	**L**anguage **M**odel
LMM	**L**og-linear **M**ixture **M**odel
LUNA	spoken **L**anguage **Un**derstanding in multilingu**A**l communication systems
LVCSR	**L**arge **V**ocabulary **S**peech **R**ecognition
MBR	**M**inimum **B**ayes **R**isk
MCE	**M**inimum **C**lassification **E**rror
MFCC	**M**el **F**requency **C**epstral **C**oefficients

ML	**M**aximum **L**ikelihood
MLP	**M**ulti **L**ayer **P**erceptron
MMI	**M**aximum **M**utual **I**nformation
MPE	**M**inimum **P**honeme **E**rror
MWE	**M**inimum **W**ord **E**rror
NAB	**N**orth **A**merican **B**usiness
NIST	**N**ational **I**nstitute of **S**tandards and **T**echnology
NN	**N**eural **N**etwork
OCR	**O**ptical **C**haracter **R**ecognition
PAC	**P**robably **A**pproximately **C**orrect
PCA	**P**rincipal **C**omponent **A**nalysis
PLP	**P**erceptual **L**inear **P**rediction
PP	Language Model **P**er**P**lexity
Rprop	**R**esilient **Prop**agation
RWTH	**R**heinisch **W**estfälische **T**echnische **H**ochschule
SAT	**S**peaker **A**daptive **T**raining
SVM	**S**upport **V**ector **M**achine
TC-STAR	**T**echnology and **C**orpora for Speech to Speech Translation
TDP	**T**ime **D**istortion **P**enalty
USPS	**US P**ostal **S**ervice
VTLN	**V**ocal **T**ract **L**ength **N**ormalization
WER	**W**ord **E**rror **R**ate
WFST	**W**eighted **FST**
WSJ	**W**all **S**treet **J**ournal

Bibliography

[Abdel-Haleem 06] Y.H. Abdel-Haleem: *Conditional random fields for continuous speech recognition*. Ph.D. thesis, Faculty of Engineering, University of Sheffield, Sheffield, UK, 2006.

[Afify 05] M. Afify: Extended Baum-Welch reestimation of Gaussian mixture models based on reverse Jensen inequality. In *Interspeech*, pp. 1113 – 1116, Lisbon, Portugal, Sept. 2005.

[Allauzen & Mohri 03] C. Allauzen, M. Mohri: Efficient algorithms for testing the twins property. *Journal of Automata, Languages and Combinatorics*, Vol. 8, No. 2, 2003.

[Alleva & Huang+ 96] P. Alleva, X.D. Huang, M.Y. Hwang: Improvements on the pronunciation prefix tree search organization. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Vol. 1, pp. 133–136, Atlanta, GA, USA, May 1996.

[Altun & Tsochantaridis+ 03] Y. Altun, I. Tsochantaridis, T. Hofmann: Hidden Markov support vector machines. In *International Conference on Machine Learning (ICML)*, Washington, DC, USA, Aug. 2003.

[Anastasiadis & Magoulas+ 05] A.D. Anastasiadis, G.D. Magoulas, M.N. Vrahatis: New globally convergent training scheme based on the resilient propagation algorithm. *Neurocomputing*, Vol. 64, pp. 253 – 270, 2005.

[Anderson 82] J. Anderson: Logistic discrimination. In P. Krishnaiah, L. Kanal, editors, *Handbook of statistics 2*, pp. 169–191. North-Holland, 1982.

[Armijo 66] L. Armijo: Minimization of functions having Lipschitz continuous first derivatives. *Pacific Journal of Mathematics*, Vol. 16, No. 1-3, 1966.

[Axelrod & Goel+ 07] S. Axelrod, V. Goel, R. Gopinath, P. Olsen, K. Visweswariah: Discriminative estimation of subspace constrained Gaussian mixture models for speech recognition. *IEEE Transactions on Speech and Audio Processing*, Vol. 15, No. 1, Jan. 2007.

[Bahl & Brown+ 86] L. Bahl, P. Brown, P. de Souza, R. Mercer: Maximum mutual information estimation of hidden Markov model parameters for speech recognition. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 49–52, Tokyo, Japan, May 1986.

[Bahl & Jelinek+ 83] L.R. Bahl, F. Jelinek, R.L. Mercer: A maximum likelihood approach to continuous speech recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 5, pp. 179–190, March 1983.

[Bahl & Padmanabhan+ 96] L.R. Bahl, M. Padmanabhan, D. Nahamoo, P.S. Gopalakrishnan: Discriminative training of Gaussian mixture models for large vocabulary speech recognition systems. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 613–616, Atlanta, GA, USA, May 1996.

[Bahl & Padmanabhan 98] L.R. Bahl, M. Padmanabhan: A discriminant measure for model complexity adaptation. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 453–456, Seattle, WA, USA, May 1998.

[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. In *ASA Meeting*, Washington, DC, USA, April 1976.

[Bauer 01] J. Bauer: *Diskriminative Methoden zur automatischen Spracherkennung fr Telefon-Anwendungen*. Dissertation, Technische Universitt Mnchen, Mnchen, 2001.

[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.

[Bayes 63] T. Bayes: An essay towards solving a problem in the doctrine of chances. *Philosophical Transactions of the Royal Society of London*, Vol. 53, pp. 370–418, 1763. Reprinted in *Biometrika*, vol. 45, no. 3/4, pp. 293–315, December 1958.

[Bellman 57] R.E. Bellman: Dynamic programming. Princeton University Press, Princeton, NJ, USA, 1957.

[Ben-David & Simon 01] S. Ben-David, H. Simon: Efficient learning of linear perceptrons. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 189–195. MIT Press, Dec. 2001.

[Berger & Della Pietra+ 96] A. Berger, S. Della Pietra, V. Della Pietra: A maximum entropy approach to natural language processing. *Computational Linguistics*, Vol. 22, No. 1, pp. 39–71, 1996.

[Beulen & Ortmanns+ 99] K. Beulen, S. Ortmanns, C. Elting: Dynamic programming search techniques for across-word modeling in speech recognition. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 609–612, Phoenix, AZ, March 1999.

[Beyerlein 97] P. Beyerlein: Discriminative model combination. In *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 238 – 245, Santa Barbara, CA, Dec. 1997.

[Beyerlein 98] P. Beyerlein: Discriminative model combination. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Vol. 1, pp. 481 – 484, Seattle, WA, USA, May 1998.

[Beyerlein 00] P. Beyerlein: *Diskriminative Modellkombination in Spracherkennungssystemen mit grossem Wortschatz*. Ph.D. thesis, RWTH Aachen University, Oct. 2000.

[Bisani & Ney 03] M. Bisani, H. Ney: Multigram-based grapheme-to-phoneme conversion for LVCSR. In *Interspeech*, pp. 933–936, Geneva, Switzerland, Sept. 2003.

[Bisani & Ney 04] M. Bisani, H. Ney: Bootstrap estimates for confidence intervals in ASR performance evaluation. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 409 – 412, Montreal, Canada, May 2004.

[Bishop 06] C. Bishop: *Pattern Recognition and Machine Learning*. Springer, 2006.

[Bocchieri 93] E. Bocchieri: Vector quantization for the efficient computation of continuous density likelihoods. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 692–695, Minneapolis, MN, April 1993.

[Bottou 91] L. Bottou: *Une approche théorique de l'apprentissage connexionniste - applications à la reconnaissance de la parole*. Ph.D. thesis, Université de Paris XI, 1991.

[Bourlard & Morgan 94] H. Bourlard, N. Morgan: *Connectionist speech recognition*. Kluwer Academic Publishers, 1994.

[Boyd & Vandenberghe 04] S. Boyd, L. Vandenberghe: *Convex optimization*. Cambridge, 2004.

[Cardin & Normandin+ 93] R. Cardin, Y. Normandin, E. Millien: Inter-word coarticulation modeling and MMIE training for improved connected digit recognition. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 243–246, Minneapolis, MN, USA, April 1993.

[Chan & Woodland 04] H. Chan, P. Woodland: Improving broadcast news transcription by lightly supervised discriminative training. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Montreal, Quebec, Canada, May 2004.

[Chang & Luo+ 08] T.H. Chang, Z.Q. Luo, L. Deng, C.Y. Chi: A convex optimization method for joint mean and variance parameter estimation of large-margin CDHMM. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Las Vegas, NV, USA, April 2008.

[Chen & Rosenfeld 99] S. Chen, R. Rosenfeld: A Gaussian prior for smoothing maximum entropy models. Technical Report CMUCS-99-108, Computer Science Department, Carnegie Mellon University, 1999.

[Chen & Zhu+ 04] B. Chen, Q. Zhu, N. Morgan: Learning long-term temporal features in LVCSR using neural networks. In *Interspeech*, Jeju Island, Korea, Oct. 2004.

[Chou & Juang+ 92] W. Chou, B.H. Juang, C.H. Lee: Segmental GPD training of HMM based speech recognizer. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 473–476, San Francisco, CA, USA, March 1992.

[Chou & Lee+ 93] W. Chou, C.H. Lee, B.H. Juang: Minimum error rate training based on *N*-best string models. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 652–655, Minneapolis, MN, 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. In *International Conference on Spoken Language Processing (ICSLP)*, pp. 439–442, Yokohama, Japan, Sept. 1994.

[Chow 90] Y.L. Chow: Maximum mutual information estimation of HMM parameters for continuous speech recognition using the N-best algorithm. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 701–704, Albuquerque, NM, USA, April 1990.

[Cohn 07] T. Cohn: *Scaling conditional random fields for natural language processing*. Ph.D. thesis, Department of Computer Science and Software Engineering, University of Melbourne, 2007.

[Cover & Thomas 91] T. Cover, J. Thomas: *Elements of information theory*. Wiley, 1991.

[Darroch & Ratcliff 72] J. Darroch, D. Ratcliff: Generalized iterative scaling for log-linear models. *Annals of Mathematical Statistics*, Vol. 43, pp. 1470 – 1480, 1972.

[Davis & Mermelstein 80] S. Davis, P. Mermelstein: Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. ASSP-28, No. 4, pp. 357 – 366, Aug. 1980.

[Della Pietra & Della Pietra+ 97] S. Della Pietra, V. Della Pietra, J. Lafferty: Inducing features of random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 19, No. 4, pp. 380–393, 1997.

[Dempster & Laird+ 77] A. Dempster, N. Laird, D. Rubin: Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, Vol. 39, No. B, pp. 1 – 38, 1977.

[Devillers & Maynard+ 04] L. Devillers, H. Maynard, S. Rosset et al.: The French Media/Evalda project: The evaluation of the understanding capability of spoken language dialog systems. In *International Conference on Language Resources and Evaluation (LREC)*, Lisbon, Portugal, May 2004.

[Doddington & Przybocki+ 00] G.R. Doddington, M.A. Przybocki, A.F. Martin, D.A. Reynolds: The NIST speaker recognition evaluation – overview, methodology, systems, results, perspective. *Speech Communication*, Vol. 31, No. 2–3, pp. 225–254, June 2000.

[Doumpiotis & Byrne 04] V. Doumpiotis, W. Byrne: Pinched lattice minimum Bayes risk discriminative training for large vocabulary continuous speech recognition. In *Interspeech*, pp. 1717 – 1720, Jeju Island, Korea, Oct. 2004.

[Doumpiotis & Byrne 05] V. Doumpiotis, W. Byrne: Lattice segmentation and minimum Bayes risk discriminative training for large vocabulary continuous speech recognition. *Speech Communication*, Vol. 2, pp. 142–160, 2005.

[Dreuw & Heigold+ 09] P. Dreuw, G. Heigold, H. Ney: Confidence-based discriminative training for model adaptation in offline Arabic handwriting recognition. In *International Conference on Document Analysis and Recognition (ICDAR)*, Barcelona, Spain, July 2009.

[Dreuw & Jonas+ 08] P. Dreuw, S. Jonas, H. Ney: White-space models for offline Arabic handwriting recognition. In *International Conference on Pattern Recognition (ICPR)*, Tampa, FL, USA, Dec. 2008.

[Dreuw & Rybach+ 09] P. Dreuw, D. Rybach, C. Gollan, H. Ney: Writer adaptative training and writing variant model refinement for offline Arabic handwriting recognition. In *International Conference on Document Analysis and Recognition (ICDAR)*, Barcelona, Spain, July 2009.

[Du & Liu+ 06] J. Du, P. Liu, F. Soong, J.L. Zhou, R.H. Wang: Minimum divergence based discriminative training. In *Interspeech*, Pittsburgh, PA, USA, Sept. 2006.

[Duda & Hart+ 01] R.O. Duda, P.E. Hart, D.G. Stork: *Pattern Classification*. John Wiley & Sons, New York, NY, USA, 2001.

[Dupont & Denis+ 05] P. Dupont, F. Denis, Y. Esposito: Links between probabilistic automata and hidden Markov models: Probability distributions, learning models and induction algorithms. *Pattern Recognition*, Vol. 38, pp. 1349 – 1371, 2005.

[Eisele & Haeb-Umbach+ 96] T. Eisele, R. Haeb-Umbach, D. Langmann: A comparative study of linear feature transformation techniques for automatic speech recognition. In *International Conference on Spoken Language Processing (ICSLP)*, pp. 252–255, Philadelphia, PA, USA, Oct. 1996.

[Eisner 01] J. Eisner: Expectation semirings: Flexible EM for finite-state transducers. In *International Workshop on Finite-State Methods and Natural Language Processing (FSMNLP)*, Helsinki, Finland, Aug. 2001.

[Evermann & Chan+ 05] G. Evermann, H. Chan, M. Gales, B. Jia, D. Mrva, P. Woodland, K. Yu: Training LVCSR systems on thousands of hours of data. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Philadelphia, PA, USA, April 2005.

[Fahlman 88] S. Fahlman: An empirical study of learning speed in back-propagation networks. Technical report, Carnegie Mellon University, 1988.

[Fisher 36] R.A. Fisher: The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, Vol. 7, No. 179-188, 1936.

[Fosler-Lussier & Morris 08] E. Fosler-Lussier, J. Morris: CRANDEM systems: Conditional random field acoustic models for hidden Markov models. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Las Vegas, NV, USA, April 2008.

[Fritsch 97] J. Fritsch: ACID/HNN: A framework for hierarchical connectionist acoustic modeling. In S. Furui, B.H. Juang, W. Chou, editors, *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 164–171, Santa Barbara, CA, USA, Dec. 1997.

[Fu & Juang 08] Q. Fu, B.H. Juang: An investigation of non-uniform error cost function design in automatic speech recognition. In *International Conference on Machine Learning and Applications (ICMLA)*, San Diego, CA, USA, Dec. 2008.

[Ganapathisraju 02] A. Ganapathisraju: *Support vector machines for speech recognition*. Ph.D. thesis, Mississipi State University, 2002.

[Gauvain & Lee 94] J.L. Gauvain, C.H. Lee: Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains. *IEEE Transactions on Speech and Audio Processing*, Vol. 2, No. 2, 1994.

[Generet & Ney+ 95] M. Generet, H. Ney, F. Wessel: Extensions to absolute discounting for language modeling. In *European Conference on Speech Communication and Technology (Eurospeech)*, Vol. 2, pp. 1245–1248, Madrid, Spain, Sept. 1995.

[Gibson 08] M. Gibson: *Minimum Bayes risk acoustic model estimation and adaptation*. Ph.D. thesis, University of Sheffield, UK, 2008.

[Gibson & Hain 06] M. Gibson, T. Hain: Hypothesis spaces for minimum Bayes risk training in large vocabulary speech recognition. In *Interspeech*, Pittsburgh, PA, USA, Sept. 2006.

[Gopalakrishnan & Kanevsky+ 88] P. Gopalakrishnan, D. Kanevsky, A. Nadas, D. Nahamoo, M. Picheny: Decoder selection based on cross-entropies. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, New York, NY, USA, April 1988.

[Gopalakrishnan & Kanevsky+ 91] P. Gopalakrishnan, D. Kanevsky, A. Nadas, 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, 1991.

[Gunawardana 01] A. Gunawardana: Maximum mututal information estimation of acoustic HMM emission densities. CLSP Research Note No. 40, Center for Language and Speech Processing, Johns Hopkins University, Baltimore, MD, 2001.

[Gunawardana & Mahajan+ 05] A. Gunawardana, M. Mahajan, A. Acero, J. Platt: Hidden conditional random fields for phone classification. In *Interspeech*, pp. 117 – 120, Lisbon, Portugal, Sept. 2005.

[Häb-Umbach & Ney 92] R. Häb-Umbach, H. Ney: Linear discriminant analysis for improved large vocabulary continuous speech recognition. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Vol. 1, pp. 13 – 16, San Francisco, CA, March 1992.

[Häb-Umbach & Ney 94] R. Häb-Umbach, H. Ney: Improvements in beam search for 10000-word continuous-speech recognition. *IEEE Transactions on Speech and Audio Processing*, Vol. 2, No. 2, pp. 353–356, April 1994.

[Hahn & Lehnen+ 08] S. Hahn, P. Lehnen, C. Raymond, H. Ney: A comparison of various methods for concept tagging for spoken language understanding. In *International Conference on Language Resources and Evaluation (LREC)*, Marrakech, Morocco, May 2008.

[Hahn & Lehnen+ 09] S. Hahn, P. Lehnen, G. Heigold, H. Ney: Optimizing CRFs for SLU tasks in various languages using modified training criteria. In *Interspeech*, Brighton, England, Sept. 2009.

[Hampel 86] F. Hampel: *Robust statistics - The approach based on influence functions*. Wiley, 1986.

[Hastie & Tibshirani+ 01] T. Hastie, R. Tibshirani, J. Friedman: *The elements of statistical learning*. Springer-Verlag, 2001.

[He & Deng+ 06] X. He, L. Deng, W. Chou: A novel learning method for hidden Markov models in speech and audio processing. In *IEEE Workshop on Multimedia Signal Processing (MMSP)*, pp. 80–85, Victoria, BC, USA, Oct. 2006.

[He & Deng+ 08] X. He, L. Deng, W. Chou: Discriminative learning in sequential pattern recognition – a unifying review for optimization-oriented speech recognition. *IEEE Signal Processing Magazine*, Vol. , 2008.

[Heigold & Deselaers+ 08a] G. Heigold, T. Deselaers, R. Schlüter, H. Ney: GIS-like estimation of log-linear models with hidden variables. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 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. In *International Conference on Machine Learning (ICML)*, pp. 384–391, Helsinki, Finland, July 2008.

[Heigold & Dreuw+ 10] G. Heigold, P. Dreuw, S. Hahn, R. Schlüter, H. Ney: Margin-based discriminative training for string recognition. *IEEE Journal of Selected Topics in Signal Processing - Statistical Learning Methods for Speech and Language Processing*, Vol., pp. accepted for publication, Dec. 2010.

[Heigold & Lehnen+ 08] G. Heigold, P. Lehnen, R. Schlüter, H. Ney: On the equivalence of Gaussian and log-linear HMMs. In *Interspeech*, Brisbane, Australia, Sept. 2008.

[Heigold & Li+ 09] G. Heigold, G.Z.X. Li, , P. Nguyen: A flat direct model for speech recognition. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Taipei, Taiwan, April 2009.

[Heigold & Macherey+ 05] G. Heigold, W. Macherey, R. Schlüter, H. Ney: Minimum exact word error training. In *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, San Juan, Puerto Rico, November – December 2005.

[Heigold & Rybach+ 09] G. Heigold, D. Rybach, R. Schlüter, H. Ney: Investigations on convex optimization using log-linear HMMs for digit string recognition. In *Interspeech*, Brighton, England, Sept. 2009.

[Heigold & Schlüter+ 07] G. Heigold, R. Schlüter, H. Ney: On the equivalence of Gaussian HMM and Gaussian HMM-like hidden conditional random fields. In *Interspeech*, Antwerp, Belgium, Aug. 2007.

[Heigold & Schlüter+ 09] G. Heigold, R. Schlüter, H. Ney: Modified MPE/MMI in a transducer-based framework. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Taipei, Taiwan, April 2009.

[Heigold & Wiesler+ 10] G. Heigold, S. Wiesler, M. Nußbaum, P. Lehnen, R. Schlüter, H. Ney: Discriminative HMMs, log-linear models, and CRFs: What is the difference? In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Dallas, Texas, USA, March 2010.

[Hermansky 90] H. Hermansky: Perceptual linear predictive (PLP) analysis of speech. *Journal of the Acoustical Society of America*, Vol. 87, No. 4, pp. 1738 – 1752, June 1990.

[Hermansky & Ellis+ 00a] H. Hermansky, D. Ellis, S. Sharma: Tandem connectionist feature extraction for conventional HMM systems. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Istanbul, Turkey, June 2000.

[Hermansky & Ellis+ 00b] H. Hermansky, D. Ellis, S. Sharma: Tandem connectionist feature stream extraction for conventional HMM systems. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 1635–1638, Istanbul, Turkey, June 2000.

[Hifny & Gao 08] Y. Hifny, Y. Gao: Discriminative training using the trusted expectation maximization. In *Interspeech*, Brisbane, Australia, Sept. 2008.

[Hifny & Renals+ 05] Y. Hifny, S. Renals, N.D. Lawrence: A hybrid MaxEnt/HMM based ASR system. In *Interspeech*, pp. 3017 – 3020, Lisbon, Portugal, Sept. 2005.

[Hifny & Renals 09] Y. Hifny, S. Renals: Speech recognition using augmented conditional random fields. *IEEE Transactions on Audio, Speech and Language Processing*, Vol. 17, No. 2, pp. 354 – 365, 2009.

[Hoffmeister & Klein+ 06] B. Hoffmeister, T. Klein, R. Schlüter, H. Ney: Frame based system combination and a comparison with weighted ROVER and CNC. In *Interspeech*, pp. 537–540, Pittsburgh, PA, USA, Sept. 2006.

[Hoffmeister & Plahl+ 07] B. Hoffmeister, C. Plahl, P. Fritz, G. Heigold, J. Lööf, R. Schlüter, H. Ney: Development of the 2007 RWTH Mandarin LVCSR system. In *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, Kyoto, Japan, Dec. 2007.

[Hon & Lee 91] H.W. Hon, K.F. Lee: Recent progress in robust vocabulary-independent speech recognition. In *DARPA Speech and Natural Language Processing Workshop*, pp. 258–263, Pacific Grove, Feb. 1991.

[Hsiao & Tam+ 09] R. Hsiao, Y.C. Tam, T. Schultz: Generalized Baum-Welch algorithm for discriminative training on large vocabulary continuous speech recognition systems. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Taipei, Taiwan, April 2009.

[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.

[Huber 81] P. Huber: *Robust statistics*. Wiley, 1981.

[Hwang & Peng+ 07] M.Y. Hwang, G. Peng, W. Wang, A. Faria, A. Heidel, M. Ostendorf: Building a highly accurate Mandarin speech recognizer. In *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 490–495, Kyoto, Japan, Dec. 2007.

[Jaakkola & Meila+ 99] T. Jaakkola, M. Meila, T. Jebara: Maximum entropy discrimination. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 470 – 476, Denver, CO, USA, November – December 1999.

[Jaynes 03] E. Jaynes: *Probability Theory: The Logic of Science*. Cambridge, 2003.

[Jebara 02] T. Jebara: *Discriminative, generative, and imitative learning*. Ph.D. thesis, Massachusetts Institute of Technology, 2002.

[Jelinek 69] F. Jelinek: A fast sequential decoding algorithm using a stack. *IBM Journal of Research and Development*, Vol. 13, pp. 675–685, Nov. 1969.

[Jelinek 76] F. Jelinek: Continuous speech recognition by statistical methods. *Proceedings of the IEEE*, Vol. 64, No. 10, pp. 532–556, April 1976.

[Jiang & Li 07] H. Jiang, X. Li: Incorporating training errors for large margin HMMs under semi-definite programming framework. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Honolulu, HI, USA, April 2007.

[Jordan & Jacobs 94] M. Jordan, R. Jacobs: Hierarchical mixtures of experts and the EM algorithm. *Neural Computation*, Vol. 6, pp. 181–214, 1994.

[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, 1992.

[Kaiser & Horvat+ 00] J. Kaiser, B. Horvat, Z. Kacic: A novel loss function for the overall risk criterion based discriminative training of HMM models. In *Interspeech*, Vol. 2, pp. 887 – 890, Bejing, China, Oct. 2000.

[Kaiser & Horvat+ 02] J. Kaiser, B. Horvat, Z. Kacic: Overall risk criterion estimation of hidden Markov model parameters. *Speech Communication*, Vol. 38, pp. 383–398, 2002.

[Kanevsky 04] D. Kanevsky: Extended Baum Welch transformations for general functions. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 821–824, Montreal, Quebec, Canada, May 2004.

[Kanthak & Ney 04] S. Kanthak, H. Ney: FSA: An efficient and flexible C++ toolkit for finite state automata using on-demand computation. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 510 – 517, Barcelona, Spain, July 2004.

[Kanthak & Schütz+ 00] S. Kanthak, K. Schütz, H. Ney: Using SIMD instructions for fast likelihood calculation in LVCSR. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 1531–1534, Istanbul, Turkey, June 2000.

[Kapadia & Valtchev+ 93] S. Kapadia, V. Valtchev, S.J. Young: MMI training for continuous phoneme recognition on the TIMIT database. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 491–494, Minneapolis, MN, 1993, April 1993.

[Katagiri & Juang+ 98] S. Katagiri, B.H. Juang, C.H. Lee: Pattern recognition using a family of design algorithms based upon the generalized probabilistic descent method. *Proceedings of the IEEE*, Vol. 86, No. 11, pp. 2345 – 2373, 1998.

[Katz 87] S.M. Katz: Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE Transactions on Speech and Audio Processing*, Vol. 35, pp. 400–401, March 1987.

[Katz & Meier+ 02] M. Katz, H.G. Meier, H. Dolfing, D. Klakow: Robustness of linear discriminant analysis in automatic speech recognition. In *International Conference on Pattern Recognition*, Vol. 3, pp. 30371 – 30374, Québec, Canada, Aug. 2002.

[Kershaw & Robinson+ 96] D. Kershaw, T. Robinson, M. Hochberg: Context-dependent classes in a hybrid recurrent network-HMM speech recognition system. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 750 – 756, Denver, CO, USA, Nov. 1996.

[Keysers & Deselaers+ 07] D. Keysers, T. Deselaers, C. Gollan, H. Ney: Deformation models for image recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 29, No. 8, 2007.

[Kingsbury 09] B. Kingsbury: Lattice-based optimization of sequence classification criteria for neural-network acoustic modeling. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 3761–3764, Taipei, Taiwan, April 2009.

[Kubala 95] F. Kubala: Design of the 1994 CSR benchmark tests. In *ARPA Human Language Technology Workshop*, pp. 41–46, Austin, TX, USA, Jan. 1995.

[Kumar & Andreou 98] 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.

[Kuo & Gao 06] H.K.J. Kuo, Y. Gao: Maximum entropy direct models for speech recognition. *IEEE Transactions on Audio, Speech and Language Processing*, Vol. 14, No. 3, pp. 873 – 881, 2006.

[Kuo & Zweig+ 07] H. Kuo, G. Zweig, B. Kingsbury: Discriminative training of decoding graphs for large vocabulary continuous speech recognition. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Honolulu, HI, USA, April 2007.

[Lafferty & McCallum+ 01] J. Lafferty, A. McCallum, F. Pereira: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *International Conference on Machine Learning (ICML)*, pp. 282 – 289, San Francisco, CA, USA, June – July 2001.

[Lauritzen 96] S. Lauritzen: *Graphical models*. Oxford University Press Inc., 1996.

[Lauritzen & Dawid+ 90] S. Lauritzen, A. Dawid, B. Larsen, H.G. Leimer: Independence properties of directed Markov fields. *NETWORKS*, Vol. 20, pp. 491–505, 1990.

[Layton & Gales 06] M. Layton, M. Gales: Augmented statistical models for speech recognition. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Toulouse, France, May 2006.

[Layton & Gales 07] M. Layton, M. Gales: Acoustic modelling using continuous rational kernels. *Journal of VLSI Signal Processing Systems for Signal, Image, and Video Technology*, Vol. 48, No. 1-2, pp. 67–82, 2007.

[Levenshtein 66] V.I. Levenshtein: Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics - Doklay*, Vol. 10, No. 10, pp. 707 – 710, 1966.

[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.

[Li 07] X. Li: *Regularized Adaptation: Theory, Algorithms and Applications*. Ph.D. thesis, University of Washington, 2007.

[Li & Eisner 09] Z. Li, J. Eisner: First- and second-order expectation semirings with applications to minimum-risk training on translation forests. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Singapore, Aug. 2009.

[Li & Jiang 06] X. Li, H. Jiang: Solving large margin estimation of HMMs via semidefinite programming. In *Interspeech*, Pittsburgh, PA, Sept. 2006.

[Li & Yan+ 07] J. Li, Z. Yan, C.H. Lee, R. Wang: A study on soft margin estimation for LVCSR. In *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, Kyoto, Japan, Dec. 2007.

[Li & Yan+ 08] J. Li, Z.J. Yan, C.H. Lee, R.H. Wang: Soft margin estimation with various separation levels for LVCSR. In *Interspeech*, Brisbane, Australia, Sept. 2008.

[Li & Yuan+ 06] J. Li, M. Yuan, C.H. Lee: Soft margin estimation of hidden Markov model parameters. In *Interspeech*, Pittsburgh, PA, Sept. 2006.

[Likhododev & Gao 02] A. Likhododev, Y. Gao: Direct models for phoneme recognition. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Orland, FL, USA, May 2002.

[Lin & Yvon 05] S.S. Lin, F. Yvon: Discriminative training of finite state decoding graphs. In *Interspeech*, Lisbon, Portugal, Sept. 2005.

[Liu & Jiang+ 05] C. Liu, H. Jiang, X. Li: Discriminative training of CDHMMs for maximum relative separation margin. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 101 – 104, Philadelphia, PA, USA, April 2005.

[Lööf & Bisani+ 06a] J. Lööf, M. Bisani, C. Gollan, G. Heigold, B. Hoffmeister, C. Plahl, R. R. Schlüter, H. Ney: The 2006 RWTH parliamentary speeches transcription system. In *TC-STAR Workshop on Speech-to-Speech Translation*, pp. 133–138, Barcelona, Spain, June 2006.

[Lööf & Bisani+ 06b] J. Lööf, M. Bisani, C. Gollan, G. Heigold, B. Hoffmeister, C. Plahl, R. Schlüter, H. Ney: The 2006 RWTH parliamentary speeches transcription system. In *Interspeech*, pp. 105 – 108, Pittsburgh, PA, Sept. 2006.

[Lööf & Gollan+ 07] J. Lööf, C. Gollan, S. Hahn, G. Heigold, B. Hoffmeister, C. Plahl, D. Rybach, R. Schlüter, H. Ney: The RWTH 2007 TC-STAR evaluation system for European English and Spanish. In *Interspeech*, Antwerp, Belgium, Aug. 2007.

[Lööf & Schlüter+ 07] J. Lööf, R. Schlüter, H. Ney: Efficient estimation of speaker-specific projecting feature transforms. In *Interspeech*, Antwerp, Belgium, Aug. 2007.

[Lowerre 76] B. Lowerre: *A Comparative Performance Analysis of Speech Understanding Systems*. Ph.D. thesis, Carnegie Mellon University, Pittsburgh, PA, 1976.

[Ma & Lee 07] C. Ma, C.H. Lee: A study on word detector design and knowledge-based pruning and rescoring. In *Interspeech*, Antwerp, Belgium, Aug. 2007.

[Macherey 10] W. Macherey: *Discriminative training and acoustic modeling for automatic speech recognition*. Ph.D. thesis, RWTH Aachen University, 2010.

[Macherey & Haferkamp+ 05] W. Macherey, L. Haferkamp, R. Schlüter, H. Ney: Investigations on error minimizing training criteria for discriminative training in automatic speech recognition. In *Interspeech*, pp. 2133–2136, Lisbon, Portugal, Sept. 2005.

[Macherey & Ney 03] W. Macherey, H. Ney: A comparative study on maximum entropy and discriminative training for acoustic modeling in automatic speech recognition. In *Interspeech*, pp. 493 – 496, Geneva, Switzerland, Sept. 2003.

[Macherey & Och+ 08] W. Macherey, F. Och, I. Thayer, J. Uszkoreit: Lattice-based minimum error rate training for statistical machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Waikiki, Honolulu, HI, USA, Aug. 2008.

[Macherey & Schlüter+ 04] W. Macherey, R. Schlüter, H. Ney: Discriminative training with tied covariance matrices. In *Interspeech*, pp. 681 – 684, Jeju Island, Korea, Oct. 2004.

[Mahajan & Gunawardana+ 06] M. Mahajan, A. Gunawardana, A. Acero: Training algorithms for hidden conditional random fields. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Toulouse, France, May 2006.

[Malouf 02] R. Malouf: A comparison of algorithms for maximum entropy parameter estimation. In *Conference on Natural Language Learning (CoNLL)*, pp. 49–55, August – September 2002.

[Marasek & Gubrynowicz 08] K. Marasek, R. Gubrynowicz: Design and data collection for spoken Polish dialogs database. In *International Conference on Language Resources and Evaluation (LREC)*, Marrakech, Morocco, May 2008.

[Märgner & Abed 07] V. Märgner, H. Abed: ICDAR 2007 Arabic handwriting recognition competition. In *International Conference on Document Analysis and Recognition (ICDAR)*, pp. 1274–1278, Curitiba, Brazil, Sept. 2007.

[Märgner & Pechwitz+ 05] V. Märgner, M. Pechwitz, H. Abed: ICDAR 2005 Arabic handwriting recognition competition. In *International Conference on Document Analysis and Recognition (ICDAR)*, pp. 70–74, Seoul, Korea, Aug. 2005.

[Matusov & Kanthak+ 05] E. Matusov, S. Kanthak, H. Ney: On the integration of speech recognition and statistical machine translation. In *Interspeech*, pp. 3177–3180, Sept. 2005.

[McCallum & Freitag+ 00] A. McCallum, D. Freitag, F. Pereira: Maximum entropy Markov models for information extraction and segmentation. In *International Conference on Machine Learning (ICML)*, Stanford, CA, USA, June 2000.

[McDermott & Hazen+ 07] E. McDermott, T. Hazen, J.L. Roux, A. Nakamura, S. Katagiri: Discriminative training for large vocabulary speech recognition using minimum classification error. *IEEE Transactions on Audio, Speech and Language Processing*, Vol. 15, No. 1, pp. 203 – 223, 2007.

[McDermott & Katagiri 97] E. McDermott, S. Katagiri: String-level MCE for continuous phoneme recognition. In *European Conference on Speech Communication and Technology*, pp. 123–126, Rhodes, Greece, Sept. 1997.

[McDermott & Katagiri 05] E. McDermott, S. Katagiri: Minimum classification error for large scale speech recognition tasks using weighted finite state transducers. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Philadelphia, PA, USA, April 2005.

[McDermott & Nakamura 08] E. McDermott, A. Nakamura: Flexible discriminative training based on equal error group scores obtained from an error-indexed forward-backward algorithm. In *Interspeech*, Brisbane, Australia, Sept. 2008.

[McDermott & Watanabe+ 09] E. McDermott, S. Watanabe, A. Nakamura: Margin-space integration of MPE loss via differencing of MMI functionals for generalized error-weighted discriminative training. In *Interspeech*, Brighton, England, Sept. 2009.

[Merialdo 88] B. Merialdo: Phonetic recognition using hidden Markov models and maximum mutual information. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 111–114, New York, NY, USA, April 1988.

[Minka 03] T. Minka: A comparison of numerical optimizers for logistic regression. Technical report, Microsoft Research, Cambridge, UK, 2003.

[Mohri 01] M. Mohri: Generic epsilon-removal algorithm for weighted automata. In S. Yu, A. Paun, editors, *International Conference on Automata (CIAA)*, Vol. 2088 of *Lecture Notes in Computer Science*, pp. 230 – 242, London Ontaria, Canada, July 2001. Springer-Verlag, Berlin-NY.

[Mohri 03] M. Mohri: Edit-distance of weighted automata: General definitions and algorithms. *International Journal of Foundations of Computer Science*, Vol. 14, No. 6, pp. 957 – 982, 2003.

[Mohri 04] M. Mohri: *Weighted finite-state transducer algorithms: An overview*. in Carlos Martín-Vide, Victor Mitrana, and Gheorghe Paun, editors, Formal Languages and Applications, Springer, Berlin, 2004.

[Mohri 09] M. Mohri: Weighted automata algorithms. In M. Droste, W. Kuich, H. Vogler, editors, *Handbook of weighted automata*, pp. 213–254. Springer, 2009.

[Mohri & Pereira+ 00a] M. Mohri, F. Pereira, M. Riley: The design principles of a weighted finite-state transducer library. *Theoretical Computer Science*, Vol. 231, No. 1, pp. 17–32, Jan. 2000.

[Mohri & Pereira+ 00b] M. Mohri, F. Pereira, M. Riley: Weighted finite-state transducers in speech recognition. In *ISCA Tutorial and Research Workshop, Automatic Speech Recognition: Challenges for the new Millenium (ASR2000)*, Paris, France, Sept. 2000.

[Mohri & Riley 97] M. Mohri, M. Riley: Weighted determinization and minimization for large vocabulary speech recognition. In *European Conference on Speech Communication and Technology (Eurospeech)*, Rhodes, Greece, Sept. 1997.

[Molau 03] S. Molau: *Normalization in the Acoustic Feature Space for Improved Speech Recognition*. Ph.D. thesis, RWTH Aachen, Aachen, Germany, 2003.

[Morris & Fosler-Lussier 09] J. Morris, E. Fosler-Lussier: CRANDEM: conditional random fields for word recognition. In *Interspeech*, pp. 3063 – 3066, Brighton, England, Sept. 2009.

[Mykowiecka & Marasek+ 07] A. Mykowiecka, K. Marasek, M. Marciniak, J. Rabiega-Wiśniewska, R. Gubrynowicz: Annotation of Polish spoken dialogs in LUNA project. In *Language & Technology Conference: Human Language Technologies as a Challenge for Computer Science and Linguistics (LTC)*, Poznan, Poland, Oct. 2007.

[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 Transactions on Acoustics, Speech, and Signal Processing*, Vol. assp-31, pp. 814 – 817, Aug. 1983.

[Nádas & Nahamoo+ 88] A. Nádas, D. Nahamoo, M. Picheny: On a model-robust training method for speech recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. 36, pp. 1432 – 1436, 1988.

[Nakamura & McDermott+ 09] A. Nakamura, E. McDermott, S. Watanabe, S. Katagiri: A unified view for discriminative objective functions based on negative exponential of difference measure between strings. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Taipei, Taiwan, April 2009.

[Ney 84] H. Ney: The use of a one-stage dynamic programming algorithm for connected word recognition. *IEEE Transactions 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. In L. Torres, E. Masgrau, M.A. Lagunas, editors, *Signal Processing V: Theories and Applications, Fifth European Signal Processing Conference*, pp. 65–72. Elsevier Science Publishers B. V., Barcelona, Spain, 1990.

[Ney 09] H. Ney: Selected topics in human language technology and pattern recognition. Technical report, RWTH Aachen University, Aachen, Germany, 2009. Lecture script.

[Ney & Aubert 94] H. Ney, X. Aubert: A word graph algorithm for large vocabulary continuous speech recognition. In *International Conference on Spoken Language Processing (ICSLP)*, Vol. 3, pp. 1355–1358, Yokohama, Japan, Sept. 1994.

[Ney & Essen+ 94] H. Ney, U. Essen, R. Kneser: On structuring probabilistic dependencies in language modeling. *Computer Speech and Language*, Vol. 2, No. 8, pp. 1–38, 1994.

[Ney & Häb-Umbach+ 92] H. Ney, R. Häb-Umbach, B.H. Tran, M. Oerder: Improvements in beam search for 10000-word continuous speech recognition. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Vol. 1, pp. 9–12, San Francisco, CA, March 1992.

[Ney & Martin+ 97] H. Ney, S.C. Martin, F. Wessel: Statistical language modeling using leaving-one-out. In S. Young, G. Bloothooft, editors, *Corpus Based Methods in Language and Speech Processing*, pp. 1–26. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1997.

[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. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 833–836, Dallas, TX, USA, April 1987.

[Ng & Jordan 02] A. Ng, M. Jordan: On discriminative vs. generative classifiers: A comparison of logistic regression and naive Bayes. In *Advances in Neural Information Processing Systems (NIPS)*, Dec. 2002.

[Nocedal & Wright 99] J. Nocedal, S. Wright: *Numerical Optimization*. Springer, 1999.

[Nopuswanchai & Povey 03] R. Nopuswanchai, D. Povey: Discriminative training for hmm-based offline handwritten character recognition. In *International Conference on Document Analysis and Recognition (ICDAR)*, Edinburgh, Scotland, Aug. 2003.

[Normandin 91] Y. Normandin: *Hidden Markov Models, Maximum Mutual Information, and the Speech Recognition Problem*. Ph.D. thesis, McGill University, Montreal, Canada, 1991.

[Normandin 96] Y. Normandin: Maximum mutual information estimation of hidden Markov models. In K.K.P. C.-H. Lee, F. K. Soong, editor, *Automatic Speech and Speaker Recognition*, pp. 57–81. Kluwer Academic Publishers, Norwell, MA, USA, 1996.

[Normandin & Cardin$^+$ 94] Y. Normandin, R. Cardin, R.D. Mori: High–performance connected digit recognition using maximum mutual information estimation. *IEEE Transactions on Speech and Audio Processing*, Vol. 2, No. 2, pp. 299–311, 1994.

[Normandin & Lacouture$^+$ 94] Y. Normandin, R. Lacouture, R. Cardin: MMIE training for large vocabulary continuous speech recognition. In *International Conference on Spoken Language Processing*, pp. 1367–1370, Yokohama, Japan, Sept. 1994.

[Normandin & Morgera 91] Y. Normandin, S. Morgera: An improved MMIE training algorithm for speaker-independent, small vocabulary, continuous speech recognition. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 537–540, Toronto, Canada, May 1991.

[Och 03] F. Och: Minimum error rate training in statistical machine translation. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, Sapporo, Japan, July 2003.

[Och & Ney 02] F. Och, H. Ney: Discriminative training and maximum entropy models for statistical machine translation. In *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. In *ARPA Spoken Language Technology Workshop*, pp. 405–410, Plainsboro, NJ, March 1994.

[Omar & Hasegawa-Johnson 03] M.K. Omar, M. Hasegawa-Johnson: Maximum conditional mutual information projection for speech recognition. In *Interspeech*, Geneva, Switzerland, Sept. 2003.

[Ortmanns 98] S. Ortmanns: *Effiziente Suchverfahren zur Erkennung kontinuierlich gesprochener Sprache*. Ph.D. thesis, RWTH Aachen, Aachen, Germany, Nov. 1998.

[Ortmanns & Ney 95] S. Ortmanns, H. Ney: An experimental study of the search space for 20000-word speech recognition. In *European Conference on Speech Communication and Technology (Eurospeech)*, Vol. 2, pp. 901–904, Madrid, Spain, Sept. 1995.

[Ortmanns & Ney$^+$ 96] S. Ortmanns, H. Ney, A. Eiden: Language-model look-ahead for large vocabulary speech recognition. In *International Conference on Spoken Language Processing (ICSLP)*, Vol. 4, pp. 2095–2098, Philadelphia, PA, Oct. 1996.

[Ortmanns & Ney$^+$ 97a] 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.

[Ortmanns & Ney+ 97b] S. Ortmanns, H. Ney, T. Firzlaff: Fast likelihood computation methods for continuous mixture densities in large vocabulary speech recognition. In *European Conference on Speech Communication and Technology (Eurospeech)*, Vol. 1, pp. 139–142, Rhodes, Greece, Sept. 1997.

[Ostrowski 60] A. Ostrowski: *Solution of equations and systems of equations*. Academic Press, New York, 1960.

[Pallett & Fiscus+ 93] D.S. Pallett, J.G. Fiscus, W.M. Fisher, J.S. Garofolo: Benchmark tests for the DARPA spoken language program. In *ARPA Human Language Technology Workshop*, pp. 7–18, Princeton, NJ, USA, March 1993.

[Pallett & Fiscus+ 95] D.S. Pallett, J.G. Fiscus, W.M. Fisher, J.S. Garofolo, B.A. Lund, M.A. Przybocki: 1994 benchmark test for the ARPA spoken language program. In *ARPA Human Language Technology Workshop*, pp. 5–36, Austin, TX, USA, Jan. 1995.

[Paul 91] D.B. Paul: Algorithms for an optimal A^* search and linearizing the search in the stack decoder. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Vol. 1, pp. 693–696, Toronto, Canada, May 1991.

[Pechwitz & Maddouri+ 02] M. Pechwitz, S.S. Maddouri, V. Mägner, N. Ellouze, H. Amiri: ICDAR 2007 Arabic handwriting recognition competition. In *Colloque International Francophone sur l'Ecrit et le Document (CIFED)*, Hammamet, Tunis, Oct. 2002.

[Pitz 05] M. Pitz: *Investigations on Linear Transformations for Speaker Adaptation and Normalization*. Ph.D. thesis, RWTH Aachen University, 2005.

[Plahl & Hoffmeister+ 08] C. Plahl, B. Hoffmeister, M.Y.H.D. Lu, G. Heigold, J. Lööf, R. Schlüter, H. Ney: Recent improvements of the RWTH GALE Mandarin LVCSR system. In *Interspeech*, pp. 2426–2429, Brisbane, Australia, Sept. 2008.

[Plahl & Hoffmeister+ 09] C. Plahl, B. Hoffmeister, G. Heigold, J. Lööf, R. Schlüter, H. Ney: Development of the GALE 2008 Mandarin LVCSR system. In *Interspeech*, Brighton, England, Sept. 2009.

[Povey 04] D. Povey: *Discriminative Training for Large Vocabulary Speech Recognition*. Ph.D. thesis, Cambridge, England, 2004.

[Povey & Gales+ 03] D. Povey, M. Gales, P. Woodland: Discriminative MAP for acoustic model adaptation. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Hong Kong, China, April 2003.

[Povey & Kanevsky+ 08] D. Povey, D. Kanevsky, B. Kingsbury, B. Ramabhadran, G. Saon, K. Visweswariah: Boosted MMI for model and feature-space discriminative training. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Las Vegas, NV, USA, April 2008.

[Povey & Kingsbury+ 05] D. Povey, B. Kingsbury, L. Mangu, G. Saon, H. Soltau, G. Zweig: fMPE: Discriminatively trained features for speech recognition. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Philadelphia, PA, May 2005.

[Povey & Kingsbury 07] D. Povey, B. Kingsbury: Evaluation of proposed modifications to MPE for large scale discriminative training. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Honolulu, HI, USA, April 2007.

[Povey & Woodland 99] D. Povey, P. Woodland: Frame discrimination training for HMMs for large vocabulary speech recognition. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 333 – 336, Phoenix, AZ, March 1999.

[Povey & Woodland 00] D. Povey, P. Woodland: Frame discrimination training of HMMs for large vocabulary speech recognition. In *CUED/F-INFENG/TR332*, Cambridge, UK, May 2000.

[Povey & Woodland 02] D. Povey, P.C. Woodland: Minimum phone error and I-smoothing for improved discriminative training. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Vol. 1, pp. 105 – 108, Orlando, FL, May 2002.

[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 ASSP Magazine*, Vol. 3, No. 1, pp. 4–16, 1986.

[Rabiner & Juang 97] L.R. Rabiner, B.H. Juang: *Fundamentals of Speech Recognition*. Prentice-Hall Signal Processing Series, Englewood Cliffs, NJ, 1997.

[Rabiner & Schafer 78] L.R. Rabiner, R.W. Schafer: *Digital Processing of Speech Signals*. Prentice-Hall, Englewood Cliffs, NJ, 1978.

[Ramasubramansian & Paliwal 92] V. Ramasubramansian, K.K. Paliwal: Fast k-dimensional tree algorithms for nearest neighbor search with application to vector quantization encoding. *IEEE Transactions on Speech and Audio Processing*, Vol. 40, No. 3, pp. 518–528, March 1992.

[Ramshaw & Marcus 95] L. Ramshaw, M. Marcus: Text chunking using transformation-based learning. In *Proceedings of the 3rd Workshop on Very Large Corpora*, pp. 84–94, Cambridge, MA, USA, June 1995.

[Rao & Rao 98] C. Rao, M. Rao: *Matrix algebra and its applications to statistics and econometrics*. Word Scientific, 1998.

[Reichl & Ruske 95] W. Reichl, G. Ruske: Discriminative training for continuous speech recognition. In *European Conference on Speech Communication and Technology*, pp. 537–540, Madrid, Spain, Sept. 1995.

[Riedmiller & Braun 93] M. Riedmiller, H. Braun: A direct adaptive method for faster backpropagation learning: The Rprop algorithm. In *IEEE International Conference on Neural Networks (ICNN)*, pp. 586 – 591, San Francisco, CA, USA, March – April 1993.

[Riezler 98] S. Riezler: *Probabilistic Constraint Logic Programming*. Ph.D. thesis, Universität Tübingen, Germany, 1998.

[Riezler & Kuhn+ 00] S. Riezler, J. Kuhn, D. Prescher, M. Johnson: Lexicalized stochastic modeling of constraint-based grammars using log-linear measures and EM training. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 480–487, Hong Kong, Oct. 2000.

[Rigoll & Willett 98] G. Rigoll, D. Willett: A NN/HMM hybrid for continuous speech recognition with a discriminant nonlinear feature extraction. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 9–12, Seattle, WA, USA, May 1998.

[Ristad & Yianilos 98a] E.S. Ristad, P.N. Yianilos: Learning string edit distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 20, No. 5, pp. 522 – 532, 1998.

[Ristad & Yianilos 98b] E.S. Ristad, P.N. Yianilos: Towards EM-style algorithms for *a posteriori* optimization of normal mixtures. In *IEEE Symposium on Information Theory*, Aug. 1998.

[Robinson & Fallside 91] T. Robinson, F. Fallside: A recurrent error propagation network speech recognition system. *Computer Speech and Language*, Vol. 5, No. 3, pp. 259–274, 1991.

[Robinson & Hochberg+ 96] T. Robinson, M. Hochberg, S. Renals: The use of recurrent networks in continuous speech recognition. In K.K.P. C.-H. Lee, F. K. Soong, editor, *Automatic Speech and Speaker Recognition*, pp. 233–258. Kluwer Academic Publishers, Norwell, MA, USA, 1996.

[Rosenfeld 94] R. Rosenfeld: *Adaptive Statistical Language Modeling: A Maximum Entropy Approach*. Ph.D. thesis, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA, 1994.

[Rybach & Gollan+ 09] D. Rybach, C. Gollan, G. Heigold, B. Hoffmeister, J. Lööf, R. Schlüter, H. Ney: The RWTH Aachen University open source speech recognition system. In *Interspeech*, Brighton, UK, Sept. 2009.

[Sakoe 79] H. Sakoe: Two-level DP-matching - a dynamic programming-based pattern matching algorithm for connected word recognition. *IEEE Transactions on Speech and Audio Processing*, Vol. 27, pp. 588–595, Dec. 1979.

[Salakhutdinov & Roweis+ 03] R. Salakhutdinov, S. Roweis, Z. Ghahramani: On the convergence of bound optimization algorithms. In *Conference in Uncertainty in Artificial Intelligence (UAI)*, Acapulco, Mexico, Aug. 2003.

[Saon & Povey 08] G. Saon, D. Povey: Penalty function maximization for large margin HMM training. In *Interspeech*, Brisbane, Australia, Sept. 2008.

[Saul & Lee 02] L. Saul, D. Lee: Multiplicative updates for classification by mixture models. In T.G. Dietterich, S. Becker, and Z. Ghahramani, editor, *Advances in Neural Information Processing Systems (NIPS)*. MIT Press, 2002.

[Schlüter 00] R. Schlüter: *Investigations on Discriminative Training Criteria*. Ph.D. thesis, RWTH Aachen University, Aachen, Germany, Sept. 2000.

[Schlüter & Macherey+ 97] R. Schlüter, W. Macherey, S. Kanthak, H. Ney, L. Welling: Comparison of optimization methods for discriminative training criteria. In *European Conference on Speech Communication and Technology (Eurospeech)*, pp. 15 – 18, Rhodes, Greece, Sept. 1997.

[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, pp. 287 – 310, 2001.

[Schlüter & Müller+ 99] R. Schlüter, B. Müller, F. Wessel, H. Ney: Interdependence of language models and discriminative training. In *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 119–122, Keystone, CO, USA, Dec. 1999.

[Schlüter & Zolnay+ 06] R. Schlüter, A. Zolnay, H. Ney: Feature combination using linear discriminant analysis and its pitfalls. In *Interspeech*, Pittsburgh, PA, USA, Sept. 2006.

[Schölkopf & Smola+ 99] B. Schölkopf, A. Smola, K.R. Müller: Fisher discriminant analysis with kernels. In *IEEE Neural Networks for Signal Processing Workshop*, pp. 41–48, aug 1999.

[Schölkopf & Tsuda+ 04] B. Schölkopf, K. Tsuda, J.P. Vert: *Kernel methods in computational biology*. MIT Press, Cambridge, MA, USA, 2004.

[Schützenberger 77] M.P. Schützenberger: Sur une variante des fonctions séquentielles. *Theoretical Computer Science*, Vol. 4, No. 1, pp. 47 – 57, 1977.

[Schwartz & Austin 91] R. Schwartz, S. Austin: A comparison of several approximate algorithms for finding multiple (*N*-best) sentence hypotheses. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Vol. 1, pp. 701–704, Toronto, Canada, May 1991.

[Schwartz & Chow 90] R. Schwartz, Y.L. Chow: The *N*-best algorithm: An efficient and exact procedure for finding the *N* most likely sentence hypotheses. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 81–84, Albuquerque, NM, April 1990.

[Sha & Saul 06] F. Sha, L. Saul: Large margin Gaussian mixture modeling for phonetic classification and recognition. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Toulouse, France, May 2006.

[Sha & Saul 07a] F. Sha, L. Saul: Comparison of large margin training to other discriminative methods for phonetic recognition by hidden Markov models. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Honolulu, HI, USA, April 2007.

[Sha & Saul 07b] F. Sha, L. Saul: Large margin hidden Markov models for automatic speech recognition. In *Advances in Neural Information Processing Systems (NIPS)*, Cambridge, MA, dec 2007.

[Sim & Gales 06] K. Sim, M. Gales: Minimum phone error training of precision matrix models. *IEEE Transactions on Speech and Audio Processing*, Vol. 14, No. 3, pp. 882–889, 2006.

[Sindhwani & Keerthi 06] V. Sindhwani, S.S. Keerthi: Large scale semi-supervised linear SVMs. In *Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 477–484, New York, NY, USA, aug 2006. ACM Press.

[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. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 593–596, Phoenix, Arizona, USA, March 1999.

[Stadermann 06] J. Stadermann: *Automatische Spracherkennung mit hybriden akustischen Modellen*. Ph.D. thesis, Munich, Germany, 2006.

[Steinbiss & Ney+ 93] V. Steinbiss, H. Ney, R. Häb-Umbach, B. Tran, U. Essen, R. Kneser, M. Oerder, H. Meier, X. Aubert, C. Dugast, D. Geller: The Philips research system for large-vocabulary continuous-speech recognition. In *European Conference on Speech Communication and Technology (Eurospeech)*, pp. 2125–2128, Berlin, Germany, Sept. 1993.

[Sutton & McCallum 07] C. Sutton, A. McCallum: An introduction to conditional random fields for relational learning. In L. Getoor, B. Taskar, editors, *Introduction to Statistical Relational Learning*. MIT Press, 2007.

[Tahir & Heigold+ 09] M. Tahir, G. Heigold, C. Plahl, R. Schlüter, H. Ney: Log-linear framework for linear feature transformations in speech recognition. In *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, Merano, Italy, Dec. 2009.

[Taskar & Guestrin+ 03] B. Taskar, C. Guestrin, D. Koller: Max-margin Markov networks. In *Advances in Neural Information Processing Systems (NIPS)*, dec 2003.

[Valtchev 95] V. Valtchev: *Discriminative Methods in HMM-based Speech Recognition*. Ph.D. thesis, St. John's College, University of Cambridge, Cambridge, 1995.

[Valtchev & Odell+ 96] V. Valtchev, J.J. Odell, P.C. Woodland, S.J. Young: Lattice-based discriminative training for large vocabulary speech recognition. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 605–608, Atlanta, GA, USA, May 1996.

[Valtchev & Odell+ 97] V. Valtchev, J.J. Odell, P.C. Woodland, S.J. Young: MMIE training of large vocabulary recognition systems. *Speech Communication*, Vol. 22, No. 4, pp. 303 – 314, 1997.

[Vapnik 95] V. Vapnik: *The nature of statistical learning theory*. Springer-Verlag, 1995.

[Venkataraman & Stolcke+ 04] A. Venkataraman, A. Stolcke, W. Wang, D. Vergyri, J. Zheng, V. Gadde, R. Ramana: An efficient repair procedure for quick transcriptions. In *Interspeech*, Vol. 2, pp. 1961–1964, Jeju Island, Korea, Oct. 2004.

[Vidal 97] E. Vidal: Finite-state speech-to-speech translation. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 111–114, Munich, Germany, April 1997.

[Vintsyuk 71] T.K. Vintsyuk: Elementwise recognition of continuous speech composed of words from a specified dictionary. *Kibernetika*, Vol. 7, pp. 133–143, March 1971.

[Viterbi 67] A. Viterbi: Error bounds for convolutional codes and an asymptotically optimal decoding algorithm. *IEEE Transactions on Information Theory*, Vol. 13, pp. 260–269, 1967.

[Walter 99] W. Walter: *Analysis 1 & 2*. Springer, 1999.

[Wang 06] L. Wang: *Discriminative linear transforms for adaptation and adaptive training*. Ph.D. thesis, Cambridge University, 2006.

[Wang & Schuurmans+ 02] S. Wang, D. Schuurmans, Y. Zhao: The latent maximum entropy principle. In *IEEE International Symposium on Information Theory (ISIT)*, Lausanne, Switzerland, June – July 2002.

[Weisstein 09] E. Weisstein: Convolution. From MathWorld – A Wolfram Web Resource, http://mathworld.wolfram.com/Convolution.html, 2009.

[Wessel 02] F. Wessel: *Word Posterior Probabilities for Large Vocabulary Continuous Speech Recognition*. Ph.D. thesis, RWTH Aachen, Aachen, Germany, 2002.

[Wessel & Macherey+ 98] F. Wessel, K. Macherey, R. Schlüter: Using word probabilities as confidence measures. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 225–228, Seattle, WA, USA, May 1998.

[Wessel & Schlüter+ 01] F. Wessel, R. Schlüter, K. Macherey, H. Ney: Confidence measures for large vocabulary continuous speech recognition. *IEEE Transactions on Speech and Audio Processing*, Vol. 9, No. 3, pp. 288–298, March 2001.

[Weston & Watkins 99] J. Weston, C. Watkins: Support vector machines for multi-class pattern classification. In *European Symposium on Artificial Neural Networks (ESANN)*, Bruges, Belgium, April 1999.

[Wiesler & Nußbaum-Thom+ 09] S. Wiesler, M. Nußbaum-Thom, G. Heigold, R. Schlüter, H. Ney: Investigations on features for log-linear acoustic models in continuous speech recognition. In *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, Merano, Italy, Dec. 2009.

[Woodland & Povey 00] P.C. Woodland, D. Povey: Large scale discriminative training for speech recognition. In *Automatic Speech Recognition (ASR)*, 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 and Language*, Vol. 16, No. 1, pp. 25–48, 2002.

[Wu 83] C. Wu: On the convergence properties of the EM algorithm. *The Annals of Statistics*, Vol. 11, No. 1, pp. 95–103, 1983.

[Yaman & Deng+ 07] S. Yaman, L. Deng, D. Yu, Y. Wang, A. Acero: A discriminative training framework using N-best speech recognition transcriptions and scores for spoken utterance classification. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Honolulu, HI, USA, April 2007.

[Yin & Jiang 07] Y. Yin, H. Jiang: A fast optimization method for large margin estimation of HMMs based on second order cone programming. In *Interspeech*, Antwerp, Belgium, Aug. 2007.

[Young 92] S.J. Young: The general use of tying in phoneme based HMM recognizers. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Vol. 1, pp. 569–572, San Francisco, CA, March 1992.

[Yu & Deng+ 06] D. Yu, L. Deng, X. He, A. Acero: Use of incrementally regulated discriminative margins in MCE training for speech recognition. In *Interspeech*, Pittsburgh, PA, Sept. 2006.

[Yu & Deng+ 07] D. Yu, L. Deng, X. He, A. Acero: Large-margin minimum classification error training for large-scale speech recognition tasks. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Honolulu, HI, USA, April 2007.

[Yu & Deng+ 08] D. Yu, L. Deng, X. He, A. Acero: Large-margin minimum classification error training: A theoretical risk minimization perspective. *Computer Speech and Language*, Vol. 22, pp. 415–429, 2008.

[Yu & Deng+ 09] D. Yu, L. Deng, A. Acero: Using continuous features in the maximum entropy model. *Pattern Recognition Letters*, Vol. 30, No. 14, pp. 1295–1300, 2009. doi:10.1016/j.patrec.2009.06.005.

[Zhang & Jin+ 03] J. Zhang, R. Jin, Y. Yang, A. Hauptmann: Modified logistic regression: An approximation to SVM and its applications in large-scale text categorization. In *International Conference on Machine Learning (ICML)*, Aug. 2003.

[Zheng & Stolcke 05a] J. Zheng, A. Stolcke: fMPE-MAP: Improved discriminative adaptation for modeling new domains. In *Interspeech*, pp. 2125–2128, Lisbon, Portugal, Sept. 2005.

[Zheng & Stolcke 05b] J. Zheng, A. Stolcke: Improved discriminative training using phone lattices. In *Interspeech*, pp. 2125–2128, Lisbon, Portugal, Sept. 2005.

[Zweig & Nguyen 09] G. Zweig, P. Nguyen: Maximum mutual information multi-phone units in direct modeling. In *Interspeech*, Brighton, England, Sept. 2009.

Curriculum Vitae

Personal Information

Name:	Georg Heigold
Date of birth:	April 22, 1974
Place of birth:	Lucerne, Switzerland
Nationality:	Swiss

Education

1981 – 1987	Primary school in Reussbühl, Switzerland
1987 – 1994	Secondary school in Reussbühl, Switzerland (Matura)
1995 – 2000	B.A. & M.A. in Physics, ETH Zurich, Switzerland major fields: solid state physics, neutron scattering minor field: probability and statistics

Working Experience

2000 – 2003	De La Rue International Limited, Berne, Switzerland Software developer (document security group)
2004 – 2009	Chair of Computer Science 6 (Human Language Technology and Pattern Recognition, RWTH Aachen University Research assistant and Ph.D. student (statistical speech recognition)
Summer 2008	Internship at Microsoft research Redmond lab