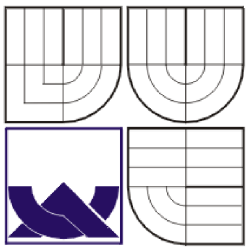
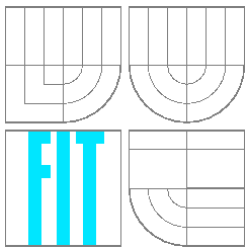


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

STUDY OF LINEAR TRANSFORMATIONS APPLIED TO TRAINING OF CROSS-DOMAIN ADAPTED LARGE VOCABULARY CONTINUOUS SPEECH RECOGNITION SYSTEMS

TITLE

DISERTAČNÍ PRÁCE
DOCTORAL THESIS

AUTOR PRÁCE — MARTIN KARAFIÁT
AUTHOR

VEDOUCÍ PRÁCE — JAN ČERNOCKÝ
SUPERVISOR

BRNO 2008

Abstract

This thesis investigates into two important issues of acoustic modeling for automatic speech recognition (ASR). The first topic are robust discriminative transforms in feature extraction. Two approaches of smoothing the popular Heteroscedastic Linear Discriminant Analysis (HLDA) were investigated: Smoothed HLDA (SHLDA) and Maximum A-Posteriori (MAP) adapted SHLDA. Both variants perform better than the basic HLDA. Moreover, we have found, that removing the silence class from the HLDA estimations (Silence-reduced HLDA) is equally effective and cheaper in computation. The second part deals with using heterogeneous data resources in ASR training. For a task, where little data is available for the target domain (meetings – 16kHz "wide-band" (WB) speech), techniques that allow to make use of abundant data from other domain, yet different in the acoustic channel (telephone data – 8kHz "narrow-band" – NB) were investigated. We successfully implemented an adaptation with WB data transformed to the NB domain based on Constrained Maximum Likelihood Linear Regression (CMLLR). A solution of how to apply this transform for HLDA and speaker-adaptive trained (SAT) systems was given using maximum likelihood. Finally, integration of this method with discriminative approaches was investigated and successfully solved. All experimental results are presented on standard data from NIST Rich Transcription (RT) 2005 evaluations.

Keywords

LVCSR system, meeting recognition, linear transform, Adaptation, cross domain adaptation, HLDA, CMLLR, MLLR, narrow band - wide band

Bibliographic citation

Martin Karafiát: *Study of Linear Transformations Applied to Training of Cross-Domain Adapted Large Vocabulary Continuous Speech Recognition Systems*, Doctoral thesis, Brno, Brno University of Technology, Faculty of Information Technology, 2008

Abstrakt

Tato disertační práce se zabývá dvěma významnými problémy z oblasti automatického rozpoznávání řeči (automatic speech recognition - ASR). Prvním tématem jsou robustní diskriminativní transformace používané pro výpočet příznaků. Ověřili jsme dvě varianty vyhlazování populární Heteroscedastické lineární discriminační analýzy (HLDA): vyhlazenou HLDA (Smoothed HLDA - SHLDA) a Maximum A-Posteriori adaptovanou HLDA. Obě varianty poskytují lepší výsledky než základní HLDA. Zjistili jsme rovněž, že pokud se při odhadu HLDA omezí nebo zcela odstraní úseky ticha (Silence-Reduced HLDA), jsou výsledky srovnatelné a metoda je podstatně méně náročná na výpočetní výkon. Druhá část disertace se zabývá použitím heterogenních dat pro trénování ASR systémů. Zkoumali jsme techniky, které pro úlohu, kde je k disposici omezené množství trénovacích dat (meetingy - 16kHz, "široké pásmo", "wide-band", WB) umožní využití dat z oblasti, kde je jich k disposici dostatek (telefonní data - 8kHz, "úzké pásmo" , "narrow-band", NB). Úspěšně jsme implementovali adaptaci s WB daty transformovanými do NB oblasti pomocí Constrained Maximum Likelihood lineární regrese (CMLLR). Pomocí metody maximum likelihood jsme ukázali, jak tuto transformaci použí společně s HLDA a SAT (speaker-adaptive) trénovanými systémy. V závěru jsme studovali a úspěšně využily integraci této techniky s diskriminativními přistupy k trénování. Všechny experimentální výsledky jsou presentovány na standardních datech z NIST Rich Transcription (RT) 2005 evaluací.

Klíčová slova

LVCSR systém, meeting recognition, linearni transformace, Adaptace, Adaptace napříč doménami, HLDA, CMLLR, MLLR.

Bibliografická citace

Martin Karafiát: *Aplikace lineárních transformací pro trénování systémů rozpoznávání spojité řeči s velkým slovníkem adaptovaný napříč doménami*, Disertační práce, Brno, Vysoké Učení Technické v Brně, Fakulta informačních technologií, 2008

Prohlášení

Prohlašuji, že jsem tuto disertační prácí vypracoval samostatně pod vedením Doc. Dr. Ing. Jana Černockého. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal. Některé v zavěru popsané aplikace fonémového rozpoznavače byly řešeny s dalšími členy skupiny Speech@FIT. Toto je vždy explicitně uvedeno.

Acknowledgments

First, I would like to thank my supervisor Jan Černocký for his endless patience, support and guidance. I am grateful to him for allowing me the freedom to explore various topics in the field speech recognition and for his constructive criticism and suggestions throughout the work on this thesis.

I would like to thank Thomas Hain for having taught me how to make order in work and guide my steps in the field of large speech recognition systems.

I would like to thank my colleagues in Speech Group at Faculty of Information Technology in Brno: František Grézl, Petr Schwarz, Pavel Matějka and others. Special thanks must go to my colleague Lukáš Burget for great support and help.

My research has been supported by Faculty of Information Technology of Brno University of Technology, in part by EC projects Multi-modal meeting manager (M4), No. IST-2001-34485, Augmented Multi-party Interaction (AMI), No. 506811, AMIDA (FP6-033812), by Grant Agency of Czech Republic projects No. 102/02/0124, No. 102/08/0707 and No. 102/05/0278.

Contents

List of Tables

List of Figures

List of Abbreviations

ANN	Artificial Neural Network
CMLLR	Constrained Maximum Likelihood Linear Regression
CMN	Cepstral Mean Normalization
CVN	Cepstral Variance Normalization
CTS	Conversation Telephone speech
DCT	Discrete Cosine Transform
DTW	Dynamic Time Warping
EM	Estimation Maximization
GMM	Gaussian mixture model
HLDA	Heteroscedastic Linear Discriminant Analysis
HMM	Hidden Markov Model
LDA	Linear Discriminant Analysis
LM	Language model
MAP	Maximum a Posteriori
MFCC	Mel Frequency Cepstral Coefficients
ML	Maximum Likelihood
MLLR	Maximum Likelihood Linear Regression
MLLT	Maximum Likelihood Linear Transform
MPE	Minimum Phone Error
NB	Narrow band (8kHz)
PCA	Principal Component Analysis
PLP	Perceptual Linear Prediction Coefficients
SHLDA	Smoothed Heteroscedastic Linear Discriminant Analysis
VTLN	Vocal Tract Length Normalisation

WB	Wide band (16kHz)
WER	Word Error Rate

Chapter 1

Introduction

Human effort to save and keep information is so old as a human being is. First saved data was typically for religious purpose. The storage media was the human memory and conversion of text to lyrics and songs was often used to increase the accuracy. During the human evolution, many other media have been used in extension (written or printed text, photography, film...). And obviously the purpose has strongly expanded into many fields - entertainment, education, description of anything, security.... But the sense is still same - keep to remember.

"Keep as much as possible" could well describe 20th century by boom of digitization and computer age. Almost all kinds of information are stored in digital form which makes it quickly accessible. The increasing amount of stored information requires development of automatic indexing methods allowing quick search.

In last few years, significant importance was put on audio/visual data. Two different kinds of information are combined here. The video records (seeing modality) is useful source of information but it is quite difficult to find interesting points. The audio records (hearing modality) can be easily processed by speech recognition system. The recognition output can be further used by indexer or as an input for other information retrieval techniques: summarization, spoken term detection...

Due to two modalities on the input, this approach is also called "multimodal approach" - a weakness of one modality could be complemented by strength of other. Obviously, the complexity of such system can be huge and it can touch many science branches. In this work, we focus only on one part of the whole system - speech recognition.

The used data were taken from "meeting sessions". We can imagine that a few more or less intelligent people are sitting around the table and spontaneously discussing some technical problem. Obviously, the language has to be same but dialects can differ and the speakers are frequently non-native. The whole discussion is recorded by video camera and microphones.

1.1 Speech recognition history

The speech recognition is quite old task. The history goes into 50ties of the last century. The first speech-to-text translation systems were able to recognize just syllables [44] or isolated digits [8],[13] and they were speaker-dependent. In 80ties, statistical approaches, particularly Hidden Markov Models (this technique will be described in section 2), were introduced. They significantly pushed forward the progress and these systems were able to recognize continuous speech and they were speaker independent. The size of dictionary significantly increased over thousand words and, consequently, first large vocabulary systems appeared.

From the 1990ties until now, **evaluations** of speech technologies are pursued on various tasks. For example WSJ - read paragraphs from Wall Street Journal, ATIS - spontaneous speech with automatic air travel system. One of the most challenging tasks was, and still is, the Switchboard - transcription of spontaneous telephone conversations also called conversation telephone speech (CTS). This evaluation was introduced by U.S. National Institute for Standards in Technology (NIST) and it was called "HUB-5". On contrary to the read speech, this task is linguistically and acoustically unconstrained. The speech has variable speaker rate and conversation style. It also contains large amount of disfluencies such as unfinished words, hesitations, laugh, coughs.... A need to cover variability of spontaneous speech led to extensive recording of this kind of data (Switchboard-I, Fisher...). The first experiments showed, not surprisingly, significant dependence of word error rate (WER) on the amount of training data [25].

1.2 Meeting recognition

In comparison to telephone conversations, the meetings speech differs in channel. Different microphones are used and the bandwidth is different. Telephone speech is naturally recorded in low bandwidth (8kHz) and meetings are recorded in wider band (16kHz). But the meeting speech is quite similar to CTS in conversation style, therefore similar problems exists. The development on meeting speech recognition was the cause of the Rich Transcription (RT) NIST evaluation in 2005 which is running till now. Big proportion of non-native speakers, different channel parameters... increase the demands for relevant acoustic data[1]. Recordings of this kind of conversation took place at several sites (M4/AMI/AMIDA series of projects[2]...) but still not enough data is available.

The meeting is usually recorded in two ways:

- Using independent headset microphones (IHM) - microphones are put directly on the speakers. It gives the clearest signal, and problems with crosstalks (more speakers talking at same time) are limited. But the speakers do not find too comfortable to keep microphones on their bodies.

[1] The data required for the training of language model can be partly derived from in-domain texts.

[2] www.m4project.org, www.amiproject.org

- Using multiple distant microphones (MDM) - microphones are spread on the table or room. It is more comfortable for speakers but the quality of records and also the recognition performance is worse.

Setting up the microphones into a microphone array offers the opportunity to improve the quality of the signal by using localization techniques. Obviously, the complexity of system then increases.

In this work, we focus on the IHM speech only. MDM speech recognition is an important research topic but it is beyond the scope of this thesis.

1.3 Goals of this thesis

The aim is to increase the robustness of acoustic techniques in meeting speech recognition, especially by the use of Heteroscedastic Linear Discriminant Analysis (HLDA). We propose MAP-Smoothed and Silence Reduced HLDA modifications.

Further, we focus on effective porting of telephone speech data resources into the meeting domain. A common problem is different bandwidth. The standard approach is to downsample the wide-band data, which is not very efficient due to the loss of information from the upper band. We investigate substitution of the downsampling by adaptation which does not remove any information. Next, we focus on using this approach together with advanced techniques like HLDA, Speaker Adaptive Training (SAT) and discriminative training. The solution is not trivial, so mathematical development and extensive experimental work is presented.

1.4 Scope of chapters

Basic speech recognition system structure is introduced in chapter 2. The problems are formulated and basic terms are explained. The focus is put on Hidden Markov Models (HMM) and feature extraction used in this thesis.

Chapter 3 introduces linear transforms widely used in speech processing and speech recognition, especially already mentioned HLDA.

In chapter 4, our baseline recognition systems are described. The specifications of training and test data sets follow.

Chapter 5 investigates into HLDA, and its modification with more robust estimates of statistics is proposed. Next, positive effect by weighting of silence class in HLDA estimation is presented (Silence Reduced HLDA).

In chapter 6, channel normalization techniques followed by speaker normalization (VTLN) and adaptation (MLLR and CMLLR) are introduced. The speaker adaptive training is described too.

Chapter 7 examines the use of well trained models on telephone data in meeting recognition system. Already mentioned technique using adaptation instead of downsampling is proposed and investigated.

Finally, the thesis is concluded and plans for future work are outlined in chapter 8.

1.5 Original contributions of this thesis

In our opinion, the original contributions — "claims of this thesis" — can be summarized as follows:

- Proposal and testing of robust methods of HLDA estimation in LVCSR:
  - Smoothing based on the amount of data available for each class - MAP-SHLDA.
  - Balancing of silence class in Silence-Reduced HLDA.
- Linear transforms employed as narrow-band → wide-band (NB-WB) adaptations:
  - MLLR adaptation of narrow-band models to wide-band speech.
  - CMLLR feature transform used to convert WB speech to NB models.
  - NB-WB transform in HLDA estimation - use of MAP for interpolation between NB and WB statistics.
  - NB-WB transform in speaker adaptive training.
  - NB-WB transform with discriminative approaches - MPE-MAP.

Chapter 2

Basis of speech recognition

Speech recognition system is a complex machine, usually composed of four main parts:

1. **Signal processing** - Extraction of features from speech waveforms by mapping to a representation (a sequence of feature vectors) that is more easily modeled. This procedure is mostly based on a spectral analysis of a short time interval. The typical examples of spectral analysis based features are well known Mel Frequency Cepstral Coefficients *MFCC* and Perceptual Linear Predictive *PLP* features. More details can be found later in section 2.1.

   Due to the research done in the last decade, feature extraction from long time context started to be significant too [51],[27].

2. **Acoustic classification** - Use of statistical tools to obtain likelihood or probability that input features are generated by a model. Models can represent whole words or just phonemes. The most common method used for acoustic modeling is Hidden Markov Model (HMM). More detailed description can be found in section 2.2.

3. **Language modeling** - Use of information given by language model to generate probability of given word sequence. The language model is able to suppress meaningless words sequences, respectively enhance reasonable ones. See section 2.5 for more details.

4. **Decoding** - Use of output information from *Acoustic models* and *Language model* to search for the most accurate output sentence. The complexity of searching space grows exponentially with size of dictionary. Therefore, a lot of optimization is necessary in case of a large vocabulary task. The description of the decoding process is not core of this work. Interested reader can find more information in [43].

A simple example of a speech recognition process is shown in figure 2.1. At first, the features are extracted from speech waveforms by the *Signal processing* block. The acoustics models are applied to obtain likelihood of acoustic match of feature vector to each particular model. The

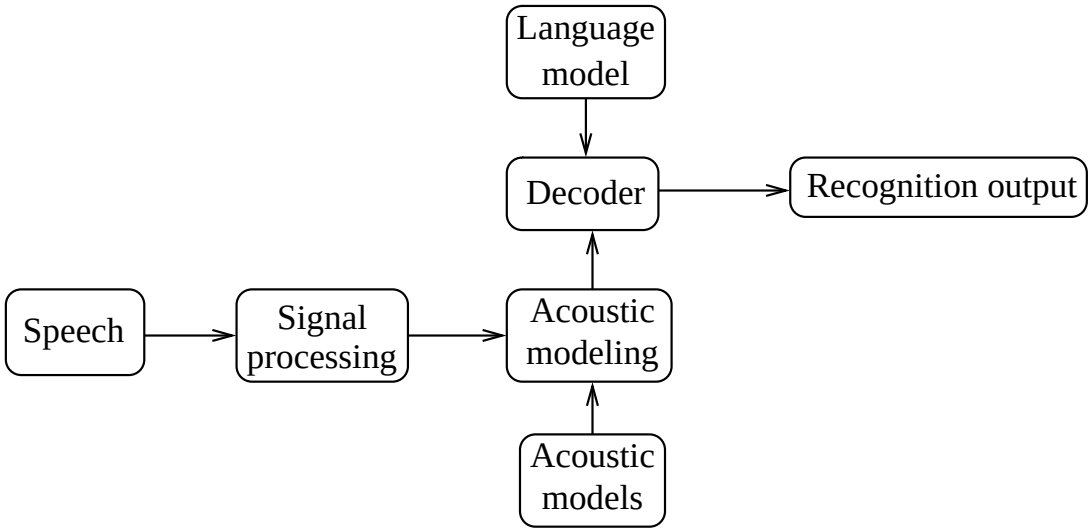


Figure 2.1: Example of recognition process.

outputs from acoustic models together with language model likelihood make up the input for the decoder. It combines these likelihoods with information from history (older likelihoods, recognized words...) for the generation of the output of the recognizer.

The output is expected in the following typical forms:

- One–best output - the sequence giving the best overall likelihood is generated.
- N–Best output - N the most probable hypotheses are generated.
- Lattice (Graph of hypotheses) - allows more hypotheses in parallel. In comparison with the N–best string output, a lattice can be stored more efficiently. It consists of nodes and links with assigned words and other associated information (times, acoustic likelihood, language model likelihood).

N–Best or lattice outputs are often used in two-(or more)-pass decoding process. Weak language and acoustic models produce wide lattices (or N–best strings) in the first pass which significantly limits the search space. The lattices are just "rescored" using strong models and new, more accurate, one-best hypothesis is generated.

2.1 Feature extraction

The purpose of feature extraction is to transform speech waveform into a form suitable for the recognizer. The standard feature extraction can be divided into the following steps:

1. **Segmentation** - Speech signal is divided into segments, where speech is regarded as stationary. A multiplication of speech waveform with rectangular or more often Hamming

window can be used for this purpose. The typical value for window size is 25ms and time shift is 10ms.

2. **Spectrum** - The power or magnitude Fourier spectrum is computed for every speech segment.

3. **Auditory-like modifications** - Modifications inspired by physiological and psychological findings about human perception of loudness and different sensitivity for different frequencies are performed on spectra of each speech frame [39, 26]. The type of modification has influence on the output feature kind.

4. **Decorrelation and dimensionality reduction** - In case of using HMM with diagonal covariance modeling (see below), the features have to be decorrelated and nuisance dimensions have to be removed. Empirically chosen Discrete Cosine Transform (DCT) is commonly used for this purpose. Some examples, where different kind of transforms are used to improve feature extraction, can be found in [4, 28, 35].

5. **Derivatives** – Feature vectors are usually augmented with first and second order derivatives of their time trajectories (delta and acceleration coefficients). These coefficients describe changes and speed of changes of the feature vector coefficients in the time.

2.1.1 Perceptual linear predictive analysis

The Perceptual linear predictive coefficients (PLP) are popular features in speech recognition area. Since these features are used in our experiments, a brief description of this method is given here.

The PLP analysis was first introduced by Hermansky [26]. The main idea of this technique is to take advantage of three principal characteristics derived from the physical and acoustic properties of the human ear for estimating the audible spectrum. These are:

1. Spectral resolution of the critical band.

2. Equal-loudness curve.

3. Intensity-loudness power law.

The whole method could be described in the following steps:

1. The short-term power spectrum $P(\omega)$ of speech segment (after Hamming windowing) is taken.

2. **Critical Band Spectral Resolution:**

- Bark frequency domain: The spectrum $P(\omega)$ is transformed to the Bark frequency domain (Ω) by:

$$\Omega(\omega) = 6 \ln \left(\frac{\omega}{1200\pi} + ((\frac{\omega}{1200\pi})^2 + 1)^{\frac{1}{2}} \right) \tag{2.1}$$

where ω represents the angular frequency measured in rad/s. The resulting warped spectrum is than convoluted with the power spectrum of the simulated critical-band:

$$\Psi(\Omega) = \begin{cases} 0 & for \quad \Omega < -1.3 \\ 10^{2.5(\Omega-0.5)} & for \quad -1.3 < \Omega < -0.5 \\ 1 & for \quad -0.5 < \Omega < 0.5 \\ 10^{-1.0(\Omega-0.5)} & for \quad 0.5 < \Omega < 2.5 \\ 0 & for \quad \Omega > 2.5 \end{cases} \tag{2.2}$$

The discrete convolution of $\Psi(\Omega)$ with $P(\Omega)$ produces the samples of the critical band power spectrum:

$$\Theta(\Omega_i) = \sum_{\Omega=-1.3}^{2.5} P(\Omega - \Omega_i)\Psi(\Omega) \tag{2.3}$$

The sampling interval is chosen so that the integral number of spectral samples covers the whole analysis band. Normally, 18 spectral samples are chosen to cover the 0-16.9 Bark band (0-5 kHz) in 0.994 Bark-steps.

- Mel-frequency domain: A conversion to Mel-frequency domain is a traditional approach based on psycho-acoustic findings, where better resolution in a spectrum is preserved for lower frequencies than for higher frequencies.

  The relationship between frequency and Mel scale domain is:

$$f_m = 1127.01048 \ln(1 + f/700)$$

  Mel filter bank is applied to smooth the spectrum: Energies in the spectrum are integrated by a set of a band limited triangular weighting functions. Their shape can be seen in figure 2.2. These weighting functions are equidistantly distributed over the Mel scale. A vector of filter bank energies for one frame can be seen as a smoothed and down-sampled version of the spectrum.

  Mel scale used instead of Bark scale in the PLP analysis generates so called ML-PLP features. A previous work [52] mentions better robustness using of ML-PLP compared to standard PLPs with Bark-scale. Therefore, this implementation is used in this work.

3. **Equal-loudness preemphasis:** The simulated equal-loudness curve is used to pre-emphasize the output from critical band analysis $\Theta(\Omega(\omega))$:

$$\Xi(\Omega(\omega)) = E(\omega)\Theta(\Omega(\omega)) \tag{2.4}$$

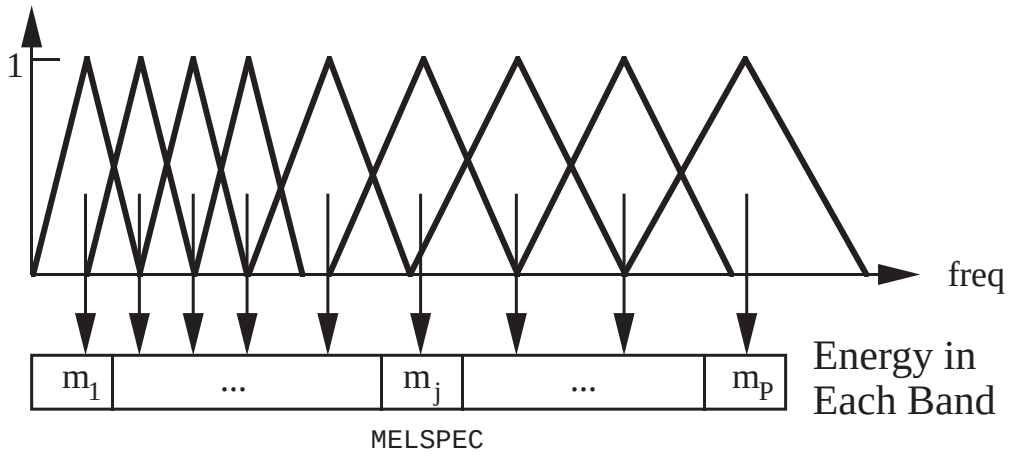


Figure 2.2: Mel scale filter bank (reproduced from [54]).

The function $E(\omega)$ represents an approximation of the human auditive sensitivity to react to different frequencies at about 40 dB level. Detailed equations of this approximation can be found in [26].

4. **Intensity-Loudness Power Law** is an approximation to the power law of hearing and simulates the non-linear relationship between the sound intensity and the perceived loudness:

$$\Phi(\Omega) = \Xi(\Omega)^{0.33} \tag{2.5}$$

This operation also reduces the spectral amplitude variation of the critical-band.

5. **Autoregressive Modeling:** In this step, the expression $\Phi(\Omega)$ is approximated by the spectrum of an all-pole model. Then, the inverse discrete Fourier transform (IDFT) is applied to obtain the autocorrelation function of $\Phi(\Omega)$. The first M+1 values are used for solving the Yule-Walker equation for obtaining the autoregressive coefficients of the all-pole model of order M. These coefficients can be transformed to other types of parameters for further analysis.

2.2 Introduction to Hidden Markov Models

Hidden Markov Models are the most popular acoustic modeling method used in speech recognition systems. In this chapter, the basic principles only will be described. Very good and compact introduction to HMM can be found in [54]. Detailed HMM description and derivation of all formulae are in [29, 18, 30, 49].

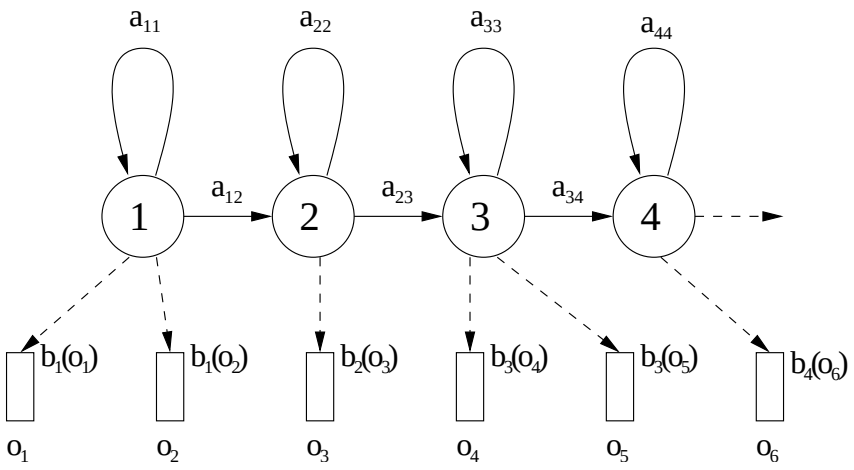


Figure 2.3: Typical Hidden Markov Model of a word (reproduced from [54]).

2.2.1 HMM based speech recognition

From statistical point of view, the goal of speech recognition is to find the most likely word sequence $\mathbf{W}' = w_1, w_2, \ldots, w_n$ given the sequence of observations (sequence of feature vectors) $\mathbf{O} = \mathbf{o}(1), \mathbf{o}(2), \ldots, \mathbf{o}(T)$. This can be expressed as:

$$\mathbf{W}' = \arg\max_{\mathbf{W}} \{P(\mathbf{W}|\mathbf{O})\}, \tag{2.6}$$

where $\mathbf{W}$ is arbitrary word sequence.

It is not easy to directly estimate the posterior probability $P(\mathbf{W}|\mathbf{O})$. According to Bayes Rule, the probability can be expressed as:

$$P(\mathbf{W}|\mathbf{O}) = \frac{p(\mathbf{O}|\mathbf{W})P(\mathbf{W})}{p(\mathbf{O})}, \tag{2.7}$$

where $p(\mathbf{O})$ is a probability (or a probability density) of the observation sequence, which stays constant over different word sequences, $\mathbf{W}$, and, therefore, it can be ignored when making decision about the most likely sequence, $\mathbf{W}'$. $P(\mathbf{W})$ is the prior probability of word sequence $\mathbf{W}$, which is usually estimated using *language model* (see section 2.5).

The recognition problem can be rewritten as:

$$\mathbf{W}' = \arg\max_{\mathbf{W}} \{p(\mathbf{O}|\mathbf{W})P(\mathbf{W})\}. \tag{2.8}$$

The likelihood $p(\mathbf{O}|\mathbf{W})$ can be modeled and evaluated using Hidden Markov Models. An example of HMM with left-to-right topology, typically used to model a single word in speech recognition, is shown in figure 2.3. HMM is a generative model, which can be seen as a finite state machine making transition from state i to state j with probability a_{ij} and generating a feature vector, $\mathbf{o}(t)$, based on distribution $b_j(\mathbf{o})$ at every discrete time step, t. Having a set of models $\{M_1, M_2, \ldots\}$ representing individual words (or even smaller linguistic units such as

phonemes) $\{w_1, w_2, \dots\}$, model M representing a word sequence $\mathbf{W}$ can be simply created by concatenating appropriate individual models. The compound model is typically constraint so that the first and the last feature vector must be generated by the first and the last HMM state, respectively. In speech recognition, of course, we do not use HMM to generate anything. However, model M representing word sequence W allows us to evaluate likelihood $p(\mathbf{O}|M)$, which represents the desired likelihood $p(\mathbf{O}|W)$. State-dependent observation distribution $b_s(\mathbf{o})$ is typically modeled using mixture of multivariate Gaussians with diagonal covariance matrices

$$b_s(\mathbf{o}) = \sum_m c_{sm} b_{sm}(\mathbf{o}), \tag{2.9}$$

where c_{sm} is weight of m^{th} Gaussian component associated with s^{th} HMM state and b_{sm} is the Gaussian component:

$$b_{sm}(\mathbf{o}) = \frac{1}{(2\pi)^n \prod_{k=1}^n \sigma_{smk}^2} e^{-\sum_{k=1}^n \frac{(o_k - \mu_{smk})^2}{2\sigma_{smk}^2}}, \tag{2.10}$$

where μ_{smk} and σ_{smk}^2 are k^{th} coefficients of mean and variance vector of the Gaussian component. Likelihood of observation sequence, $\mathbf{O}$, given a state sequence, $\mathbf{S} = s(1), s(2), \dots, s(T)$, and model M is

$$p(\mathbf{O}|\mathbf{S}, M) = \prod_{t=1}^T b_{s(t)}\left(\mathbf{o}(t)\right) \tag{2.11}$$

and probability of a state sequence, $\mathbf{S}$, given a model, M, is

$$P(\mathbf{S}|M) = \prod_{t=2}^T a_{s(t-1)s(t)}. \tag{2.12}$$

Since the state sequence is not directly observable (discrete random variable $s(t)$ is hidden), the likelihood $p(\mathbf{O}|M)$ is expressed as a sum over all possible state sequences:

$$p(\mathbf{O}|M) = \sum_{\mathbf{S}} p(\mathbf{O}|\mathbf{S}, M) P(\mathbf{S}|M). \tag{2.13}$$

It can be computed using the Baum-Welch algorithm (see section 2.3.3). This likelihood is often approximated by the likelihood of the best state sequence $\mathbf{S}$ only:

$$\hat{p}(\mathbf{O}|M) = \max_{\mathbf{S}} \{p(\mathbf{O}|\mathbf{S}, M) P(\mathbf{S}|M)\}, \tag{2.14}$$

This likelihood, so called *Viterbi likelihood* (or incorrectly *Viterbi probability*), can be computed more efficiently than the true likelihood. Viterbi decoding, which is an algorithm based on Dynamic Programming allowing to efficiently compute Viterbi probability, is used to find the most likely sequence of composite HMM states. Such state sequence uniquely identifies the most likely sequence of words.

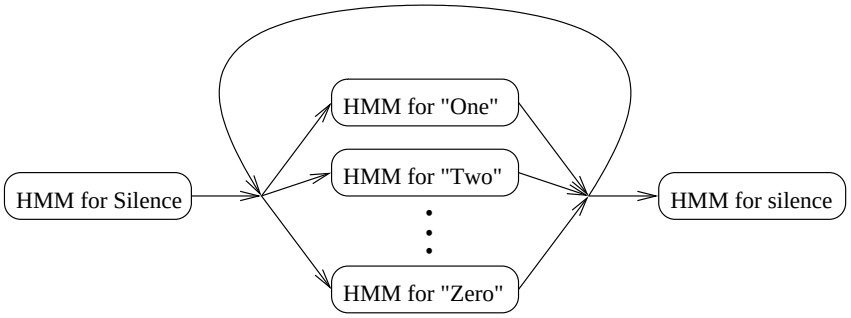


Figure 2.4: Recognition network used for recognition of connected digits.

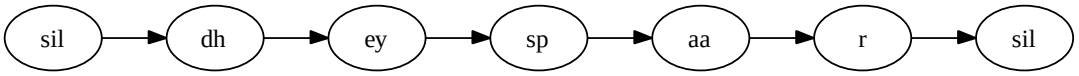


Figure 2.5: Monophone expansion of sentence “THEY ARE”.

2.3 Application of HMM in recognition systems

In more sophisticated recognition systems such as a large vocabulary recognizer, it is impossible to work with word HMMs. The models of words are composed of smaller units, usually phonemes, according to given dictionary.

Three main kinds of word-phoneme expansions are used:

- monophones - context independent phoneme models. It gives simple implementation but not good performance due to context dependency of phonemes.
- within-word context dependent phonemes - the words are expanded into context dependent phoneme unit, but with no respect of phonemes outside of word boundaries.
- cross-word context dependent phonemes - on contrary to within-word, the cross-word expansion goes through the word boundaries.

The context expansion commonly considers neighboring context only, so called triphones. Wider context size, like quinphones, are used in some state-of-the-art systems too [53]. Figure 2.5 shows monophone expansion of sentence “THEY ARE”. The cross-word triphone expansion is shown in figure 2.6. Typically, more pronunciation variants are defined for each word. The example is shown in figure 2.7.

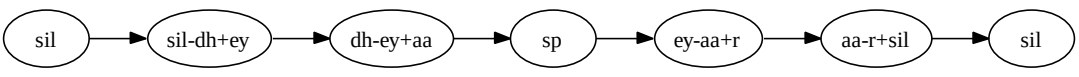


Figure 2.6: Cross-word triphone expansion of sentence “THEY ARE”.

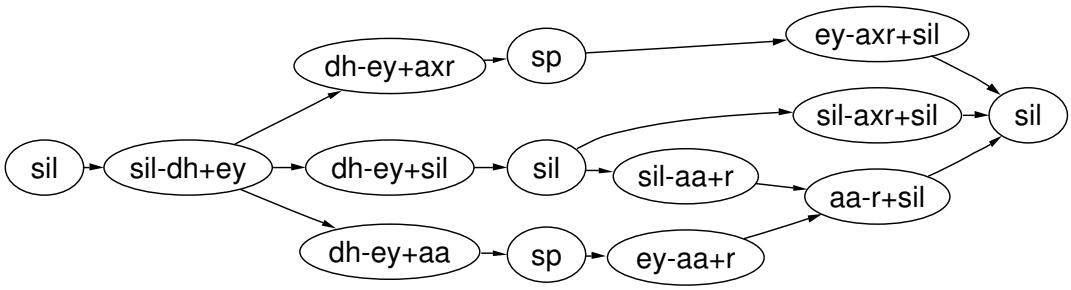


Figure 2.7: Cross-word triphone expansion of sentence "THEY ARE" with more pronunciation variants.

2.3.1 Forced alignment

The forced alignment is a recognition using a given data transcription, typically on word level. A recognition network is composed for each sentence according to this transcription and dictionary. It allows to find the most probable phoneme sequence from the dictionary and also time boundaries of the words and phonemes. This is commonly used during the training process: data are re-aligned by currently trained models. It produces more precise phoneme-level transcription for further training.

2.3.2 Decision tree clustering

Context dependent triphones provide better acoustics modeling than independent ones, but unseen or rarely occurring triphones can not be properly trained. This problem can be solved by decision tree clustering technique:

1. First, a set of binary questions has to be created. It could be data driven or based on linguistic rules, which is more common. For example: Is the left context "e"? Is the right context a vowel? Is the right context an unvoiced stop?

2. Selection of a state and merging all context variants is done - for example - all central triphone states derived from "s".

3. Next, the binary questions are applied. The joint state from the point 2 is split according to the particular question. For each split, it is checked if the partition generates a positive effect (increase of likelihood) and if there is enough training data for both new states. The best split is chosen and we iterate the algorithm on the newly created states. The splitting stops if none of questions can generate new states fulfilling the two conditions above.

   Note, that the minimum increase of likelihood and minimum training observations per state are the main constants which control the whole process and the resulting number of clusters.

4. Return to point 2 for next state.

5. With the help of successful decisions trees, we can fill the list of triphones for unseen triphones.

2.3.3 Estimation of HMM parameters using maximum likelihood

Before Hidden Markov Models can be used for recognition, the set of their parameters, $\lambda = \{a_{ij}, \boldsymbol{\mu}_j, \boldsymbol{\sigma}_j^2\}$, must be determined. For simplification, only one-Gaussian models are considered.

The usual practice is to estimate the parameters from training speech data for which the correct transcription is known. The most popular training scheme is maximum likelihood (ML) parameter estimation: Let $M_{\boldsymbol{\lambda}}^{\mathbf{W}}$ denote a compound model representing model sequence $\mathbf{W}$, where $\boldsymbol{\lambda}$ are parameters of all models from which the compound models is constructed. The goal is to find such setting of values, $\boldsymbol{\lambda}'$, that maximizes the likelihood of training data:

$$\boldsymbol{\lambda}' = \arg\max_{\boldsymbol{\lambda}} \left\{ \prod_{r=1}^{R} P(\mathbf{O}^r | M_{\boldsymbol{\lambda}}^{\mathbf{W}^r}) \right\}, \tag{2.15}$$

where R is number of training utterances, $\mathbf{O}^r$ is observation sequence for r^{th} utterance and $\mathbf{W}^r$ is its transcription. Well known Baum-Welch algorithm [2], which is based on the standard EM (Estimation Maximization) [9], can be used to search for ML estimates of HMM parameters. This algorithm guarantees to increase (or at least not to decrease) the likelihood of training data. It collects set of statistics over the training data using current set of parameters $\boldsymbol{\lambda}$. Those statistics are further used to estimate new set of parameters $\hat{\boldsymbol{\lambda}}$. This is an iterative process which converges to the parameter settings $\boldsymbol{\lambda}'$.

The algorithm estimates an "occupation" probability $\gamma_j^r(t)$, which is the posterior probability of being in state j in time t of utterance r by computing likelihoods based on two different partial probabilities: forward probability $\alpha_j^r(t)$ which expresses a likelihood of all possible state sequences up to and including state j and time t of utterance r. Backward probability $\beta_j^r(t)$ expresses all possible state sequences from state j and time t to the end of utterance r.

The forward probability is computed by the following recursion:

$$\alpha_j^r(t) = \sum_{k=1}^{S} \alpha_k^r(t-1) a_{jk} b_j(\mathbf{o}^r(t)) \tag{2.16}$$

where S is the final state. The initial conditions are set to $\alpha_1(1) = 1$ and $\alpha_j(1) = a_{1j} b_j(\mathbf{o}(1))$ for $j > 1$.

The backward probability is computed as:

$$\beta_j^r(t) = \sum_{k=1}^{S} \beta_k^r(t+1) a_k j b_j(\mathbf{o}^r(t+1)) \tag{2.17}$$

with initial conditions $\beta_j^r(T^r) = a_{jS}$ where T^r is number of frames of utterance r.

Then, the occupation probability $\gamma_j^r(t)$ is given by

$$\gamma_j^r(t) = \frac{\alpha_j^r(t) \beta_j^r(t)}{p(\mathbf{O}^r)} \tag{2.18}$$

where the probability of the data $p(\mathbf{O}^r) = \alpha_S^r(T^r) = \beta_1^r(1)$.

These occupation probabilities make it possible to gather sufficient statistics for re-estimation formulae.

$$\gamma_j = \sum_{r=1}^{R}\sum_{t=1}^{T^r}\gamma_j^r(t) \tag{2.19}$$

$$\theta_j(\mathbf{O}) = \sum_{r=1}^{R}\sum_{t=1}^{T^r}\gamma_j^r(t)\mathbf{o}^r(t) \tag{2.20}$$

$$\theta_j(\mathbf{O}^2) = \sum_{r=1}^{R}\sum_{t=1}^{T^r}\gamma_j^r(t)(\mathbf{o}^r(t))^2 \tag{2.21}$$

Then, the following formulae are used to re-estimate Gaussian parameters:

$$\hat{\boldsymbol{\mu}}_j = \frac{\theta_j(\mathbf{O})}{\gamma_j} \tag{2.22}$$

$$\hat{\boldsymbol{\sigma}}_j = \frac{\theta_j(\mathbf{O}^2)}{\gamma_j} - \boldsymbol{\mu}_j^2 \tag{2.23}$$

The extension to Gaussian mixture models is straightforward. The statistics $\gamma_{jm}, \theta_{jm}(\mathbf{O}), \theta_{jm}(\mathbf{O}^2)$ need to be collected for each mixture component m. The Gaussian parameters are updated by equations similar to 2.22 and 2.23.

Finally, the mixture weights c_{jm} are updated according to:

$$c_{jm} = \frac{\gamma_{jm}}{\sum_m \gamma_{jm}}. \tag{2.24}$$

2.4 Estimation of HMM parameters using discriminative approach

In addition to Baum-Welch algorithm (which was described in previous section) other training schemes exist where HMM parameters are not estimated in ML fashion. Maximum Mutual Information (MMI) [41], Minimum Classification Error (MCE) [7], Minimum Phone Error (MPE), and Minimum Word Error(MWE) [48] are examples of discriminative training schemes. Instead of maximizing the likelihood, an objective function, which is believed to be closer related to the recognition performance, is maximized.

Let us introduce the MPE training as it is used in this thesis later. Only basic principles of MPE will be given, and for simplification, only one-Gaussian per state models are considered. More detailed information can be found in [48],[46].

For R training observation sequences $\{\mathbf{O}^1, \ldots, \mathbf{O}^r, \ldots \mathbf{O}^R\}$ we known word and phone transcriptions $\mathbf{W}^r$. The $\mathbf{W}^r$ is a subset of space of all possible hypotheses $\mathbb{W}$. The MPE criterion for parameter set $\boldsymbol{\lambda}$ is defined by

$$F_{MPE}(\boldsymbol{\lambda}) = \sum_{r=1}^{R} \frac{\sum_{\mathbf{W}\in\mathbb{W}} p_{\boldsymbol{\lambda}}(\mathbf{O}^r|\mathbf{W})^\kappa P(\mathbf{W}) A(\mathbf{W})}{\sum_{\mathbf{W}\in\mathbb{W}} p_{\boldsymbol{\lambda}}(\mathbf{O}^r|\mathbf{W})^\kappa P(\mathbf{W})}, \tag{2.25}$$

where $A(\mathbf{W})$ is measure of number of phones correctly recognized in hypothesis $\mathbf{W}$. $P(\mathbf{W})$ is probability of hypothesis $\mathbf{W}$ given by language model and κ is a scaling constant.

For each training utterance, MPE criterion averages over all hypothesis from $\mathbb{W}$ weighted by the respective accuracy $A(\mathbf{W})$. Therefore, the correct transcriptions $\mathbf{W}^r$ are used only for measuring the accuracy $A(\mathbf{W})$.

Related to MPE criterion is MWE criterion, the only difference is in measure of accuracy $A(\mathbf{W})$ on word level instead of phone one. The MWE was found more effective in maximizing accuracy on training set than MPE, but slightly worse results on test set were presented [48].

The MPE framework is done in context of word level recognition too like MWE. It works on word level but the accuracy measure is based on how many phones are correct.

The space of hypotheses $\mathbb{W}$ could be enormously huge, therefore it has to be limited to reasonable size. Lattices are used for this purpose. Typically, the ML trained models are taken as the input. Those models are used for generation of word-phoneme lattices over the training data. These lattices contain the time information also on phone level which is further used to speed up the processing.

2.4.1 Calculation of $A(\mathbf{W})$

The function of $A(\mathbf{W})$ ideally equals the number of correct phones minus the number of insertions. It could be expressed as a sum of phone level accuracies $A(q)$ over all phones q in $\mathbf{W}$, where $A(q)$ is defined as:

$$A(q) = \begin{cases} 1 & \text{if correct phone} \\ 0 & \text{if substitution} \\ -1 & \text{if insertion} \end{cases} \tag{2.26}$$

This, however, calls for a dynamic programing to obtain the alignment of each hypothesis and reference sequence, which is quite inefficient. If time boundaries of phones in both sequences are known, the following approximation can be used to avoid this extra-computation:

Consider a phone q from hypothesis $\mathbf{W}$ and phone z from reference transcript which overlap in time: $e(q, z)$ is overlapped proportion of q in reference phone z. Then, the approximated $A(q)$ can be computed according to:

$$A(q) = \max_z \begin{cases} -1 + 2e(q, z) & \text{if } z \text{ and } q \text{ are the same phone} \\ -1 + e(q, z) & \text{if } z \text{ and } q \text{ are different} \end{cases} \tag{2.27}$$

If phone q overlapped with more than one phone from reference, more "partial" $A(q)$ are computed and the highest value is taken.

2.4.2 MPE estimation and update

The whole optimization of objective function can be found in [46]. But the most important value is the differential of 2.25 w.r.t. log likelihood of phoneme arc q from the lattice:

$$\gamma_q^{MPE} = \frac{1}{\kappa} \frac{\delta F_{MPE}}{\delta \log p(\mathbf{O}|q)}, \tag{2.28}$$

which can be computed as

$$\gamma_q^{MPE} = \gamma_q(c(q) - c_{avg}^r), \tag{2.29}$$

where γ_q is the posterior probability of the arc q derived by forward-backward algorithm over the arcs. $c(q)$ is average accuracy $A(\mathbf{W})$ of hypothesis passing through the arc q, and c_{avg}^r is the average $A(\mathbf{W})$ of all the hypothesis in the recognition lattice for r'th training file (all averages are weighted by posteriors of the hypotheses).

If time information about word and phones boundaries is given in lattice, so called, **exact-match algorithm** can be effectively used to speed up the computation of forward-backward algorithm:

Let's say that each phone arc q has a known start time s_q and end time e_q. Forward-backward computation within a single arc q gives the arc-likelihood $p(\mathbf{O}|q)$ and also occupation probabilities γ_{qjm} for Gaussian m of state j. When all arcs are processed, the arc-likelihoods $p(\mathbf{O}|q)$ are used in the lattice node-level forward-backward computation. It estimates a posterior probability of arc γ_q: probability of going through the arc q.

When the γ_q^{MPE} and state occupation probabilities are calculated, the statistics for MPE update can be collected:

$$\gamma_{jm} = \sum_{q=1}^{Q} \sum_{t=s_q}^{e_q} \gamma_{qjm}(t) \gamma_q^{MPE} \tag{2.30}$$

$$\theta_{jm}(\mathbf{O}) = \sum_{q=1}^{Q} \sum_{t=s_q}^{e_q} \gamma_{qjm}(t) \gamma_q^{MPE} \mathbf{o}(t) \tag{2.31}$$

$$\theta_{jm}(\mathbf{O}^2) = \sum_{q=1}^{Q} \sum_{t=s_q}^{e_q} \gamma_{qjm}(t) \gamma_q^{MPE} \mathbf{o}(t)^2 \tag{2.32}$$

The new model parameters can be updated by Extended Baum-Welch formula which was introduced for optimization of MMI objective function [42]. For MPE, it could be simplified to:

$$\hat{\boldsymbol{\mu}}_{jm} = \frac{\theta_{jm}(\mathbf{O}) + D\boldsymbol{\mu}_{jm}}{\gamma_{jm} + D} \tag{2.33}$$

$$\hat{\boldsymbol{\sigma}}_{jm} = \frac{\theta_j(\mathbf{O}^2) + D(\boldsymbol{\sigma}_{jm}^2 + \boldsymbol{\mu}_{jm}^2)}{\gamma_{jm} + D} - \hat{\boldsymbol{\mu}}_{jm}^2 \tag{2.34}$$

$$\tag{2.35}$$

where $\boldsymbol{\mu}_{jm}$ and $\boldsymbol{\sigma}_{jm}$ are original Gaussian parameters. Constant D is set on Gaussian level to twice the value needed to ensure a positive variance update or to a global constant E multiplied

by γ_{jm}^{den}, which is sum of negative contributions from equation 2.30. E is commonly set to 1 or 2.

2.4.3 I-smoothing

The I-smoothing is a technique which allows an interpolation between MPE and ML estimates based on amount of data available for each Gaussian, where ML statistics are regarded as a prior.

The ML statistics are collected using correct transcriptions, equations 2.19-2.32, and added into the MPE statistics according to:

$$\gamma'_{jm} = \gamma_{jm} + \tau \tag{2.36}$$

$$\theta'_{jm}(\mathbf{O}) = \frac{\theta_{jm}(\mathbf{O})\gamma_{jm} + \theta_{jm}^{ML}(\mathbf{O})\tau}{\gamma_{jm} + \tau} \tag{2.37}$$

$$\theta'_{jm}(\mathbf{O}^2) = \frac{\theta_{jm}(\mathbf{O}^2)\gamma_{jm} + \theta_{jm}^{ML}(\mathbf{O}^2)\tau}{\gamma_{jm} + \tau} \tag{2.38}$$

where τ is a control constant.

I-smoothing has been found very effective in discriminative training approaches, not only in MPE.

2.4.4 MPE-MAP

Similarly to I-smoothing, MPE-MAP incorporates a prior information into the MPE update. In the I-smoothing, the priors are estimated in ML fashion which could be inaccurate if enough data is not available. Therefore, in this case it is preferable to use ML-MAP to estimate more robust priors.

The MPE-MAP operates in two levels. First, new priors are estimated using ML-MAP and unadapted mean $\tilde{\boldsymbol{\mu}}_{jm}$ and variance $\tilde{\boldsymbol{\sigma}}_{jm}$ are used as priors:

$$\boldsymbol{\mu}_{jm}^{map} = \frac{\theta_{jm}^{ML}(\mathbf{O}) + \tau\tilde{\boldsymbol{\mu}}_{jm}}{\gamma_{jm}^{ML} + \tau} \tag{2.39}$$

$$\boldsymbol{\sigma}_{jm}^{map2} = \frac{\theta_{jm}^{ML}(\mathbf{O}^2) + \tau(\tilde{\boldsymbol{\mu}}_{jm}^2 + \tilde{\boldsymbol{\sigma}}_{jm}^2)}{\gamma_{jm}^{ML} + \tau} - \boldsymbol{\mu}_{jm}^{map2} \tag{2.40}$$

Then, the ML-MAP parameters are used as the priors in MPE-MAP update. For the MPE-MAP, mean is given by:

$$\hat{\boldsymbol{\mu}}_{jm} = \frac{\theta_{jm} + D\boldsymbol{\mu}_{jm} + \tau^I\boldsymbol{\mu}_{jm}^{map}}{\gamma_{jm} + D + \tau^I} \tag{2.41}$$

The MPE-MAP is an iterative process where in each stage $\boldsymbol{\mu}_{jm}^{map}$ and $\boldsymbol{\sigma}_{jm}^{map2}$ values re-computed using new ML statistics. It introduces two new values: τ which controls the prior distribution for MPE-MAP estimated by ML-MAP, and τ^I which controls the weight of prior in the discriminative update.

2.5 Language models

The role of language model (LM) is to select the hypothesis which is the most likely the right sequence of speech elements (sentence) of a given language. The complexity of used language model depends on complexity of the problem being solved (continuous speech vs. limited number of commands). Statistical models derived from data are usually used for this purpose (N-grams). HMM based speech recognition provides unified framework, where acoustic and language models are jointly used to find the most probable word sequence (see section 2.2.1).

Although LM are essential for good recognition performance, this thesis does not deal with LM and interested reader can refer to [30, 40, 6] for more information.

Chapter 3

Linear transforms in feature-space

In our experiments described later (chapter 5), linear transforms are used to decorrelate and reduce dimensionality of features.

3.1 Heteroscedastic Linear Discriminant Analysis

The Heteroscedastic Linear Discriminant Analysis [31] can be used to derive linear projection de-correlating feature vectors and performing the dimensionality reduction. For HLDA, each feature vector that is used to derive the transformation must be assigned to a class. When performing the dimensionality reduction, HLDA allows to preserve useful dimensions, in which feature vectors representing individual classes are best separated (Figure 3.1). HLDA allows to derive such projection that best de-correlates features associated with each particular class [31, 15].

To perform de-correlation and dimensionality reduction, n-dimensional feature vectors are projected into first $p < n$ rows, $\mathbf{a}_{k=1...p}$, of $n \times n$ HLDA transformation matrix, $\mathbf{A}$. An efficient iterative algorithm [15] is used in our experiments to estimate matrix $\mathbf{A}$, where individual rows are periodically re-estimated using:

$$\hat{\mathbf{a}}_k = \mathbf{c}_k \mathbf{G}^{(k)-1} \sqrt{\frac{T}{\mathbf{c}_k \mathbf{G}^{(k)-1} \mathbf{c}_k^T}} \tag{3.1}$$

where $\mathbf{c}_i$ is the i^{th} row vector of co-factor matrix $\mathbf{C} = |\mathbf{A}|\mathbf{A}^{-1}$ for the current estimate of $\mathbf{A}$ and

$$\mathbf{G}^{(k)} = \begin{cases} \sum_{j=1}^{J} \frac{\gamma_j}{\mathbf{a}_k \hat{\mathbf{\Sigma}}^{(j)} \mathbf{a}_k^T} \hat{\mathbf{\Sigma}}^{(j)} & k \leq p \\ \frac{T}{\mathbf{a}_k \hat{\mathbf{\Sigma}} \mathbf{a}_k^T} \hat{\mathbf{\Sigma}} & k > p \end{cases} \tag{3.2}$$

where $\hat{\mathbf{\Sigma}}$ and $\hat{\mathbf{\Sigma}}^{(j)}$ are estimates of global covariance matrix and covariance matrix of j^{th} class, γ_j is number of training feature vectors belonging to j^{th} class and T is the total number of training feature vectors.

In our experiments, the classes are defined by each Gaussian mixture component m of each state s. The selection, that feature vector $\mathbf{o}(t)$ belongs to class j, is given by the value of

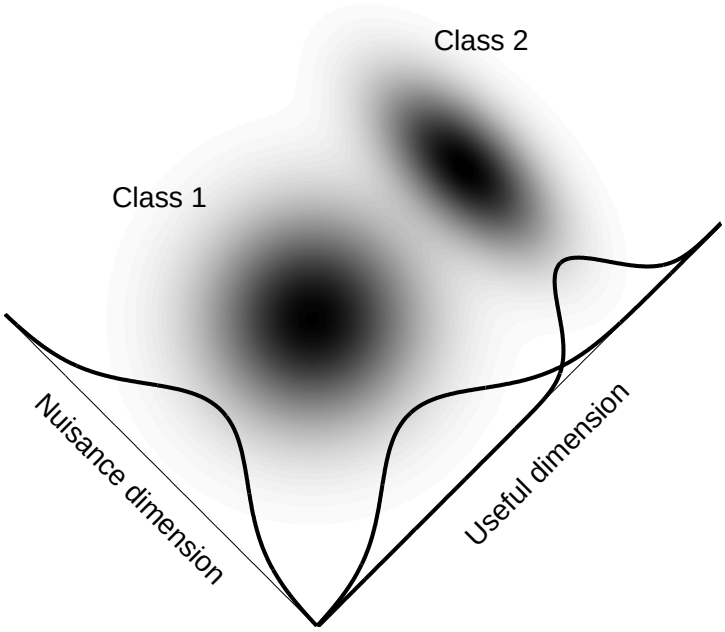


Figure 3.1: Heteroscedastic Linear Discriminant Analysis for 2-Dimensional Data (reproduced from [5]).

occupation probability $\gamma_j(t)$. These occupation probabilities, are used to re-estimate transition probabilities, and mixture component weights according to standard Baum-Welch algorithm. In this algorithm, occupation probabilities, $\gamma_j(t)$, and feature vectors $\mathbf{o}(t)$ are used to estimate n-dimensional mean vector, $\boldsymbol{\mu}_j$, and full covariance $n \times n$ matrix, $\boldsymbol{\Sigma}_j$, of each Gaussian mixture component, j, according to equations:

$$\gamma_j = \sum_{t=1}^{T} \gamma_j(t), \tag{3.3}$$

$$\hat{\boldsymbol{\mu}}_j = \frac{\sum_{t=1}^{T} \gamma_j(t)\mathbf{o}(t)}{\gamma_j}, \tag{3.4}$$

$$\hat{\boldsymbol{\Sigma}}_j = \frac{\sum_{t=1}^{T} \gamma_j(t)(\mathbf{o}(t) - \hat{\boldsymbol{\mu}}_j)(\mathbf{o}(t) - \hat{\boldsymbol{\mu}}_j)^T}{\gamma_j} \tag{3.5}$$

where T is the number of feature vectors used for training. Note that these equations correspond to 2.22 and 2.23 except for estimation of full covariance matrix. New HLDA projection, $\mathbf{A}$, is then derived using the occupation probabilities and the estimated class covariance matrices, $\hat{\boldsymbol{\Sigma}}_j$.

To obtain the correct estimates of HMM parameters in feature space corresponding to the newly derived transformation, $\mathbf{A}_p$, p-dimensional mean vector, $\hat{\boldsymbol{\mu}}_j^{HLDA}$, and variance vector, $\hat{\boldsymbol{\sigma}}_j^{HLDA}$, of each Gaussian mixture component are updated according to:

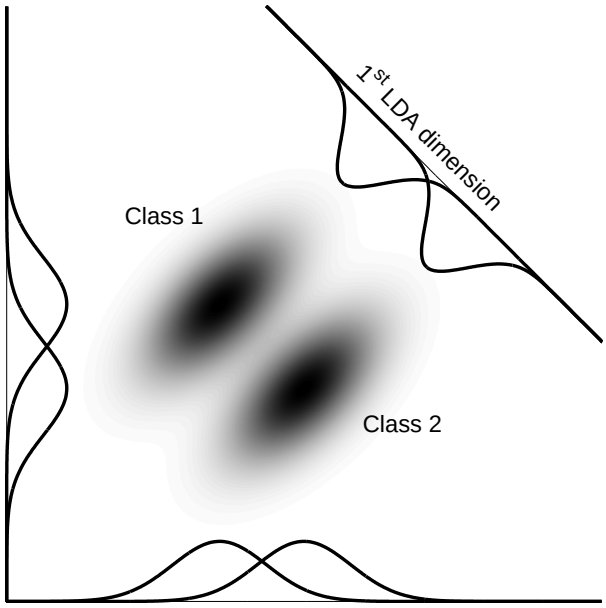


Figure 3.2: Linear Discriminant Analysis for 2-Dimensional Data (reproduced from [5]).

$$\hat{\boldsymbol{\mu}}_j^{HLDA} = \mathbf{A}_p \hat{\boldsymbol{\mu}}_j, \tag{3.6}$$

$$\hat{\boldsymbol{\sigma}}_j^{HLDA} = diag(\mathbf{A}_p \hat{\boldsymbol{\Sigma}}_j \mathbf{A}_p^T), \tag{3.7}$$

where $\mathbf{A}_p$ is matrix consisting of the first p rows of matrix $\mathbf{A}$.

In the special case, where $p = n$ (no dimensionality reduction is performed), HLDA transformation equals to Maximum Likelihood Linear Transform (MLLT) [19], which is also often referred to as diagonalization transform.

Note, that an alternative definition of HLDA (sometime referred to as HDA) proposed by Saon [50] also exists, which is, however, not derived in the maximum likelihood framework.

3.2 LDA

Well known Linear Discriminant Analysis (LDA) can be seen as special case of HLDA, where it is assumed that covariance matrices of all classes are the same (see Figure 3.2). In contrast to HLDA, closed form solution exists in this case.

Base vectors of LDA transformations are given by eigen vectors of matrix $\boldsymbol{\Sigma}_{ac} \times \boldsymbol{\Sigma}_{wc}^{-1}$. The within-class covariance matrix, $\boldsymbol{\Sigma}_{wc}$, which represents the unwanted variability in data, is estimated as weighted average of covariance matrices of all classes:

$$\hat{\boldsymbol{\Sigma}}_{wc} = \frac{1}{N} \sum_j \gamma_j \hat{\boldsymbol{\Sigma}}_j, \tag{3.8}$$

where N is the number of classes.

The across-class covariance matrix $\mathbf{\Sigma}_{ac}$ represents the wanted variability in data and it is computed as the covariance matrix of weighted mean vectors of all classes:

$$\hat{\mathbf{\Sigma}}_{ac} = \frac{1}{T}\sum_{j}\gamma_j(\hat{\boldsymbol{\mu}}_j - \hat{\boldsymbol{\mu}})(\hat{\boldsymbol{\mu}}_j - \hat{\boldsymbol{\mu}})^T = \hat{\mathbf{\Sigma}} - \hat{\mathbf{\Sigma}}_{wc}, \tag{3.9}$$

where $\hat{\boldsymbol{\mu}}$ is the global mean vector and $\hat{\mathbf{\Sigma}}$ is the global covariance matrix.

Again, projection to only several eigen vectors corresponding to largest eigen values can be performed in order to reduce the dimensionality of features.

Chapter 4

Description of recognition system

To get accurate acoustic models, it is important to train on large corpora close to the target task. Meetings should be used for acoustic model training but data in this domain is still sparse. As was already mentioned, meeting speech is spontaneous discussion. The wide-band (WB) 16 kHz sampling rate is typically used for recording. The conversational telephone speech (CTS) speaking style is close to meeting data and training data is more abundant. Therefore, development was made also with CTS data, although it is naturally recorded in 8 kHz, narrow band (NB). Experiments allowing for use of CTS corpora in the training of acoustic models for meeting will be presented later in chapter 7. Hence, two recognition systems were created, one for meeting speech and the second for telephone speech. Both are described later in this chapter.

When the recognition systems were built, meeting and telephone speech test sets were processed by corresponding systems to produce wide lattices. Consequently, newly developed models and approaches could be efficiently tested by acoustic rescoring of these lattices.

4.1 Description of basic HMM training

The features were 13th order ML-PLP cepstral coefficients, including the 0th one, with the first and second derivatives added. This gives a standard 39 dimension feature vector. Cepstral mean and variance normalization was applied (described later in 6.1.1). Cross-word triphones were chosen as the smallest modeled units. Each HMM contained 3 emitting states in the left-to-right topology.

Training transcriptions were created from word level transcriptions and given dictionary. The first pronunciation variant was taken.

To initialize the system, monophone models were first trained from scratch using Baum-Welch algorithm (see section 2.3.3) and mixture splitting. In more detail:

1. Initialize set of 1 Gaussian models by global mean and variance.
2. Run 4 Baum-Welch iterations.
3. Mixture splitting - split 2 mixtures with the highest mixture weight.

4. Go to step 2 if requested number of Gaussian was not reached.

As monophones are used just for initialization, 16 Gaussians mixtures were chosen for output. The final models were fixed and they are shortly called "mono".

In the next step, the transition to triphones was done:

1. Take "mono" models as an input.
2. Merge all non-silence Gaussian mixtures to produce set of 1 Gaussian models.
3. Clone according to all triphones occurring in the training data.
4. Run 4 Baum-Welch iterations and save statistics for decision tree algorithm.
5. Use decision tree clustering to create tied-state triphone (see 2.3.2). The number of resulting tied states was set depending on the amount of training data.
6. HMM training and mixture splitting similar to that for monophones followed. The final models are shortly called "xwrd.1". The 16 Gaussian models were selected again.

The estimation of decision tree statistics in step 4 is given by simple 1-Gaussian models which could be inaccurate. Also, the data transcriptions can be regenerated using forced alignment and the current models. Therefore we increased the accuracy of the system by running the whole triphone training again with two changes:

- Transcriptions were created using forced alignment.
- Step 4 was changed - the 1-Gaussian triphone models from step 3 were updated according to forward-backward procedure which was not based on models from the previous iteration but on the final models from step 6. This technique is called 2-model reestimation.

This process can run iteratively using always output models from the previous iteration, which produces model selections shortly called "xwrd.2","xwrd.3"...

4.2 Conversational telephone speech recognition system

4.2.1 Acoustic models

The CTS system was trained on ctstrain04 training set, a subset of the h5train03 set, defined at the University of Cambridge. It contains about 278 hours of well transcribed speech data from Switchboard I,II and Call Home English (see Table 4.1).

All CTS models selections were tested on the Hub5 Eval01 test set (defined during 2001 NIST CTS evaluation) composed of 3 subsets of 20 conversations from Switchboard-1, Switchboard-2 and Switchboard-cellular corpora, for a total length of more than 6 hours of audio data (see Table 4.2).

Database	Amount of data [hours]	
	h5train03	ctstrain04
Switchboard I	263.61	248.52
Switchboard II - cellular	16.18	15.27
Call Home English	15.77	13.93
Total	295.56	277.72

Table 4.1: CTS training data description.

Database	Amount of data [hours]
Switchboard I	2
Switchboard II	2
Switchboard II - cellular	2.2
Total	6.2

Table 4.2: Description of eval01 test set.

The "basic" models were trained by 4 whole iterations of algorithm in section 4.1. Table 4.3 shows the effect of re-clustering. The final "xwrd.4" basic CTS system contained 7598 tied states and 16 Gaussian mixtures per state.

4.2.2 Language model

The Language model used in the decoding was estimated by interpolation from Switchboard I,II + Call Home English and Hub4 (Broadcast news) transcriptions (see Table 4.4). The size of recognition vocabulary was 50k words.

models	WER [%]
mono	52.6
xwrd.1	37.0
xwrd.2	36.8
xwrd.3	36.7
xwrd.4	36.7

Table 4.3: Effect of reclustering in triphone model training.

Corpus	Count of words	LM weights	
		2gram	3gram
SWB + CallHome	3.5M	0.733	0.639
Hub4 LM96	220M	0.266	0.360

Table 4.4: Number of words and weights per corpus for CTS language model.

4.2.3 System flow-chart

Final eval01 test set lattices were generated by simple system based on "xwrd.4" models (see section 4.2.1) using a bigram language model (see section 4.2.2). Those lattices were further expanded by trigram language model. This is shown in figure 4.1

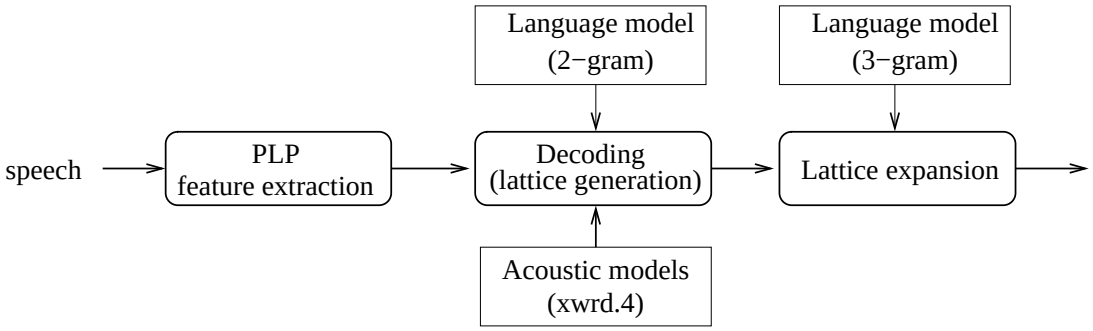


Figure 4.1: CTS lattice generation

4.3 Meeting speech recognition system

The meeting speech recognition system had more sophisticated structure than the CTS one. It was developed for NIST Rich Transcription evaluation 2005. Just brief information will be given in this section. More details can be found in [22].

4.3.1 Acoustic models

The acoustics models were trained on 112h of meeting data collected from all meeting corpora currently available by Linguistic Data Consorcium (LDC), see Table 4.5.

Meeting models were tested on the NIST Rich Transcriptions 2005 test set (rt05) composed of 5 subsets which were made available by AMI, NIST, Interactive System Labs (ISL), International Computer Science Institute (ICSI) and Virginia Polytechnic (VT), see Table 4.6. The total length of all speech segments was about 2h.

Corpora	Size [h]
ICSI meeting corpus	73
NIST data	13
ISL	10
AMI preliminary development set	16
Total	112h

Table 4.5: Description of meeting data for the training.

Database	Amount of data [hours]
AMI	0.39
ISL	0.46
ICSI	0.45
NIST	0.41
VT	0.35
Total	2.06

Table 4.6: Description of rt05 test set.

The "basic" models were trained using similar procedure as the CTS models above but the following advanced techniques were used to obtain more accurate system:

- VTLN - vocal tract length normalization. Speaker normalization technique, it will be described later in section 6.1.2
- SHLDA - Smoothed Heteroscedastic linear discriminant analysis. Maximum likelihood feature transform, it will be described later in section 5.1
- MPE training - Minimum Phone Error discriminative training.

4.3.2 Language model

The Language model used in NIST RT05 system was based on interpolation of various corpora, see Table 4.4. The size of recognition vocabulary was again 50k words.

4.3.3 System flow-chart

The NIST RT05 test data were processed by system shown in figure 4.2.

Several advanced techniques were used:

Corpus	words [MW]
Swbd/CHE	3.5
Fisher	10.5
Web (Swbd)	163
Web (Fisher)	484
Web (Fisher topics)	156
BBC - THISL	33
HUB4-LM96	152
SDR99-Newswire	39
ICSI/NIST/ISL/AMI	1.5
Web(ICSI)	128
Web (AMI)	100
Web (CHIL)	70

Table 4.7: Numbers of words per corpus used for training of meeting language model.

- CMN/CVN - Cepstral Mean and Variance Normalization - simple feature normalization, it will be described later in section 6.1.1
- MLLR - Maximum Likelihood Linear Regression - speaker adaptation technique. It estimates a transform (or set of transforms) which rotate model parameters in order to increase likelihood on adaptation (test) data. It will described in more detail in section 6.2.2.
- Confusion Networks (CN) - The technique converts lattice into the so called "Confusion Network" or "sausage lattice". The confusion network could be imagined as a sequence of "sausages" where each "sausage" contains a list of words with known posterior probability. More details can be found in [37],[36].
- Consensus decoding, also called Minimum word error decoding, picks up the word with highest probability from each "sausage". It optimizes Word Error Rate (WER). This is main advantage of this approach because the decoders itself gives output which optimizes the best Sentence Error Rate (SER).

The system works in six passes:

1. First, the PLP features were extracted from the waveforms and speech activity segments were detected. The first decoding run with "basic" ML trained models and 3-gram LM. It generated one-best output strings.
2. VTLN warping factor for each speaker was searched out by iterative feature coding and aligning of first pass output. VTLN PLP features were generated and second decoding pass run. The same 3-gram LM but more sophisticated VTLN SHLDA MPE trained acoustics models were taken. Therefore, it produced more reliable one-best output string than the first pass.

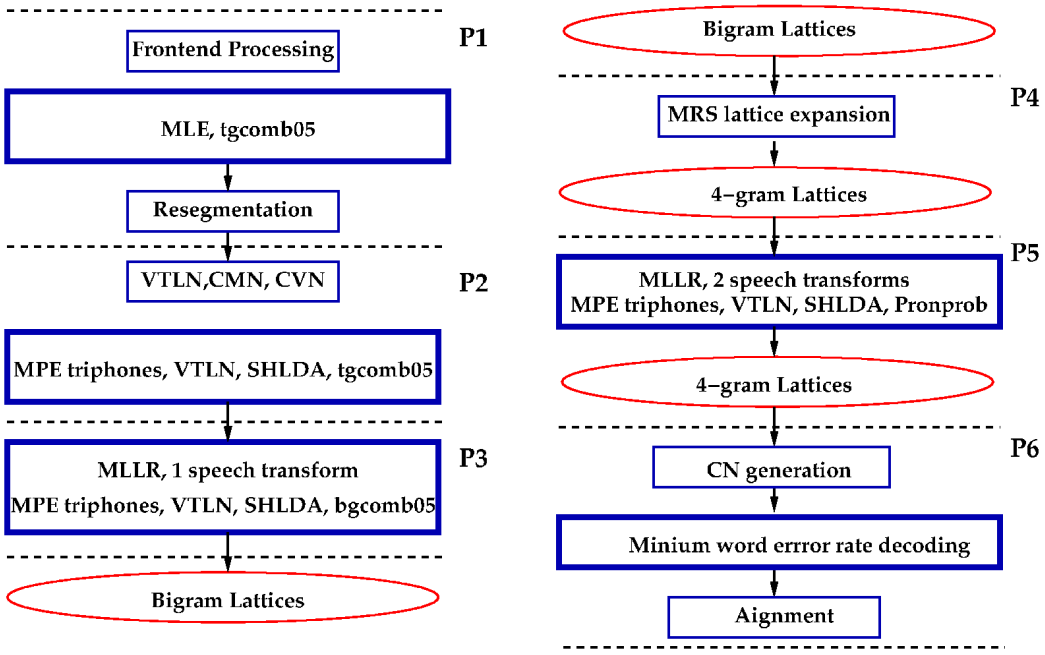


Figure 4.2: AMI RT05 system description (reproduced from [22]).

3. The output from previous pass was used for speaker adaptation based on MLLR. 2 adaptation transforms (silence, voice) were estimated. These transforms were further applied on current VTLN SHLDA MPE models and wide lattices were generated by decoding with 2-gram LM.
4. The 2-gram lattices from previous pass were expanded by more accurate 3-gram and 4-gram Language models.
5. More accurate one-best output generated from the 4-gram lattices was used to reestimate the MLLR transforms. New sets of lattices were generated by acoustic rescoring of current 4-gram lattices using new transforms and VTLN SHLDA MPE models.
6. Confusion networks were generated from final set of lattices. Minimum word error decoding produced the final one-best output string but it did not contain any time information about word boundaries which is required by NIST scoring script. Therefore, forced alignment run to add this time information back.

The lattices used in this work were taken from pass 4.

Chapter 5

Study of the HLDA

HLDA estimation algorithm requires estimation of full covariance statistics for each class. A Gaussian is usually considered as the class in a standard HMM system. This can however lead to noisy estimation of statistics even in case of well tuned system. Therefore, a smoothing techniques will be introduced in this chapter to obtain more robust HLDA estimation.

In our experiments, we added the third derivatives into the PLP feature stream, which gave us 52 dimensional feature vectors. **HLDA** transform was then trained to perform the projection from 52 to 39 dimension. The statistics were projected into the new space and HMM models were updated. A few additional Baum-Welch iterations were run to better settle HMM into the new space.

In chapter 7, we will investigate using of CTS models in meeting system, therefore the development run on both tasks. It is important to know if the best approach for telephone speech generalizes also for meetings. The testing was performed on eval01 test set (for CTS system) and on rt05 test set (for meeting system). For meeting system, VTLN was applied in advance.

Table 5.1 shows a comparison of no HLDA, HLDA without dimensionality reduction (also called Maximum Likelihood Linear Transform) and HLDA system projecting 52 dimensional space to 39.

HLDA approach	CTS system - WER [%]	Meeting system - WER[%]
no HLDA	36.7	30.3
MLLT	35.0	29.0
HLDA	**34.8**	**28.8**

Table 5.1: Comparison of HLDA and MLLT systems on eval01 and rt05 test sets.

5.1 Smoothed HLDA - SHLDA

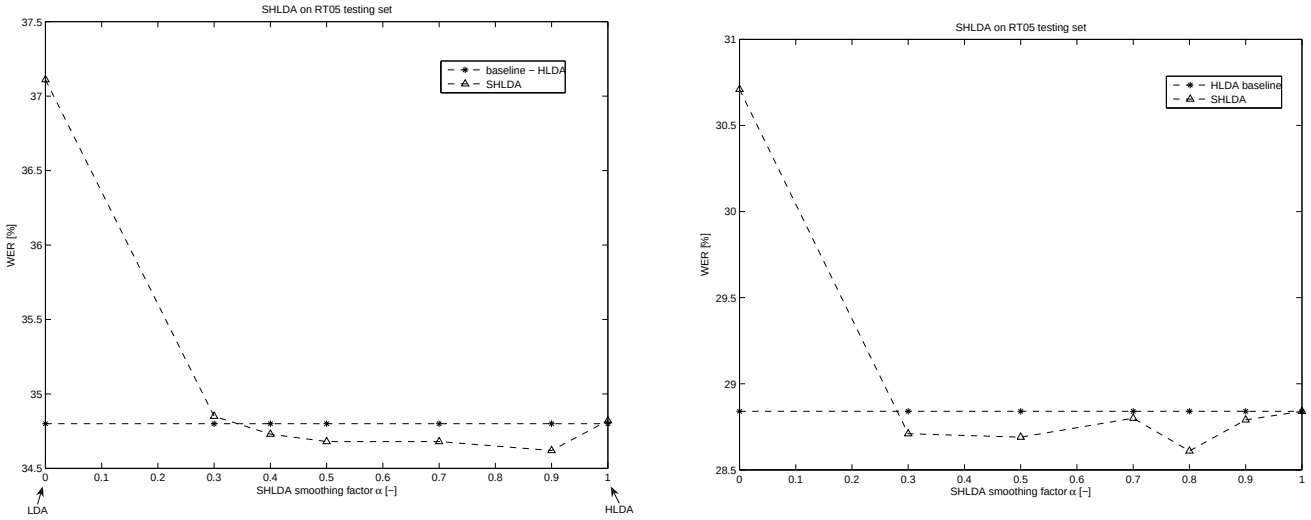
SHLDA is a technique based on combination of HLDA and LDA proposed in [3], where class covariance matrices are estimated more robustly, and at the same time, (at least the major) differences between covariance matrices of different classes are preserved. Smoothed HLDA (SHLDA) differs from HLDA only in the way of estimating class covariance matrices. In the case of SHLDA, the estimate of class covariance matrices is given by:

$$\check{\mathbf{\Sigma}}_j = \alpha\hat{\mathbf{\Sigma}}_j + (1-\alpha)\mathbf{\Sigma}_{WC} \tag{5.1}$$

where $\check{\mathbf{\Sigma}}_j$ is "smoothed" estimate of covariance matrix of class j. $\hat{\mathbf{\Sigma}}_j$ is the original estimate of covariance matrix given by equation 3.5, $\mathbf{\Sigma}_{WC}$ is estimate of within-class covariance matrix (see equation 3.8) and α is smoothing factor — a value in the range of 0 to 1. Note that for α equal to 0, SHLDA becomes LDA and for α equal to 1, SHLDA becomes HLDA.

Results of Smoothed HLDA in CTS and meeting systems

Figure 5.1a shows the dependency of WER on SHLDA smoothing factor α for eval01 test set. Pure LDA failed, probably due to bad assumption of the same Gaussian distribution in all classes. The best system performance 34.6% was obtained for smoothing factor $\alpha = 0.9$. The relative improvement of this system is 7.9% compared to the non-HLDA and 0.6% compared to the clean HLDA setup.



(a) CTS system

(b) Meeting system

Figure 5.1: Dependency of WER on SHLDA smoothing factor α.

The results for RT05 test set are shown in the Figure 5.1b. We obtained similar curves as for CTS system. The best system performance 28.6% was obtained for smoothing factor $\alpha = 0.8$. The relative improvement over the clean HLDA setup was 0.8%.

5.2 MAP smoothed HLDA - MAP-SHLDA

SHLDA gives more robust estimation than standard HLDA but optimal smoothing factor α depends on the amount of data for each class. In extreme case, α should be set to 0 (HLDA) if infinite amount of training data is available. With decreasing amount of data, optimal α value will slide up to LDA direction.

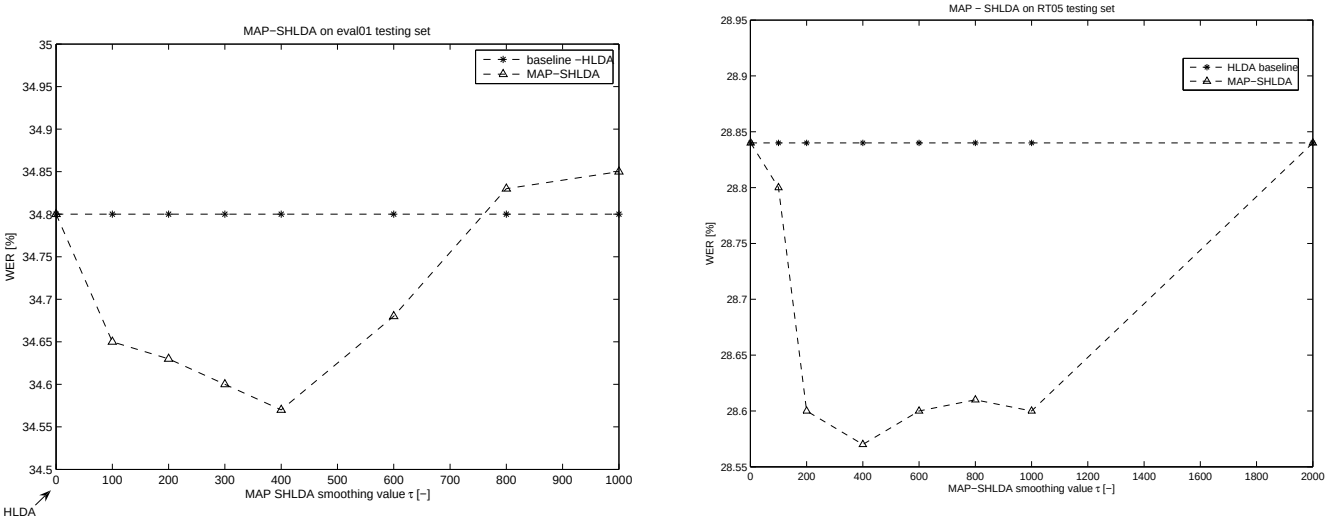
To add more robustness into the smoothing procedure, we defined maximum a posteriori (MAP) smoothing similar to classical MAP adaptation of Gaussian parameters introduced in [17]. The within-class covariance matrix $\mathbf{\Sigma}_{WC}$ is considered as the prior and an estimate of the class covariance matrix is given by:

$$\check{\mathbf{\Sigma}}_j = \mathbf{\Sigma}_{WC}\frac{\tau}{\gamma_j + \tau} + \hat{\mathbf{\Sigma}}_j\frac{\gamma_j}{\gamma_j + \tau} \tag{5.2}$$

where τ is a control constant and γ_j is occupation count for class j. Obviously, if insufficient data is available for current class, the prior resource $\mathbf{\Sigma}_{WC}$ is considered as more reliable than the class estimation $\hat{\mathbf{\Sigma}}_j$. In case of infinite data, only the class estimation of covariance matrix $\hat{\mathbf{\Sigma}}_j$ is used for further processing.

MAP-SHLDA results for CTS and meeting systems

Figure 5.2a shows results of this technique for CTS eval01 test set. The best performance 34.6% was obtained again for $\tau = 400$, which is 0.7% relative improvement compared to the clean HLDA setup.

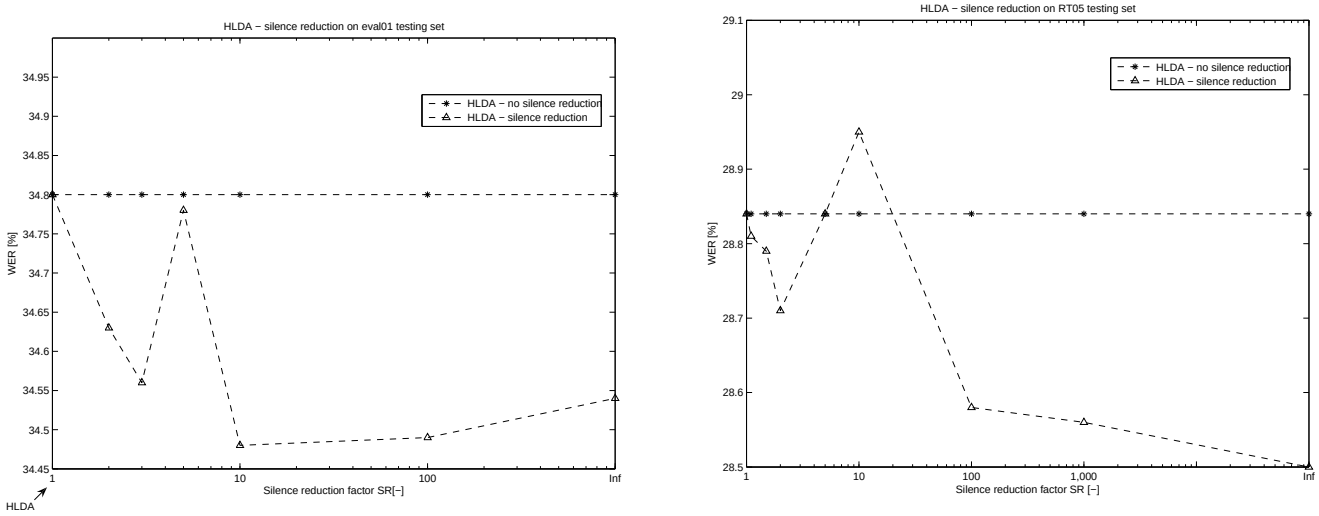


(a) CTS system

(b) Meeting system

Figure 5.2: Dependency of WER on MAP-SHLDA smoothing factors.

Figure 5.2b show results of this technique for meeting RT05 test set. The best performance



(a) CTS system

(b) Meeting system

Figure 5.3: Dependency of WER on Silence Reduction factor SR.

28.6% was obtained for $\tau = 400$. The relative improvement of this system is 0.9% compared to the clean HLDA setup.

5.3 Silence Reduction in HLDA estimation - SR-HLDA

From the point of view of transformation estimation, silence is a "bad" class as its distributions differ significantly from all speech classes. Moreover, training data (even if end-pointed) contains significant proportion of silence. An estimation of two HLDA transforms solves this problem but it makes the implementation more difficult.

Rather than discarding the silence frames, the occupation counts, γ_j, of silence classes ζ, which take part in computation of HLDA estimation in equation 3.2, are scaled by factor $1/SR$.

$$\hat{\gamma}_j = \frac{\gamma_j}{SR} \quad \text{if } j \in \zeta \tag{5.3}$$

$SR = \infty$ corresponds to complete elimination of silence statistics.

Results with SR-HLDA in CTS and meeting systems

Figure 5.3a presents nice effect of silence reduction on eval01 test set. The best result is 34.5%, which is 0.9% relative improvement over the standard HLDA. It was obtained by the Silence Reduction factor $SR = 10$.

Figure 5.3b shows the results of silence reduction in HLDA system on rt05 test set. The elimination of all silence statistics gives the best performance 28.5%. This is 1.2% relative improvement over the standard HLDA.

System	CTS system - WER [%]	Meeting system - WER [%]
(no HLDA)	36.71	30.30
standard HLDA	34.80	28.84
SHLDA	34.62	28.61
MAP-SHLDA	34.57	28.57
SR-HLDA	34.48	28.50

Table 5.2: Comparison of HLDA systems on eval01 and RT05 test sets.

This approach is giving quite good results but for some SR values it causes a degradation of performance. Removing all silence statistics gives good results for both tests without any need of tuning.

5.4 Summary of HLDA results

Table 5.2 summarizes the performances of all already presented techniques. Smoothed HLDA (SHLDA) and MAP variant of SHLDA, taking into account the amounts of data available for estimation of statistics for different classes, perform both better than the basic HLDA. We have however found, that removing the silence class from HLDA statistics (Silence-reduced HLDA) is equally effective and cheaper in computation. Testing SHLDA and MAP-SHLDA on the top of SR-HLDA did not bring any further improvement, therefore we stick with SR-HLDA, especially with complete silence removal, as the most suitable transformation in our next LVCSR experiments.

Chapter 6

Normalization and adaptation techniques for LVCSR

Speech recorded in real environment differs from clean speech. The technical side of the problem includes microphone and channel variations. Non-native speech is expected in meetings which also produces huge speaker variance. Therefore, robustness is the major issue for our task. In this chapter, basic channel and speaker normalization techniques will be described as well as more advanced speaker adaptations.

6.1 Feature normalization

6.1.1 Cepstral mean and variance normalization

Published works [45] show a negative impact of noise to the speech recognition due to mean shift and variance reduction in the distribution of cepstral features.

Cepstral mean and variance normalization is well know normalization technique to decrease the sensitivity of cepstral features to channel distortions. The mean and variance normalized feature vector $\hat{\mathbf{o}}_k(t)$ is computed according to:

$$\hat{\mathbf{o}}_k(t) = \frac{\mathbf{o}_k(t) - \hat{\boldsymbol{\mu}}_k}{\hat{\boldsymbol{\sigma}}_k}, k = 1..K. \tag{6.1}$$

Where k is index of speech segment and K is total number of segments. $\hat{\boldsymbol{\mu}}_k$ and $\hat{\boldsymbol{\sigma}}_k$ are estimates of the mean and standard deviation of cepstral features over the segment k. Usually, k is given by index of each sentence generated by speech/non-speech segmenter. But previous experiments on the conversational telephone speech show better performance obtained with $\hat{\boldsymbol{\mu}}_k$ and $\hat{\boldsymbol{\sigma}}_k$ estimated on the whole conversation side due to poor mean and variance estimates in some short segments [24].

models	WER [%]
baseline	36.7
baseline + VTLN on test	36.0
VTLN train + VTLN on test	34.2

Table 6.1: WER reduction by VTLN on eval01.

6.1.2 Vocal tract length normalization

Vocal tract length normalization (VTLN) is speaker based normalization based on warping of frequency axis by speaker dependent warping factor [32]. It is typically made by simple multiplication of frequency and warping factor.

The normalization of vocal tract among speakers has strong positive effect in reduction of cross-speaker variability. The warping factor is typically found by search procedure which compares likelihoods at different warping factors: the features are repeatedly coded using all warping factors in a search range, typically 0.8-1.2. After each coding, forced alignment is used to measure the likelihood. The warping factor producing the best likelihood is then chosen.

The VTLN can by applied in both training and test. Obviously, applying VTLN in both training and recognition leads to the best performance, as we are using "sharp" speaker invariant models. Table 6.1 shows the effect of VTLN: 7% relative improvement against baseline is reached using VTLN training and testing and only 2% relative gain is shown if VTLN is applied on test data only.

The common way to cope with difference of male and female speakers are gender-dependent (GD) HMMs. But it causes a data split in the HMM training and the complexity of recognition system increases. VTLN minimizes gender differences across data. Consequently, the need of GD models in VTLN system is disputable. Moreover, experiments show that system trained on all the data can outperform gender-dependent system if VTLN is applied [20]. No need of GD models is additional advantage which significantly simplifies the system.

Figure 6.1 shows male/female histograms of warping factors on CTS training set. The strong dependency of VTLN on gender is obvious.

Our system does not use gender dependent models and it is trained on all the data.

6.2 Adaptation

Speech recognition systems can be dived into two categories:

- "speaker-independent" - trained on a big amount of data containing recordings of many speakers.
- "speaker-dependent" - trained on one speaker data.

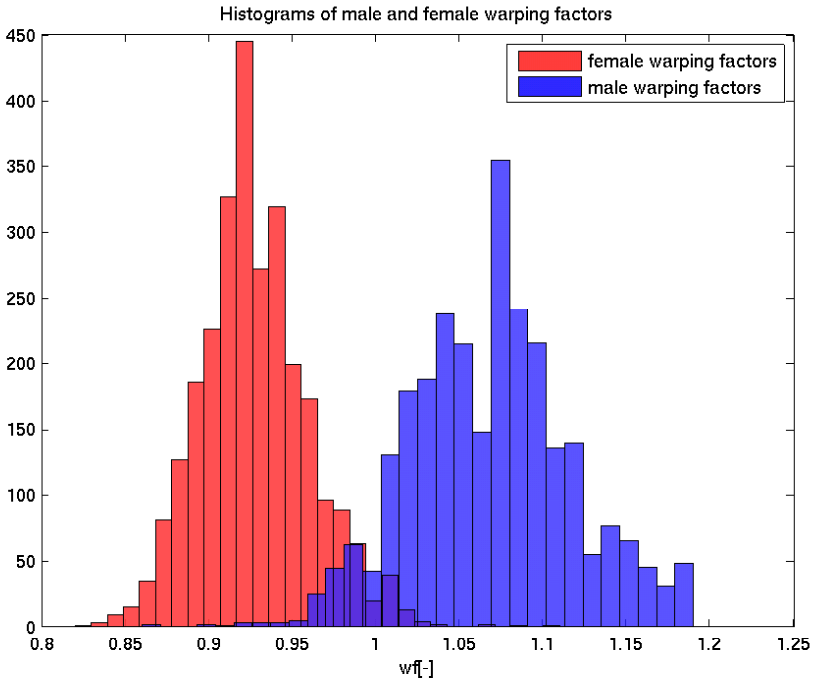


Figure 6.1: Histograms of male and female warping factors on the CTS training set.

A "speaker-dependent" system is more accurate but the problem is the sparsity of training data. It is usually not possible to train speaker-dependent models because the standard training techniques (section 2.2) expect enough data to estimate HMM parameters. Common solution is to **adapt** speaker-independent models on speaker-specific data.

In this work, two different adaptation principles are used: maximum a posteriori (MAP) approach and adaptation using linear transforms. Therefore, a quick introduction will be given in the following sections.

6.2.1 Maximum a posteriori (MAP) adaptation

Maximum a posteriori approach incorporates prior knowledge about model distribution into the training process. This technique is introduced and well described in [17], which presents an implementation of MAP into the standard maximum likelihood estimation of HMM parameters using EM algorithm.

The solution for mean updates turns to be simply a weighted sum of prior means and observed data. The update formula for state j and mixture component m is:

$$\hat{\boldsymbol{\mu}}_{jm} = \frac{\gamma_{jm}}{\gamma_{jm} + \tau}\bar{\boldsymbol{\mu}}_{jm} + \frac{\tau}{\gamma_{jm} + \tau}\boldsymbol{\mu}_{jm} \tag{6.2}$$

where τ is the controlling value (the weight of the prior knowledge to the adaptation data) and γ_{jm} is sum of the occupations likelihoods computed by standard forward-backward algorithm.

It represents the amount of adaptation data for state j and mixture m and is defined defined as:

$$\gamma_{jm} = \sum_{t=1}^{T} \gamma_{jm}(t).$$

$\boldsymbol{\mu}_{jm}$ is prior mean and $\bar{\boldsymbol{\mu}}_{jm}$ is the mean of the observed adaptation data $\mathbf{o}(t)$:

$$\bar{\boldsymbol{\mu}}_{jm} = \frac{\sum_{t=1}^{T} \gamma_{jm}(t)\mathbf{o}(t)}{\gamma_{jm}} \tag{6.3}$$

In the adaptation process, the speaker-independent parameters are chosen as the prior parameters and speaker-dependent data represents observed adaptation data.

The MAP approach can be used in various training techniques. In this work, MAP approach is mainly used later in chapter 7 to allow for using of CTS models in training of the meeting system.

6.2.2 Maximum likelihood linear regression (MLLR)

MLLR is linear transform operating in the space of model parameters which maximizes the likelihood of the adaptation data. On contrary to MAP, where only parameters which have enough data are adapted, MLLR estimates a transform or set of transforms to rotate **all** model parameters. Therefore, less data is needed to make this technique effective.

The transform is estimated by optimization of the following EM criterion [9, 34]:

$$\begin{aligned} Q(M, \hat{M}) = K- \\ \frac{1}{2} \sum_{m=1}^{M} \sum_{t=1}^{T} \gamma_m(t)[K_m + log(|\hat{\boldsymbol{\Sigma}}_m|) + \\ +(\mathbf{o}(t) - \hat{\boldsymbol{\mu}}_m)^T \hat{\boldsymbol{\Sigma}}_m^{-1}(\mathbf{o}(t) - \hat{\boldsymbol{\mu}}_m)], \end{aligned} \tag{6.4}$$

where $\hat{\boldsymbol{\mu}}_m$ and $\hat{\boldsymbol{\Sigma}}_m$ are transformed mean and variance for Gaussian component m, M is the total number components, $\gamma_m(t)$ is probability of being in component m at time t. K is a constant depending only on the transition probabilities. K_m is a normalization constant associated with component m. $\mathbf{O}_T = \mathbf{o}(1).....\mathbf{o}(T)$ is the adaptation data.

The transformed mean $\hat{\boldsymbol{\mu}}_m$ and variance $\hat{\boldsymbol{\Sigma}}_m$ are:

$$\hat{\boldsymbol{\mu}}_m = \mathbf{A}_m \boldsymbol{\mu}_m + \mathbf{b}_m, \tag{6.5}$$

$$\hat{\boldsymbol{\Sigma}}_m = \mathbf{H}_m \boldsymbol{\Sigma}_m \mathbf{H}_m^T \tag{6.6}$$

where $\mathbf{A}_m$ and $\mathbf{H}_m$ are transformation matrices and $\mathbf{b}_m$ is adaptation bias of mean $\boldsymbol{\mu}_m$ tied with mixture m.

In this thesis, estimation of mean adaptation transform only will be given. The adaptation of variances can be found in [16].

If transformation matrices were specific for each Gaussian parameter, it would cause complete reestimation. The problem of unseen adaptation data would not be solved. Therefore, the transforms are tied among the Gaussians, so if some adaptation data are unavailable, the transform is still applied. The classes could be created manually (for example silence/speech) or Gaussians can be grouped using a regression class tree [33]. If small amount of adaptation data is available, all parameters can share the same transforms.

To decrease the number of matrix parameters needed to be estimated and also the amount of adaptation data required, a block diagonal matrix can be used instead of full transformation matrix:

$$\mathbf{A} = \begin{pmatrix} \mathbf{A}_s & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_\Delta & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{A}_{\Delta^2} \end{pmatrix} \tag{6.7}$$

The sub-matrix $\mathbf{A}_s$ is specific to static parameters, $\mathbf{A}_\Delta$ for deltas and $\mathbf{A}_{\Delta^2}$ for accelerations, and the rest is set to zero.

Estimation of the Mean transform

Equation 6.5 can be rewritten to

$$\hat{\boldsymbol{\mu}}_m = \mathbf{W}_m \boldsymbol{\xi}_m \tag{6.8}$$

where

$$\mathbf{W}_m = \begin{bmatrix} \mathbf{A}_m & \mathbf{b}_m \end{bmatrix} \tag{6.9}$$

$$\boldsymbol{\xi}_m = \begin{bmatrix} 1 & \boldsymbol{\mu}_m^T \end{bmatrix}^T \tag{6.10}$$

The solution for particular transform $\mathbf{W}_m$ which is shared among R Gaussians $\{m_1, \ldots, m_R\}$ can be found by differentiating the criterion in 6.4 w.r.t $\mathbf{W}_m$ and setting it to zero. The details can be found in [34]. This leads to general equation for estimation of the transformation matrix $\mathbf{W}_m$:

$$\sum_{t=1}^{T} \sum_{r=1}^{R} \gamma_{m_r}(t) \boldsymbol{\Sigma}_{m_r}^{-1} \mathbf{o}(t) \boldsymbol{\xi}_{m_r}^T = \sum_{t=1}^{T} \sum_{r=1}^{R} \gamma_{m_r}(t) \boldsymbol{\Sigma}_{m_r}^{-1} \mathbf{W}_m \boldsymbol{\xi}_{m_r} \boldsymbol{\xi}_{m_r}^T \tag{6.11}$$

The left side of the equation 6.11 is independent on the transformation matrix $\mathbf{W}_m$ and it will be referred to as $\mathbf{Z}$:

$$\mathbf{Z} = \sum_{t=1}^{T} \sum_{r=1}^{R} \gamma_{m_r}(t) \boldsymbol{\Sigma}_{m_r}^{-1} \mathbf{o}(t) \boldsymbol{\xi}_{m_r}^T \tag{6.12}$$

The right side of the equation 6.11 will be referred to as $\mathbf{Y}$ and it could be further rewritten to:

$$\mathbf{Y} = \sum_{r=1}^{R} \mathbf{V}^{(r)} \mathbf{W}_m \mathbf{D}^{(r)} \tag{6.13}$$

where $\mathbf{V}^{(r)}$ is inverse covariance matrix weighted by occupation probabilities:

$$\mathbf{V}^{(r)} = \sum_{t=1}^{T} \gamma_{m_r}(t) \boldsymbol{\Sigma}_{m_r}^{-1} \tag{6.14}$$

and $\mathbf{D}^{(r)}$ is outer product of the extended mean vectors $\boldsymbol{\xi}_{m_r}$:

$$\mathbf{D}^{(r)} = \boldsymbol{\xi}_{m_r} \boldsymbol{\xi}_{m_r}^{T} \tag{6.15}$$

If the elements of matrices $\mathbf{Y}$, $\mathbf{V}^{(r)}$, $\mathbf{W}^{(r)}$ and $\mathbf{D}^{(r)}$ are denoted by y_{ij}, v_{ij}, w_{ij} and d_{ij}, equation 6.13 can be rewritten as:

$$y_{ij} = \sum_{p=1}^{n} \sum_{q=1}^{n+1} w_{pq} \left[\sum_{r=1}^{R} v_{ip}^{(r)} d_{qj}^{(r)} \right] \tag{6.16}$$

Generally, HMMs uses diagonal covariance and D is symmetric, therefore:

$$\sum_{r=1}^{R} v_{ip}^{(r)} d_{qj}^{(r)} = \begin{cases} \sum_{r=1}^{R} v_{ii}^{(r)} d_{qj}^{(r)} & \text{when } i = p \\ 0 & \text{when } i \neq p \end{cases}. \tag{6.17}$$

Therefore,

$$y_{ij} = \sum_{q=1}^{n+1} w_{iq} g_{jq}^{(i)}, \tag{6.18}$$

where $g_{jq}^{(i)}$ are elements of matrix $\mathbf{G}^{(i)}$ given by,

$$\mathbf{G}^{(i)} = \sum_{r=1}^{R} v_{ii}^{(r)} \mathbf{D}^{(r)} \tag{6.19}$$

The $\mathbf{G}^{(i)-1}$ and $\mathbf{Z}$ are completely independent from transformation matrix $\mathbf{W}_m$ and both can be computed from occupation probabilities and model parameters. Therefore, i'th row of matrix $\mathbf{W}_m$, can be directly computed from the following equation by Gaussian elimination or LU decomposition:

$$\mathbf{w}_i^T = \mathbf{G}^{(i)-1} \mathbf{z}_i^T \tag{6.20}$$

where $\mathbf{z}_i$ is i'th row of matrix $\mathbf{Z}$.

Multiple iterations of MLLR

The estimation of adaptation matrix depends on Gaussian posterior probability $\gamma_m(t)$ on adaptation data. See equation 6.11 or equations 6.14 and 6.12 with statistics $\mathbf{V}$ and $\mathbf{Z}$ needed for MLLR update. Therefore multiple iterations can improve MLLR estimation [33]. The adaptation transforms from previous iteration are used to estimate better Gaussian alignment $\gamma_m(t)$ for the current update.

6.2.3 Constrained Maximum likelihood linear regression

MLLR allows different transforms for means and variances but in **Constrained MLLR**, mean and variance transforms are constrained to be the same [10]:

$$\hat{\boldsymbol{\mu}} = \mathbf{A}'\boldsymbol{\mu} + \mathbf{b}', \tag{6.21}$$

$$\hat{\boldsymbol{\Sigma}} = \mathbf{A}'\boldsymbol{\Sigma}\mathbf{A}'^{T} \tag{6.22}$$

Then, the EM criterion in equation 6.4 can be rewritten to:

$$\begin{aligned} Q(M, \hat{M}) = K - \frac{1}{2}\sum_{m=1}^{M}\sum_{t=1}^{T}\gamma_m(t)[K_m + \\ + log(|\boldsymbol{\Sigma}_m|) - log(|\mathbf{A}|^2) + \\ + (\hat{\mathbf{o}}(t) - \boldsymbol{\mu}_m)^T\boldsymbol{\Sigma}_m^{-1}(\hat{\mathbf{o}}(t) - \boldsymbol{\mu}_m)], \end{aligned} \tag{6.23}$$

where

$$\hat{\mathbf{o}}(t) = \mathbf{A}'^{-1}\mathbf{o}(t) + \mathbf{A}'^{-1}\mathbf{b}' = \mathbf{A}\mathbf{o}(t) + \mathbf{b}. \tag{6.24}$$

The solution of equation 6.23 can be found in [10, 9, 14].

It is obvious that CMLLR can be applied online by transforming the features only. Therefore, there is no need to transform model parameters [14] which is significant advantage of this method. Note, that this attribute of CMLLR causes that it is also called feature-space MLLR (FMLLR).

6.3 Speaker Adaptive Training

Speaker adaptive training (SAT) is a technique used to suppress cross-speaker variance [1]. The original algorithm estimates a set of MLLR transforms to adapt global model to the speaker-dependent training data. These transforms are further applied during the training. But this is not straight-forward in practical implementation due to storage of all Gaussian statistics for each speaker in memory during the iteration [38].

The application of CMLLR instead of MLLR, which we used, solves this problem. The features are online transformed by speaker specific transform and no additional statistics need to be kept [14].

The speaker adaptive training can be easily described in the following steps:

1. Take speaker-independent models as the input.
2. Speaker-dependent CMLLR transforms are estimated using the input models and training data.
3. The input models are further retrained with CMLLR transforms applied.
4. The accuracy in the last iteration is checked.
5. Models from the last iteration are taken as input models and the process is repeated from step 2 until the accuracy stabilizes.

Chapter 7

Narrow band - wide band adaptation

As was already mentioned, the amount of training data has a crucial effect on the accuracy of HMM-based meeting recognition systems but data in the meeting domain is still sparse. The common approach is to use other corpora for the training of acoustic models. One possibility to improve the system performance is to perform adaptation of models trained on considerably larger amounts of data. Typical domains with large amounts of recorded material are broadcast news (BN) or conversational telephone speech (CTS). This data differs from the meeting domain, so one would try to adapt to either different recording environments or to different speaking style. As the speaking style is often the cause for greater variability, adaptation to database with similar speaking style is preferred. Hence for the meeting domain, adaptation of models trained on CTS data is appropriate [11].

Conversational telephone speech speaking style matches well with meetings but CTS is naturally recorded with low bandwidth. Therefore, an adaptation to meeting domain is not trivial as the standard bandwidth for meeting recordings is 16 kHz (wide-band, WB).

7.1 Adaptation of CTS model to downsampled data (NB-NB)

The intuitive way to circumvent the problem of different band-widths is to downsample meeting data to NB and use directly the CTS models, see Figure 7.1.

The downsampled meeting data still significantly differs from the CTS, mainly in channel parameters. Therefore, **adaptation** of CTS models to downsampled meeting training data significantly improves the system. Figure 7.2 presents MAP adaptation to the downsampled data.

7.1.1 Experiments with downsampling

To find out the degradation caused by downsampling the data, the HMMs were also trained on downsampled meeting training data. The comparison is shown in the first two lines of Table 7.1. A degradation of 0.4% can be seen. Direct decoding of downsampled test data with

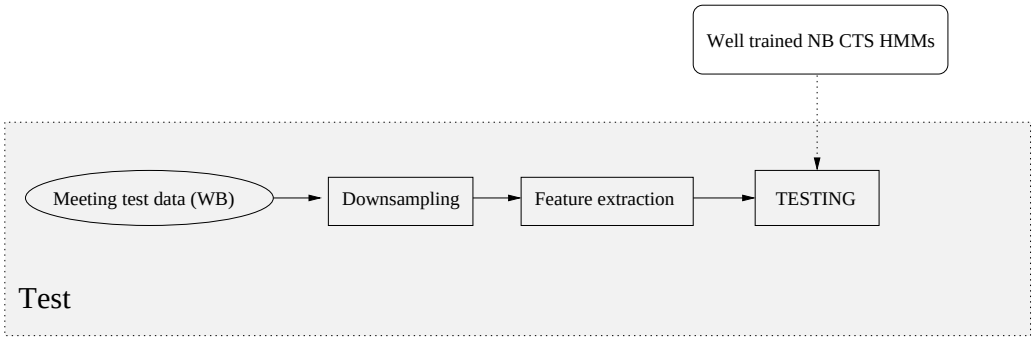


Figure 7.1: Simple system based on downsampling of WB data and CTS models.

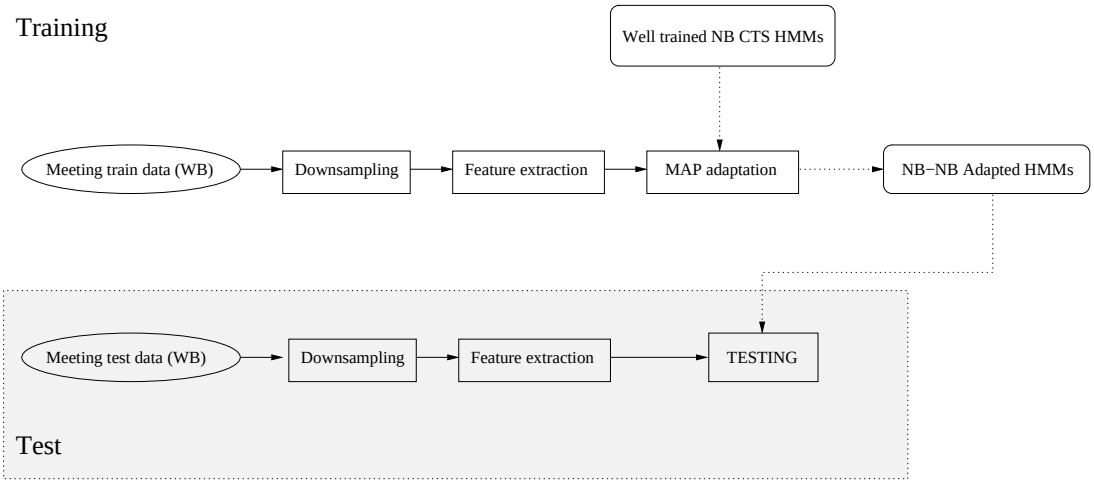


Figure 7.2: Simple system based on downsampling of WB data and adapted CTS models.

CTS models (Figure 7.1) does not perform well probably due to data mismatch - 2.2% worse than WB meeting system. But the adapted CTS system improves this result significantly. We tried to use just MAP adaptation (Figure 7.2) and also a cascade of MLLR followed by MAP adaptation, which outperforms WB baseline by 0.8% absolute.

The main **disadvantage** of this approach is the loss of the upper band (4-8 kHz) while it is known to contain useful information [23]. The solution will be given in the next section.

7.2 Introduction into NB-WB adaptation

The loss of information by downsampling of WB data can be solved by global transformation based on Maximum Likelihood Linear Regression (MLLR) to perform a NB to WB conversion and its application on CTS models instead of downsampling the data. With this approach, even though the upper band information cannot be recovered for CTS data, we can still make use of the richer information in actual target domain recordings.

Training set	Adaptation	WER [%]
WB meeting	none	30.3
NB meeting	none	30.7
CTS	none	32.5
CTS-NB	MAP	29.8
CTS-NB	MLLR MAP	29.5

Table 7.1: Performance of non-adapted and downsampled systems.

Once the CTS models are rotated into the WB domain, it is possible to use any adaptation technique to better settle the rotated CTS models into the WB domain. We used an Maximum a Posteriori (MAP) adaptation [17]. The basic idea of this process is shown in Figure 7.3.

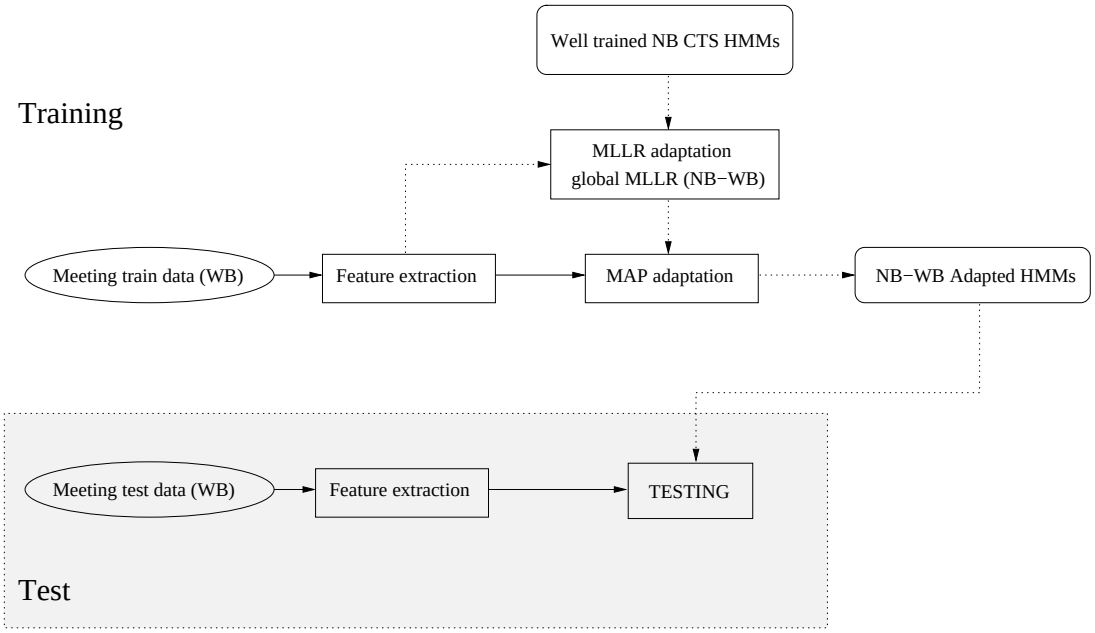


Figure 7.3: NB→WB adapted system using MLLR.

7.3 MLLR as a transformation between wide-band and narrow-band

The initial MLLR NB→WB single transform was estimated using CTS models and WB data. This is, however, not too accurate due to data mismatch. Therefore, the iterative MLLR (see section 6.2.2) was used. Therefore, the process run iteratively and the following iterations

use MLLR NB→WB from previous iteration to obtain better alignment for estimating MLLR statistics.

Table 7.2 shows the performance of CTS models rotated by various kinds of NB→WB MLLR transforms. The "single" represents one single transform, "speech/non-speech" represents two transforms estimated one for speech, one for silence. Experiments to compare block-diagonal and full transformation matrices are reported too. Adding more transforms than one global gives no improvement. After quite some iterations (12-16) the process stabilizes. The full transformation matrix gives 0.3% absolute improvement.

MLLR iteration	WER [%]		
	single block-diag	speech/non-speech block-diag	single full
4	35.2	35.2	35.3
8	34.4	34.3	34.0
12	34.1	34.1	33.9
16	34.1	34.1	**33.8**
20	-	-	33.9

Table 7.2: NB→WB - Performance of CTS models on the WB meeting data with various kinds and quality of NB→WB MLLR.

When the NB→WB transformation is applied on the CTS models, it is possible to use any adaptation technique to adapt the CTS models into the transformed space.

Figure 7.4 shows the performance using standard MAP with a fixed prior using full MLLR matrix from 16th iteration. It runs iteratively - models from previous iteration were used to estimate statistics for current iteration till the process converged.

In our experiments, we use MAP adaptation applied iteratively, so output HMMs from previous iteration are taken as a prior for the current iteration [23]. The first iteration is standard MAP. The MLLR-adapted CTS prior is used to align WB data, which is not very accurate. The solution is to use the models from the previous iteration similarly as above. But we tried to go further - take models from previous iteration as the MAP prior. This gives better Gaussian alignment and also smoother convergence to WB system. There is however a risk of overtraining, so the optimal τ value has to be set higher than in standard MAP; the number of iterations then provides the adaptation control. This approach changes the tuning of adaptation control value τ to just finding the best performing iteration.

Figure 7.5 shows the MAP performance with different τ values and fixed block-diagonal MLLR transform from 12th iteration. We can see that the value of τ is not too important, but higher values give more smooth convergence to the best WER. Beyond the optimum number of iterations, the system tends to be overtrained. In comparison with standard MAP, this gives a bit better results, more than 0.1% absolute.

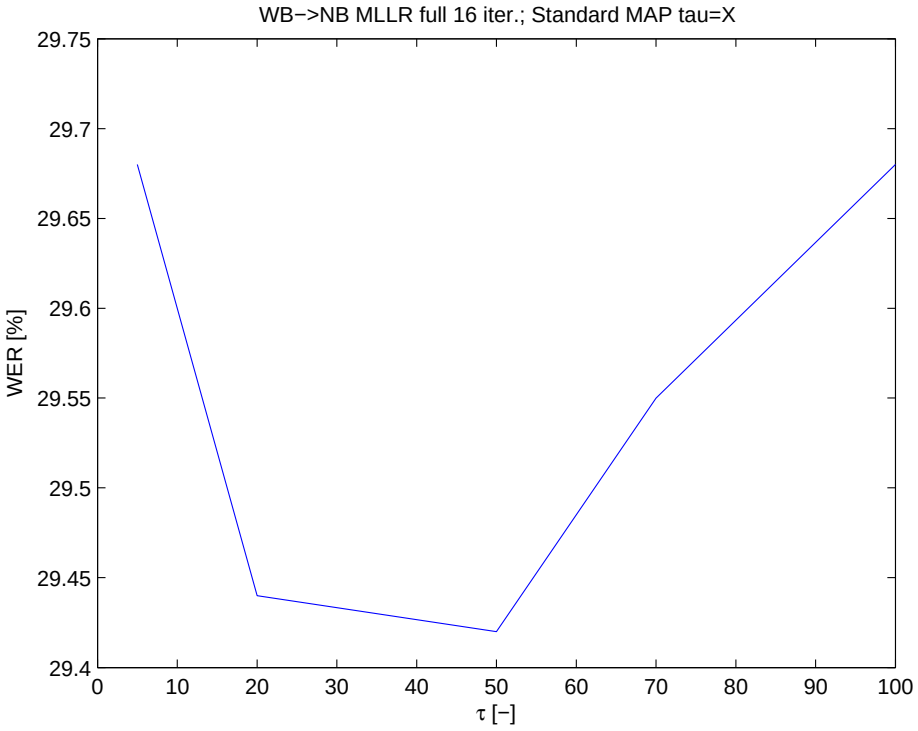


Figure 7.4: NB→WB - τ constant in standard MAP with full MLLR matrix taken from 16th iteration.

The comparison of full MLLR matrix and block-diagonal one is shown in Table 7.3. A 0.3% improvement given by full transformation matrix is lost in MAP adaptation process. We found this a bit disappointing, but the overall results of MLLR-MAP combination are very good.

7.3.1 Summary of MLLR NB→WB adaptation

The comparison with traditional non-adapted system is summarized in Table 7.4. We can see that MLLR WB→NB adapted systems outperform the best downsampled system by 0.2%.

Although promising results were obtained, there is a main **disadvantage**: MLLR is difficult to implement with advanced techniques such as Heteroscedastic Linear Discriminative Analysis (HLDA) (section 3.1) and Speaker Adaptive Training (SAT) (section 6.3). Therefore we propose a single constrained MLLR (CMLLR) (section 6.2.3) for **feature space transformation**[1] from WB to NB. This method allows for straight forward implementation and requires little extra computation in decoding.

[1]Note, that the transforms proposed above transformed the model parameters, not features.

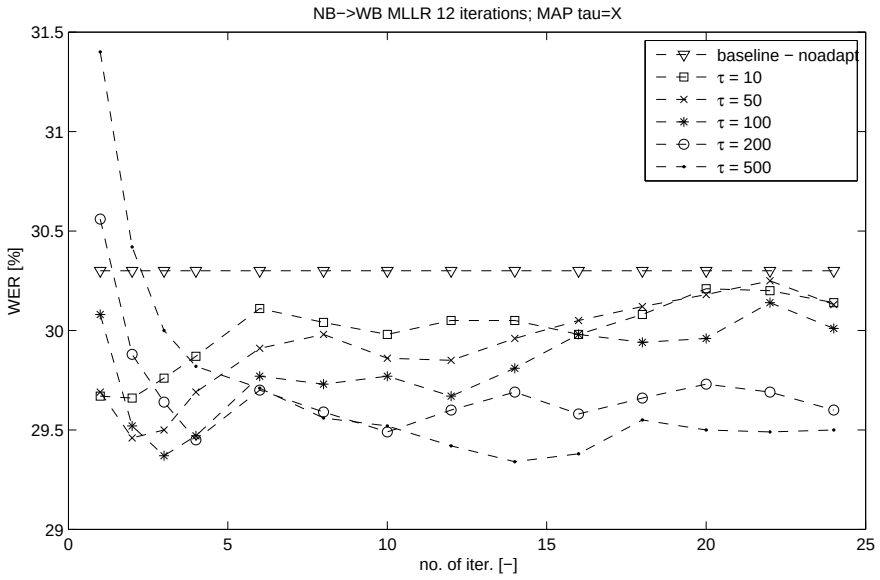


Figure 7.5: NB→WB - τ constant in the iterative MAP with block-diagonal single MLLR taken from 12 iteration.

	Adaptation	
	MLLR	MLLR MAP
block-diagonal	34.1	**29.3**
full	33.8	**29.3**

Table 7.3: NB→WB - Comparison of MLLR and MLLR-MAP with block-diagonal and full MLLR matrix.

7.4 CMLLR as a transformation between wide-band and narrow-band

The CMLLR transformation is estimated to adapt CTS models to meeting WB data. This can equally be interpreted as a projection of WB meeting data into NB CTS domain. This does not seem an obvious choice as one constrains the increased richness of WB meeting data. However, the alternative, i.e. transforming NB CTS data into the WB space, clearly can only add distortion, but add no information. Hence better model training on the larger amounts of data is given priority. Using a transformation matrix to make the meeting data more like CTS data may however preserve some of the characteristics only visible with higher bandwidth. The basic idea of this process is shown in Figure 7.6.

The initial CMLLR WB→NB single transform was estimated in the same way as MLLR

Training set	Adaptation	WER [%]
WB	none	30.3
CTS-NB	MLLR MAP	29.5
CTS-WB	MLLR$_{NB \rightarrow WB}$,MAP	**29.3**

Table 7.4: Comparison of downsampled and WB→NB MLLR MAP systems.

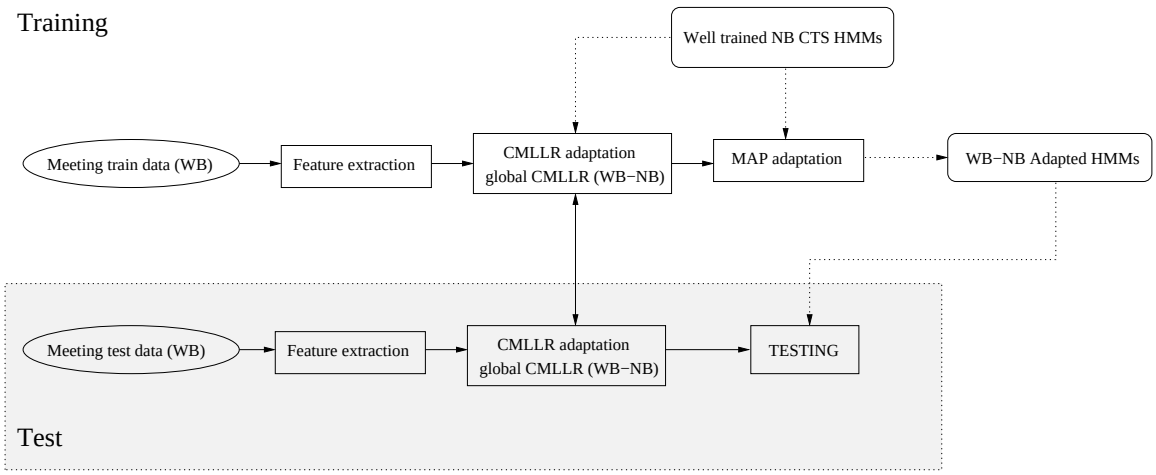


Figure 7.6: WB→NB adapted system based on CMLLR.

above. We use block-diagonal transform. Table 7.5 shows performance of CTS models directly applied on WB→NB rotated test data. After 12 CMLLR iterations, like with the MLLR, we do not have any improvement, so this transformation was used.

Next, an iterative MAP followed to better settle the models into the target space. Figure 7.7 shows the performance with different τ values with fixed number of 12 CMLLR iterations. Similar behavior as in MLLR approach is observed.

The accuracies of unadapted systems (training on the meeting data only), WB→NB adapted and downsampled systems are shown in Table 7.6. In comparison to MLLR approach, the CMLLR gives 1.5% worse results on system without MAP adaptation due to constrains in transformation matrix on means and variances. But after MAP, CMLLR gives 0.2% better performance, so it seems to be more useful for other adaptation. The best performance of WB→NB system is 29% which is a 4.4% relative improvement over the non-adapted WB system and 2.7% over NB-NB adapted system.

CMLLR iterations	WER [%]
4	40.0
8	35.7
12	**35.3**
16	35.4
20	35.4

Table 7.5: WB→NB - Performance of CTS models on the WB meeting data with different WB→NB CMLLR quality.

Training set	Adaptation	WER [%]
WB meeting	none	30.3
NB meeting	none	30.7
CTS-NB	CMLLR MAP	29.8
CTS-WB	MLLR$_{NB\rightarrow WB}$,MAP	29.3
CTS-WB	CMLLR$_{WB\rightarrow NB}$,MAP	**29.1**

Table 7.6: Performance of WB→NB systems.

7.5 WB→NB transform in HLDA estimation

7.5.1 WB→NB system based on HLDA from CTS

The easiest way to train WB→NB HLDA system is to take HLDA transforms and HMMs already trained on CTS data and adapt them directly to the WB→NB transformed features similarly as in the basic system above. This process is displayed in Figure 7.8. The upper branch shows standard HLDA estimation in the CTS domain, described already in chapters 3 and 5. Then, the models and HLDA matrix are fixed for adaptation into the meeting domain.

Obviously, this is not optimal as the HLDA is trained on CTS but the target data are meetings.

7.5.2 Adaptation of statistics

It is useful to estimate statistics required for HLDA estimation from both data sets, to take an advantage of the meeting data also for HLDA matrix estimation. We use MAP adaptation of statistics, so the CTS full-covariance statistics $\boldsymbol{\Sigma}^{(m)}_{(CTS)}, \boldsymbol{\mu}^{(m)}_{(CTS)}, \gamma^{(m)}_{(CTS)}$ are considered as priors and the WB→NB transformed WB statistics $\hat{\boldsymbol{\Sigma}}^{(m)}_{(WB)}, \hat{\boldsymbol{\mu}}^{(m)}_{(WB)}, \hat{\gamma}^{(m)}_{(WB)}$ are taken for the adaptation. These statistics can be directly obtained by collecting over the transformed features or by WB→NB

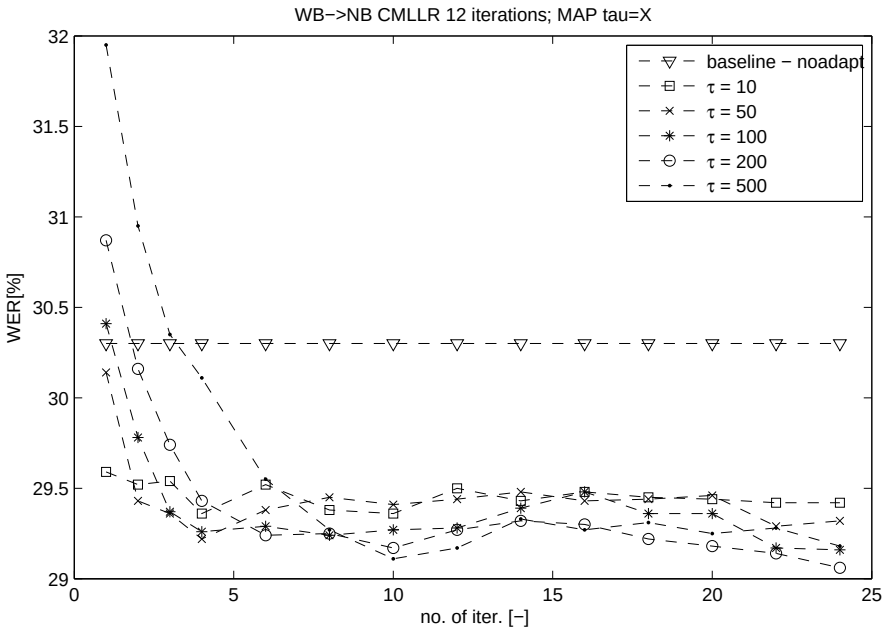


Figure 7.7: WB→NB - τ constant in the iterative MAP with fixed number of 12 CMLLR iterations.

rotation of the original ones:

$$\hat{\boldsymbol{\mu}}^{(m)}_{(WB)} = \mathbf{A}_{(WB\rightarrow NB)}\boldsymbol{\mu}^{(m)}_{(WB)} + \mathbf{b}_{(WB\rightarrow NB)}, \tag{7.1}$$

$$\hat{\boldsymbol{\Sigma}}^{(m)}_{(WB)} = \mathbf{A}_{(WB\rightarrow NB)}\boldsymbol{\Sigma}^{(m)}_{(WB)}\mathbf{A}_{(WB\rightarrow NB)}{}^{T} \tag{7.2}$$

$$\hat{\gamma}^{(m)}_{(WB)} = \gamma^{(m)}_{(WB)}, \tag{7.3}$$

where $\mathbf{A}_{(WB\rightarrow NB)}$ and $\mathbf{b}_{(WB\rightarrow NB)}$ are given by $WB\rightarrow NB$ CMLLR transform.

The estimation of an arbitrary covariance matrix $\boldsymbol{\Sigma}^{(m)}$ is given by:

$$\boldsymbol{\Sigma}^{(m)} = \frac{\sum_{t=1}^{T}\gamma^{(m)}(t)\mathbf{o}(t)\mathbf{o}(t)^{T}}{\gamma^{(m)}} - \boldsymbol{\mu}^{(m)}\boldsymbol{\mu}^{(m)T}, \tag{7.4}$$

where $\gamma_{(m)}$ is occupation count of component m given by:

$$\gamma^{(m)} = \sum_{t=1}^{T}\gamma^{(m)}(t) \tag{7.5}$$

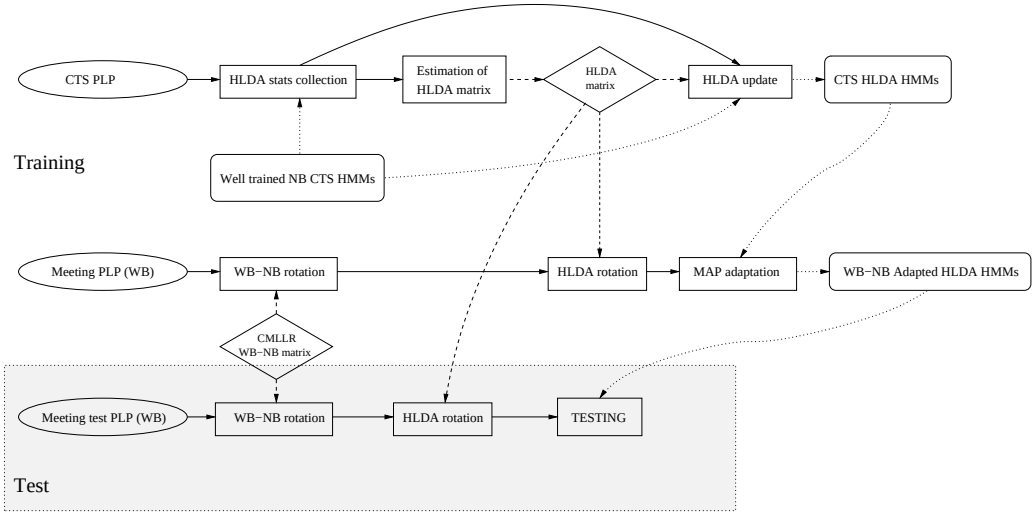


Figure 7.8: WB→NB system using HLDA from CTS.

The MAP adaptation of the statistics is given by:

$$\check{\boldsymbol{\mu}}^{(m)} = \frac{\hat{\gamma}^{(m)}_{(WB)}\hat{\boldsymbol{\mu}}^{(m)}_{(WB)} + \tau\boldsymbol{\mu}^{(m)}_{(CTS)}}{\hat{\gamma}^{(m)}_{(WB)} + \tau} \tag{7.6}$$

$$\check{\boldsymbol{\Sigma}}^{(m)} = \frac{(\hat{\boldsymbol{\Sigma}}^{(m)}_{(WB)} + \hat{\boldsymbol{\mu}}^{(m)}_{(WB)}\hat{\boldsymbol{\mu}}^{(m)^T}_{(WB)})\hat{\gamma}^{(m)}_{(WB)}}{\hat{\gamma}^{(m)}_{(WB)} + \tau} + \tag{7.7}$$

$$+\frac{(\boldsymbol{\Sigma}^{(m)}_{(CTS)} + \boldsymbol{\mu}^{(m)}_{(CTS)}\boldsymbol{\mu}^{(m)^T}_{(CTS)})\tau}{\hat{\gamma}^{(m)}_{(WB)} + \tau} - \check{\boldsymbol{\mu}}^{(m)}\check{\boldsymbol{\mu}}^{(m)^T}$$

$$\check{\gamma}^{(m)} = \gamma^{(m)}_{(CTS)} \tag{7.8}$$

where τ is the a control constant and $\check{\boldsymbol{\mu}}^{(m)}, \check{\boldsymbol{\Sigma}}^{(m)}, \check{\gamma}^{(m)}$ are the resulting adapted statistics. In our experiment, Gaussian component occupation counts $\gamma^{(m)}$ are simply copied from CTS system. The part of our future plan is to do it in more "clever" way, for example similarly as adaptation of Gaussian component weights.

In the next step, HLDA is estimated from these statistics and HMMs are updated by projecting the statistics through HLDA (see equations 3.6 and 3.7). The standard iterative MAP adaptation follows to settle updated HMMs. This process is shown in Figure 7.9.

7.5.3 Experiments

HLDA transform is used to perform dimensionality reduction from 52 dimensional space to 39. Therefore, WB→NB CMLLR has to be also 52 dimensional. To be able to estimate this transform, 52 dimensional CTS models need to be trained. "Single pass retraining" technique was used for this purpose. It works with one model set (39 dimensional CTS models) and

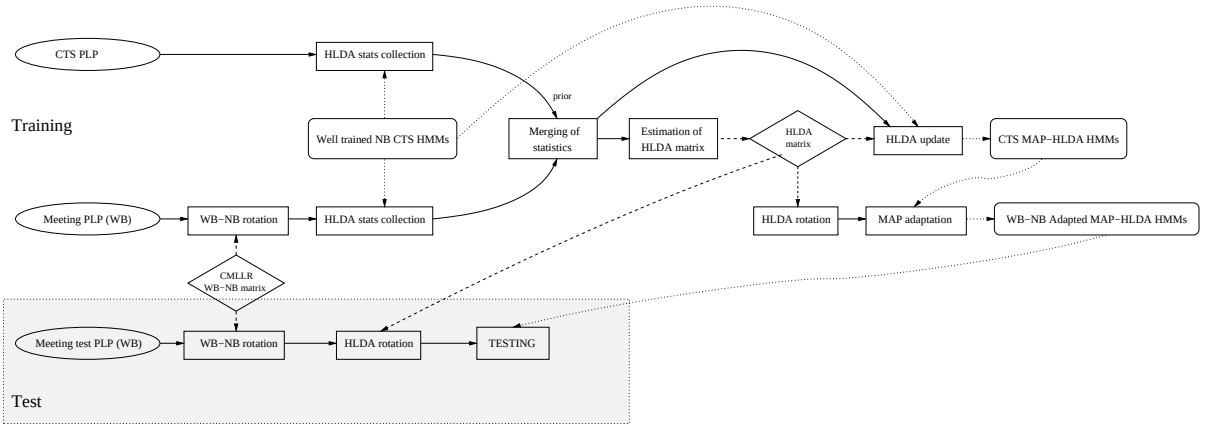


Figure 7.9: WB→NB system using HLDA based on merged statistics from the CTS and meeting training set.

two feature streams: the first, compatible with current model set (39 dimensional PLPs), is used to generate full Gaussian alignment $\gamma_{jm}(t)$ and the second, incompatible to model set (52 dimensional PLPs), is used for collection of statistics $\theta_{jm}(\mathbf{O}), \theta_{jm}(\mathbf{O}^2)$ (see equations 2.19, 2.21). The new models are updated using statistics of the second stream (see equations 2.22-2.24). When the models are trained, new 52 dimensional WB→NB CMLLR can be estimated by the same iterative approach as in section 7.4.

Full covariance statistics have to be accumulated for both data sets to estimate HLDA matrix. For the merging procedure, it is important to collect them by the same clustered models. The WB→NB transform was applied on WB data and all statistics were collected with the CTS models. Consequently, the WB statistics were collected in rotated space, thus equations 7.1,7.2 did not need to be applied.

Equations 7.6-7.8 were used to merge the statistics. The new HLDA matrix was estimated and CTS HMMs were updated. Further, the iterative MAP was applied to settle the HMMs into the new space.

Figure 7.10 shows the dependency of WER on the τ value during MAP merging of statistics. For $\tau =$ "Inf", it presents the system based on CTS HLDA described in previous section (7.5.1). The WER is 28.3%. This is an interesting result because it still gives almost 1% relative improvement over the non-adapted HLDA meeting system although no meeting data was used in HLDA estimation. It is caused by significantly bigger amount of CTS data than meetings. The system generates the best accuracy 27.8% with $\tau = 200$. It gives a 2.5% relative improvement over the non-adapted HLDA system (see Table 7.7).

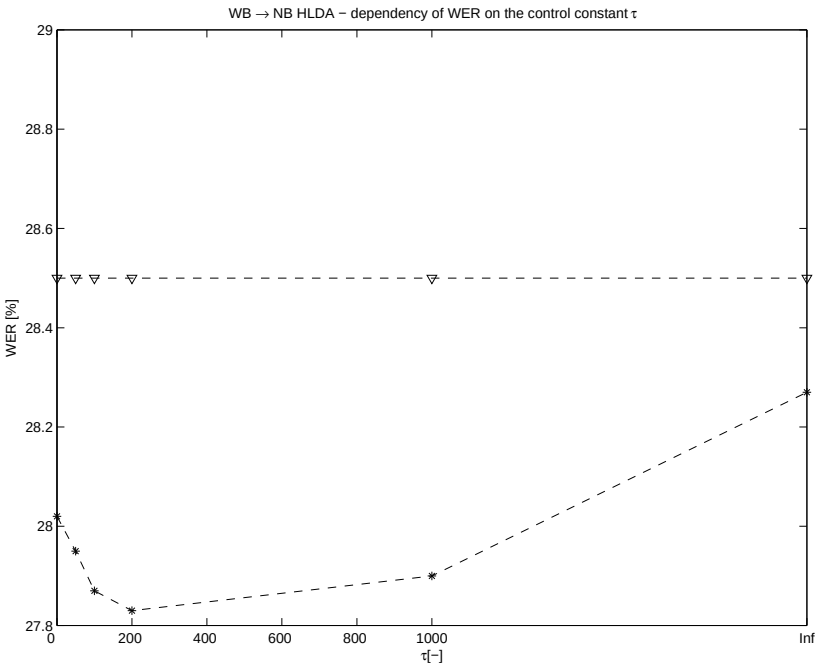


Figure 7.10: WB→NB HLDA - the τ value in MAP adaptation of statistics.

7.5.4 Experiments with downsampled data

For comparison with standard downsampling approach, the same experiments run also in the downsampled NB domain. First, NB non-adapted HLDA system was trained and evaluated. Table 7.7 shows 1.2% degradation of accuracy by downsampling.

Using an adapting scheme, the WB→NB CMLLR feature rotation in Figures 7.9 and 7.8 was replaced by downsampling of waveforms and feature extraction from this data. Table 7.7 presents both kinds of HLDA estimation in CTS-NB adapted system: the HLDA taken directly from CTS system and HLDA estimated from the MAP adapted statistics like in previous section 7.5.2. It is interesting that HLDA taken from CTS for NB-NB adapted system gives better performance than adapted HLDA. It seems that statistics collected over the downsampled meeting data do not contain any additional information for HLDA estimation.

7.5.5 HLDA conclusions

We have proved that when using WB data, CMLLR transforms outperforms the down-sampling and that the HLDA estimates significantly improve (from 28.6% to 27.8%, almost 1% absolute).

System	HLDA adaptation	WER [%]
WB	non-adapted HLDA	28.5
NB	non-adapted HLDA	29.7
WB-NB	CMLLR$_{WB \rightarrow NB}$, HLDA from CTS	28.3
WB-NB	CMLLR$_{WB \rightarrow NB}$, MAP HLDA	**27.8**
CTS-NB	HLDA from CTS	28.6
CTS-NB	MAP HLDA	29.0

Table 7.7: Performance of HLDA systems.

7.6 WB→NB transform in Speaker Adaptive Training

Speaker adaptive training (SAT) was further used in addition to HLDA to improve the accuracy. As it was introduced in section 6.3, SAT is a technique used to suppress cross-speaker variance. The implementation in the WB→NB adapted system differs primarily in using adaptation instead of reestimation. Consequently, the initial seed can not be taken from the final iteration of training of input models. On contrary, prior models have to be used.

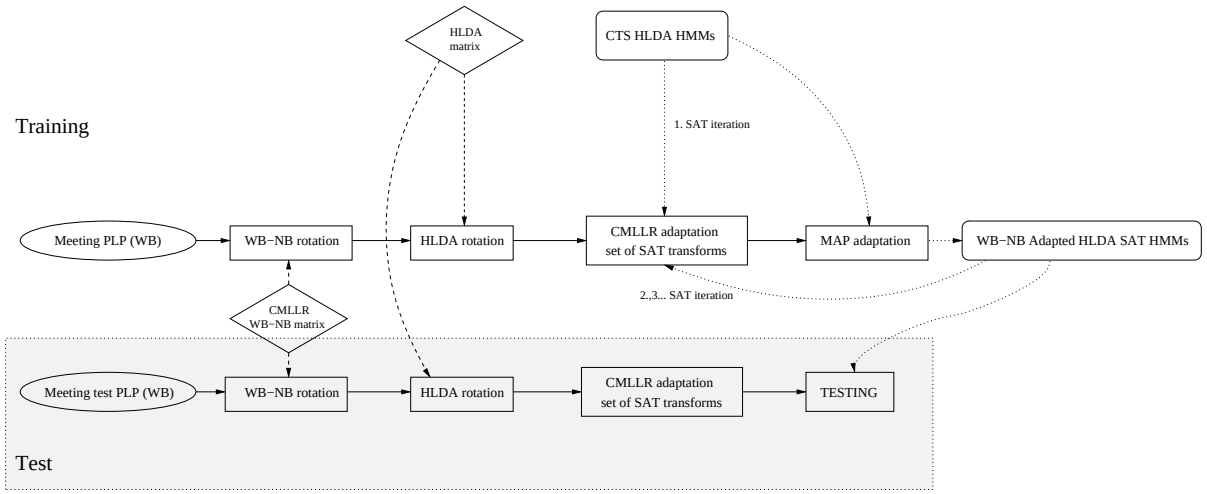


Figure 7.11: WB→NB system in Speaker Adaptive Training.

An implementation of WB→NB transform during the SAT training is shown in Figure 7.11. It is a straightforward procedure which could be described in the following steps:

1. Choose proper CTS HLDA prior model.

2. Rotate the WB data by WB→NB transform and HLDA transform $\mathbf{A}_{(HLDA)}$. These ma-

Prior	SAT training	WER [%]
CTS_MAP-HLDA	-	26.8
CTS_MAP-HLDA	yes	26.6
CTS_MAP-HLDA_SAT	yes	**26.5**

Table 7.8: Performance with different prior models. The CMLLR adaptation was applied also in testing.

trices can be multiplied and feature rotation can be written as:

$$\hat{\mathbf{o}}(t) = \mathbf{Z}\mathbf{o}(t) + \mathbf{z}, \tag{7.9}$$

where

$$\mathbf{Z} = \mathbf{A}_{(HLDA)}\mathbf{A}_{(WB \rightarrow NB)} \tag{7.10}$$

$$\mathbf{z} = \mathbf{A}_{(HLDA)}\mathbf{b}_{(WB \rightarrow NB)} \tag{7.11}$$

3. Use the prior to estimate SAT CMLLR transforms for each speaker in the training data $\hat{\mathbf{o}}(t)$.
4. Take the prior and run iterative MAP using the rotated data $\hat{\mathbf{o}}(t)$ transformed by the respective SAT CMLLR transform.
5. Estimate a new set of SAT CMLLR transforms using the final models and go to step 4.

This process can be repeated iteratively until the accuracy stops to increase. In our experiments, we have not noticed any improvement after the second iteration.

We experimented with two kinds of priors:

- CTS_MAP-HLDA prior model from section 7.5, using MAP-HLDA with $\tau = 200$.
- The prior above was further SAT retrained in CTS domain. We called it CTS_MAP-HLDA_SAT.

Table 7.8 presents 0.1% absolute improvement using SAT retrained prior. This is coherent with the conclusion from section 7.3 that a better prior gives just small improvement after adaptation.

7.6.1 Comparison with downsampling

The comparison of proposed SAT implementations and traditional approach using downsampling lies in replacing of WB→NB CMLLR by downsampling of data and feature extraction. It is

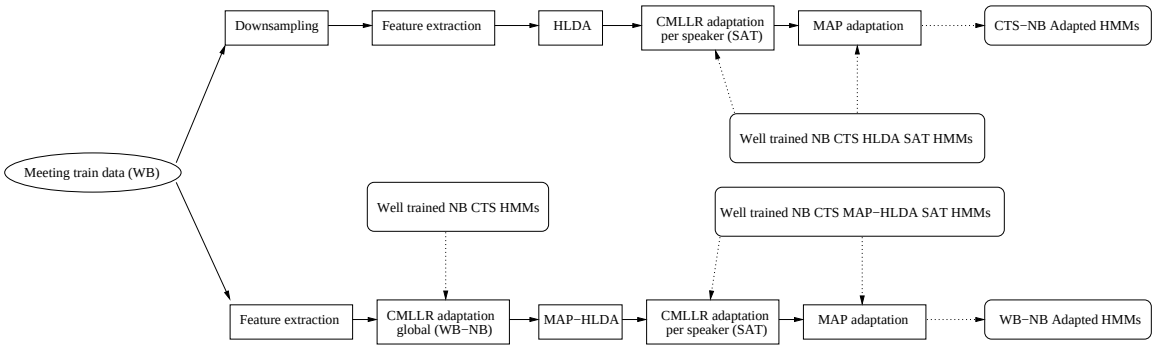


Figure 7.12: Downsampled and WB→NB adapted HLDA SAT system.

System	Adaptation	WER [%]
WB	none	27.5
NB	none	28.8
NB-NB	CMLLR_{SAT}	27.9
WB-NB	$\text{CMLLR}_{WB \rightarrow NB}$, CMLLR_{SAT}	26.5

Table 7.9: Results of HLDA SAT systems.

shown in Figure 7.12. The upper branch presents the traditional approach and the lower branch the WB→NB system.

Table 7.9 shows the accuracy of SAT systems. The best performance 26.6% is obtained by WB→NB HLDA SAT system which is a 3.3% relative improvement over the non-adapted HLDA SAT system and 4.6% relative improvement over NB-NB adapted system.

7.7 Discriminative training of WB→NB adapted system

Discriminative approaches are getting widely used in training of acoustic models for state-of-the-art recognition systems. We decided to improve our system by using discriminative MAP adaptation. Several discriminative criteria are available but usually the best performance is achieved by using the Minimum Phone Error (MPE) criterion. As introduced in section 2.4.4, MPE-MAP adaptation is an iterative process, where each iteration consists of two steps: First, a given prior model is adapted using standard (ML-)MAP adaptation. However, the resulting model is used only as a prior for the following MPE update, where the parameters of the current model are shifted to make compromise between improving MPE objective function and obeying the prior distribution. Therefore, we need to distinguish two models that serve as the input for MPE-MAP adaptation: the (fixed) prior model and the starting point model, which is to be iteratively updated. It is usual practice to set the starting point to be equal

Prior	Starting point	Adaptation	WER [%]
CTS_MAP-HLDA_MPE	CTS_MAP-HLDA_MPE	MPE-MAP	27.2
CTS_MAP-HLDA_MPE	-	ML-MAP	27.0
CTS_MAP-HLDA_MPE	CTS_MAP-HLDA_MPE.ML-MAP	MPE-MAP	**25.6**

Table 7.10: MPE-MAP: Effect of prior and starting point.

to the prior. However, the problem for the practical implementation of WB→NB system lies in quite significant difference between the CTS prior models and WB→NB rotated adaptation data. Therefore, we first adapt the CTS prior to rotated adaptation data using iterative ML-MAP [2] to obtain good starting point, which is further iteratively adapted using MPE-MAP (still with CTS model fixed as the prior). Although each MPE-MAP iteration also contains a single iteration of ML-MAP adaptation, performing the iterative ML-MAP before starting the discriminative adaptation turned out to be essential for successful use of MPE-MAP.

7.7.1 Discriminative adaptation of WB→NB HLDA system

For simplicity, in the first experiment with discriminatively adapted models, we do not make use of SAT, just MAP-HLDA. It is important for MPE-MAP adaptation to have a relevant prior information about the target domain distributions. Therefore, the CTS MAP-HLDA models described in previous sections were further trained using MPE to get a better prior model. In section 7.7, we also mentioned the importance of having proper model that serves as a starting point for MAP-MPE. Therefore, experiments were conducted exploring the influence of the starting point and the adaptation approach.

First, we compared ML-MAP and MPE-MAP using CTS_HLDA_MPE prior and starting point models. Discriminatively trained prior was used here, therefore the floating prior approach in ML-MAP used above was inappropriate. The discriminative information is lost in further iterations, so the prior was fixed and the models from the previous iterations were used only for alignment in the current iteration.

In Table 7.10, we can see that the MPE-MAP using CTS_HLDA_MPE starting point does not give any improvement, even 0.2% degradation of accuracy due to starting point too different from the target data. In the next step, we decided to use ML-MAP adapted models as the starting point for MPE-MAP adaptation. This approach yielded 1.6% absolute improvement compared to CTS_MAP-HLDA_MPE starting point.

The final adaption scheme is shown in Figure 7.13.

[2] On contrary to iterative MAP described in section 7.3, the prior does not change over the iterations and stays fixed to CTS model.

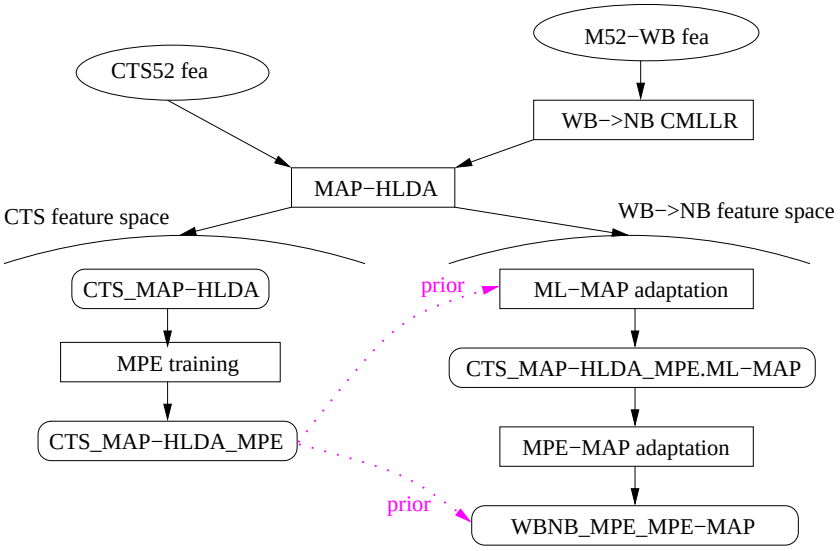


Figure 7.13: Adaptation scheme of MPE-MAP adaptation into the WB→NB features.

7.7.2 Discriminative training of WB→NB adapted HLDA SAT system

The discriminative extension of the speaker adaptive training of the WB→NB system is based on sections 7.6 and 7.7.1.

A need for good CTS prior led to additional MPE training of the CTS_MAP-HLDA_SAT models. It produced new models set referred CTS_MAP-HLDA_SAT_MPE. When processing the meeting data, SAT transforms were estimated based on the CTS WB→NB resulting models from section 7.6. The transforms remained fixed for further processing.

Using an equivalent setup as that in section 7.7.1, we adapted CTS_MAP-HLDA_SAT models into WB→NB rotated domain using iterative ML-MAP with application of the above SAT transforms. These models, further referred to as NBWB_MPE_ML-MAP_SAT, are used as the starting point for the final MPE-MAP adaptation (see Figure 7.14).

To investigate the effect of the CTS prior, the NBWB_MPE_ML-MAP_SAT models were also re-trained only using MPE instead of adapting using MPE-MAP. Table 7.11 shows that a 1.5% absolute improvement is obtained by MPE training of ML-MAP adapted models. Incorporation of the CTS prior and use of MPE-MAP adaptation gives 0.3% additional improvement.

As the ML-MAP adapted model NBWB_MPE_ML-MAP_SAT turns out to be very good starting point for consequent MPE-MAP adaptation, another option is to use this also as the prior for the MPE-MAP. Comparing the last two lines in the Table 7.11, we can see that keeping the original prior CTS_MAP-HLDA_SAT_MPE trained only on CTS data is the better option.

The final models were successfully used in the AMI LVCSR system for NIST 2006 Rich Transcription evaluation.

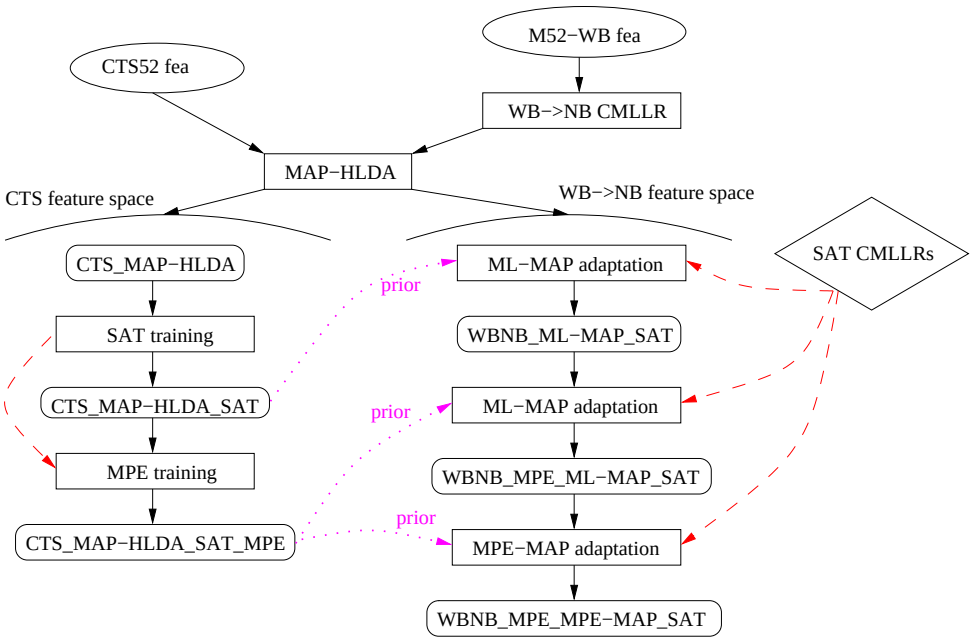


Figure 7.14: SAT - adaptation scheme of MPE-MAP adaptation into the WB→NB features.

Prior	Starting point	Adaptation	WER[%]
CTS_MAP-HLDA_SAT_MPE	-	ML-MAP	25.7
-	NBWB_MPE_ML-MAP_SAT	MPE	24.2
NBWB_MPE_ML-MAP_SAT	NBWB_MPE_ML-MAP_SAT	MPE-MAP	24.1
CTS_MAP-HLDA_SAT_MPE	NBWB_MPE_ML-MAP_SAT	MPE-MAP	**23.9**

Table 7.11: MPE-MAP in the SAT: Effect of prior and starting point.

7.8 WB→NB adapted system trained on increased amount of data

The availability of new meeting data resources, especially the release of full AMI corpus[3], led to further improving of the current LVCSR system. Moreover, we were able to check how the techniques generalize on the new data. The CTS training data size was increased too by Fisher corpus. The new data sizes are described in Table 7.12. We see that the additional data represents about 70 hours of meetings and more than 1700 hours of telephone speech.

7.8.1 CTS system development

The previous work on training from large databases [12] showed no yield over 1000 hours using ML training. But the discriminative techniques were still improving the system significantly.

[3]Detailed information on AMI corpus is available at `http://corpus.amiproject.org`

Task	IHM	MDM	CTS
standard data	112h	63h	278h
boosted data	183h	127h	2000h

Table 7.12: Amount of data in the original and data boosted systems.

Data amount	orig WB→NB CMLLR	new WB→NB CMLLR
CTS 52d 280h	36.3	-
CTS 52d 1000h	35.4	34.0

Table 7.13: CTS 52d models: Effect of WB→NB CMLLR and training data size. It was tested by acoustic rescoring of rt05 lattices.

Therefore, the training data was split to produce two training sets of 1000h and 2000h.

First, the VTLN warping factors were estimated using 280h VTLN models. The ML models were trained on 1000h using the same techniques as for 270h system: decision tree clustering produced 10000 tied-states and mixture-up training run to produce 20-Gaussian models.

These models were retrained using single pass retraining to 52 dimensional space and new WB→NB global CMLLR transform was estimated. Table 7.13 shows improvement given by new CTS models and the transform.

The HLDA transform was estimated taking into account the further adaptation to the meeting domain. Hence, full covariance statistics were collected for both data sets and merged using MAP criteria (see section 7.5). The model parameters were projected into the new space and further trained using SAT. Next, the final CTS SAT models were further trained discriminatively using MPE criteria on the full 2000h set.

Table 7.14 shows the effect of amount of training data. As expected, ML training on 2000 hours does not give any improvement over that on 1000h (note however, that the decision trees were not re-done for the larger set). MPE training using 2000h shows a substantial gain of 3.7% absolutely against ML and 0.5% compared to 1000h MPE models. All model sets make use of MAP-HLDA. As these are SAT models, speaker based adaptation was applied in all test cases.

Data amount	280h	1000h	2000h
ML HLDA SAT	31.3	29.6	29.6
MPE HLDA SAT	28.0	26.4	25.9

Table 7.14: CTS system: Dependency of WER on the training data size. It was tested by acoustic rescoring of eval01 lattices.

Data amount	112h	182h
ML HLDA SAT	27.5	25.8
MPE HLDA SAT	24.5	23.4

Table 7.15: Unadapted meeting system: Dependency of WER on the training data size.

Data amount	112h / 278h	183h / 2000h
CTS SAT prior		
ML-MAP	26.5	25.1
CTS SAT MPE prior		
ML-MAP	25.7	23.8
MPE-MAP	23.9	22.1

Table 7.16: WB → NB: Effect of training data and adaptation approach.

7.8.2 WB→NB adaptation using SAT and new data

All experiments to adapt a new 2000h CTS MPE HLDA SAT models into the WB→NB rotated domain used the same algorithm as described above.

First, an unadapted baseline system was trained just on the new meeting data which yielded 1.7% absolute improvement in ML training over the original system and more than 1% when using MPE (see Table 7.15).

To capitalize on these gains, the 2000h CTS MPE HLDA SAT models were adapted in the WB→NB rotated domain according to the scheme in section 7.7.2: First, MPE starting point models were trained using ML-MAP and MPE-MAP adaptation followed.

Table 7.16 shows a 1.8% absolute gain due to adding training data and 1.3% improvement by adaptation from CTS.

Chapter 8

Conclusion and future work

The recognition of meeting speech is an important research issue and has been in the center of interest of several EC-sponsored projects: M4[1], CHIL[2], AMI[3], and AMIDA[4]. This work has been done in tight cooperation with the meeting recognition team in the series of M4/AMI/AMIDA [21] and concentrated on feature extraction and acoustic modeling. It has investigated into two important problems in building of recognition system for meeting data:

- Improving of robustness of HLDA estimation by smoothing.
- Making use of additional data resources in the training.

8.1 Robust HLDA

Two approaches of HLDA smoothing were tested: Smoothed HLDA (SHLDA) and MAP variant of SHLDA, taking into account the amounts of data available for estimation of statistics for different classes. Both variants perform better than the basic HLDA. Moreover, we have found, that removing the silence class from the HLDA estimations (Silence-reduced HLDA) was equally effective and cheaper in computation. Testing SHLDA and MAP-SHLDA on the top of SR-HLDA did not bring any further improvement.

8.2 NB-WB adaptation

We successfully implemented an adaptation technique where WB data is transformed to the NB domain by CMLLR feature transform. Here, the well trained CTS models are taken as priors for adaptation. A solution of how to apply this transform for HLDA and SAT systems was given using maximum likelihood. A 4.6% relative improvement against adaptation in the

[1] http://www.dcs.shef.ac.uk/spandh/projects/m4/
[2] http://chil.server.de
[3] http://www.amiproject.org/
[4] http://www.amidaproject.org/

downsampled domain was obtained. Next, ML-MAP was replaced by the discriminative MPE-MAP scheme, where a 2.4% relative improvement over the non-adapted meeting system was shown.

Finally, the Fisher corpora were included for improving the CTS prior model and also some new meeting data resources were added. In the final MPE-MAP implementation, we obtained a 5.6% relative improvement over the non-adapted meeting system.

8.3 Future work

In HLDA, the improvements obtained by smoothing techniques in HLDA show that these approaches are performing well but the differences are quite small. Therefore we want to focus on areas which suffer from higher insufficiency of data, such as long-span features [55], where the proposed approaches should lead to significant improvements.

In the second field, we have shown that speech data from sources different from the target domain can be advantageously used to improve the performance of a recognition system. The future of meeting recognition is definitely in processing speech from multiple multiple distant microphones (MDM), as they are much more practical for users than independent head-set microphones (IHM). MDM speech was processed by our speech recognition system but the powerful adaptation approaches presented in this thesis were not yet implemented for MDM. Therefore, our research will focus into this area. The potential for improvement is even greater than for IHM, as MDM speech corpora are even less available due to removing of crosstalks (more speakers talking in same time) and the channel variability in MDM speech is greater than for IHM.

Bibliography

[1] T. Anastasakos, J. McDonough, R. Schwartz, and J. Makhoul. A compact model for speaker-adaptive training. In *Proc. ICSLP '96*, volume 2, pages 1137–1140, Philadelphia, PA, 1996.

[2] L. E. Baum. An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes. *Inequalities*, 3:1–8, 1972.

[3] L. Burget. Combination of speech features using smoothed heteroscedastic linear discriminant analysis. In *8th International Conference on Spoken Language Processing*, Jeju Island, Korea, oct 2004.

[4] L. Burget and H. Heřmanský. Data driven design of filter bank for speech recognition. In *Proc. International conference on Text Speech and Dialogue*, Železná Ruda, Czech Republic, September 2001. Springer.

[5] Lukas Burget. *Speech Recognition System Complementarity and System Combination*. PhD thesis, Brno University of Technology, 2004.

[6] Stanley F. Chen. An empirical study of smoothing techniques for language modeling. In *Proc. of the 34th annual meeting on Association for Computational Linguistics*, pages 310–318, 1996.

[7] W. Chou, C-H. Lee, , and B-H. Juang. Minimum error rate training based on N−best string models. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, volume II, pages 652–655, Minneapolis, USA, April 1993.

[8] K. H. Davis, R. Biddulph, and S. Balashek. Automatic recognition of spoken digits. *Journal of the Acoustic Society of America*, 24(6):627–642, 1952.

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

[10] V. Digalakis and L. Neumeyer. Speaker adaptation using combined transformation and Bayesian methods. In *Proc. ICASSP '95*, pages 680–683, Detroit, MI, 1995.

[11] Andreas Stolcke et al. Further progress in meeting recognition: The ICSI-SRI spring 2005 speech-to-text evaulation system. In *Proc. Rich Transcription 2005 Spring Meeting Recognition Evaluation Workshop*, Edinburgh, UK, July 2005.

[12] G. Evermann, H.Y. Chan, M.J.F. Gales, B. Jia, D. Mrva, P.C. Woodland, and K. Yu. Training LVCSR systems on thousands of hours of data. In *Proc. IEEE Int. Conf. on Acoustic, Speech and Signal processing (ICASSP)*, volume 1, pages 209–212, Philadelphia, PA, USA, march 2005.

[13] J. W. Forgie and C. D. Forgie. Results obtained from a vowel recognition computer program. *Journal of the Acoustic Society of America*, 31(11):1480–1489, 1959.

[14] M. Gales. Maximum likelihood linear transformations for HMM-based speech recognition. Technical report, Cambridge University, 1997.

[15] M.J.F. Gales. Semi-tied covariance matrices for hidden markov models. *IEEE Trans. Speech and Audio Processing*, 7:272–281, 1999.

[16] M.J.F. Gales and P.C. Woodland. Mean and variance adaptation within the MLLR framework. In *Computer Speech and Language, Vol. 10, pp. 249 264*, 1996.

[17] J. Gauvain and C. Lee. Maximum a posteriori estimation for multivariate Gaussian mixture. *IEEE Trans. Speech and Audio Processing*, 2:291–298, 1994.

[18] B. Gold and N. Morgan. *Speech and Audio Signal Processing.* John Wiley & Sons, New York, 1999.

[19] R. Gopinath. Maximum likelihood modeling with Gaussian distributions for classification. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, volume II, pages 661–664, Seattle, Washington, USA, May 1998.

[20] Reinhold Haeb-Umbach, Xavier Aubert, Peter Beyerlein, Dietrich Klakow, Meinhard Ullrich, Andreas Wendemuth, and Patricia Wilcox. Acoustic modeling in the Philips hub-4 continuous-speech recognition system. In *Proc. of DARPA Speech recognition workshop*, Lansdowne, VA, 1998.

[21] T. Hain. Conversational multi-party speech recognition using remote microphones. `http://www.amiproject.org/ami-scientific-portal/documentation/annual-reports/pdf/SOTA-Conversational-multiparty-ASR-using-remote-mics-Jan2007.pdf`.

[22] T. Hain, L. Burget, J. Dines, G. Garau, M. Karafiat, M. Lincoln, I. McCowan, D. Moore, V. Wan, R. Ordelman, and Steve Renals. The 2005 AMI system for the transcription of speech in meetings. In *Proc. Rich Transcription 2005 Spring Meeting Recognition Evaluation Workshop*, Edinburgh, UK, July 2005.

[23] T. Hain, J. Dines, G. Gaurau, M. Karafiat, D. Moore, V. Wan, R.J.F. Ordelman, and S. Renals. Transcription of conference room meetings: an investigation. In *Proceedings of Interspeech 2005*, Lisabon, Portugal, 2005.

[24] T. Hain, P.Woodland, T. Niesler, and E. Whittaker. The 1998 HTK system for transcription of conversational telephone speech. In *Proc. IEEE ICASSP*, 1999.

[25] Thomas Hain, Philip Woodland, Gunnar Evermann, and Dan Povey. The CU-HTK March 2000 Hub5e transcription system. In *Proc. Speech Transcription Workshop, 2000.*, 2000.

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

[27] H. Hermansky, D. P. W. Ellis, and S. Sharma. Tandem connectionist feature extraction for conventional HMM systems. In *Proc. ICASSP 2000*, Turkey, 2000.

[28] H. Hermansky and N. Malayath. Spectral basis functions from discriminant analysis. In *Proc. International Conference on Spoken Language Processing*, Sydney, Australia, November 1998. ISCA.

[29] X. Huang, A. Acero, and H. Hon. *Spoken Language Processing.* Prentice Hall, 2001.

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

[31] N. Kumar. *Investigation of Silicon-Auditory Models and Generalization of Linear Discriminant Analysis for Improved Speech Recognition.* PhD thesis, John Hopkins University, Baltimore, 1997.

[32] L. Lee and R. Rose. Speaker normalization using efficient frequency warping procedures. In *Proc. ICASSP 1996*, pages 339–341, Atlanta, GA, USA, May 1996.

[33] C. Leggetter and P.C. Woodland. Flexible speaker adaptation using maximum likelihood linear regression. In *Proc. Eurospeech'95*, pages 1155–1158, Madrid, Spain, 1995.

[34] C.J. Leggetter and P.C. Woodland. Maximum likelihood linear regression for speaker adaptation of continuous density HMMs. *Computer, Speech and Language*, 9:171–186, 1995.

[35] N. Malayath. *Data-Driven Methods for Extracting Features from Speech.* Ph.d. thesis, Oregon Graduate Institute, Portland, USA, 2000.

[36] L. Mangu. *Finding Consensus in Speech Recognition.* Ph.d. thesis, Johns Hopkins University, USA, 2000.

[37] L. Mangu, E. Brill, and A. Stolcke. Finding consensus among words: Lattice-based word error minimization. In *Eurospeech*, pages 495–498, Budapest, Hungary, 1999.

[38] S. Matsoukas, R. Schwartz, H. Jin, and L. Nguyen. Practical implementations of speaker-adaptive training. In *Proceedings of DARPA Speech Recognition Workshop*, 1997.

[39] Brian C.J. Moore. *An introduction to the psychology of hearing.* Academic press, Boston, USA, 1997.

[40] H. Ney and S. Martin. Maximum likelihood criterion in language modeling. In K. Ponting, editor, *Computational Models of Speech Pattern Processing.* NATO ASI Series, Berlin, 1999.

[41] Y. Normandin. Maximum mutual information estimation of hidden Markov models. In C.H. Lee, F.K. Soong, and K.K. Paliwal, editors, *Automatic Speech and Speaker Recognition: Advanced Topics*, pages 57–81. Kluwer Academic Publishers, Norwell, MA, 1996.

[42] Yves Normandin. *Hidden Markov models, maximum mutual information estimation, and the speech recognition problem.* PhD thesis, McGill University, Montreal, Quebec, Canada, 1991.

[43] Julian James Odell. *The Use of Context in Large Vocabulary Speech Recognition.* PhD thesis, Cambridge University Engineering Department, March 1995.

[44] H. F. Olson and H. Belar. Phonetic typewriter. *Journal of the Acoustic Society of America*, 28(6):1072–1081, 1956.

[45] J.P. Openshaw and J.S. Masan. On the limitations of cepstral features in noise. In *Proc. ICASSP 1994*, Adelaide, SA, Australia, April 1994.

[46] D. Povey. *Discriminative training for large vocabulary speech recognition.* PhD thesis, University of Cambridge, 2003.

[47] D. Povey, M.J.F. Gales, D.Y. Kim, and P.C. Woodland. MMI-MAP and MPE-MAP for acoustic model adaptation. In *Proc. Eurospeech 2003*, Geneva, Switzerland, 2003.

[48] D. Povey and P.C. Woodland. Minimum phone error & i-smoothing for improved discriminative training. In *Proc. ICASSP 2002*, volume 1, pages 105–108, USA, 2002.

[49] L. Rabiner and B. H. Juang. *Fundamentals of speech recognition Signal Processing.* Prentice Hall, Engelwood cliffs, NJ, 1993.

[50] G. Saon, M. Padmanabhan, R. Gopinath, and S. Chen. Maximum likelihood discriminant feature spaces. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, volume 2, pages 1129–1132, Istanbul, Turkey, June 2000.

[51] S. Sharma and H. Hermansky. Temporal patterns (TRAPS) in ASR of noisy speech. In *Proc. IEEE Int. Conf. on Acoustic, Speech and Signal processing (ICASSP)*, pages 289–292, Phoenix, Arizona, 1999.

[52] P. Woodland, M.J.F. Gales, D. Pye, and S.J. Young. The development of the HTK broadcast news transcription system: An overview. *Speech Communication*, 37:47–67, 2002.

[53] P.C. Woodland, C.J. Leggetter, J.J. Odell, V. Valtchev, and S.J. Young. The 1994 HTK large vocabulary speech recognition system. In *Proc. IEEE ICASSP*, Detroit, USA, 1995.

[54] S. Young. *The HTK Book*. Entropics Ltd., 1999.

[55] Bing Zhang, Spyros Matsoukas, Jeff Ma, and Richard Schwartz. Long span features and minimum phoneme error heteroscedastic linear discriminant analysis. In *Proc. of DARPA EARS RT-04 workshop*, Palisades, NY, December 2004.