Brno University of Technology
Faculty of Information Technology
Department of Computer Graphics and Multimedia

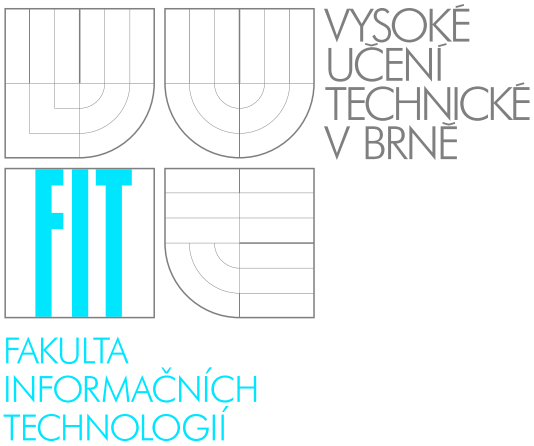
DOCTORAL THESIS

Complementarity of Speech Recognition Systems and System Combination

Lukáš Burget

September 2004

BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY



Complementarity of Speech Recognition Systems and System Combination

by

Lukáš Burget
Brno University of Technology, 2004

Submitted to the *Faculty of Information Technology* in partial fulfillment of the requirements for the degree of Doctor of Philosophy

Submitted:	September 1, 2004
State Doctoral Exam:	June 26, 2001
Thesis Supervisor:	Dr. Jan Černocký
	Associate Professor
	Faculty of Information Technology
	Brno University of Technology

Abstract

In the past, many speech recognition systems differing in feature extraction method, classification method, training algorithm, etc. have been developed. A powerful technique to obtain better recognition results is combination of such systems at different levels (feature-level combination, ROVER-combination of recognition outputs and others). The choice of systems for combination was however done ad-hoc or by an exhaustive search over all possible combinations.

This thesis addresses primarily the problem of choice of systems suitable for combination. We assume that systems good for combination must produce complementary outputs. To evaluate this complementarity, we first define complementarity measures for pairs of recognition systems based on simultaneous and dependent errors the two systems make. ROVER-like alignment of their text outputs is used to count the errors and to derive the measures. These measures are then extended to a definition of complementarity measures of a set of recognition systems.

To verify experimentally the coherence of proposed measures with actual performance of combined systems, a small, yet representative data-set, based on AURORA database is defined. The coherence of measures with the recognition results is confirmed for the following three combination methods: ROVER-like combination of text outputs of recognition systems (similar technique as that used for derivation of complementarity measures), feature-based combination of recognition systems and combination of likelihoods in a multi-stream HMM.

For feature-level combination, this thesis addresses also the suitability of linear transforms for de-correlation and dimensionality reduction of the feature space. PCA, LDA, HLDA are studied and plausible reasons of failure of these approaches are discussed. Based on this analysis, two robust modifications of HLDA - Smoothed HLDA (SHLDA) and Clustered HLDA (CHLDA) balancing the advantages of HLDA and LDA are suggested. Experiments have shown their superiority over PCA, LDA and HLDA.

Abstrakt

Během minulých let bylo vyvinuto značné množství systémů pro rozpoznávání řeči (rozpoznávačů), které se liší v metodách výpočtu příznaků, v metodách klasifikace, trénování, atd. Technikou, která většinou zlepšuje výsledky rozpoznávání, je kombinace systémů na různých úrovních (příznaky, kombinace výsledků rozpoznávání typu ROVER, atd.). Výběr rozpoznávačů vhodných ke kombinování je většinou náhodný nebo založený na prozkoušení všech možných kombinací.

Tato disertační práce se zaměřuje zejména na výběr rozpoznávačů vhodných pro kombinování. Předpokládáme, že systémy vhodné ke kombinování musí mít komplementární výstupy. Abychom tuto komplementaritu dokázali kvantifikovat, definujeme nejprve míry komplementarity pro páry rozpoznávačů. Tyto míry jsou založeny na počtu simultánních a závislých chyb na výstupech obou systémů. Pro časové srovnání těchto dvou výstupů byla použita metoda ROVER. Tyto míry jsou pak rozšířeny - definujeme míry komplementarity pro sadu rozpoznávačů.

Pro experimentální ověření souvislosti mezi námi definovanými mírami a skutečnými úspěšnostmi kombinovaných rozpoznávačů jsme výběrem z databáze AURORA definovali malou, ale reprezentativní databázi řečových signálů. Koherence navrhovaných měr a úspěšností je prokázána pro následující tři kombinační techniky: 1) kombinace textových výsledků rozpoznávání založená na metodě ROVER (podobná metoda, která sloužila pro definici měr komplementarity), 2) kombinace na úrovni příznaků, 3) kombinace na úrovni funkcí hustoty rozdělení pravděpodobnosti v HMM (multi-stream).

Pro kombinaci systémů na úrovni příznaků se tato práce dále zabývá vhodností lineárních transformací pro dekorelaci a omezení počtu rozměrů příznakového prostoru. Studovali jsme metody PCA, LDA a HLDA a shrnujeme pravděpodobné důvody jejich selhání. Na základě této analýzy jsme definovali dvě robustní modifikace metody HLDA: vyhlazenou HLDA (smoothed HLDA - SHLDA) a shlukovou HLDA (clustered HLDA - CHLDA), které vyvažují výhody metod HLDA a LDA. V experimentech jsme prokázali, že SHLDA a CHLDA poskytují lepší výsledky než výše zmiňované metody PCA, LDA and HLDA.

Acknowledgments

First, I would like to thanks 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 to Hynek Heřmanský for allowing me to spend two years in his Anthropic Signal Processing Group at OGI in Portland. I am grateful for challenging discussions with him, insisting always on the question "why we are doing that" rather than "what we are doing".

I would like to thank to my colleagues in Speech Group at Faculty of Information Technology in Brno: František Grézl, Petr Schwarz, Martin Karafiát, Pavel Matějka and others and also to my past colleagues from Anthropic Signal Processing Group: Pratibha Jain, Sachin Kajarekar, Sunil Sivadas, Andre Adami and Pavel Chytil for their friendship, assistance and many fruitful discussions that this theses have greatly benefited from. Special thanks must go to my colleague and friend Petr Motlíček for his great help when finishing this thesis.

Thanks to my girlfriend Zdeňka for her patience and everything she had to stand for several past years. Final thanks go to my parents. Without their constat encouragement and support I would not be where I am today.

My research has been supported by Faculty of Electrical Engineering and Communication and by Faculty of Information Technology of Brno University of Technology, in part by industrial grant from Qualcomm, by DARPA N66001-00-2-8901/0006, by EC projects Multi-modal meeting manager (M4), No. IST-2001-34485, Augmented Multi-party Interaction (AMI), No. 506811 and by Grant Agency of Czech Republic under project No. 102/02/0124.

Contents

1 Introduction **1**

1.1 Scope of chapters 2

1.2 Original contributions of this thesis 3

2 Background **5**

2.1 Tasks in speech processing 5

2.2 Speech recognition 6

2.2.1 Feature extraction 6

2.2.2 Acoustic classification 6

2.2.3 Language models 7

2.3 Introduction to Hidden Markov Models 7

2.3.1 HMM based speech recognition 7

2.3.2 Estimation of HMM parameters 10

2.4 Feature extraction 12

2.4.1 Mel frequency cepstral coefficients 13

2.5 Dimensionality reduction and decorrelation 15

2.5.1 Principal Component Analysis 15

2.5.2 Linear Discriminant Analysis 17

2.5.3 Heteroscedastic Linear Discriminant Analysis 20

3 Complementarity of recognition systems **25**

3.1 Introduction 25

3.2 Terminology 26

3.3 ROVER - Recognizer Output Voting Error Reduction 27

3.3.1 ROVER Alignment 27

3.4 Complementarity of two recognition systems 29

3.4.1 Alignment for identification of error dependency 29

3.4.2 Counting simultaneous and dependent errors 31
3.4.3 Measurement of error dependency between two systems 32
3.4.4 Properties of error dependency measures 32
3.4.5 Experimental setup 34
3.4.6 Analysis of LBWER and DWER matrices 37
3.4.7 Redundancy of a system in the system set 40
3.5 Complementarity measures for set of systems 43
3.5.1 Average Lower Bound Word Error Rate (ALBWER) 43
3.5.2 Average Dependent Word Error Rate (ADWER) 44
3.5.3 Average Sum of LBWER and DWER (ALBWERDWER) . . . 44
3.5.4 Geometric Average of Lower Bound Word Error Rate (GLBWER) 44
3.5.5 Experimental setup 45
3.5.6 Correlation between combined system WER and system set complementarity measures 48
3.6 Discussion and conclusions 55

4 Feature level system combination 61
4.1 Introduction 61
4.2 Combination of feature streams 62
4.3 Postprocessing using PCA 63
4.4 Postprocessing using LDA and HLDA 65
4.4.1 Classes given by labels 66
4.4.2 Classes given by occupation probabilities 66
4.4.3 HLDA in the Maximum Likelihood framework 68
4.5 Robust estimation of statistics 69
4.5.1 Assumption of block diagonal covariance matrix 69
4.5.2 PCA stabilization 70
4.5.3 PCA stabilization preserving feature vector coefficient independency assumptions 71
4.5.4 Smoothed HLDA 73
4.5.5 Clustered HLDA 73
4.6 Feature combination experiments 75
4.6.1 Experimental setup 75
4.6.2 Size of required statistics 77
4.6.3 How to read and compare experimental results 77
4.6.4 Experiments based on PCA 81

4.6.5 Experiments based on LDA and HLDA with classes given by HMM state labels 84
4.6.6 Experiments based on LDA and HLDA with classes given by mixture occupation probabilities 92
4.7 Discussion 93
4.8 Conclusions 97

5 Combination based on Multi-stream HMM 99
5.1 Introduction 99
5.2 Multi-stream Hidden Markov Models 99
5.3 Experimental Setup 100
5.4 Results 101
5.5 Discussion and conclusions 102

6 Conclusion and future work 109

A Matlab implementation of HLDA 113

B Description of developed software 117

List of Tables

3.1 DWER matrix for both systems from figure 3.2. 33
3.2 Word Error Rates of individual recognizers. 36
3.3 LBWER matrix for set of nine systems. 38
3.4 DWER matrix for set of nine systems. 38
3.5 Percentage of *dependent errors* in *simultaneous errors*. 40
3.6 ROVERing 8 of 9 systems. 41
3.7 Five best ROVER combinations. 42
3.8 Five best ROVER combinations - WER for different conditions. 42
3.9 WER of individual *systems with missing bands MFCC*. 47
3.10 Correlation coefficients representing correlations between individual complementarity measures and ROVER WER. 50

4.1 Numbers of covariance matrix $\hat{\mathbf{\Sigma}}_x$ non-zero coefficients / dimensionality of feature vector in the *smoothed space* for different pairs of combined base features. 78
4.2 *Average decorrelating system WER* and *Average combining system WER*. 80
4.3 WER of systems using feature combination based on PCA. 81
4.4 WER of systems using feature combination based on PCA. Only feature vectors representing clean speech are used for estimation of PCA transformation. 82
4.5 WER of systems using feature combination based on PCA. Alternative method of scaling concatenated feature vector coefficients given by equation 4.2 is used. 83
4.6 WER of systems using feature combination based on PCA. Feature frames corresponding to silence parts of utterances are not used for estimation of PCA transformation. 84

4.7 WER of systems using feature combination based on PCA. No assumption is made on concatenated feature vector coefficients independency; full covariance matrix is used to derive PCA transformation. 86
4.8 WER of systems using feature combination based on LDA. Classes are given by HMM state labels. 87
4.9 WER of systems using feature combination based on HLDA. Classes are given by HMM state labels. 88
4.10 WER of systems using feature combination based on SHLDA for $\alpha = 0.75$. Classes are given by HMM state labels. 90
4.11 WER of systems using feature combination based on CHLDA. Classes are given by HMM state labels. Two clusters are considered: HMM states representing non-speech parts of utterances and states representing speech parts. 91
4.12 WER of systems using feature combination based on LDA. Soft assignment of speech frames to classes is given by Gaussian mixture occupation probabilities. 93
4.13 WER of systems using feature combination based on HLDA. Soft assignment of speech frames to classes is given by Gaussian mixture occupation probabilities. 94
4.14 WER of systems using feature combination based on SHLDA for $\alpha = 0.5$. Soft assignment of speech frames to classes is given by Gaussian mixture occupation probabilities. 94

5.1 WER of multi-stream systems combining *different features*. 104
5.2 WER of multi-stream systems combining *missing bands MFCC*. 105

List of Figures

2.1 Typical Hidden Markov Model of a word. 8
2.2 Recognition network used for conected digits recognition. 10
2.3 Block diagram showing steps of MFCC computation. 14
2.4 Outputs of individual steps of MFCC computation. 14
2.5 Principal Component Analysis for 2-dimensional features. 16
2.6 Covariance matrix estimated from log filter bank output vectors. . . . 18
2.7 Spectral basis derived using PCA. 18
2.8 Linear Discriminant Analysis for 2-Dimensional Data. 19
2.9 Heteroscedastic Linear Discriminant analysis. 21

3.1 ROVER method block diagram. 27
3.2 Different systems with the same DWER matrix. 33
3.3 LBWER matrix for set of nine systems. 39
3.4 DWER matrix for set of nine systems. 39
3.5 LBWER matrix for *systems with different features*. 46
3.6 LBWER matrix for *systems with missing bands MFCC*. 47
3.7 ROVER WER for combinations of 3, 8 and all 11 systems. 49
3.8 Correlation between average WER and ROVER WER. 51
3.9 Correlation between ALBWER and ROVER WER. 53
3.10 Correlation between ALBWER' and ROVER WER. 54
3.11 Correlation between ADWER and ROVER WER. 56
3.12 Correlation between ADWER' and ROVER WER. 57
3.13 Correlation between GLBWER and ROVER WER. 58

4.1 Numbers of covariance matrix $\hat{\mathbf{\Sigma}}_x$ non-zero coefficients in the *smoothed space* for different pairs of combined base features. 78
4.2 WER of systems using feature combination based on PCA. 85

4.3 WER of systems using feature combination based on LDA, HLDA and SHLDA. Classes are given by HMM state labels. 89
4.4 WER of systems using feature combination based on CHLDA. Classes are given by HMM state labels. Two clusters are considered: HMM states representing non-speech parts of utterances and states representing speech parts. 91
4.5 WER of systems using feature combination based on LDA, HLDA and SHLDA. Soft assignment of speech frames to classes is given by Gaussian mixture occupation probabilities. 95

5.1 WER of multi-stream systems combining *different features*. 104
5.2 WER of multi-stream systems combining *missing bands MFCC*. 105
5.3 Correlation between average WER and Multi-stream system WER. . . 106
5.4 Correlation between ALBWER and Multi-stream system WER. 107

List of Abbreviations

ADWER	Average Dependent Word Error Rate
ALBWER	Average Lower Bound Word Error Rate
ALBWERDWER	Average Sum of LBWER and DWER
ANN	Artificial Neural Network
CHLDA	Clustered Heteroscedastic Linear Discriminant Analysis
DCT	Discrete Cosine Transform
DTW	Dynamic Time Warping
DWER	Dependent Word Error Rate
EM	Estimation Maximization
GLBWER	Geometric Average of Lower Bound Word Error Rate
GMM	Gaussian mixture model
HLDA	Heteroscedastic Linear Discriminant Analysis
HMM	Hidden Markov Model
LBWER	Lower Bound Word Error Rate
LDA	Linear Discriminant Analysis
LPC	Linear Predictive Coding
LPCC	Linear Prediction Cepstral coefficients
MCE	Minimum Classification Error
MFCC	Mel Frequency Cepstral Coefficients
ML	Maximum Likelihood
MMI	Maximum Mutual Information

PCA	Principal Component Analysis
PLP	Perceptual Linear Prediction Coefficients
ROVER	Recognizer Output Voting Error Reduction
SHLDA	Smoothed Heteroscedastic Linear Discriminant Analysis
SNR	Signal to Noise Ratio
TRAPS	TempoRAl PatternS
WER	Word Error Rate

Chapter 1

Introduction

Many approaches have been developed in the past to perform speech recognition, which differ in feature extraction method, classification method, training algorithm, and so on. Different approaches often utilize different complementary information about speech, and an attempt to preserve one part of information often leads to the loss of another. Therefore, it is not possible to say, which recognition technique is the right one. It has been proved that combination of different techniques can be a powerful way to improve the recognition performance [46, 42, 8, 19, 45, 61]. Success with a combination is however limited by the complementarity of information in combined techniques. Often, it is not easy to guess what is the level of complementarity for two or more recognition techniques. Then, in order to find the best combination, many recognition systems combining different available techniques must be built, trained and evaluated. It would be beneficial to have an apparatus allowing to analyze complementarity of different approaches without exhaustive evaluation of all different combined systems. This problem was previously addressed in [17], where mutual information between two feature stream was estimated to analyze complementarity of the two streams. Another approach, which is more closely related to our work, was proposed in [28]. Here, diversity of decisions made by different classifiers was used to guess how well will the classifiers combine.

The aim of this work is to provide the technique allowing for measuring complementarity of whole recognition systems. Complementarity measures are proposed, which are based on comparison of errors found in symbol sequences obtained at the output of individual recognizers. In several experiments, proposed measures are shown to be useful for selection of systems suitable for combination. For all our experiments, a pool of recognition systems to be combined is defined. Individual systems differ only

in feature extraction. Therefore, we can expect that the complementarity measures derived from the outputs of systems reflect the complementarity of different features. Three sets of experiments using three very different types of combination techniques are carried out. In all cases, a correlation between complementarity measures and actual performance of corresponding combined system is shown.

In the second and largest set of experiments, the combination is performed directly at the feature level. Here, vectors from different feature streams are concatenated and further processed by a linear transformation in order to decorrelate features and reduce their dimensionality. Various methods of deriving the linear transformation are examined, namely Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA) and Heteroscedastic Linear Discriminant Analysis (HLDA). All these methods rely on statistics, which may not be properly estimated when only limited amount of training data is available. We propose Smoothed HLDA (SHLDA) and Clustered HLDA (CHLDA) as modifications of HLDA method, which are based on more robust estimation of statistic. In our experiments, SHLDA and CHLDA turn out to perform superiorly to all other examined methods.

1.1 Scope of chapters

Main problems from speech processing domain are formulated and basic terms are explained in chapter 2. Most of current speech recognition systems are based on Hidden Markov Models. An introduction to Hidden Markov Models, which are also used in our experiments, is given in section 2.3. Important source of complementary information useful for speech recognition can be found in different methods of extraction of speech features from acoustic signal. Methods of feature extraction are addressed in section 2.4. Statistical methods allowing for decorrelation and dimensionality reduction of features (PCA, LDA, HLDA) are introduced in section 2.5.

In chapter 3, we propose a method of measuring the complementarity of recognition systems allowing to select such systems whose combination is the most beneficial. First, a measure of complementarity of system pair is proposed, which is further extended to measurement of complementarity of a whole set of systems. Applicability of the measures are verified in experiments, where recognition system using different features are combined by ROVER technique.

Chapter 4 deals with the problem of feature combination. Statistical methods mentioned above are used to decorrelate and to reduce dimensionality of combined

feature vectors. Modifications of HLDA using more robust estimation of statistics (SHLDA and CHLDA) are proposed and shown to be superior to all other tested methods. Applicability of the proposed complementarity measures are again verified in the feature combination experiments.

The complementarity measures are once more tested in experiments where recognition systems are combined using multi-stream approach. These experiments are described in chapter 5.

Finally, the thesis are concluded and plans for future work are listed in chapter 6.

1.2 Original contributions of this thesis

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

- Definition of complementarity measures for pairs of recognition systems based on simultaneous and dependent errors the two systems make. ROVER-like alignment of their text outputs is used to count the errors and derive the measures.
- Definition of complementarity measures of a *set* of recognition systems based on matrices of measures.
- Definition of a small, yet representative data-set, based on AURORA database, allowing for fast evaluations of proposed complementarity measures and combination techniques.
- Evaluation of coherence of proposed complementarity measures with recognition results (word error rates) of combined systems for three types of system combination:
  - ROVER-like combination of text outputs of recognition systems (similar technique as that used for derivation of complementarity measures) – "combination at the end of the recognition chain".
  - feature-based combination of recognition systems (two feature streams are used, concatenated and post-processed by a de-correlating transform) – "combination at the beginning of the recognition chain".
  - combination of likelihoods in a multi-stream HMM – "combination in the middle of the recognition chain".

- Investigation into the use of known de-correlating transforms (PCA, LDA, HLDA) for combination of concatenated feature streams, including different training strategies of LDA. Discussion on plausible causes of failure of simple approaches.
- Definition of robust variants of HLDA: Smoothed HLDA (SHLDA) and Clustered HLDA (CHLDA) balancing the advantages of HLDA (relaxed assumptions on statistical properties of the data) and robust estimation of necessary statistics (covariance matrices) of LDA.

Chapter 2

Background

2.1 Tasks in speech processing

Speech processing can be divided into several principal domains, such as:

- **Speech recognition** - speech signal is translated to the stream of symbols (phonemes, words) which represents information in the utterance.
- **Speaker recognition** - determination which one speaker from a set of possible speakers has pronounced given utterance.
- **Speaker verification** - verification whether the speaker who has pronounced a utterance is the given person or not.
- **Speech synthesis** - artificial generation of speech never before pronounced by any human speaker.
- **Speech coding** - conversion of the speech into an efficient representation (for the transmission or storage purposes) which allows for the reconstruction of original speech.

This work is mainly focused on speech recognition, however, described techniques can find their application in all the principal domains mentioned above.

2.2 Speech recognition

Most of current systems for automatic speech recognition [36, 38, 27] consist of three basic "building blocks":

2.2.1 Feature extraction

In this phase, speech signal is converted into discrete sequence of feature vectors, which is assumed to contain only that information about given utterance that is important for its correct recognition. Feature extraction is performed in order to reduce dimensionality of original speech signal data and to preprocess that signal into a form fitting requirements of following classification stage. An important property of feature extraction is the suppression of information irrelevant for correct classification such as information about speaker (e.g. fundamental frequency) and information about transmission channel (e.g. characteristic of a microphone). Currently most popular features are Mel Frequency Cepstral coefficients (see section 2.4.1) and Perceptual Linear Prediction coefficients (PLP) [30].

2.2.2 Acoustic classification

The role of classifier is to find a mapping between sequences of speech feature vectors and recognized fundamental speech elements (words in a vocabulary, phonemes). Such mapping can be found for example by a pattern recognizer based on Dynamic Time Warping (DTW). In this simple approach no statistical models are created, but the sequences of feature vectors are stored directly as references. The recognition is performed by a time-alignment of tested word with all references followed by finding the best match. However, DTW algorithm is not practical in case of building speaker-independent recognizer because there is no simple way to find representative references when we have many patterns from many speakers in the training database. For this reason, current speech recognition sysytems are mostly based on Hidden Markov Models (HMM) [64, 43, 24, 59]. The tested word features are not compared with references (as in the case of DTW), but rather statistical acoustic models are created. Parameters of this set of models are estimated from training utterances and their associated transcriptions. After this process, trained models can be used for recognition of unknown utterances. The output of the classifier is a set of possible sequences of speech elements (hypotheses) and their probabilities. Basic ideas of HMM are given in section 2.3.

2.2.3 Language models

The role of language models is a selection of hypothesis which is most likely the right sequence of speech elements (sentence) of a given language. The complexity of a used language model depends on complexity of the problem being solved (continuous speech vs. limited number of commands). Statistical models derived from data are also very often used for this purpose (N-grams). HMM based speech recognition provide an unified framework, where acoustic and language models are jointly used to find the most probable word sequence (see section 2.3.1). However, language models are not in the interest of this work, and therefore, they are not described in greater details in this text. Interested reader can refer to [43, 56].

2.3 Introduction to Hidden Markov Models

During the last decades, methods based on the statistical models and specially on Hidden Markov Models (HMM) became dominant in speech recognition. The sequence of feature vectors derived in feature extraction phase is considered to be a sequence of random vectors (random process). HMM is a powerful statistical method allowing for modeling and efficient evaluation of distribution of such random vector sequences.

In this chapter, a brief introduction to Hidden Markov Models will be given. However, this text is not intended to be a best tutorial on this topic. Rather, it should be seen as collection of basic HMM concepts, equations and other notes that will serve for future references from following sections. A general description of HMM is not given, often only the special cases used in our experiments are considered. Very good and compact introduction to HMM can be found in [65]. Detailed HMM description and derivation of all formulae are in [37, 24, 43, 59].

2.3.1 HMM based speech recognition

From statistical point of view, the goal of speech recognition is to find the most likely sequence of words $\mathbf{W} = w_1, w_2, \ldots, w_n$ 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:

$$\mathbf{W} = \arg\max_{\mathbf{W}'}\{P(\mathbf{W}'|\mathbf{O})\}. \tag{2.1}$$

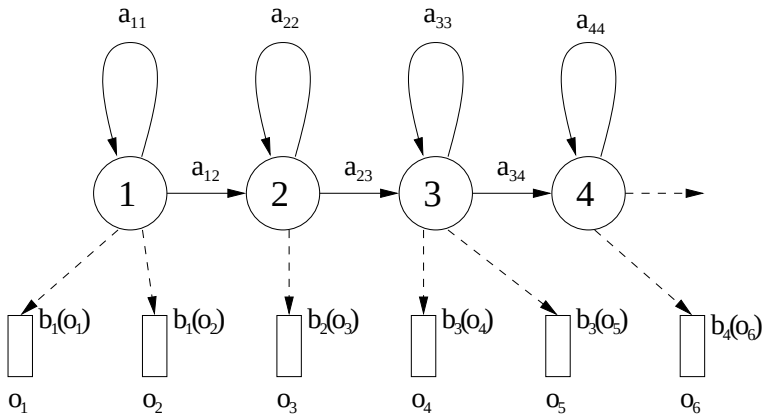


Figure 2.1: Typical Hidden Markov Model of a word.

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:

$$P(\mathbf{W}|\mathbf{O}) = \frac{p(\mathbf{O}|\mathbf{W})P(\mathbf{W})}{p(\mathbf{O})}, \tag{2.2}$$

where $p(\mathbf{O})$ is a probability (or a probability density) of an 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 a word sequence $\mathbf{W}$, which is usually estimated using *language model*. In our experiments, discussed in chapters 3, 4, 5, the task is to recognize sequences of English digits, and all the digit sequences are assumed to have the same prior probability. In such case, the term P($\mathbf{W}$) can be ignored too and the decision about the most likely word sequence will be based only on the term $p(\mathbf{O}|\mathbf{W})$, which is called likelihood. Recognition problem can be then reformulated (equation 2.1 can be rewritten) as:

$$\mathbf{W} = \arg\max_{\mathbf{W}'}\{p(\mathbf{O}|\mathbf{W}')\}. \tag{2.3}$$

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 2.1. HMM is a generative model, which can be seen as a finite state machine making transition from a state i to a 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 $\{M_1, M_2, \dots\}$ 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 is assumed to equal to 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

$$b_s(\mathbf{o}) = \sum_m c_{sm} b_{sm}(\mathbf{o}), \tag{2.4}$$

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

$$b_{sm}(\mathbf{o}) = \frac{1}{(2\pi)^n \prod_{k=1}^{n} \sigma_{smk}^2} e^{-\prod_{k=0}^{n} \frac{(o_k - \mu_{smk})^2}{2\sigma_{smk}^2}}, \tag{2.5}$$

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

$$p(\mathbf{O}|\mathbf{S}, M) = \prod_{t=1}^{T} b_{s(t)}\left(\mathbf{o}(t)\right) \tag{2.6}$$

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

$$P(\mathbf{S}|M) = \prod_{t=2}^{T} a_{s(t-1)s(t)}. \tag{2.7}$$

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

$$p(\mathbf{O}|M) = \sum_{\mathbf{S}} p(\mathbf{O}|\mathbf{S}, M) P(\mathbf{S}|M), \tag{2.8}$$

which can be efficiently computed using an algorithm based on Dynamic Programming. True model evaluation according to formula 2.8 is often approximated by so called *Viterbi probability*

$$\hat{p}(\mathbf{O}|M) = \max_{\mathbf{S}} \{p(\mathbf{O}|\mathbf{S}, M) P(\mathbf{S}|M)\}, \tag{2.9}$$

which is, in fact, the likelihood of the best state sequence, $\mathbf{S}$. *Viterbi probability* can be computed more efficiently than the true likelihood especially in the case of continuous speech recognition: It was mentioned above that the task in our experiments is

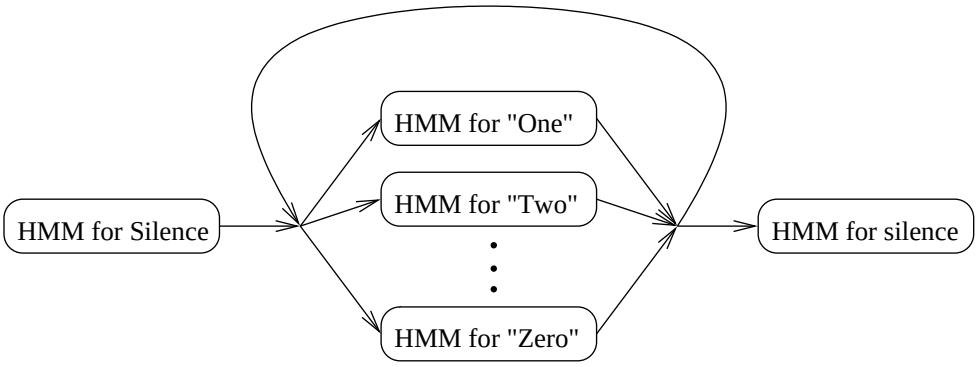


Figure 2.2: Recognition network used for conected digits recognition.

a recognition of connected digits. Instead of constructing and evaluating a composite left-to-right model for each possible sequence of digits, only one composite model is created according to figure 2.2. The loop transition allows to re-enter any of digit models, which are all placed in parallel. Now, Viterbi approximation can be advantageously used to recognize the most likely sequence of digits. 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 model states. Such state sequence uniquely identifies the most likely sequence of words.

2.3.2 Estimation of HMM parameters

Before Hiden Markov Models can be used for recognition, the set of their parameters, $\mathbf{\Phi} = \{a_{ij}, c_{sm}, \boldsymbol{\mu}_{sm}, \boldsymbol{\sigma}^2_{sm}\}$, must be determined. 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 $M^{\mathbf{W}}_{\mathbf{\Phi}}$ denote a compound model representing word sequence $\mathbf{W}$, where $\mathbf{\Phi}$ are parameters of all word models from which the compound models is constructed. The goal is to find such setting of values, $\mathbf{\Phi}$, that maximizes the likelihood of training data:

$$\mathbf{\Phi} = \arg\max_{\mathbf{\Phi}'} \{ \prod_{u}^{U} P(\mathbf{O}^u | M^{\mathbf{W}^u}_{\mathbf{\Phi}'}) \}, \tag{2.10}$$

where U is number of training utterances, $\mathbf{O}^u$ is observation sequence for u^{th} utterance and $\mathbf{W}^u$ is its transcription. Well known Baum-Welch algorithm [5], which is based on the standard EM (Estimation Maximization) algorithm [15], can be used to iteratively search for ML estimates of HMM parameters: Using current set of parameters $\mathbf{\Phi}$, the

algorithm allows to estimate new set of parameters, $\hat{\mathbf{\Phi}}$, which is guaranteed to increase (or at least not to decrease) the likelihood of training data. The following formulae are used to re-estimate Gaussian component weights, c_{sm}, mean vectors, $\boldsymbol{\mu}_{sm}$, variances, $\boldsymbol{\sigma}^2_{sm}$:

$$\hat{c}_{sm} = \frac{\sum_{u=1}^{U}\sum_{t=1}^{T^u}\gamma^u_{sm}(t)}{\sum_{u=1}^{U}\sum_{t=1}^{T^u}\sum_{m}\gamma^u_{sm}(t)}, \tag{2.11}$$

$$\hat{\boldsymbol{\mu}}_{sm} = \frac{\sum_{u=1}^{U}\sum_{t=1}^{T^u}\gamma^u_{sm}(t)\mathbf{o}^u(t)}{\sum_{u=1}^{U}\sum_{t=1}^{T^u}\gamma^u_{sm}(t)}, \tag{2.12}$$

$$\hat{\boldsymbol{\sigma}}^2_{sm} = \frac{\sum_{u=1}^{U}\sum_{t=1}^{T^u}\gamma^u_{sm}(t)(\mathbf{o}^u(t)-\hat{\boldsymbol{\mu}}_{sm})^2}{\sum_{u=1}^{U}\sum_{t=1}^{T^u}\gamma^u_{sm}(t)}, \tag{2.13}$$

where T_u is number of observations in sequence $\mathbf{O}^u$ and $\gamma^u_{sm}(t)$ is the posterior probability of generating t^{th} observation of u^{th} utterance, $\mathbf{o}^u(t)$, by m^{th} Gaussian component associated with s^{th} HMM state. This probability is estimated using current set of parameters, $\mathbf{\Phi}$:

$$\begin{aligned}\gamma^u_{sm}(t) &= P(s(t)=s, m(t)=m|\mathbf{O}^u, M^{\mathbf{W}^u}_{\mathbf{\Phi}}) \\ &= \frac{p(\mathbf{O}^{ut}_1, s(t)=s|M^{\mathbf{W}^u}_{\mathbf{\Phi}})p(\mathbf{O}^{uT}_{t+1}|s(t)=s, M^{\mathbf{W}^u}_{\mathbf{\Phi}}) - \sum_{i\neq m} c_{si}b_{si}(\mathbf{o}^u(t))}{p(\mathbf{O}^u|M^{\mathbf{W}^u}_{\mathbf{\Phi}})}.\end{aligned} \tag{2.14}$$

Here, $p(\mathbf{O}^t_1, s(t)=s|M)$, which is often refereed as *forward probability*, is a likelihood of generating partial observation sequence $\mathbf{O}^t_1 = \mathbf{o}(1), \mathbf{o}(2), \ldots \mathbf{o}(t)$ by model M ending-up in state s. The term $p(\mathbf{O}^T_{t+1}|s(t)=s, M)$, which is often refereed as *backward probability*, is likelihood of generating the rest of observations, $\mathbf{O}^T_{t+1}$, by model M given that the previous observation vector, $\mathbf{o}(t)$, was generated from state s. The term $-\sum_{i\neq m} c_{si}b_{si}(\mathbf{o}(t))$ expresses that observation $\mathbf{o}(t)$ is not generated by any but m^{th} Gaussian component associated with state s. Occupation probability, $\gamma^u_{sm}(t)$, can be efficiently computed using so-called *forward-backward algorithm*, which is again based on Dynamic Programming principle. Transition probabilities, a_{ij} are re-estimated using formula:

$$\hat{a}_{ij} = \frac{\sum_{u=1}^{U}\sum_{t=1}^{T^u}\zeta^u_{ij}(t)}{\sum_{u=1}^{U}\sum_{t=1}^{T^u}\sum_{s=1}^{S}\zeta^u_{is}(t)}, \tag{2.15}$$

where $\zeta_{ij}(t)$ is the probability of making transition from state i to state j right before generating t^{th} observation. This probability can be again expressed in terms of forward and backward probabilities:

$$\begin{aligned} \zeta_{ij}^{u}(t) &= P(s(t\text{–}1)=i, s(t)=j|\mathbf{O}^{u}, M_{\mathbf{\Phi}}^{\mathbf{W}^{u}}) \\ &= \frac{p(\mathbf{O}_{1}^{ut-1}, s(t\text{–}1)=i|M_{\mathbf{\Phi}}^{\mathbf{W}^{u}}) a_{ij} b_{s}(\mathbf{o}^{u}(t)) p(\mathbf{O}_{t+1}^{uT}|s(t)=j, M_{\mathbf{\Phi}}^{\mathbf{W}^{u}})}{p(\mathbf{O}^{u}|M_{\mathbf{\Phi}}^{\mathbf{W}^{u}})}. \end{aligned} \tag{2.16}$$

Note, that besides Baum-Welch algorithm, which was described here, other training schemes exist where HMM parameters are not necessarily estimated in ML fashion. Maximum Mutual Information (MMI) [57] and Minimum Classification Error (MCE) [13] are two known examples of discriminative training schemes, where HMM parameters are estimated to minimize the classification error rather than to maximize the likelihood of data.

2.4 Feature extraction

It was mentioned above that the purpose of feature extraction is the reduction of speech data size and other processing required for an adaptation of this data to classifier (HMM) needs. The standard feature extraction consists of the following steps:

- **Segmentation** – Speech signal is divided to segments where the waveform can be regarded as being stationary (the typical duration 25 ms). The classifiers generally assume that their input is a sequence of discrete parameter vectors where each parameter vector represents one such segment - frame.
- **Spectrum** – Current methods of a feature extraction are mostly based on the short term Fourier spectrum and its changes in the time, therefore the power or magnitude Fourier spectrum is computed for every speech segment.
- **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 [52, 30].
- **Decorrelation** – Some technique for vector decorrelation is used for a better adaptation of features to requirements of classifier. In the case of HMM, only a variance vector can be used for a description of output probabilities instead of a full covariance matrix if features are properly decorrelated (see section 2.3).

- **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.

2.4.1 Mel frequency cepstral coefficients

Mel frequency cepstral coefficients (MFCC) [14] are commonly used features for speech recognition. Since MFCC and their modifications are used as the features in our experiments described in chapters 3 to 5, brief description of this method is given here. Individual steps are shown on the block diagram in figure 2.3. Output of each step is shown in figure 2.4 for a segment of voiced speech (vowel 'iy').

First, speech samples are divided into overlapping frames. The usual frame length is 25 ms and the frame rate is 10 ms. Example of one such frame for English vowel 'iy' can be seen in figure 2.4a. Each frame is usually processed by pre-emphasis filter to amplify higher frequencies. This is an approximation of psychological findings about sensitivity of human hearing on different frequencies [52]. Hamming window is applied in the next step (figure 2.4b) and Fourier spectrum is computed for the windowed frame signal (figure 2.4c). Mel filter bank is then applied to smooth the spectrum: Energies in the spectrum are integrated by the set of a band limited triangular weighting functions. Their shape can be seen in figure 2.4c (dotted lines). These weighting functions are equidistantly distributed over the Mel scale according to psycho-acoustic findings, where better resolution in a spectrum is preserved for lower frequencies than for higher frequencies. A vector of filter bank energies for one frame can be seen as a smoothed and down-sampled version of spectrum (figure 2.4d). The log of integrated spectral energies is taken with agreement to the human perception of sound loudness (figure 2.4e). Temporal trajectories of the log energies of each band can be optionally filtered using RASTA-like bandpass filters [33]. The low pass character of such filter allows to remove fast energy changes which cannot be produced by the human articulatory tract. The high pass character of the filter is responsible for removing static information about the channel, since it appears as an additive constant to the filter bank band output in the log domain. The feature vector is finally decorrelated and its dimensionality is reduced by its projection to several first cosine basis (Discrete Cosine Transform).

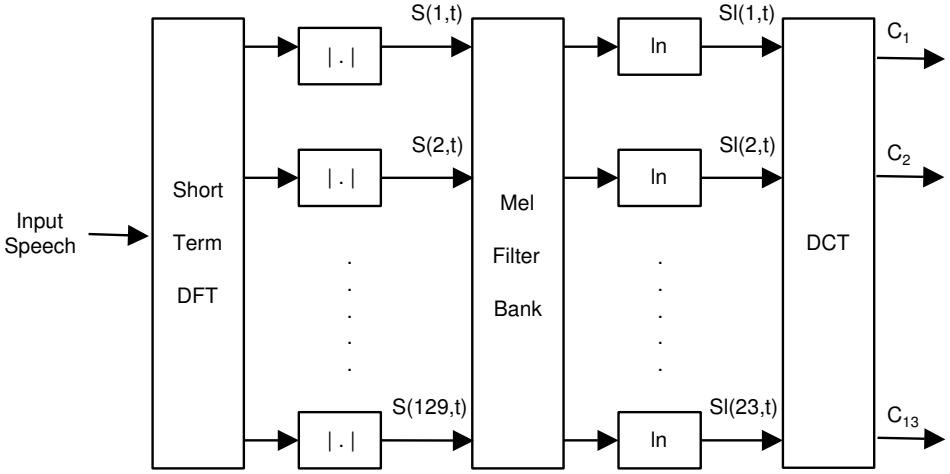


Figure 2.3: Block diagram showing steps of MFCC computation.

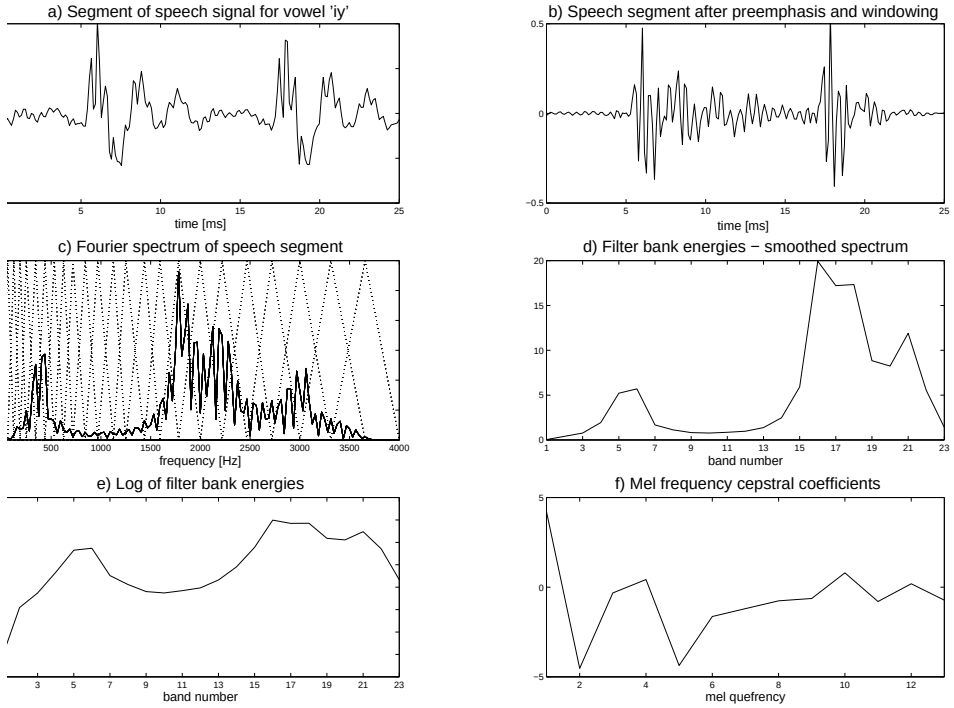


Figure 2.4: Outputs of individual steps of MFCC computation.

2.5 Dimensionality reduction and decorrelation using a linear transform

During feature extraction, sequence of feature vectors is generated containing the information important for correct recognition of speech. As already mentioned, two aims of feature extraction are to ensure low dimensionality of feature vectors and to decorrelate them. In the last step of MFCC feature extraction, projection to several basis of Discrete Cosine Transform (DCT) is performed in order to decorrelate features and to perform final reduction of dimensionality. DCT was chosen empirically as a good decorrelating transformation for MFCC. Alternatively, DCT can be replaced by one of transforms described below, which are derived using a set of training feature vectors. Other examples, where these data driven transforms are used to improve feature extraction, can be found in [10, 32, 50].

In our experiments that are described in chapter 4, these transforms are used to decorrelate and reduce dimensionality of features created as a combination of two different feature streams.

2.5.1 Principal Component Analysis

Principal Component Analysis (PCA) [16, 21] or Karhunen-Loève transform (KLT) is a technique allowing to derive linear transformation with orthonormal basis, given by matrix $\mathbf{A}$, having the following property: The first base vector (first row of matrix $\mathbf{A}$), $\mathbf{a}_1$, shows the direction of the largest variability in n-dimensional space of feature vectors $\mathbf{x}$ (n-dimensional random variable). The second base vector then shows a direction perpendicular to direction given by the first vector with the second largest variability and so on. Figure 2.5 shows an example for 2-dimensional features. PCA transform has two important properties:

- Coefficients of rotated feature vectors, $\bar{\mathbf{x}} = \mathbf{A}\mathbf{x}$, are decorrelated (features have diagonal covariance matrix). Under the assumption that features obey multivariate Gaussian distribution, coefficients of rotated feature vectors are statistically independent each of other.

- Projection to only several first base vectors, $\mathbf{a}_i$, where $i = 1 \ldots p < n$, which is preserving most of the variability of original features, can be performed for the purpose of dimensionality reduction.

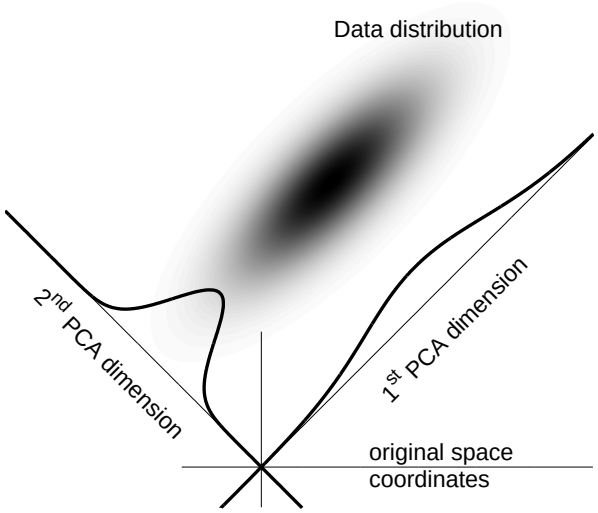


Figure 2.5: Principal Component Analysis for 2-dimensional features. The "gray cloud" represents distribution of features, the Gaussian curves represent the new distributions of uncorrelated features in both PCA directions.

For rotated features, $\bar{\mathbf{x}} = \mathbf{A}\mathbf{x}$, it is easy to show that

$$\bar{\mathbf{\Sigma}} = \mathbf{A}\mathbf{\Sigma}\mathbf{A}^T, \tag{2.17}$$

where $\bar{\mathbf{\Sigma}}$ and $\mathbf{\Sigma}$ are covariance matrices for rotated and original features, respectively. Since $\bar{\mathbf{\Sigma}}$ must be a diagonal matrix for PCA transformation, equation 2.17 corresponds to the standard problem of finding eigen vectors and eigen values of the covariance matrix, $\mathbf{\Sigma}$. Eigen values correspond to diagonal elements of matrix $\bar{\mathbf{\Sigma}}$, which are variances of coefficients of target (uncorrelated) features, $\bar{\mathbf{x}}$. Therefore, i^{th} base vector of PCA transformation, $\mathbf{a}_i$, is given by the eigen vector corresponding to i^{th} largest eigen value. The covariance matrix, $\mathbf{\Sigma}$ can be estimated from training data according to the well known equation:

$$\hat{\mathbf{\Sigma}} = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{x}_i - \hat{\boldsymbol{\mu}})(\mathbf{x}_i - \hat{\boldsymbol{\mu}})^T, \tag{2.18}$$

where N is the number of feature vectors available for training, $\mathbf{x}_i$ is the i-th training feature vector and $\hat{\boldsymbol{\mu}}$ is the estimated mean vector, that is

$$\hat{\boldsymbol{\mu}} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i. \tag{2.19}$$

The projection to cosine basis (DCT) is used in the final stage of MFCC computation (see 2.4.1) to decorrelate features and to reduce their dimensionality. Alternatively, PCA can be used to derive a linear transformation replacing DCT. The vectors of log Mel filter bank outputs derived from TIMIT database were used to estimate the covariance matrix shown in figure 2.6. The values on the matrix diagonal represent variances in individual bands. The values out of the diagonal represent correlations between bands. High correlation can be seen especially between several neighboring low-frequency bands. The figure 2.7 shows the sorted eigen values and the first five eigen vectors (PCA basis) of the covariance matrix. For comparison, the cosine basis are plotted (dotted line) together with corresponding PCA base vector (solid line). The eigen values in figure 2.7a indicate that almost all the variability is preserved by projecting features to only several first basis. The visible similarity between PCA and DCT basis can be seen as a justification for use of DCT in MFCC computation.

2.5.2 Linear Discriminant Analysis

Similarly to PCA, Linear Discriminant Analysis (LDA) [16, 21, 40, 32] is a data driven technique looking for linear transformation allowing for dimensionality reduction of features. Unlike PCA, the aim of LDA is to preserve information important for *discrimination* between feature vectors belonging to different classes. Therefore, for each training feature vector, we need also information about the class to which the vector belongs. LDA allows to derive linear transformation with bases sorted by their importance for discrimination between classes. Therefore, for the purpose of dimensionality reduction, we can project features only into several first basis, which preserve almost all the variability in data important for the discrimination of classes. Note, that LDA like a PCA ensures the decorrelation of features. Moreover, it does not decorrelate only overall training data, as it is in the case of PCA, but features belonging to each particular class are also decorrelated. However, assumption that features belonging to each particular class obey Gaussian distribution and that all the classes share the same covariance matrix must be fulfilled for the optimal functionality of LDA. Figure 2.8 demonstrates the effect of LDA on 2-dimensional feature vectors belonging to two classes. Two “gray clouds” represent distributions of data for two different classes. A large overlap of the distributions can be seen in the directions of the original coordinates. However, classes are well separated in the direction corresponding to the first LDA base vector. Since this example deals only with two classes and since LDA assumes that distributions of all classes are Gaussian with the same covariance matrix

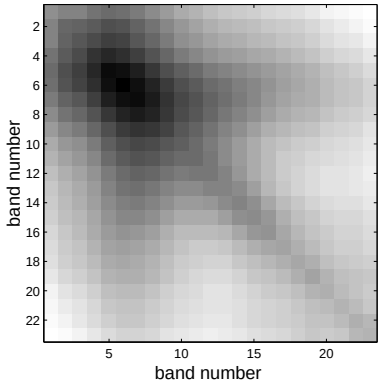


Figure 2.6: Covariance matrix estimated from log filter bank output vectors.

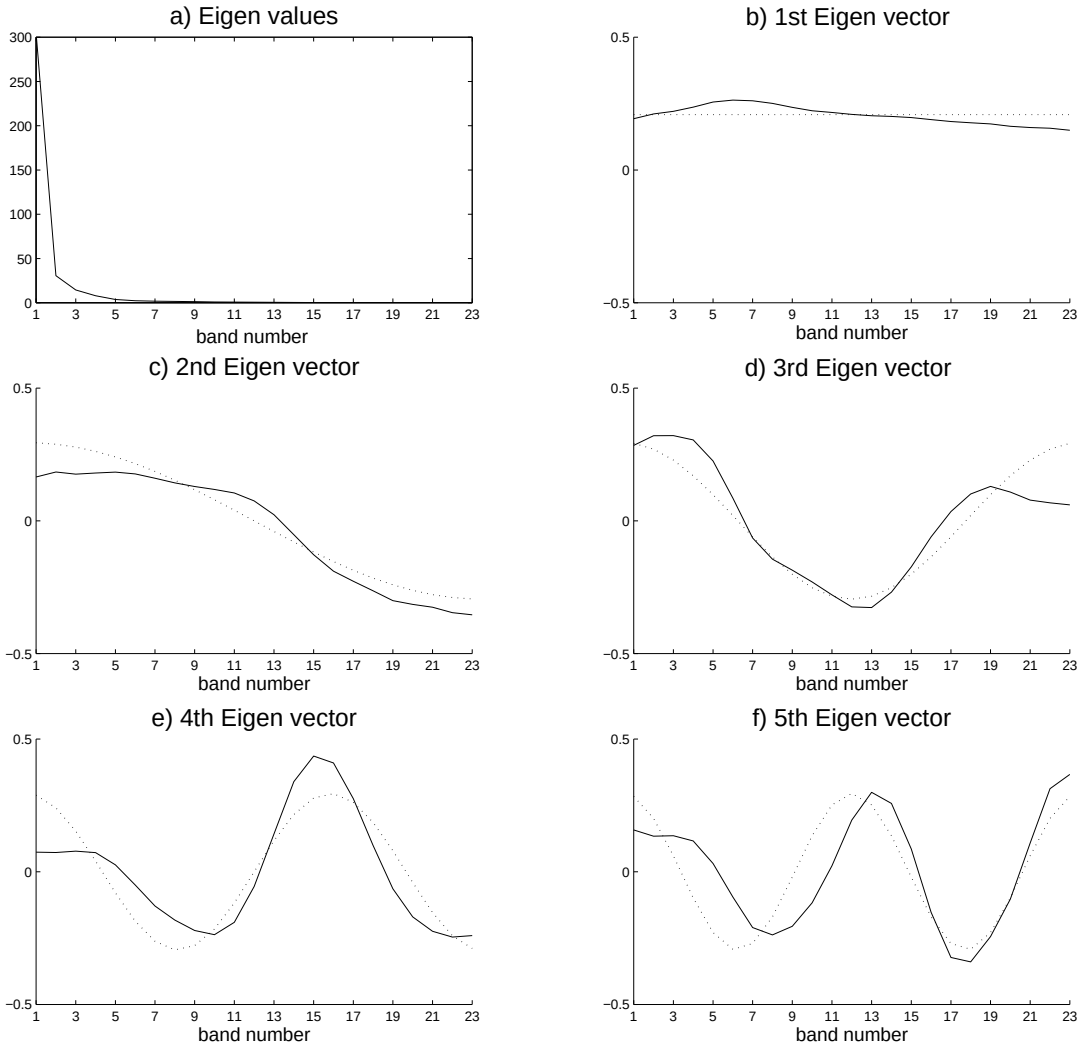


Figure 2.7: Spectral basis derived using PCA.

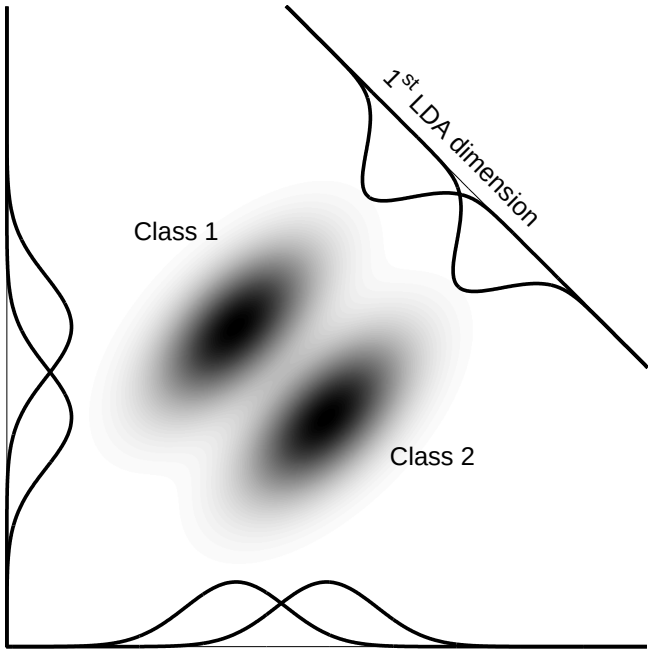


Figure 2.8: Linear Discriminant Analysis for 2-Dimensional Data.

($\mathbf{\Sigma}^{(1)} = \mathbf{\Sigma}^{(2)}$) no other direction can be obtained for a better discrimination.

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

$$\hat{\mathbf{\Sigma}}_{wc} = \frac{1}{N} \sum_{j=1}^{J} N_j \hat{\mathbf{\Sigma}}^{(j)}, \tag{2.20}$$

where J is the number of classes, N_j is the number of training vectors belonging to class j, N is the total number of all training vectors and $\hat{\mathbf{\Sigma}}^{(j)}$ is the estimate of covariance matrix for j^{th} class:

$$\hat{\mathbf{\Sigma}}^{(j)} = \frac{1}{N_j} \sum_{i=1}^{N_j} (\mathbf{x}_i^{(j)} - \hat{\boldsymbol{\mu}}^{(j)})(\mathbf{x}_i^{(j)} - \hat{\boldsymbol{\mu}}^{(j)})^T, \tag{2.21}$$

where $\mathbf{x}_i^{(j)}$ is the i^{th} training vector belonging to class j and $\hat{\boldsymbol{\mu}}^{(j)}$ is the estimated mean vector for class j:

$$\hat{\boldsymbol{\mu}}^{(j)} = \frac{1}{N_j} \sum_{i=1}^{N_j} \mathbf{x}_i^{(j)}. \tag{2.22}$$

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

$$\hat{\mathbf{\Sigma}}_{ac} = \frac{1}{N}\sum_{j=1}^{J} N_j(\hat{\boldsymbol{\mu}}^{(j)} - \hat{\boldsymbol{\mu}})(\hat{\boldsymbol{\mu}}^{(j)} - \hat{\boldsymbol{\mu}})^T = \hat{\mathbf{\Sigma}} - \hat{\mathbf{\Sigma}}_{wc}, \tag{2.23}$$

where $\hat{\boldsymbol{\mu}}$ is the global mean vector estimate given by equation 2.19.

Like in the case of PCA, the importance of eigen vectors (ordering of LDA base vectors) is given by corresponding eigen values. Note, that for J classes, there are maximally $J-1$ non-zero eigen values (only $J-1$ LDA dimensions can be found containing discriminatory information).

2.5.3 Heteroscedastic Linear Discriminant Analysis

Heteroscedastic linear discriminant analysis (HLDA), which was first proposed by N. Kumar [48, 47], can be viewed as a generalization of LDA. HLDA again assumes that classes obey multivariate Gaussian distribution, however, the assumption of the same covariance matrix shared by all classes is relaxed. HLDA assumes that n-dimensional original feature space can be split into two statistically independent subspaces: While in p useful dimensions (containing discriminatory information), classes are well separated, in $(n-p)$ nuisance dimensions, the distributions of classes are overlapped. An example of 2-dimensional feature space with one useful and one nuisance dimension is shown in figure 2.9. The goal of HLDA is to find the transformation matrix $\mathbf{A}$ for n-dimensional vectors $\mathbf{x}$ that may be written as:

$$\bar{\mathbf{x}} = \mathbf{A}\mathbf{x} = \begin{bmatrix} \mathbf{A}_{[p]}\mathbf{x} \\ \mathbf{A}_{[n-p]}\mathbf{x} \end{bmatrix} = \begin{bmatrix} \bar{\mathbf{x}}_{[p]} \\ \bar{\mathbf{x}}_{[n-p]} \end{bmatrix}, \tag{2.24}$$

where $\mathbf{A}_{[p]}$ is a matrix consisting of the first p rows of $n \times n$ matrix $\mathbf{A}$ and $\mathbf{A}_{[n-p]}$ consists of the remaining $n-p$ rows, $\bar{\mathbf{x}}_{[p]}$ are deemed to be those useful dimensions in the rotated space and $\bar{\mathbf{x}}_{[n-p]}$ are the nuisance dimensions.

To find the optimal HLDA transformation, a model taking into account the above assumptions is created and its parameters are estimated in maximum likelihood framework. Distribution of class j is modeled by a single Gaussian in the rotated space:

$$p_j(\mathbf{x}) = \frac{\det(\mathbf{A})}{\sqrt{(2\pi)^n \det(\bar{\mathbf{\Sigma}}^{(j)})}} \exp\left(-\frac{(\mathbf{A}\mathbf{x} - \bar{\boldsymbol{\mu}}^{(j)})^T\, \bar{\mathbf{\Sigma}}^{(j)\,-1}\, (\mathbf{A}\mathbf{x} - \bar{\boldsymbol{\mu}}^{(j)})}{2}\right), \tag{2.25}$$

where $\bar{\boldsymbol{\mu}}^{(j)}$ and $\bar{\mathbf{\Sigma}}^{(j)}$ are mean vector and covariance matrix of class j in rotated space. Note that Kumar [48] distinguishes two different HLDA solutions, which assume

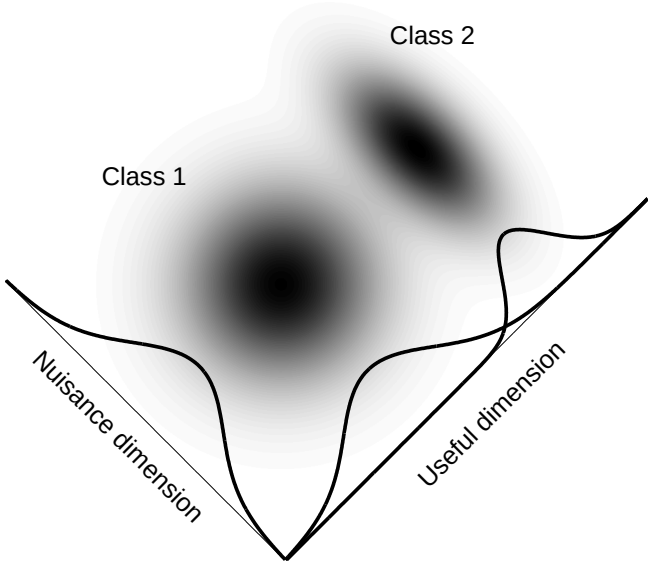


Figure 2.9: Heteroscedastic Linear Discriminant analysis.

that classes will be modeled using full covariance resp. diagonal covariance matrices in the rotated space.[1] Since the diagonal covariance modeling is the usual choice for state-of-the-art recognition systems and because it is also the choice in our experiments, we will consider only the diagonal covariance modeling case whenever talking about HLDA. Therefore, out-of-diagonal components of matrix $\mathbf{\Sigma}^{(j)}$ will be assumed to be zero in the model and $\bar{\sigma}_k^{(j)2}$ will denote the k^{th} component on the diagonal. To model the assumption that distribution of classes overlap for nuisance dimensions, parameters $\bar{\mu}_k^{(j)}$ and $\bar{\sigma}_k^{(j)2}$ will be *tied* over all classes for all $k > p$. For this model, we want to find parameters $\mathbf{A}$, $\bar{\boldsymbol{\mu}}^{(j)}$ and $\bar{\boldsymbol{\sigma}}^{(j)2}$ for all $j = 1 \dots J$ that maximize log-likelihood of the data:

$$\mathcal{L}(\{\mathbf{x_i}\}; \mathbf{A}, \{\bar{\boldsymbol{\mu}}^{(j)}, \bar{\boldsymbol{\sigma}}^{(j)\,2}\}) = \sum_j^J \sum_i^{N_j} \log p_j(\mathbf{x}_i^{(j)}) = \tag{2.26}$$

$$= \sum_j^J \sum_i^{N_j} \left[\frac{1}{2} \log \left(\frac{\det(\mathbf{A})^2}{(2\pi)^n \prod_{k=1}^n \bar{\sigma}_k^{(j)\,2}} \right) - \sum_{k=1}^n \frac{\left(\mathbf{a}_k \mathbf{x}_i^{(j)} - \bar{\mu}_k^{(j)} \right)^2}{2\, \bar{\sigma}_k^{(j)\,2}} \right],$$

[1] In the case of diagonal covariance modeling, optimal HLDA rotation not only identifies useful and nuisance dimensions, but also (if possible) decorrelates features of individual classes.

where $\mathbf{x}_i^{(j)}$ is the i^{th} training vector belonging to class j and $\mathbf{a}_k$ is k^{th} row of matrix $\mathbf{A}$. To simplify the above equation, maximum likelihood estimates of parameters $\bar{\mu}_k^{(j)}$ and $\bar{\sigma}_k^{(j)2}$ can be found in terms of fixed transformation $\mathbf{A}$ by differentiating equation 2.27 with respect to parameters $\bar{\mu}_k^{(j)}$ and $\bar{\sigma}_k^{(j)2}$ and finding the point were partial derivatives are zero:

$$\hat{\bar{\mu}}_k^{(j)} = \begin{cases} \mathbf{a}_k \hat{\boldsymbol{\mu}}^{(j)} & k \leq p \\ \mathbf{a}_k \hat{\boldsymbol{\mu}} & k > p \end{cases} \tag{2.27}$$

$$\hat{\bar{\sigma}}_k^{(j)\,2} = \begin{cases} \mathbf{a}_k \hat{\boldsymbol{\Sigma}}^{(j)} \mathbf{a}_k^T & k \leq p \\ \mathbf{a}_k \hat{\boldsymbol{\Sigma}} \mathbf{a}_k^T & k > p \end{cases} \tag{2.28}$$

where $\hat{\boldsymbol{\mu}}^{(j)}$, $\hat{\boldsymbol{\mu}}$, $\hat{\boldsymbol{\Sigma}}^{(j)}$ and $\hat{\boldsymbol{\Sigma}}$ are estimates of class/global mean vectors and covariance matrices in the original space given by equations 2.22, 2.19, 2.21 and 2.18. Substituting estimates 2.27 and 2.28 into equation 2.27, we can now express log-likelihood of the data in terms of only transformation matrix $\mathbf{A}$ and known statistics $\hat{\boldsymbol{\Sigma}}^{(j)}$ and $\hat{\boldsymbol{\Sigma}}$:

$$\mathcal{L}(\{\mathbf{x_i}\}; A) = \sum_{j=1}^{J} \frac{N_j}{2} \log \left(\frac{\det(\mathbf{A})^2}{(2\pi)^n \prod_{k=1}^{p} \mathbf{a}_k \hat{\boldsymbol{\Sigma}}^{(j)} \mathbf{a}_k^T \prod_{k=p+1}^{n} \mathbf{a}_k \hat{\boldsymbol{\Sigma}} \mathbf{a}_k^T} \right) - \frac{Nn}{2}. \tag{2.29}$$

The transformation $\mathbf{A}$ that maximizes expression 2.29 is the optimal HLDA transformation. In contrast to LDA, unfortunately, there is no closed-form solution to this problem. Kumar [48] uses standard nonlinear optimization techniques to find HLDA solution.[2] Use of this optimization, however, turns out to be not feasible for larger problems because of both computation speed and memory requirements.

Instead, simple iterative optimization scheme proposed by Gales [22, 23] can be used, which is based on generalized EM algorithm. This algorithm is very efficient, converges very quickly and is stable. First, according to equation 2.28, Gales estimates variances $\{\hat{\bar{\boldsymbol{\sigma}}}^{(j)\,2}\}$ using a current estimate of the transformation $\mathbf{A}$. Then, unlike Kumar, he substitutes only mean estimates 2.27 into equation 2.27 and expresses likelihood of the data in terms of the fixed current estimates of variances $\{\hat{\bar{\boldsymbol{\sigma}}}^{(j)\,2}\}$ as

$$\mathcal{L}(\{\mathbf{x_i}\}; \mathbf{A}, \{\hat{\bar{\boldsymbol{\sigma}}}^{(j)\,2}\}) = \frac{1}{2} \left(\log \left(\left(\mathbf{c}_i \mathbf{a}_i^T \right)^2 \right) - K - \sum_{k=1}^{n} \mathbf{a}_k \mathbf{G}^{(k)} \mathbf{a}_k^T \right), \tag{2.30}$$

where $\mathbf{c}_i$ is the i^{th} row of cofactor matrix $C = \det(\mathbf{A})\mathbf{A}^{-1}$ for current estimate of $\mathbf{A}$,

$$K = \sum_{j=1}^{J} N_j \log \left((2\pi)^n \prod_{k=1}^{n} \bar{\sigma}_k^{(j)\,2} \right) \tag{2.31}$$

[2] Matlab implementation of HLDA based on nonlinear optimization can be found directly in Kumar Thesis [48].

are all terms independent of $\mathbf{A}$ and

$$\mathbf{G}^{(k)} = \begin{cases} \sum_{j=1}^{J} \frac{N_j}{\hat{\sigma}_k^{(j)\,2}} \hat{\mathbf{\Sigma}}^{(j)} & k \leq p \\ \frac{N}{\hat{\sigma}_k^{(j)\,2}} \hat{\mathbf{\Sigma}} & k > p \end{cases} \tag{2.32}$$

By differentiating equation 2.30 with respect to $\mathbf{a}_i$ and equating to zero, we can find maximum likelihood estimate of i^{th} row of transformation matrix $\mathbf{A}$ for the fixed $\{\hat{\boldsymbol{\sigma}}^{(j)2}\}$. It can be shown [23] that such estimate is given by

$$\hat{\mathbf{a}}_k = \mathbf{c}_k \, \mathbf{G}^{(k)\,-1} \sqrt{\frac{N}{\mathbf{c}_k \, \mathbf{G}^{(k)\,-1} \, \mathbf{c}_k^T}}. \tag{2.33}$$

The whole optimization is an iterative process where variances, $\hat{\boldsymbol{\sigma}}^{(j)\,2}$, and transformation matrix, $\mathbf{A}$, are alternately re-estimated. Re-estimation of matrix $\mathbf{A}$ itself has again an iterative character. It is re-estimated row-by-row using equation 2.33, where each row is related to other rows only by the cofactors. Matlab implementation of this iterative HLDA optimization can be found in appendix A.

As described above, HLDA considers model where parameters $\bar{\mu}_k^{(j)}$ and $\bar{\sigma}_k^{(j)\,2}$ are tied over all classes for nuisance dimensions. If the model is further constrained so that $\bar{\sigma}_k^{(j)\,2}$ are tied also for useful dimensions (we assume that all classes have the same covariance matrices and differ only in mean vectors), it can be proven [48] that for such constrained model, LDA solution is also HLDA solution.

In the special case, where $p = n$ (no dimensionality reduction is performed), HLDA transformation equals to Maximum Likelihood Linear Transform (MLLT) [25], which is also often referred as diagonalization transform. The idea of feature diagonalization was recently further elaborated by several researchers, which resulted in development of more sophisticated modeling techniques such as: Maximum Likelihood Multiple Projection Schemes [22], Extended Maximum Likelihood Linear Transform (EMLLT) [58], Subspace Precision and Mean models (SPAM) [3] and Subspace Constrained Gaussian Mixture Models (SCGMM) [2].

Note that an alternative definition of HLDA (sometime refereed as HDA) proposed by Saon [60] also exists, which is, however, not derived in the maximum likelihood framework.

Chapter 3

Complementarity of recognition systems

3.1 Introduction

In the past, many approaches have been developed to perform speech recognition, which differ in **feature extraction method** (MFCC [14], PLP [30], TRAPS [41, 26]), **classification method and model** (HMM, Hybrid ANN-HMM [9], HMMs of different types and topologies), **model training framework** (ML [43], MMI [57], MCE [13]), and so on. It is not possible to say, which approach is the right one. For example, in the case of feature extraction, it is not exactly known which information should be extracted from speech. Moreover, an attempt to preserve one part of information often leads to loss of another (e.g. resolution in the time vs. resolution in the frequency). Speech recognition systems based on these different approaches often show considerable complementarity of their outputs, which means that different systems make errors in different situations. It has been proved that combination of different systems can be powerful technique to improve recognition performance [46, 42, 8, 19, 45, 61]. The level of success is however limited by the complementarity of systems combined. In this work, we propose a method to measure this complementarity allowing to select such systems whose combination is the most beneficial.

The combination can be performed at different levels. For example, in our experiments, all systems differ only in feature extraction method and they could be, therefore, combined directly at feature level, leaving the rest of the system unchanged. In this case, individual feature streams could be combined into one stream using some technique (such as PCA, LDA (see sections 2.5.1 and 2.5.2) or Tandem[1] [31, 18]) preserving the important information encoded in the original streams. In our initial set of experiments, however, outputs of individual recognizers — word (symbol) sequences — are combined using technique known as ROVER (Recognizer Output Voting Error Reduction) [20]. The proposed measures of complementarity are also based on comparing output word sequences. However, we have shown that these measures are meaningful also for other methods of system combination (see chapters 4 and 5).

The derivation of complementarity measures is based on techniques similar to those used by ROVER. Therefore, ROVER is briefly described in the next section. In section 3.4, measures of error dependency between two recognition systems are developed. In experiments, it is shown that these measures are useful for selection of systems good for combination. Measures of complementarity of *a set* of systems are proposed in section 3.5 and a correlation between these measures and actual performances of systems combined using ROVER is shown.

3.2 Terminology

In the following text, term *system* will be used to denote individual speech recognition system. Terms *system output* or *output sequence* will denote sequence of words recognized by the system. Term *combined system* will be used for combination of individual systems. ROVER is used in our experiments for system combination, therefore, *combined system output* is obtained as ROVER combination of output word sequences of individual recognition systems. Term *system set* will be used to denote set of individual systems available for combination. Where only several systems from currently used system set are combined, a term *system subset* will be used.

[1] Tandem is a technique where Artificial Neural Network is used to map (set of) features to posterior probabilities of classes (usually given by phoneme labels). These posterior probabilities are then used as features for HMM recognizer.

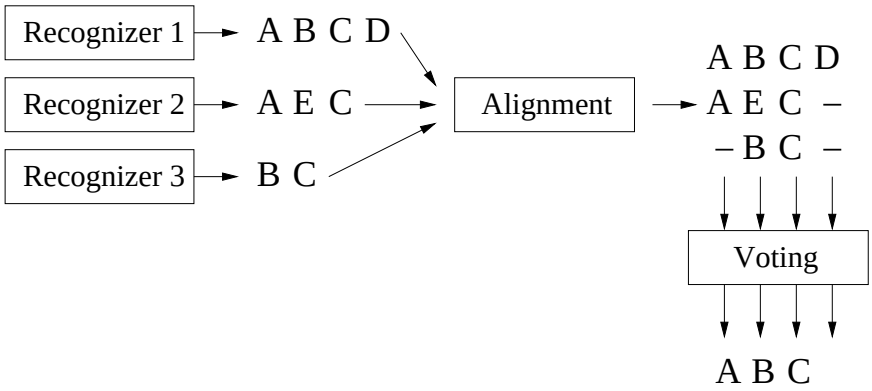


Figure 3.1: ROVER method block diagram.

3.3 ROVER - Recognizer Output Voting Error Reduction

ROVER [20] is a technique allowing to combine word (symbol) sequences taken as outputs of different recognition systems. Philosophy of this method is illustrated in figure 3.1. First, alignment of word sequences is performed to find corresponding words over different sequences. In this step, all sequences are merged into one sequence of *correspondence sets*, where each *correspondence set* is a multi-set[2] that contains corresponding words one from each individual sequence. In figure 3.1, *correspondence sets* are represented by columns of words on the output of alignment block. As can be seen in figure 3.1, there can be no corresponding word from a particular sequence in a *correspondence set*. In such case, *null word* (symbol '-' in figure) is added into the *correspondence set*. In the second step, final symbol sequence is obtained by selecting one word from each *correspondence set* using voting algorithm. In our experiments simple majority voting is used. Note, that for *correspondence set*, where *null word* is the winning one, no word is output to the final sequence.

3.3.1 ROVER Alignment

Merging of individual word sequences into one sequence of *correspondence sets* is performed iteratively. Initially, first two sequences are aligned to produce *correspondence sets* each having only two elements. The alignment is performed the same way that is commonly used for scoring performances of speech recognition systems: The reference word sequence is aligned with the recognized one, to allow for counting of insertions, deletions and substitutions. Such alignment, that minimizes the total cost is found

[2]Multi-set is a set where multiple occurrences of the same element are allowed.

using Dynamic Programming. In our implementation, the cost of each deletion and insertion is 3, the cost of a substitution is 4 and cost of a correct word is 0. Of course, when ROVER is used to combine different recognized output sequences, we do not have any reference sequence. However, assuming the first sequence being the reference will allow us to use terms: insertion and deletion in the following examples.

In the example from figure 3.1, *correspondence set* sequence created in the first iteration is:

$$\begin{array}{r} 1st\ sequence \\ 2nd\ sequence \end{array} \begin{pmatrix} A \\ A \end{pmatrix}, \begin{pmatrix} B \\ E \end{pmatrix}, \begin{pmatrix} C \\ C \end{pmatrix}, \begin{pmatrix} D \\ - \end{pmatrix}$$

and cost for this alignment is: $4 + 3 = 7$ corresponding to one substitution of the word E for B and one deletion of the word D.

In each next iteration, next word sequence is aligned with the *correspondence sets* obtained in the previous iteration. The alignment is performed the same way as the alignment of first two sequences, however, sequence of *correspondence sets* now serve as the reference and cost must be always computed with respect to all words in each *correspondence set*. In the example from figure 3.1, *correspondence set* sequence created in the second (last) iteration is:

$$\begin{array}{r} 1st\ sequence \\ 2nd\ sequence \\ 3rd\ sequence \end{array} \begin{pmatrix} A \\ A \\ - \end{pmatrix}, \begin{pmatrix} B \\ E \\ B \end{pmatrix}, \begin{pmatrix} C \\ C \\ C \end{pmatrix}, \begin{pmatrix} D \\ - \\ - \end{pmatrix}$$

and cost for this alignment is: $2 \times 3 + 4 + 3 = 13$ corresponding to two deletions of the word A, one substitution of B for E and one deletion of D.

Note that this iterative method of alignment of multiple sequences is not the optimal one and the order in which individual sequences are aligned is important. In our experiments, outputs of systems that perform the best[3] are aligned first, outputs of systems with poorer performance are added later. This has been experimentally shown to be a reasonable suboptimal solution. N-dimensional Dynamic Programming would have to be used to obtain the optimal alignment, where N is the number of sequences aligned. However, such alignment would be very computationally expensive for higher N.

[3] in terms of Word Error Rate of individual systems

3.4 Complementarity of two recognition systems

It was mentioned in section 3.1 that the improvement of recognition performance given by combination of different systems is limited by amount of complementarity of systems combined. In our experiments ROVER is used to combine systems at the level of output word sequences. Therefore, we are interested in complementarity encoded in these sequences, which is represented by independency of errors that individual systems make.

We will distinguish two types of error dependency. We will say that two systems make *simultaneous error* if both systems make error at the same time. Both systems can, however, make different errors (e.g. correct word A is recognized by first system as B and by second system as C). We will say that two systems make *dependent error* in the special case where both systems make the same error.

In this section, measures of complementarity of two systems based on counting *simultaneous and dependent errors* are proposed. Extension for measuring complementarity of a whole set of systems will be proposed in section 3.5. Measures of complementarity of two systems are estimated from a selected set of utterances in the following steps:

- For each utterance, output sequences of both systems are obtained.
- Each pair of sequences is aligned with the corresponding reference sequence according to algorithm described in section 3.4.1.
- For each pair of sequences, *simultaneous and dependent errors* are counted (see section 3.4.2).
- Counts of *simultaneous and dependent errors* are used to compute complementarity measures proposed in section 3.4.3.

3.4.1 Alignment for identification of error dependency

To identify where two systems make dependent errors, for each utterance from a given set, corresponding output word sequences of both systems are aligned with the reference word sequence. Alignment is performed in similar manner as ROVER alignment described in section 3.3.1. Output sequence of one system is aligned with the reference sequence first. However, when the second output sequence is added, the alignment is performed with respect to words only from the reference sequence, and

output of first system is taken into account, only if more than one alignment with the reference sequence having the minimal cost is available. This is best illustrated on the following example:

Let the reference sequence be only one word C. Both systems are apt to insert words so that output sequences of the first and the second system are: C, X, X, X, Z and X, X, X, C, Z, respectively. These two sequences will be referred as *sequence 1* and *sequence 2*. The alignment with minimal cost that would be used by ROVER for these three sequences is:

$$\begin{array}{l} reference \\ sequence\ 1 \\ sequence\ 2 \end{array} \begin{pmatrix} C \\ C \\ - \end{pmatrix}, \begin{pmatrix} - \\ X \\ X \end{pmatrix}, \begin{pmatrix} - \\ X \\ X \end{pmatrix}, \begin{pmatrix} - \\ X \\ C \end{pmatrix}, \begin{pmatrix} - \\ Z \\ Z \end{pmatrix}$$

For identification of error dependency, both alignment of *sequence 1* with the reference and alignment of *sequence 2* with the reference should be the same as those used for scoring of these sequences (see section 3.3.1). In contrast to *sequence 1*, *sequence 2* is not aligned with the reference in such manner in the case of ROVER alignment (word C from *sequence 2* is not aligned with word C from the reference). For this reason, in alignment used for identification of error dependency, *sequence 2* is preferably aligned to the reference sequence. The following alignment is then obtained for sequences from our example:

$$\begin{array}{l} reference \\ sequence\ 1 \\ sequence\ 2 \end{array} \begin{pmatrix} - \\ - \\ X \end{pmatrix}, \begin{pmatrix} - \\ - \\ X \end{pmatrix}, \begin{pmatrix} C \\ C \\ C \end{pmatrix}, \begin{pmatrix} - \\ X \\ - \end{pmatrix}, \begin{pmatrix} - \\ X \\ - \end{pmatrix}, \begin{pmatrix} - \\ X \\ - \end{pmatrix}, \begin{pmatrix} - \\ Z \\ Z \end{pmatrix}$$

When aligning *sequence 2*, the words from *sequence 1* are initially ignored and the word C is therefore aligned with the word C from the reference sequence. The word Z can be, however, aligned with the same cost with any *null word* following C in the reference sequence. Here, the words from *sequence 1* are also taken into account and the secondary cost minimization leads to the alignment of the words Z from *sequence 1* and *sequence 2*.

Note that the order in which two output sequences are aligned is again important. The following example demonstrates the case where different order results in different alignments. Consider sequences: B, C, A as the reference, B, C, B as *sequence X* and B, B as *sequence Y*. If *sequence X* is aligned with the reference sequence first and then

sequence Y is added, the following optimal alignment is obtained:

$$\begin{array}{c} reference \\ sequence\ X \\ sequence\ Y \end{array} \begin{pmatrix} B \\ B \\ B \end{pmatrix}, \begin{pmatrix} C \\ C \\ - \end{pmatrix}, \begin{pmatrix} A \\ B \\ B \end{pmatrix}$$

The opposite order of processing sequences Y and X can lead to the following (wrong) alignment of the reference sequence with *sequence Y*:

$$\begin{array}{c} reference \\ sequence\ Y \end{array} \begin{pmatrix} B \\ B \end{pmatrix}, \begin{pmatrix} C \\ B \end{pmatrix}, \begin{pmatrix} A \\ - \end{pmatrix}$$

Until we see *sequence X* we do not know that it is better to align second word B from *sequence Y* with word A from the reference, and the alignment with word C, which has the same cost, can be chosen. Adding *sequence X*, which already does not affect this wrong decision, leads to the following final alignment:

$$\begin{array}{c} reference \\ sequence\ Y \\ sequence\ X \end{array} \begin{pmatrix} B \\ B \\ B \end{pmatrix}, \begin{pmatrix} C \\ B \\ C \end{pmatrix}, \begin{pmatrix} A \\ - \\ B \end{pmatrix}$$

Note that the optimal alignment could be obtained using 3-dimensional Dynamic Programming.

3.4.2 Counting simultaneous and dependent errors

Once corresponding outputs of two systems are aligned with their references, *simultaneous and dependent errors* can be counted. The following example demonstrates alignment of sequences with two *simultaneous errors* where words A and D are incorrectly recognized by both systems. Moreover, in the case of word D, both systems make the same error (words are deleted) and therefore this error is also *dependent error*.

$$\begin{array}{c} reference \\ output\ 1 \\ output\ 2 \end{array} \begin{pmatrix} A \\ E \\ F \end{pmatrix}, \begin{pmatrix} B \\ B \\ B \end{pmatrix}, \begin{pmatrix} C \\ G \\ C \end{pmatrix}, \begin{pmatrix} D \\ - \\ - \end{pmatrix}$$

For measuring error dependency, we are not interested in the cases where only one system makes error (word C is incorrectly recognized only by first system).

3.4.3 Measurement of error dependency between two systems

Let N_{ref} be the total number of words in all reference sequences for the set of utterances used to estimate complementarity measures. Let $N_{sim}(i,j)$ and $N_{dep}(i,j)$ be the total number of *simultaneous errors* and *dependent errors* between i^{th} and j^{th} system, respectively. We propose the following measures of error dependency between two systems:

Lower Bound Word Error Rate (LBWER)

for two systems i and j is defined as the ratio between the number of *simultaneous errors* and the overall number of words in the set of utterances:

$$LBWER(i,j) = \frac{N_{sim}(i,j)}{N_{ref}} \times 100. \tag{3.1}$$

We can also regard this measure as an error rate of such system combining outputs of two recognizers that always selects (using an ideal confidence measure) the correct word for all the cases where only one recognizer makes error (therefore the name Lower Bound WER).

For a set of systems S and $\forall i,j \in S$, the values of $LBWER(i,j)$ form a matrix. We will call it *LBWER matrix of set S*. Note, that each value on the matrix diagonal $LBWER(i,i)$, which is the ordinary WER for the system i, is the highest value in the corresponding row and column.

Dependent Word Error Rate (DWER)

for two systems i and j is defined as the ratio between the overall number of *dependent errors* and the number of words in our set of utterances:

$$DWER(i,j) = \frac{N_{dep}(i,j)}{N_{ref}} \times 100. \tag{3.2}$$

DWER matrix of a system set is defined in the same manner as *LBWER matrix*. Note, that values on the *DWER matrix* diagonal are again ordinary WERs of individual systems.

3.4.4 Properties of error dependency measures

If a set of at least three systems has diagonal *LBWER matrix*, ROVER combination of these systems based on majority voting must result in zero WER. Systems make

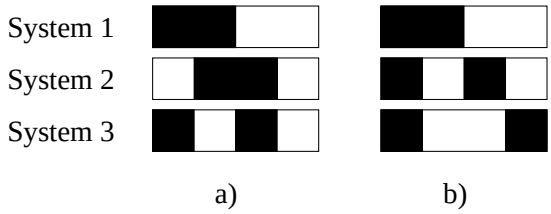


Figure 3.2: Different systems with the same DWER matrix.

	Sys. 1	Sys. 2	Sys. 3
System 1	50	25	25
System 2	25	50	25
System 3	25	25	50

Table 3.1: DWER matrix for both systems from figure 3.2.

no *simultaneous errors* in such case, and therefore a single system making an error is always outvoted by all others. Note, that this does not have to be true for a set of systems with diagonal *DWER matrix*. On the other hand, *dependent errors* measured by DWER can be seen as the worse variants of *simultaneous errors*, since any time systems make *dependent error*, we need even more correct answers to outvote the error. We can, therefore, intuitively expect that system set, in order to be good for combination, must generally have small values out of *LBWER matrix* and *DWER matrix* diagonals (but also values on diagonals representing ordinary WERs should be small, since performance of individual system is also important).

Both *LBWER* and *DWER matrices* are, however, not directly related to performance of combined system. It can be proven using the following example showing two sets of systems with identical *LBWER* and *DWER matrix* where the combination of systems from each set leads to different results:

Figure 3.2a represents a set of three systems, where each row bar corresponding to one system shows portion of correctly recognized words (white area) and incorrectly recognized words (black area). Overlapping parts of black areas of two systems correspond to *simultaneous error* between the systems. Let us assume that all *simultaneous errors* are also *dependent errors* in this example. Therefore *LBWER matrix* of this system set, which is shown in table 3.1, will be identical to *DWER matrix*. WERs of all individual systems are 50% (values in matrix diagonal), *dependent errors* for all pairs of systems are 25% (values out of diagonal). Majority voting based combination

of systems from figure 3.2a would result in WER of 75%, since all systems vote for correct word only for the portion corresponding to last quarter of row bars. In all other cases, there are always two systems making errors outvoting the correct one. The combined system therefore performs even worse than the individual systems. On the other hand, system combination not based only on majority voting can, for example, choose the output word using a word confidence measure that in ideal case always prefers the correct word. Such combination can still result in WER of 0%, since there is always at least one system producing the correct word in figure 3.2a.

Figure 3.2b represents a set of systems that have *LBWER* and *DWER matrices* identical to those obtained for system set from figure 3.2a. In this case however, majority voting based combination would result in better WER of 25% and in opposite, the mentioned word confidence measure based system combination cannot have WER better than 25% since all systems make errors for the portion corresponding to first quarter of row bars. The difference is caused by the triple error dependency (three systems make dependent error) that is already not captured in *LBWER* and *DWER matrices* representing only dependencies between pairs of systems.

Although it was shown that there is no direct relationship between performance of combined systems[4] and values from *LBWER* and *DWER matrices*, we can still expect that there is certain correlation between them. In the next section, experimental setup will be described, where recognition systems using different feature sets are combined by ROVER. *LBWER* and *DWER matrices* of these systems will be analyzed and we will observe that LBWER and DWER measures can be useful for selection of systems good for combination. In section 3.5, complementarity measures for a whole set of systems are proposed, which are based on the values from *LBWER* and *DWER matrices*. Correlation between these measures and performance of ROVER combining a set of systems is shown.

3.4.5 Experimental setup

Speech data from TI Connected Digits database [49] were used for both training and testing of all recognition systems. Limited number of clean speech utterances were selected for training (616 utterances from 4 male and 4 female speakers). Four types of noise (subway, car, exhibition, babble) from AURORA2 TI Digits database [35] were artificially added to speech data at SNR levels of 20dB and 10dB. The same 616 utterances were used to create data for all noisy conditions. Together $616 \times (1+4 \times 2) =$

[4] At least for the combination techniques mentioned in the example.

5544 utterances were used for training.

Test data set was prepared in a similar manner. Here, 912 utterances from 12 male and 12 female speakers not seen in the training data were used, 4 noises used for training and four unseen noises (train station, airport, restaurant, street) were added to test data. Additionally, SNR 0dB condition was generated for both seen and unseen noises. Together $912 \times (1 + 8 \times 3) = 22800$ utterances were used for testing.

Nine recognition systems were trained, each using different feature extraction method. The following feature extractions were performed:

- **BSL** - 15 Mel Frequency Cepstral Coefficients [14] augmented with their first and second order derivatives (delta and double-delta), filter bank applied on magnitude spectrum, 23 bands in Mel filter bank, 25 ms window length, 10 ms frame rate, 5 frames delta and delta-delta window, frame energy is represented by C0 coefficient
- **LPCC** 15 Linear Prediction Cepstral coefficients augmented with their derivatives (LPC order 15, other parameters similar to BSL features)

The name BSL stays for "baseline", since all seven remaining feature extraction methods are only modifications of BSL methods and always only one of their parameters is changed. In the following list, only the changed parameter of BSL features is described:

- **DA1** - length of delta and delta-delta window is 3 (± 1) frames
- **DA4** - length of delta and delta-delta window is 9 (± 4) frames
- **B15** - filter bank consists of 15 bands
- **B30** - filter bank consists of 30 bands
- **ENG** - log frame energy is computed as a replacement for C0
- **POW** - filter bank is applied on power spectra
- **NOE** - only coefficients C1 to C14 are used (no C0 or frame energy)

Except the feature extraction part, all recognition systems are the same. They are based on continuous HMMs with output probability density function modeled

Condition	**Clean**	**Seen noises**			**Unseen noises**			**Seen**
SNR level	-	**20dB**	**10dB**	**0dB**	**20dB**	**10dB**	**0dB**	**cond.**
System:								
POW	1.11	1.70	4.55	48.50	1.50	3.70	37.03	2.90
DA4	1.34	1.58	4.65	48.67	1.45	3.63	36.34	2.92
30B	1.76	1.62	4.68	52.55	1.57	3.77	40.38	3.00
ENG	1.37	1.69	4.72	44.11	1.63	4.12	35.97	3.00
BSL	1.37	1.75	4.74	51.18	1.58	3.77	38.51	3.04
15B	1.63	1.63	5.03	51.66	1.54	4.20	40.94	3.14
LPCC	1.44	1.62	5.59	44.50	1.64	4.41	29.97	3.36
DA1	1.89	2.06	5.39	54.87	1.80	4.30	44.73	3.52
NOE	3.59	1.71	5.47	58.52	1.97	4.80	48.66	3.59
ROVER 9	1.14	1.41	4.14	49.55	1.35	3.38	37.38	2.59

Table 3.2: Word Error Rates of individual recognizers.

by Gaussian mixture (3 mixture components). Whole word models with left-to-right topology (16 states for digits, 3 states for silence) are used.

The names of the feature extraction methods will also serve to distinguish individual systems in the following text. For example, system with BSL features will be referred as BSL system.

Table 3.2 shows WERs of all individual recognition systems for different levels of SNR for both seen and unseen conditions. All values in the table for seen and unseen noises are averaged WERs for four seen or four unseen types of noise. In the experiments, we will need a single value representing the system performance, with respect to which we can look for the optimal system combination. For this purpose, we will use WER evaluated on a subset of test data containing: clean data and data corrupted by seen noises with SNR 20dB and 10dB. This data subset will be referred to as *seen conditions test data*. WERs for individual recognition systems evaluated on this subset can be seen in the last column of table 3.2. In the last row, there are WERs for ROVER combination of all nine systems. The overall performance of ROVER is generally better than performance of any individual system, however, for certain conditions (clean speech and SNR 0dB) some systems are even outperforming ROVER.

3.4.6 Analysis of LBWER and DWER matrices

Seen conditions test data is also used to derive *LBWER* and *DWER matrices*. Here, one could object that test data should not be used for estimation of complementarity measures based on *LBWER and DWER matrices*. However, this is not a problem in our case, since, we will be interested in real correlation between combined system recognition performance and measures estimated from the same data that was used for recognition. We can also consider *seen conditions test data* to be the evaluation set that is only intended to find the best combined system. Another question is, how the error dependency statistics estimated from this data set will generalize for other test data. For this purpose, we can look at results obtained for unseen noises, which are not used for estimation of *LBWER and DWER matrices*.

For our set of nine systems, the estimate of *LBWER matrix* defined by equation 3.1 is shown in table 3.3. Values in the matrix diagonal are ordinary WERs of individual systems from the last column of table 3.2. Although *LBWER matrix* should be symmetric, we can see that corresponding values slightly differ in table 3.3. In section 3.4.1, suboptimal alignment used in our experiments for estimation of LBWER and DWER was described and the importance of the order in which output sequences of two systems are aligned with the reference sequence was noticed. Each pair of corresponding values in table 3.3 corresponds to these two different alignment orders. The differences between the corresponding values are, however, very small, which proves the proper functionality of the suboptimal alignment method used. *DWER matrix* with similar properties defined by equation 3.2 can be found in table 3.4.

In both tables 3.3 and 3.4, it can be directly observed, that values in the row and column corresponding to system DA4 are considerably smaller than other values. These lower values of LBWER and DWER indicate high complementarity of DA4 system with all other systems. Moreover, among the systems in our set, DA4 system has the second lowest WER. Therefore, it will be a hot candidate for combination. Second system that seems to be quite complementary to other systems is LPCC.

The complementarity of both systems DA4 and LPCC is probably even more visible in figure 3.3, which is graphical representation of *LBWER matrix*. The bright rows and and columns corresponding to DA4 and LPCC systems represent low LBWER values. In opposite, we can see a darker block representing LBWERs between systems POW, 30B, ENG and BSL. It indicates a higher error dependency between these systems, which, we expect, implies their lower complementarity. Figure 3.4 showing similar graphical representation of *DWER matrix*, is visually almost identical with figure 3.3.

System	POW	DA4	30B	ENG	BSL	15B	LPCC	DA1	NOE
POW	2.90	1.75	2.17	2.23	2.36	2.11	1.85	2.01	2.20
DA4	1.75	2.92	1.72	1.76	1.74	1.65	1.60	1.65	1.85
30B	2.18	1.71	3.00	2.17	2.46	2.01	1.77	2.09	2.09
ENG	2.22	1.75	2.16	3.00	2.25	2.03	1.92	2.10	2.17
BSL	2.36	1.74	2.46	2.26	3.04	2.03	1.80	2.14	2.19
15B	2.11	1.64	2.01	2.02	2.02	3.14	1.86	1.94	2.10
LPCC	1.85	1.59	1.77	1.91	1.80	1.86	3.36	1.81	1.86
DA1	2.01	1.65	2.09	2.10	2.14	1.94	1.82	3.52	2.03
NOE	2.19	1.85	2.09	2.17	2.18	2.09	1.87	2.02	3.59
Avg.	1.85	**1.52**	1.83	1.85	1.88	1.75	**1.61**	1.75	1.83

Table 3.3: LBWER matrix for set of nine systems.

System	POW	DA4	30B	ENG	BSL	15B	LPCC	DA1	NOE
POW	2.90	1.51	1.97	2.02	2.22	1.89	1.62	1.76	1.88
DA4	1.52	2.92	1.43	1.50	1.50	1.36	1.33	1.43	1.56
30B	1.98	1.43	3.00	1.93	2.30	1.73	1.46	1.85	1.71
ENG	2.03	1.50	1.93	3.00	2.03	1.77	1.62	1.83	1.79
BSL	2.22	1.50	2.30	2.04	3.04	1.78	1.51	1.91	1.84
15B	1.89	1.36	1.73	1.78	1.79	3.14	1.56	1.64	1.79
LPCC	1.61	1.32	1.46	1.61	1.51	1.53	3.36	1.51	1.48
DA1	1.77	1.42	1.84	1.84	1.89	1.63	1.51	3.52	1.75
NOE	1.88	1.55	1.70	1.79	1.84	1.79	1.51	1.75	3.59
Avg.	1.66	**1.29**	1.59	1.61	1.67	1.50	**1.35**	1.52	1.53

Table 3.4: DWER matrix for set of nine systems.

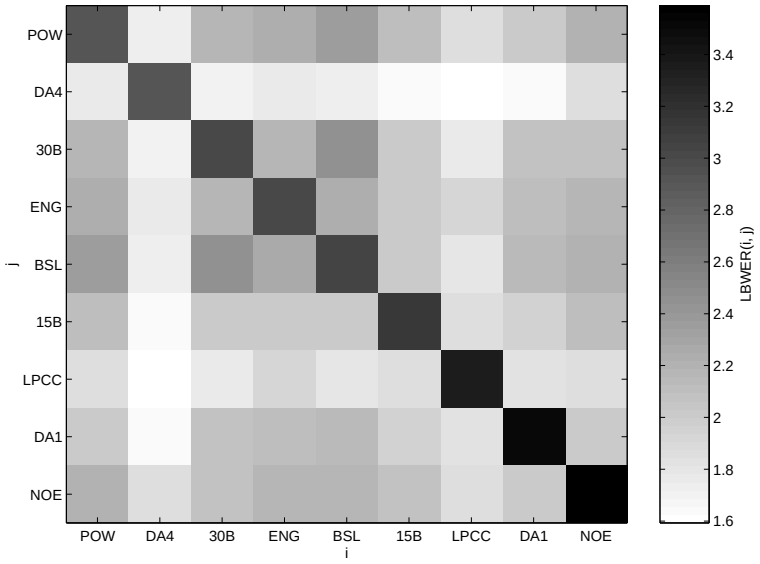


Figure 3.3: LBWER matrix for set of nine systems.

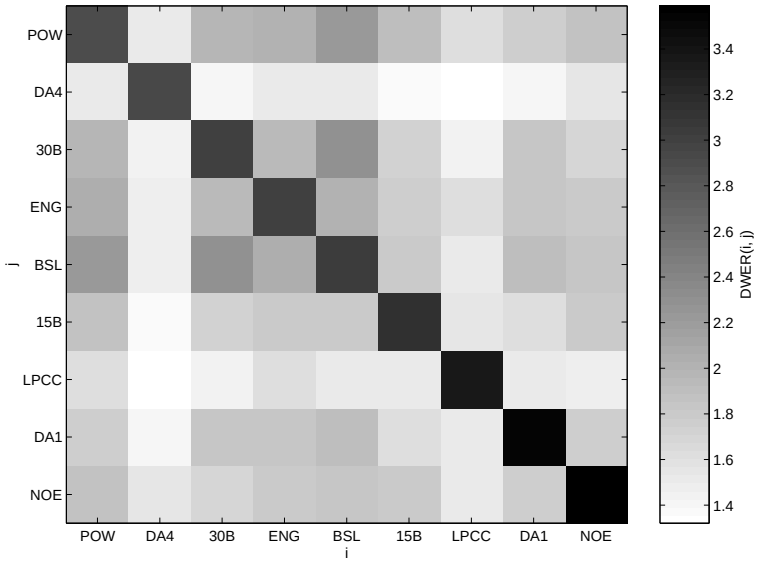


Figure 3.4: DWER matrix for set of nine systems.

System	POW	DA4	30B	ENG	BSL	15B	LPCC	DA1	NOE
POW	100.0	86.7	90.6	90.6	94.2	89.7	87.5	87.6	85.5
DA4	86.7	100.0	83.1	85.3	86.2	82.0	83.0	86.6	84.1
30B	91.0	83.7	100.0	89.3	93.5	85.9	82.6	88.4	81.9
ENG	91.4	85.5	89.3	100.0	90.3	87.5	84.7	87.2	82.4
BSL	94.2	86.0	93.2	90.2	100.0	87.8	83.7	89.0	83.9
15B	89.5	83.0	86.1	88.0	88.2	100.0	84.0	84.5	85.5
LPCC	86.9	82.9	82.5	84.4	83.8	82.2	100.0	83.0	79.7
DA1	87.9	86.3	88.0	87.9	88.6	84.1	83.1	100.0	86.4
NOE	85.8	83.9	81.6	82.7	84.2	85.6	80.7	86.5	100.0

Table 3.5: Percentage of *dependent errors* in *simultaneous errors*.

Dependent errors were defined as a special case of *simultaneous errors*. Table 3.5 shows how many percent of *simultaneous errors* are also *dependent errors* for each pair of systems. All values in the diagonal are 100%, which corresponds to the fact that if two same systems are compared, all their errors are *simultaneous errors* and at the same time also *dependent errors*. For any pair of systems, most of *simultaneous errors* are also *dependent errors* (between 79.7% and 94.2%). We can, therefore, expect that measurement of complementarity based on LBWER will not differ dramatically from that based on DWER. Still, there is certain difference in the percentage of *dependent errors* for different pairs of systems, which justifies investigating both kinds of complementarity measurements.

3.4.7 Redundancy of a system in the system set

As an objective measure of one system complementarity with all other systems in the set, we propose to simply average values in *LBWER or DWER matrix* column (or row) corresponding to the system. Ordinary WERs of the systems (values on the diagonal) are excluded from averaging[5]. These column averages can be seen in the last rows of tables 3.3 and 3.4. In both tables, we observe that the lowest values indicating high complementarity with other systems corresponds to systems DA4 and LPCC, which

[5]Including system's own WERs in averaging can be seen as additional favoritism for systems with low WER. That may be also important while selecting systems suitable for combination. The effect of including or excluding WERs from complementarity measure computation will be demonstrated in experiments described in section 3.5.

Excluded system	POW	DA4	30B	ENG	BSL	15B	LPCC	DA1	NOE
ROVER WER	**2.49**	2.66	2.61	**2.58**	**2.54**	2.61	2.63	2.62	2.61

Table 3.6: ROVERing 8 of 9 systems. Some combinations of eight systems perform even better than the combination of all nine systems with WER of 2.59%. WERs of such combined systems are indicated by bold values in the table.

is in agreement with our previous findings. In opposite, the highest value indicating low complementarity corresponds to BSL system. This is an interesting and natural finding, because all other systems (except of LPCC) use features which are derived from BSL features by modifying only one of its parameters.

The proposed measurement of one system complementarity with all other systems is verified in an experiment, where only eight of nine systems are combined using ROVER. Here, we are interested in performance degradation when excluding one particular system from combination. In table 3.2, we saw that WER of ROVER combination of all nine systems is 2.59%. Table 3.6 shows combined system WERs depending on which system is excluded from combination. The highest degradation is caused by omitting system DA4, followed by systems LPCC, which were distinguished as two systems most complementary to other systems according to proposed complementarity measures. In opposite, three least complementary systems are BSL, POW and ENG. As can be seen in table 3.6, performance of ROVER even improves when excluding one of these three systems from combination.

In the next experiment, ROVER is used to combine all possible subsets of our system set, where each individual subset consists of three to nine systems. Five subsets with the lowest combined system WER are listed in table 3.7. All listed subsets contain systems DA4, LPCC, 15B and DA1, which are the four most complementary systems according to the measure based on *LBWER and DWER matrix* column averaging. The subset with the lowest combined system WER, which consists only of five systems, contains also BSL system — the worst system for combination according to the measures. We must, however, point out that proposed measures are correct only to measure suitability of a system for its combination with *all other systems*. The measures handicap BSL system mainly because of its very low complementarity with systems POW, 30B and ENG as can be seen, for example, in figure 3.3 (dark fields in BSL row). None of these systems is, however, included in the system subset with the lowest combined system WER. In opposite, brighter fields in BSL row indicate that BSL system is quite complementary to other four systems.

System set	Combined systems								WER [%]
best		DA4		BSL	15B	LPCC	DA1		2.44
2nd best	POW	DA4			15B	LPCC	DA1		2.45
3rd best		DA4		BSL	15B	LPCC	DA1	NOE	2.45
4th best		DA4	30B	BSL	15B	LPCC	DA1	NOE	2.45
5th best		DA4	30B		15B	LPCC	DA1		2.46

Table 3.7: Five best ROVER combinations.

Condition	**Clean**	**Seen noises**			**Unseen noises**			**Seen**
SNR level	-	**20dB**	**10dB**	**00dB**	**20dB**	**10dB**	**00dB**	**cond.**
POW	1.11	1.70	4.55	48.50	1.50	3.70	37.03	2.90
ROVER 9	1.14	1.41	4.14	49.55	1.35	3.38	37.38	2.59
System set								
best	1.21	1.32	3.85	50.16	1.27	3.27	36.69	2.44
2nd best	1.11	1.27	3.96	49.12	1.26	3.27	36.27	2.45
3rd best	1.21	1.31	3.91	50.66	1.22	3.30	37.54	2.45
4th best	1.14	1.30	3.92	51.20	1.29	3.32	38.50	2.45
5th best	1.18	1.25	3.99	50.34	1.22	3.26	37.44	2.46

Table 3.8: Five best ROVER combinations - WER for different conditions.

Note that WER of the best ROVER system, which is 2.44%, corresponds to 15.9% relative WER improvement with respect to the best individual system POW (2.90%) and to 5.8% relative WER improvement with respect to ROVER combining all nine systems (2.59%).

Table 3.8 shows WER of combined systems listed in table 3.7 for individual noisy conditions. This table can be compared with table 3.2 showing WERs for individual systems and for ROVER combining all nine systems. For all five listed combined systems, the highest improvement is observed for seen noises SNR 20dB and 10dB (8/9 of data with respect to which we were searching for the optimal system combination). We also observe good generalization for unseen noises for the same SNR levels.

3.5 Complementarity measures for set of systems

In the previous section, we have shown some connection between complementarity of recognition systems, their suitability for system combination and LBWER and DWER measures. Values from *LBWER and DWER matrices* were used to make a decision which systems from a given set are complementary to others and which are redundant for system combination. However, it would be practical to have a measure assigning a single value to a system set, that would say how the systems in the set are good for combination. In the ideal case, this measure would allow to select a subset of a large set of systems whose combination would lead to the lowest WER.

Several complementarity measures for a set of systems are proposed in this section and the correlation between proposed measures and the actual WER of combined system is shown in experiments.

3.5.1 Average Lower Bound Word Error Rate (ALBWER)

In section 3.4.4, we have expressed the assumption that the smaller values out of the diagonal (and perhaps also on the diagonal) of *LBWER matrix* the better such system set should be for combination. In the previous section, average of LBWER matrix column was used as a measure of one system's complementarity with all other systems in the given set. As a natural extension, we propose to simply average **all** values from *LBWER matrix* to obtain measure of overall complementarity among systems in a set. The averaging is given by equation:

$$ALBWER(S) = \frac{\sum_{i \in S} \sum_{j \in S} LBWER(i,j)}{|S|^2}, \tag{3.3}$$

where S is a set of systems and $|S|$ denotes number of systems in this set. In this definition, WERs of individual systems (values on the diagonal) are also included in the average. Alternative definition excluding individual WERs from averaging can be expressed by the following equation:

$$ALBWER'(S) = \frac{\sum_{i \in S} \sum_{j \in S, j \neq i} LBWER(i,j)}{|S|^2 - |S|}. \tag{3.4}$$

Note that both measures ALBWER and ALBWER' become similar for higher number of elements (systems) in the set as the ratio between number of values in matrix diagonal and values out of diagonal becomes smaller.

3.5.2 Average Dependent Word Error Rate (ADWER)

This measure is defined in the same manner as ALBWER measure. The only difference is that values from the *DWER matrix* are averaged instead of *LBWER matrix* according to the following equation:

$$ADWER(S) = \frac{\sum_{i \in S} \sum_{j \in S} DWER(i,j)}{|S|^2}. \tag{3.5}$$

Again, alternative measure ADWER', where individual WERs are excluded from averaging, is given by equation:

$$ADWER'(S) = \frac{\sum_{i \in S} \sum_{j \in S, j \neq i} DWER(i,j)}{|S|^2 - |S|}. \tag{3.6}$$

3.5.3 Average Sum of LBWER and DWER (ALBWERDWER)

This measure is a combination of the previous two measures given by equation:

$$ALBWERDWER(S) = \frac{\sum_{i \in S} \sum_{j \in S} LBWER(i,j) + DWER(i,j)}{|S|^2}. \tag{3.7}$$

Every sum of LBWER and DWER in averaging can be regarded as a measure of error dependency similar to the LBWER where *dependent errors* are, however, counted twice. This is in agreement with assumption that *dependent errors* are worse than *simultaneous errors* (see section 3.4.4). Again, alternative measure ALBWERDWER' excluding individual WERs from averaging can be defined in the same manner as measure ALBWER' (equation 3.4).

3.5.4 Geometric Average of Lower Bound Word Error Rate (GLBWER)

This measure is similar to ALBWER, however, geometric average is used instead of arithmetic average. The measure is defined by the following equation:

$$GLBWER(S) = \prod_{i \in S} \prod_{j \in S} LBWER(i,j)^{\frac{1}{|S|^2}}. \tag{3.8}$$

Note that $\frac{LBWER(i,j)}{100}$ can be interpreted as a probability of *simultaneous error* made by systems i and j. Under the assumption that these probabilities are independent for each different pair of systems i and j, GLBWER measure is related to probability that all systems make *simultaneous error* at the same time.

If two particular systems in a system set make no *simultaneous error*, GLBWER measure for the set will equal to zero. This however does not imply zero WER for combined systems (at least for ROVER combination).

Measures GLBWER', GDWER, GDWER', etc. can be defined in the obvious way. In our experiments, we will show that measures based on the geometric average do not differ significantly from those based on arithmetic average for real data.

3.5.5 Experimental setup

In the experiments with system set complementarity measures, the same data that was described in section 3.4.5 is used for training and testing. *Seen conditions test data* is used for estimation of *LBWER* and *DWER matrices*. All individual systems again differ only in the feature extraction part. Otherwise each system follows the description given in section 3.4.5. Two different sets — each consisting of eleven individual systems — are used in these experiments to investigate the generalization of proposed complementarity measures.

Systems from the first set will be referred to as *systems with different features*. These systems are identical to those described in section 3.4.5. In addition, two systems using following new features, which are again derived from BSL features, are included to the system set:

- **W15** - 15 ms window is used to compute spectrum of each frame instead of 25 ms window. These features allow for better resolution in time in comparison with BSL features at price of more noisy spectral estimates.
- **W35** - 35 ms window is used to compute spectrum of each frame instead of 25 ms window. Here, feature vector of each frame represents longer time period, however we do not gain better resolution in the spectrum (as could be expected), since the spectrum of each frame is smoothed by the same 23 band Mel filter bank that is used also for BSL features.

Graphical representation of *LBWER matrix* for the set of *systems with different features* is shown in figure 3.5. Relatively low LBWER values in column corresponding to system W15 indicates good complementarity of this newly added system with other systems in the set.

Systems from the second set will be referred to as *systems with missing bands MFCC*. The features used by these systems are similar to BSL features where, however,

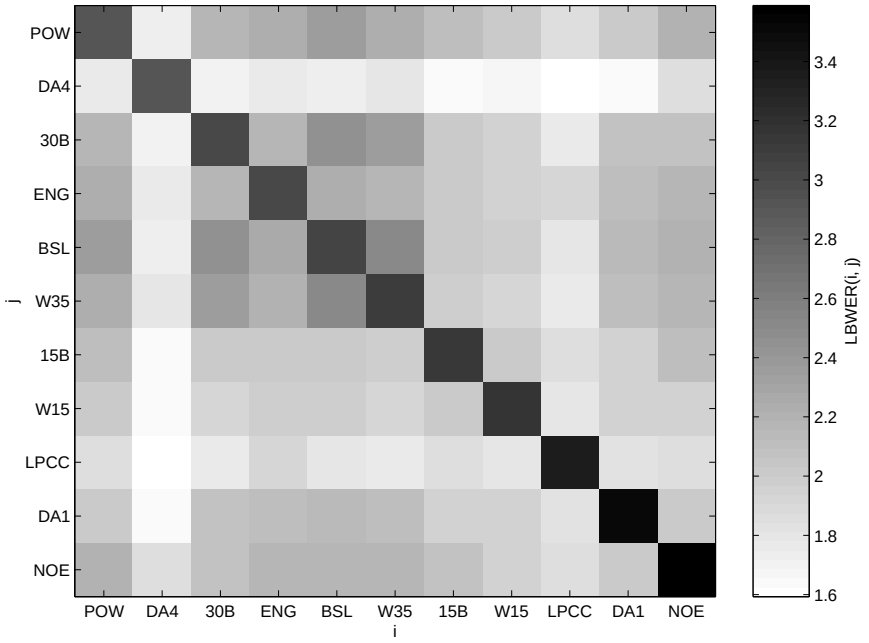


Figure 3.5: LBWER matrix for *systems with different features*.

log energies of certain bands of Mel filter bank are ignored (always three consecutive bands). Instead of DCT, PCA derived from training data is applied to decorrelate output of preserved filter bank bands. The features for individual systems differ only in selection of bands that are ignored. Eleven such systems are employed in our experiments:

- **M1-3** - 1st, 2nd and 3rd band of Mel filter bank is ignored
- **M3-5** - 3rd, 4th and 5th band of Mel filter bank is ignored
- **M5-7** - 5th, 6th and 7th band of Mel filter bank is ignored

  ⋮

- **M21-23** - 21st, 22nd and 23rd band of Mel filter bank is ignored

Graphical representation of *LBWER matrix* for the set of *systems with missing bands* is shown in figure 3.6. WERs of individual systems are presented in table 3.9.

System	**M1-3**	**M3-5**	**M5-7**	**M7-9**	**M9-11**	**M11-13**
WER [%]	3.59	4.41	3.92	3.93	3.40	3.30
System	**M13-15**	**M15-17**	**M17-19**	**M19-21**	**M21-23**	
WER [%]	3.09	3.25	3.23	3.71	3.59	

Table 3.9: WER of individual *systems with missing bands MFCC*.
tab:CM:MBSystemWERstab:CM:ED-DF

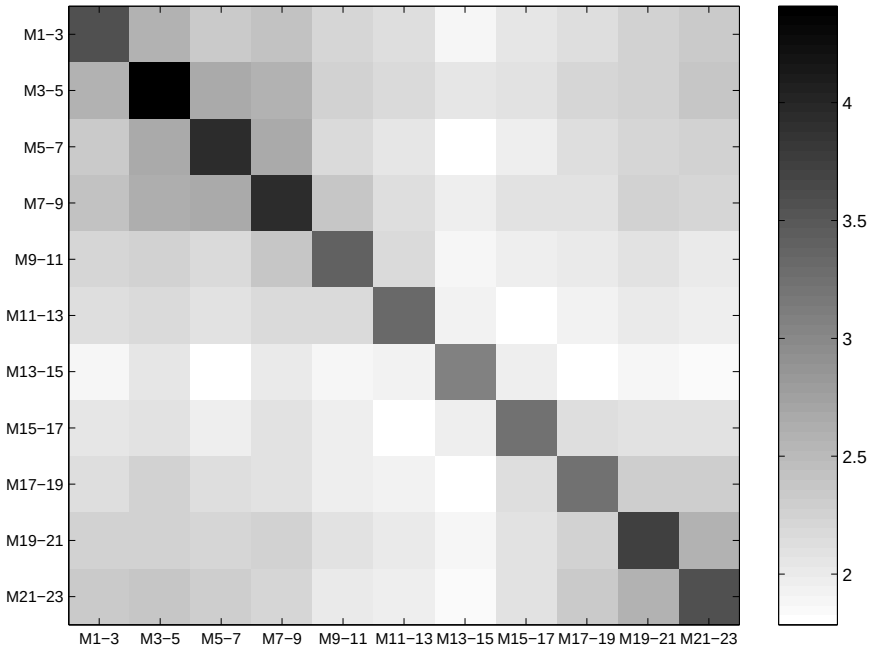


Figure 3.6: LBWER matrix for *systems with missing bands MFCC*.

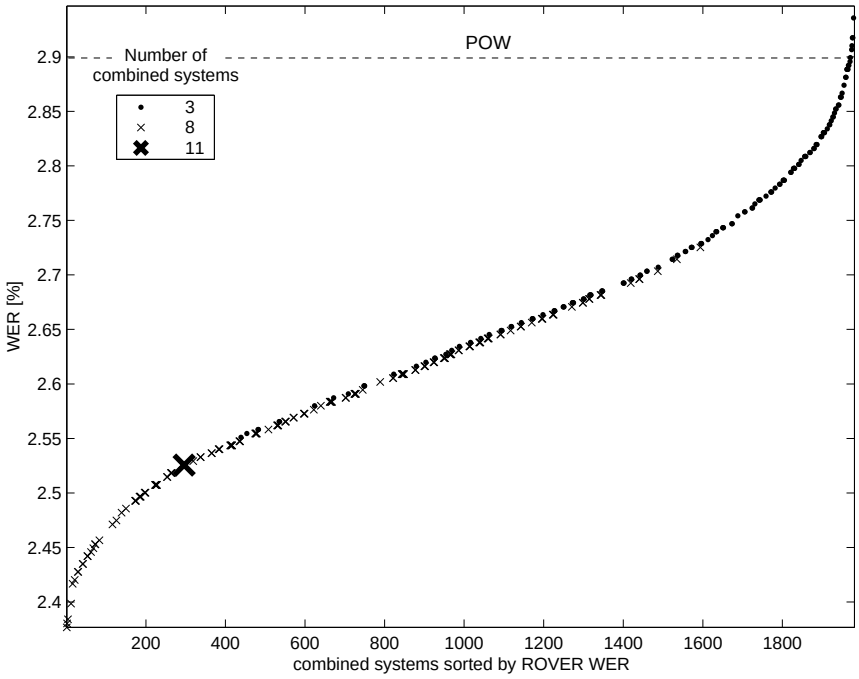
3.5.6 Correlation between combined system WER and system set complementarity measures

The following experiments were carried out to investigate a correlation between the proposed system set complementarity measures and the corresponding combined system WERs. All subsets of *systems with different features* consisting of three and eight systems were combined using ROVER and corresponding WERs were evaluated. Similarly, all subsets of three and eight systems were combined for the set of *systems with missing bands MFCC*. The combination of three and eight systems was chosen to show how complementarity measures are correlated with combined system WER for combinations of only few (three) and larger number (eight) of systems.

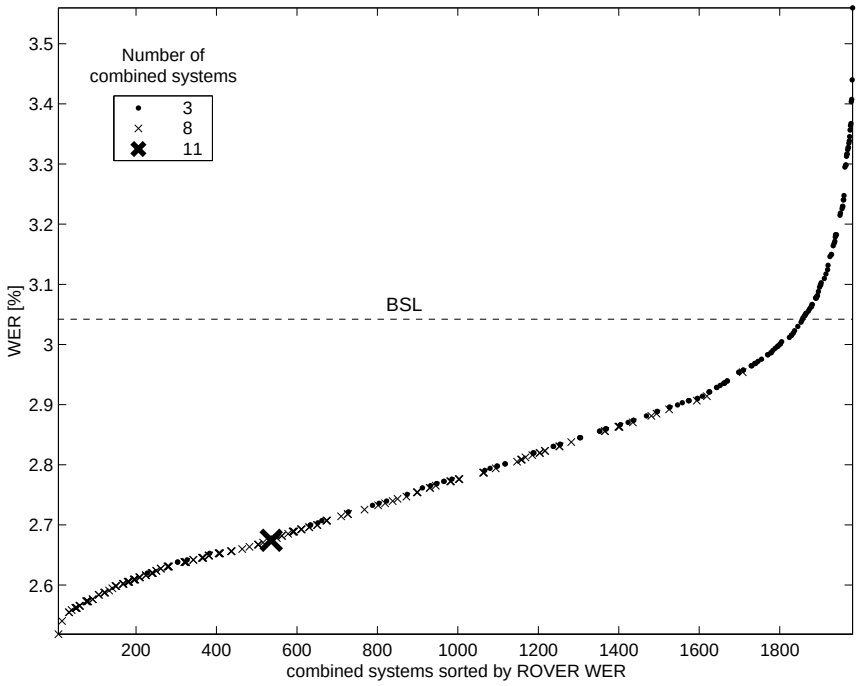
Figure 3.7a shows WERs of combined systems where the subsets of *systems with different features* are combined. Each dot corresponds to one combination of three systems and each cross corresponds to one combination of eight systems. The big bold cross represents the combination of all eleven systems. Axis Y represents WER of combined system. Combined systems are ordered according to their WER on X axis. As can be seen in figure 3.7a, in average, combinations of eight systems perform better than combinations of three systems, however, the best combinations of three systems (the dots most on the left) performs much better than the worst combinations of eight systems (crosses most on the right). There are few combinations of three systems performing worse than the best individual system POW with WER of 2.90% and in opposite, the best such combination performs almost as well as the combination of all eleven systems. Of course, the most interesting part of the figure is on the left of the big bold cross, where systems are outperforming the combination of all eleven systems. The best combined system in the figure with WER of 2.38% is one of the combinations of eight systems.

Similar figure 3.7b shows WERs of combined systems where the subsets of *systems with missing bands MFCC* are combined. When combining these systems, the goal is to outperform BSL system with WER of 3.04%, which uses the information from all bands. As can be seen in the figure, combination of all eleven systems with WER of 2.67% reaches the goal. Many combinations of three systems perform worse than BSL system, on the other hand, there are combinations of three systems outperforming even the combination of all eleven systems. The best combined system in the figure with WER of 2.51% is one of the combinations of eight systems.

Figure 3.8 shows the correlation between WER of combined system (X axis) and average of WERs of corresponding individual systems (axis Y). Again, each dot, cross



a) Different features



b) Missing band MFCC

Figure 3.7: ROVER WER for combinations of 3, 8 and all 11 systems.

System set	Different features		Missing bands MFCC	
# combined sys.	3	8	3	8
Measure:				
Avg WER	0.18	-0.28	0.88	0.73
ALBWER	0.86	0.77	0.96	0.88
ALBWER'	0.82	0.80	0.95	0.89
ADWER	0.86	0.82	0.95	0.90
ADWER'	0.79	0.84	0.86	0.91
GLBWER	0.85	0.77	0.96	0.90

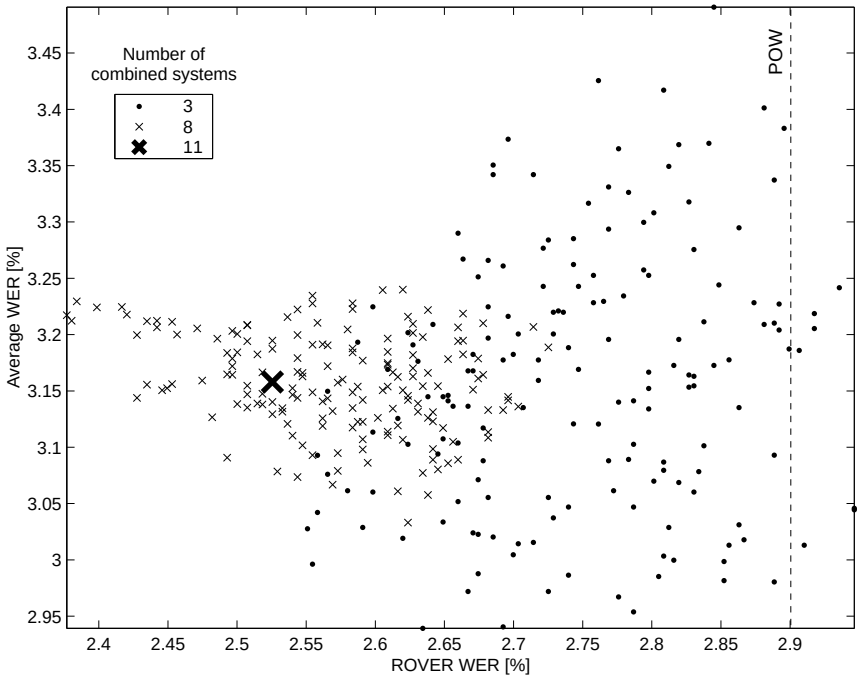
Table 3.10: Correlation coefficients representing correlations between individual complementarity measures and ROVER WER.

and big cross in the figure corresponds to one combination of three, eight and eleven systems, respectively. Figure 3.8a shows that for *systems with different features* no significant correlation can be observed. Therefore, we can conclude that for this system set, WERs of individual systems are not important for selection of systems suitable for combination. In figure 3.8b, for *systems with missing bands MFCC*, some correlation between combined system WER and average WER of individual systems can be observed.[6] As can be seen in figure 3.6, for this system set, systems with lower WER were generally more suitable for combination, however, it does not mean that average WER of individual systems is a good measure of system complementarity. We will see that the proposed complementarity measures are much more correlated with corresponding WER of combined system.

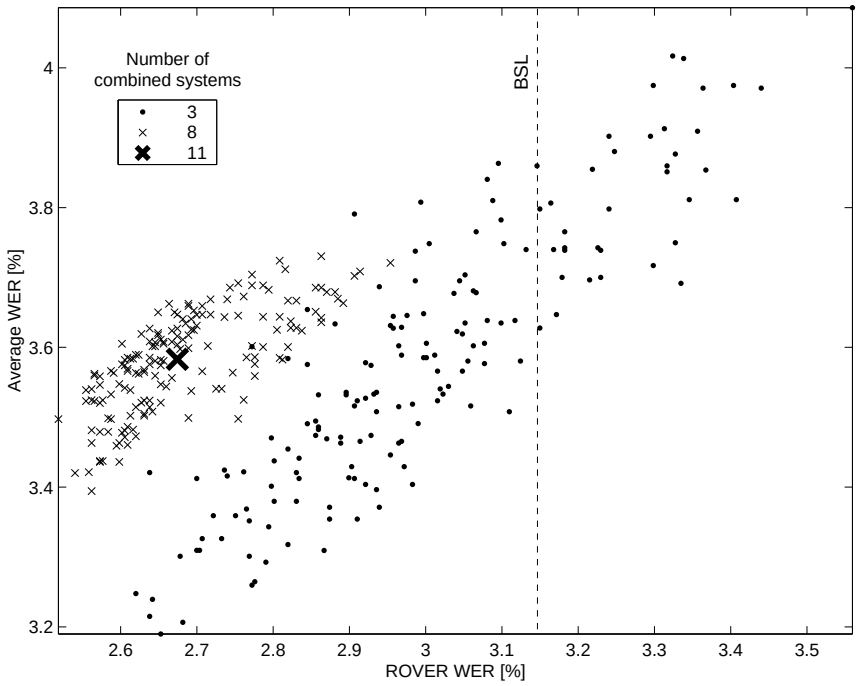
To allow for objective comparison of performances of individual complementarity measures, correlation coefficients can be found in table 3.10. These coefficients represent correlations between individual complementarity measures and ROVER WER. For example, the correlation coefficient estimated from data points from figure 3.8 (correlation between average WER and ROVER WER) can be found in the first line of the table. Note, that a coefficient equal to 0 reflects no correlation, a coefficient equal to 1 reflects absolute correlation (all data points lay in one line). A negative correlation is also possible.

In the following experiments, we will examine the correlation between proposed system set complementarity measures and corresponding combined system WER. We

[6]Note that we must look at combinations of three systems and eight systems separately.



a) Different features



b) Missing band MFCC

Figure 3.8: Correlation between average WER and ROVER WER.

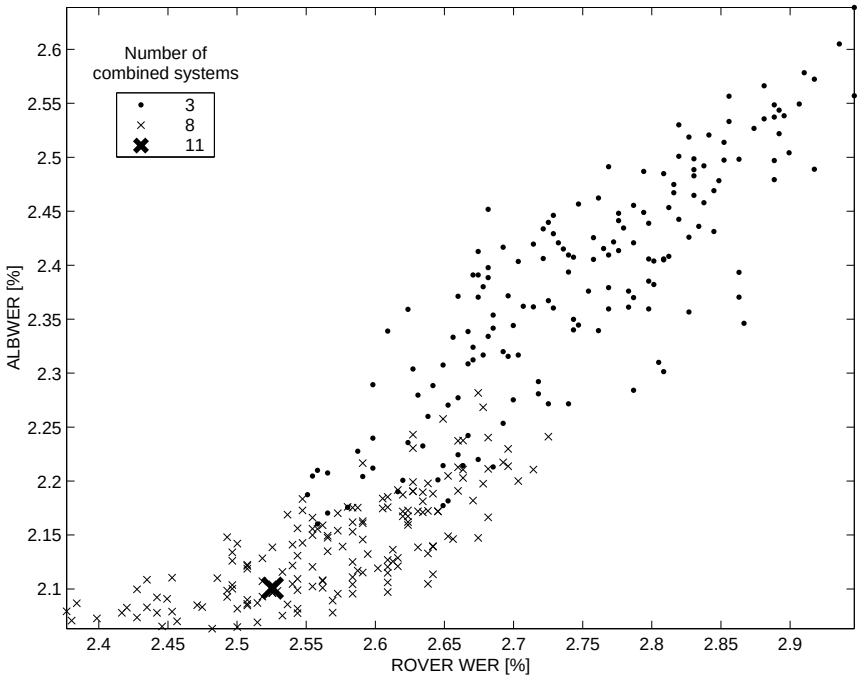
will compare different measures and make conclusions about their performances for combinations of small number and larger number of systems. Properties of the measures are again demonstrated on combinations of three and eight systems from the set of *systems with different features* and the set of *systems with missing bands MFCC*. Presented results of these experiments may not seem to be sufficient to make some of the following conclusions, however, trends similar to those presented here were observed also for different numbers of combined systems and for different sets of systems.

Figure 3.9 shows the correlation between combined system WER and corresponding *Average Lower Bound Word Error Rate (ALBWER)* measure computed according to equation 3.3. For both system sets and for combinations of three and eight systems, visible correlation is observed between ALBWER measure and combined system WER. For *systems with missing band MFCC*, much higher correlation is observed in comparison to figure 3.8b.

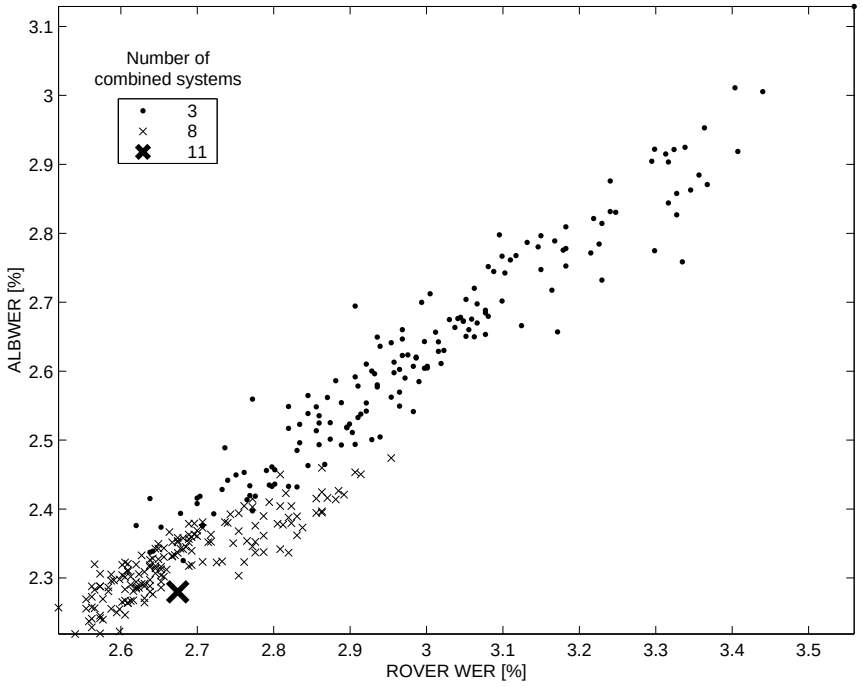
Figure 3.10 shows the correlation between combined system WER and ALBWER' measure (alternative definition of ALBWER measure excluding WERs of individual systems from averaging), which is computed according to equation 3.4. In comparison to ALBWER, this measure is less correlated with combined system WER for combinations of three systems (the dots are more spread around the line on which they would ideally lay). This could be, however, specific only to ROVER combination with majority voting used in our experiments, where voting based on decision of only few systems can be unreliable and actual WER of individual systems can by more important. In opposite, comparing figures 3.9a and 3.10a, WER of combinations of eight systems seems to be more correlated with ALBWER' measure than with ALBWER measure.

It can be seen in figure 3.10 that dots representing combinations of three systems and crosses representing combinations of eight systems are concentrated around two separate lines. Therefore, values of ALBWER' measure can not be compared for two sets with different number of systems. In other words, first, we must know how many systems we want to combine and then we can use ALBWER' measure to choose which systems will be good for combination. The same rule applies for all other proposed complementarity measures.

Figures 3.11 and 3.12 show complementarity measures based on averaging of values of *DWER matrix* according to equations 3.5 and 3.6, respectively. Again, we observed that ADWER measure is more correlated with three systems combination WERs than ADWER' and, in opposite, eight systems combination is more correlated

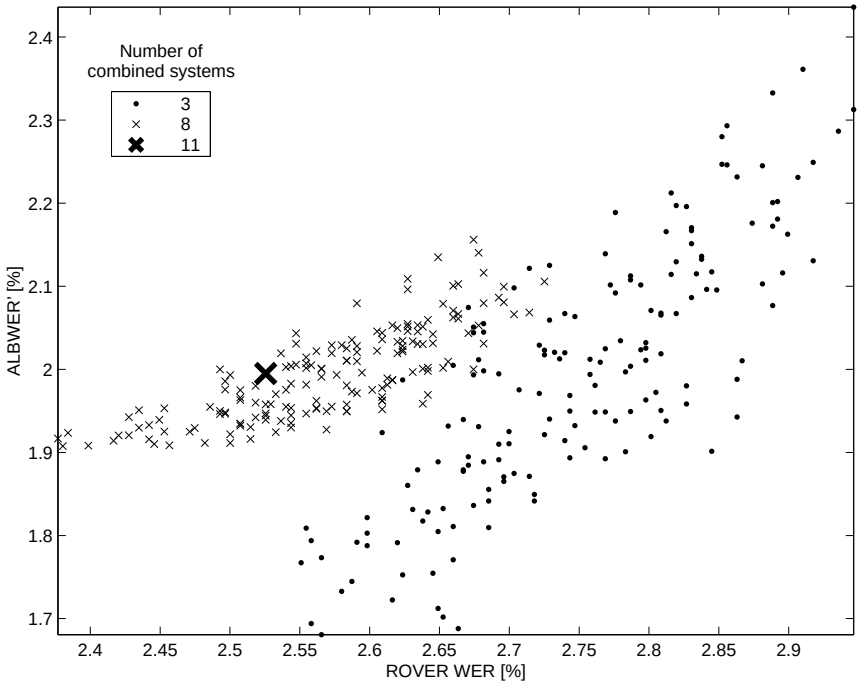


a) Different features

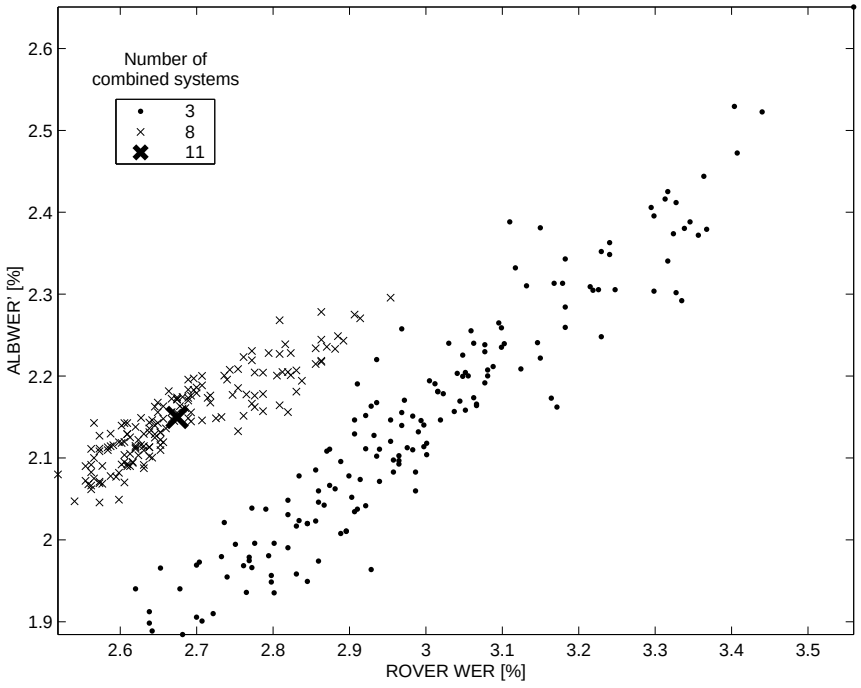


b) Missing band MFCC

Figure 3.9: Correlation between ALBWER and ROVER WER.



a) Different features



b) Missing band MFCC

Figure 3.10: Correlation between ALBWER' and ROVER WER.

with ADWER' measure. An interesting finding is that for higher number of combined systems, measures ADWER and ADWER' show higher correlation with WER of combined system than measures ALBWER and ALBWER'.

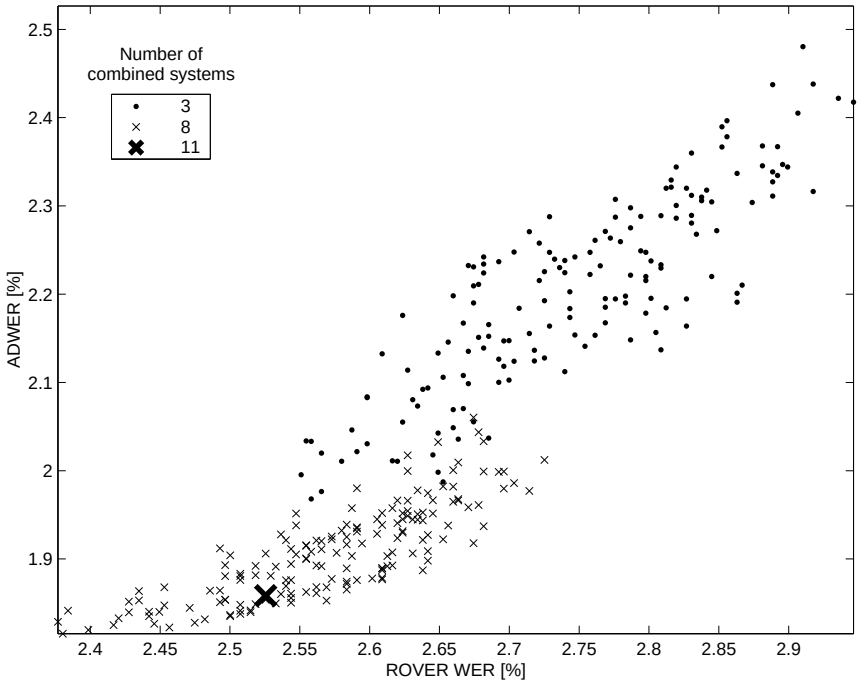
In section 3.5.3, we proposed measure ALBWERDWER averaging sums of corresponding values from *LBWER and DWER matrix*. However, experiments with this measure did not show any particular advantage. The results obtained for this measure look simply as a compromise between ALBWER and ADWER' measure.

In section 3.5.4, measures based on geometric average of values from *LBWER and DWER matrix* were proposed. Figure 3.13 demonstrate results obtained in experiment with GLBWER measure given by equation 3.8. Again, no particular advantage of using geometric average was observed. The results obtained for these measures were almost identical to those obtained for corresponding measures based on arithmetic average, especially for higher number of combined systems[7].

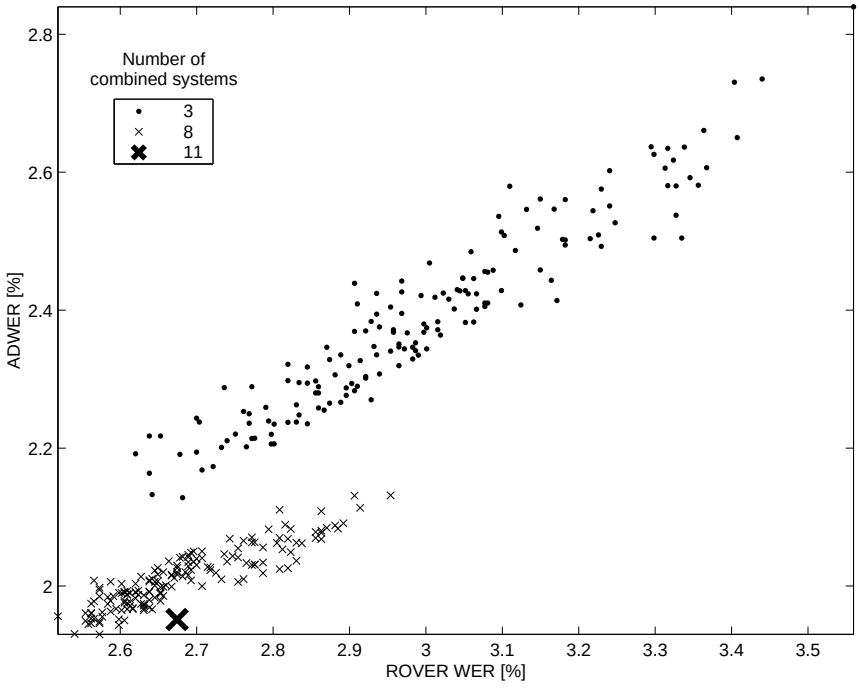
3.6 Discussion and conclusions

Combination of different systems can be a powerful technique to improve recognition performance. The success of these techniques is, however, contingent on complementarity of combined systems. Given a set of N systems, one way to determine the subset of systems most suitable for combination is to exhaustively evaluate recognition performance for all possible system combinations. In the case of ROVER-like combination of output sequences, training and recognition must be performed only once for each of N systems. Then, however, ROVER-like technique must be applied for each combination of N systems, which may be not feasible for large values of N. From this point of view, combination at the feature level is even worse case. Here, also training and recognition must be performed for each combination of N systems, which increases the whole evaluation time in order of magnitudes. For this reason, we have proposed the measures of complementarity of recognition systems, which are based on measurement of error dependency of individual system outputs. First, methods for measuring complementarity of two systems were proposed. These measures can be computed very efficiently even for large set of systems. Training and recognition must be performed only once for each of N systems, then a technique similar to ROVER is used to measure complementarity only for each pair of systems. Simple averaging of these measures is used as an extension allowing to measure the complementarity of

[7]Comparing figures 3.13 and 3.9, relative positions of crosses are almost identical.

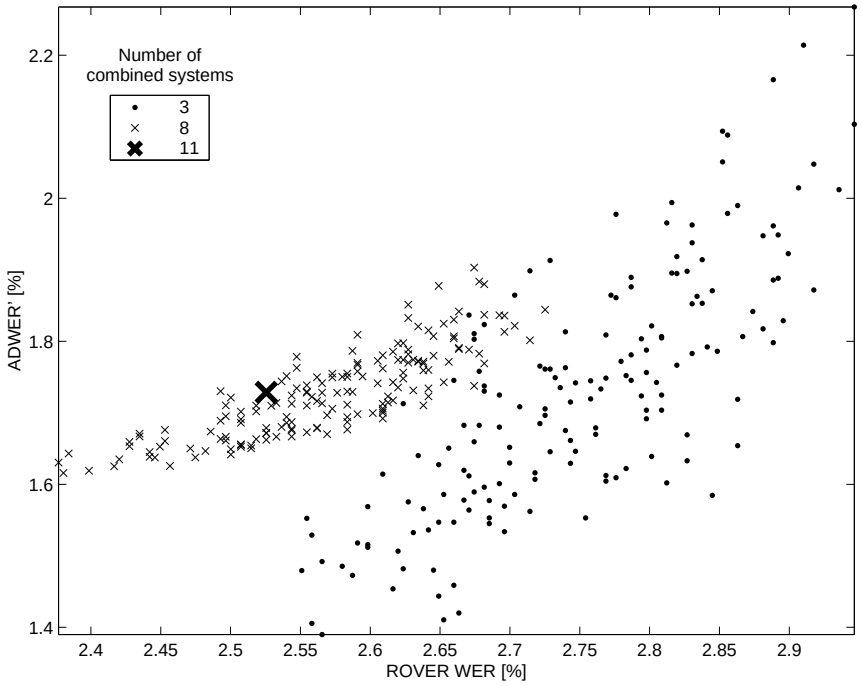


a) Different features

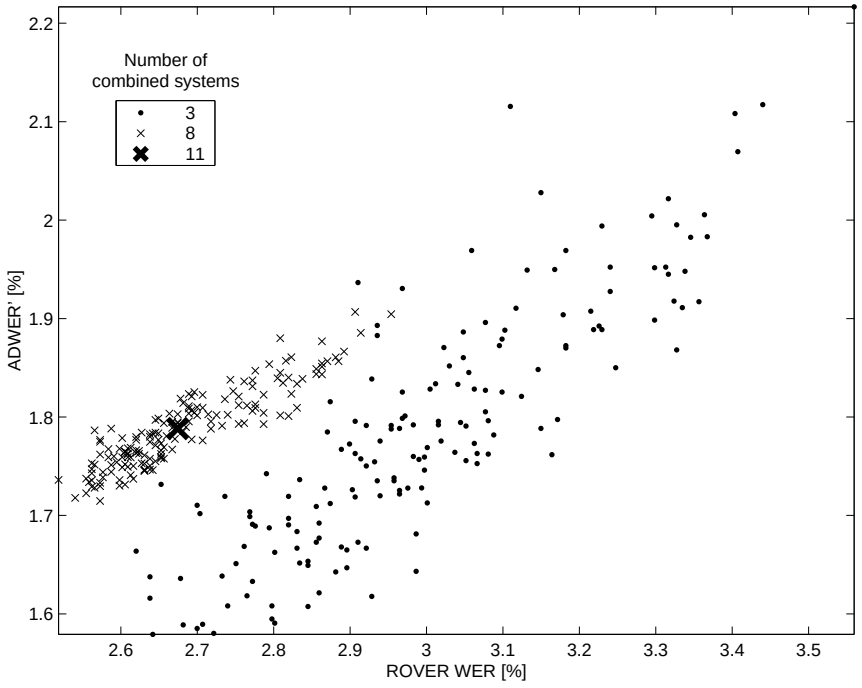


b) Missing band MFCC

Figure 3.11: Correlation between ADWER and ROVER WER.

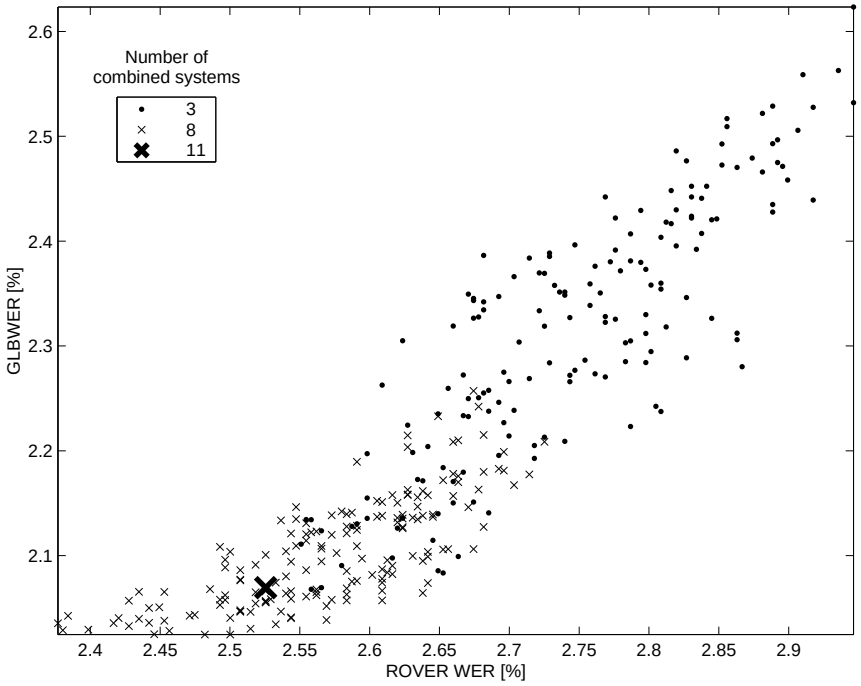


a) Different features

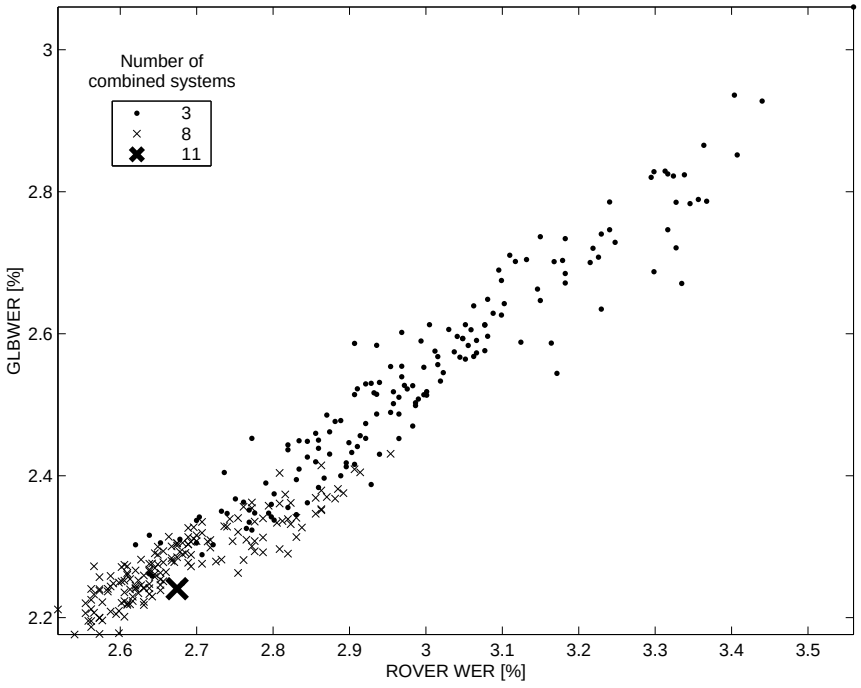


b) Missing band MFCC

Figure 3.12: Correlation between ADWER' and ROVER WER.



a) Different features



b) Missing band MFCC

Figure 3.13: Correlation between GLBWER and ROVER WER.

a system subset. The correlation between these measures and actual performances of combined systems that was shown in experiments indicates that these measures can be advantageously used to select systems suitable for combination.

In one set of our experiments, ROVER was used to combine subsets of eleven *systems with different features*. All possible combinations of three and eight systems were tested. WERs of 2.55% and 2.38% were obtained for the best combinations of three and eight systems, respectively. When one of the proposed measures was used to select complementary systems, usually the performance of the first or the second selected system combination was very close to the best one. Measures ignoring ordinary WERs of individual systems (ALBDER' and ADWER') were observed to perform better where a higher number of systems (eight system) were combined, whereas measures ALBDER and ADWER performed better for combinations of three systems (see section 3.5.6). Furthermore, for combinations of eight systems, complementarity measures based on DWER were observed to be more accurate than those based on LBWER. Therefore ADWER' measure promises to be the right measure, when a higher number of system is combined.

The proposed complementarity measures and the techniques that these measures are based on are not useful only for automatic selection of systems which should be combined. They can be also used as an analysis tool allowing to find where the complementarity of different systems is coming from and to identify complementary approaches in speech recognition. For example, in our experiments, *systems with different features* differ only in the feature extraction method. We can therefore expect that most of the complementarity seen in the outputs of these systems reflects the complementarity encoded in the different features. System DA4, which differs from others in computation of derivatives, was found to be the system most complementary to all other systems. In fact, the usage of different derivatives corresponds to a usage of more precise information about acoustic context. Therefore, extended information about the context turns out to be very important. This also justifies research works trying to utilize such information for feature extraction [34, 41, 18] or acoustic modeling [44, 51, 7]. Advantage of features using extended information about acoustic context is also demonstrated in the next chapter, which is dealing with feature combination, where successful combinations of DA4 features with any other features are presented.

Knowing that two systems are complementary, an additional analysis can be also carried out by means of a technique similar to that used for LBWER measure. We

can not only count how many times systems are "able to correct" each other, but we can also collect statistics saying under which condition (e.g. clean or noisy speech) and in which situation (e.g. for which word, phoneme) one system is "able to correct" the second system. This can be an interesting input especially for developing more sophisticated combination methods.

Although *systems with different features* differ only in feature extraction, **not all** the complementarity seen in the system outputs reflects the complementarity encoded in the different features. For example, NOE features do not contain any information complementary to BSL features. In fact, NOE system equals to BSL system with the exception that it does not use information about frame energy. Although NOE system uses less information than BSL system and its performance (WER of 3.59%) is significantly worse than performance of BSL system (WER of 3.04%), still, it is able to provide correct symbol on the output in almost one third of situations where BSL system fails (as can be seen in table 3.3 — compare BSL-NOE 2.18% LBWER with BSL 3.04% WER). The explanation is the following: If test data is corrupted the way not seen during the training, hiding the information about a corrupted feature dimension (which often is the energy) can be helpful for the classification process. Similar analysis of complementarity can therefore provide a valuable input for improving acoustic modeling and classification process, since in ideal case the classification should never fail only because an additional information becomes available.

Another example, where complementarity is "generated" by hiding different pieces of information are *systems with missing bands*. Here, each system does not use an information about some particular part of smoothed spectrum represented by filter bank energies. WERs of the individual systems range from 3.09% to 4.42% (see table 3.9). None of these systems performs better than BSL (WER of 3.01%) "seeing" the information from all bands. BSL system is however clearly outperformed by many ROVER combinations of *systems with missing bands*. WER of the best combination of eight system is 2.51%.

Chapter 4

Feature level system combination

4.1 Introduction

In the previous chapter, the method of system complementarity measurement was introduced, which can be useful for the selection of systems suitable for combination. Correctness of the proposed measures was demonstrated in experiments where correlation between the measure and the actual WER of a combined system was shown. In all these experiments, output symbol sequences of individual systems were combined using technique known as ROVER. Although proposed measurement of complementarity is also based on comparing output symbol sequences, the assumption was made that the measures should be useful also for selections of complementary systems, even if the combination of these systems is performed at different level (e.g. feature level combination).

In the following experiments, pairs of different feature streams, which were described in the previous section, are combined at the feature level and recognition systems using such combined features are evaluated. Performances of systems using different feature combination techniques are compared and the correlation between their actual WERs and proposed complementarity measures is presented.

There are two main aims of the work presented in this chapter. First, we want to examine whether the proposed complementarity measures are applicable even in the case where the point of combination is moved from the very end of the recognition chain (output word sequences) to the very beginning (input feature streams). Second, several techniques used for feature combination (such as PCA, LDA, HLDA) are compared in experiments where limited amount of training data is available. Success with the feature combination can be quite dependent on proper estimation of statistics re-

quired by the used technique. Insufficiency of training data is, therefore, an important problem, which has to be taken into account in our experiments. Besides some standard approaches increasing robustness of statistic estimation (e.g. PCA stabilization), methods based on combination of LDA and HLDA will be proposed.

In the next sections, the main concept of combination of feature streams is introduced, different combination techniques are described and problems related to robust estimation of feature statistics are discussed. Experimental setup is described in section 4.6.1. In the rest of the chapter, experiments with different feature combination techniques are described and advantages and drawbacks of individual techniques are discussed. The correlation between the proposed complementarity measures and the performances of systems using combined features is also presented.

4.2 Combination of feature streams

Assume two or more feature streams representing the same speech utterance, where the information about the speech is at some level complementary in different streams. Then the main idea of feature combination is to produce single output stream, which contains all the information encoded in the original streams that is important for the correct recognition of the utterance. Assuming that all feature streams have the same number of feature vectors (frames) and all the streams are synchronized in time (feature vectors at corresponding position in streams represent the same part of speech), the simplest way to perform the combination is to concatenate corresponding feature vectors. Resulting feature stream may not be, however, suitable for following classification process, which usually requires feature vectors of reasonable dimensionality having individual coefficients decorrelated. Simple concatenation can result in highly dimensional features containing redundant information coming from different input feature streams. Coefficients in the concatenated feature vector can be correlated even in the case where individual input features were already decorrelated. The feature concatenation is, therefore, only the first step of combination techniques presented here. These techniques then differ in postprocessing performed in order to decorrelate concatenated feature vectors and to reduce their dimensionality by removing coefficients with redundant and unimportant information. The following sections deal with these postprocessing methods and related problems.

4.3 Postprocessing using PCA

Principal Component Analysis [21, 16] (PCA) was already presented in section 2.5.1. PCA is a technique allowing to derive such linear transformation from a set of feature vectors that ensures decorrelation of their coefficients. Additionally, PCA provides the information about importance of individual coefficients (dimensions) of the decorrelated feature vectors. The importance is given by the variance of individual coefficients. This information can be used to perform dimensionality reduction by discarding less important coefficients. PCA can be therefore used for both decorrelation and reduction of dimensionality of concatenated feature vectors. However, success with PCA used for this purpose can be limited because of the following assumptions and constraints:

- PCA assumes that input feature vectors obey the (multivariate) Gaussian distribution.
- PCA transformation ensures the decorrelation only of the overall set of feature vectors on which the transformation is derived. In other words, only the global covariance matrix of the transformed (rotated) features is ensured to be diagonal. However, for classification process, it is usually desirable that features representing each particular class (e.g. one HMM state) are decorrelated (also class covariance matrices should be diagonal).
- Quality of the transformation can be quite sensitive to the selection of data used for its estimation. For example, there are usually many feature vectors representing non-speech parts of utterances in the available data. If all feature vectors are used, the resulting transformation will be mainly given by variances seen in these non-speech parts. Moreover, for many commonly used features, the assumptions of Gaussian distribution does not hold especially for non-speech parts.
- As was already mentioned, dimensionality reduction can be performed by omitting decorrelated feature vectors coefficients having low variances. Correctness of this approach is therefore contingent on the assumption that variance in the data is directly related to the amount of information important for recognition of speech.

The problem related to the assumption mentioned in the last point is particulary important in our case where vectors of different feature streams are concatenated.

Typically, PCA is used to decorrelate log energies on the output of Mel filer bank, where low-variance dimensions of decorrelated vectors will carry information about only minor changes in these log energies. Corresponding changes in the speech would not be probably even perceived by humans and therefore, they are of minor importance for speech recognition. PCA is appropriate for processing filter bank log energies because quantities represented by individual coefficients of original feature vectors are directly comparable.

Individual coefficients may not be, however, directly comparable in the case, where features are created by concatenation of feature vectors from two or more different feature streams. For example, consider that feature vectors of two streams having a lot of complementary information important for recognition are concatenated. All coefficients in one stream are scaled down into much smaller range than coefficients in the second stream. The coefficients from the first stream have therefore much smaller variances and information encoded in these coefficients will be therefore considered as less important. A possible solution for this problem is to properly scale coefficients from different feature streams to give them comparable importance before deriving PCA transformation. Scaling factors can be obtained by one of the following methods:

- For each feature vector coefficient, a different scaling factor is obtained by taking inverse of coefficient's standard deviation. Scaled coefficients are given by the following equation:

$$\hat{c}_{is}(t) = \frac{c_{is}(t)}{\sigma_{is}}, \tag{4.1}$$

  where c_{is} is i^{th} coefficient of s^{th} stream in t^{th} frame, σ_{is} is standard deviation of i^{th} coefficient of s^{th} stream.

  According to this equation, all coefficients are scaled to have the variance equal to one no mater to which original feature stream the individual coefficient belongs. However, giving the same importance to all coefficients may not be desirable in cases where variances are related to the importance of coefficients in the scope of each original feature stream. For example, if MFCC stream and LPCC stream are merged, in both streams cepstral coefficients with higher indices have usually lower variance, which indicates their smaller importance.

- The alternative method of feature coefficient scaling preserves the relative variances in the scope of each original feature stream. Here, all coefficients belonging

to one stream are scaled by the same factor according to the following equation:

$$\bar{c}_{is}(t) = \frac{c_{is}(t)}{\sqrt{\sum_{j=1}^{I_s} |\lambda_{sj}|}}, \tag{4.2}$$

where I_s is number of feature vector coefficients of s^{th} stream and λ_{sj} is j^{th} eigen value of s^{th} stream covariance matrix. The sum of the eigen values here represents the overall variance in a particular stream.

4.4 Postprocessing using LDA and HLDA

Alternatively to PCA, Linear Discriminant Analysis (LDA) [21, 16, 40] or Heteroscedastic Linear Discriminant Analysis (HLDA) [48][1] can be used to decorrelate concatenated feature vectors and to perform the dimensionality reduction. LDA and HLDA were already presented in sections 2.5.2 and 2.5.3. For LDA or HLDA, each feature vector representing a speech frame that is used to derive the transformation must be assigned to a class. When performing the dimensionality reduction, both LDA and HLDA allow to preserve such dimensions, in which feature vectors representing individual classes can be best separated. Because the importance of a dimension is given by separability of classes and not by variance of data in this dimension (as it was in the case of PCA), there is no need to perform any scaling of original feature vector coefficients before applying LDA or HLDA. Another advantage of LDA and HLDA against PCA is that LDA and HLDA transformations decorrelate also feature vectors forming each particular class [2], which makes features more suitable for following classification process.

Disadvantage of LDA is its assumption that feature vectors representing classes obey (multivariate) Gaussian distribution and that all these distributions share the same covariance matrix. Statistics necessary for deriving LDA transformations are two covariance matrices (across-class and within-class, see section 2.5.2) with dimensionality given by concatenated feature vector.

[1] Particular definition HLDA depends on the model used for modeling distributions of classes in rotated space (see section 2.5.3). In this text, only the special case of HLDA is considered, where classes are modeled using Gaussian distribution with diagonal covariance matrix.

[2] In fact, it may not be possible to fully decorrelate all classes using single linear transformation. Instead, HLDA finds such transformation that ensures maximum likelihood modeling of classes by Gaussian distributions with diagonal covariance matrices. LDA decorrelates classes if the assumption of the same covariance matrix for all classes holds.

HLDA relaxes the constraint that distributions of all classes share the same covariance matrix. On the other hand, the amount of statistics necessary for deriving HLDA transformation is much higher then in the case of LDA. These statistics are: global covariance matrix and covariance matrix of each class.

Since LDA can be seen as a special case of HLDA with the constraint of shared covariance matrix amount all classes, the abbreviation HLDA will stay for both LDA and HLDA in the rest of this section.

4.4.1 Classes given by labels

The fundamental problem, which has to be solved before using HLDA, is assignment of feature vectors (speech frames) to classes. There are several possibilities. If the speech data used to derive HLDA transformations are phonetically labelled, each class can be formed by feature vectors marked by the same phoneme label. Alternatively, using a well-trained HMM system, HMM state labels can be generated for the speech data by state-level forced alignment algorithm. HMM state labels allows to have very consistent definition of classes for both HLDA (as a part of the feature extraction) and the following HMM based recognition process. For this reason, classes given by state labels are used in most of the following experiments.

4.4.2 Classes given by occupation probabilities

In HMMs, feature distributions associated with individual states are usually modeled by mixtures of Gaussians. To obtain the representation of classes even more consistent with their representation in HMM, classes corresponding to mixture components can be used instead of classes corresponding to HMM states. Such representation of classes also ensures that the distribution of feature vectors corresponding to a particular class will be more Gaussian, which is required by HLDA. In the class assignment strategy described in the previous paragraph, "hard" assignment of feature vectors to classes was given by labels. Here, "soft" assignment is given by $\gamma_{sm}(t)$ (see equation 2.14)[3], which is the probability of occupying the m^{th} mixture component of the state s by t^{th} speech frame of training data. This probability can be computed using standard Forward-backward algorithm, which is also employed in Baum-Welch re-estimation of HMM parameters.

[3]For convenience, utterance index, which appears in equation 2.14, is omitted here, and symbol t is considered to identify both the utterance and the frame of the utterance.

During HMM training, any re-estimation of HMM parameters can result in change of $\gamma_{sm}(t)$. Therefore, HLDA transformation relying on $\gamma_{sm}(t)$ should be re-estimated as well. However, the change of HLDA transformation results in different feature extraction and HMM parameters should be re-estimated again using new features. In contrast to the case of hard classes, where HLDA transformation is derived only once before starting the HMM training, here, an iterative process is used, where HMM parameters and HLDA transformation are alternately estimated in each iteration of model training. One iteration of training then consists of the following steps:

1. Using n-dimensional concatenated feature vectors, $\mathbf{o}^c(t)$, and the current estimate of HLDA projection $p \times n$ matrix, $\mathbf{A}$, new set of feature vectors, $\mathbf{o}(t)$, (with dimensionality reduced to p) is generated according to equation:

$$\mathbf{o}(t) = \mathbf{A}\mathbf{o}^c(t). \tag{4.3}$$

2. Given a current set of HMM parameters, Gaussian mixture component occupation probabilities, $\gamma_{sm}(t)$, are estimated using Forward-backward algorithm.

3. The occupation probabilities, $\gamma_{sm}(t)$, are used to re-estimate transition probabilities and mixture component weights according to standard Baum-Welch algorithm (see formulae 2.15 and 2.11).

4. The occupation probabilities, $\gamma_{sm}(t)$, and concatenated feature vectors are used to estimate n-dimensional mean vector, $\boldsymbol{\mu}^c_{sm}$, and full covariance $n \times n$ matrix, $\boldsymbol{\Sigma}^c_{sm}$, of each Gaussian mixture component m of each state s according to the following equations:

$$\boldsymbol{\mu}^c_{sm} = \frac{\sum_{t=1}^{T} \gamma_{sm}(t)\mathbf{o}^c(t)}{\sum_{t=1}^{T} \gamma_{sm}(t)}, \tag{4.4}$$

$$\boldsymbol{\Sigma}^c_{sm} = \frac{\sum_{t=1}^{T} \gamma_{sm}(t)(\mathbf{o}^c(t) - \boldsymbol{\mu}^c_{sm})(\mathbf{o}^c(t) - \boldsymbol{\mu}^c_{sm})^T}{\sum_{t=1}^{T} \gamma_{sm}(t)}, \tag{4.5}$$

   where T is the number of feature vectors used for training. The superscript c is used to emphasize that statistics are estimated from the concatenated feature vectors, $\mathbf{o}^c(t)$, and not on features, $\mathbf{o}(t)$, which are modeled by HMM.

5. New HLDA projection, $\mathbf{A}$, is derived using the occupation probabilities and the estimated class covariance matrices, $\boldsymbol{\Sigma}^c_{sm}$ (see section 2.5.3).

6. To obtain the correct estimates of HMM parameters for features corresponding to the newly derived transformation, $\mathbf{A}$, p-dimensional mean vector, $\boldsymbol{\mu}_{sm}$, and variance vector, $\boldsymbol{\sigma}^2_{sm}$, of each Gaussian mixture component is updated according to the following equations[4]:

$$\boldsymbol{\mu}_{sm} = \mathbf{A}\boldsymbol{\mu}^c_{sm}, \tag{4.6}$$

$$\boldsymbol{\sigma}^2_{sm} = diag(\mathbf{A}\boldsymbol{\Sigma}^c_{sm}\mathbf{A}^T). \tag{4.7}$$

The initial estimate of transformation matrix $\mathbf{A}$ can be given, for example, by transformation derived using state labels.

4.4.3 HLDA in the Maximum Likelihood framework

It was noticed that HLDA transformation allows to project features from the original space to the subspace where classes are best separated. Here, one could object that for the speech recognition task, we want to discriminate between different words and not between all HMM states or even their Gaussian components, which were chosen to define classes for HLDA. For example, in the following experiments, 16 states HMMs are used to model each English digit. The objection could be that it is probably not important to discriminate between the last states of models of the words *Zero* and *O [ow]*, because both states represent the final part of phoneme *ow* at the end of the word. From this point of view, feature vectors corresponding to these two states should not be treated as two separate classes. In the case of mixture component related classes, the objection could be, for example, that we do not need to discriminate between two different mixture components of the same state.

However, the iterative algorithm presented in the previous section, where mixture component related classes are used to derive HLDA transformation, is absolutely correct from the point of view of the model training under the Maximum Likelihood framework. In fact, HLDA transformation is the transformation allowing maximum likelihood modeling of features in the rotated space using the constrained model described in section 2.5.3. It was proven that presented iterative training is stable and that it ensures to increase (or at worse not to decrease) the likelihood of described constrained model in each iteration [23].

[4]Here, we consider modeling using Gaussian mixture component with diagonal covariance matrices, which is used in our experiments.

Note that in this text, HLDA transformation is considered to be a part of feature extraction. More correctly, in the case of the iterative algorithm from the previous section, HLDA transformation matrix should be rather seen as a part of HMM parameters, which are all re-estimated in the Maximum Likelihood framework.

4.5 Robust estimation of statistics

All the techniques for postprocessing of concatenated feature vectors, that were described in previous sections, rely on statistics (e.g. global or class covariance matrices) estimated from training data. Success with the feature combination is, therefore, quite dependent on the correct estimation of these statistics. In our experiments, however, only limited amount of training data is available, which may not be sufficient to obtain good estimates. The estimation will be problematic especially for HLDA, where an estimate of covariance matrix is required for each individual class. To overcome this problem, methods increasing robustness of estimation are used in our experiments. All the used methods are described in this section.

We will often deal with covariance matrices in the following paragraphs. Of course, we distinguish different kinds of covariance matrices such as global, class, within-class or across-class covariance matrices. When we will speak about covariance matrices, the discussed problem (an assumption or an operation) will apply for any of these matrices. In equations, symbol $\mathbf{\Sigma}_x$ will be used for this purpose, where subscript x is used to emphasize that the type of covariance matrix is not specified.

4.5.1 Assumption of block diagonal covariance matrix

To reduce the number of statistics, an assumption can be made about statistical independency of certain coefficients of feature vectors. The assumption of coefficient independency implies zeros in corresponding position of covariance matrices. When statistics are estimated from feature vectors that consist of some static coefficients and their delta and acceleration coefficients, it is often considered that these three types of coefficients form three independent streams. It means that any coefficient from one stream is considered to be statistically independent of any coefficient from other streams. Although it is known that there is a considerable correlation between coefficients from these different streams (especially between static and acceleration coefficients), the assumption of their independency often improves both performance and efficiency of a method relying on the estimated statistics. Under this stream

independency assumption, covariance matrices are block diagonal:

$$\begin{bmatrix} \mathbf{\Sigma}_x & \emptyset & \emptyset \\ \emptyset & \mathbf{\Sigma}_{xd} & \emptyset \\ \emptyset & \emptyset & \mathbf{\Sigma}_{xa} \end{bmatrix}, \tag{4.8}$$

where $\mathbf{\Sigma}_x$, $\mathbf{\Sigma}_{xd}$ and $\mathbf{\Sigma}_{xa}$ are covariance matrices for static coefficients, delta coefficients and acceleration coefficients, respectively, and $\emptyset$ is a zero matrix.

In our experiments, feature vectors are created by concatenating feature vectors from two different feature streams A and B, where each original feature vector consists of static coefficients and their delta and acceleration coefficients. Each final concatenated feature vector has the following form:

$$\begin{bmatrix} \mathbf{c}^A & \mathbf{c}_d^A & \mathbf{c}_a^A & \mathbf{c}^B & \mathbf{c}_d^B & \mathbf{c}_a^B \end{bmatrix}, \tag{4.9}$$

where $\mathbf{c}^A$ and $\mathbf{c}^B$ are vectors of static coefficients from original feature stream A and B, respectively. Similarly $\mathbf{c}_d^A$, $\mathbf{c}_d^B$, $\mathbf{c}_a^A$ and $\mathbf{c}_a^B$ are vectors of delta and acceleration coefficients from both streams A and B.

In most of the feature combination experiments, such assumption of statistical independency is made that any static coefficient from either stream A or B is independent of any delta or acceleration coefficient and also that any delta coefficient is independent of any acceleration coefficient. Under such assumptions, covariance matrices of the concatenated feature vectors have the following form:

$$\begin{bmatrix} \mathbf{\Sigma}_x^{AA} & \emptyset & \emptyset & \mathbf{\Sigma}_x^{AB} & \emptyset & \emptyset \\ \emptyset & \mathbf{\Sigma}_{xd}^{AA} & \emptyset & \emptyset & \mathbf{\Sigma}_{xd}^{AB} & \emptyset \\ \emptyset & \emptyset & \mathbf{\Sigma}_{xa}^{AA} & \emptyset & \emptyset & \mathbf{\Sigma}_{xa}^{AB} \\ \mathbf{\Sigma}_x^{BA} & \emptyset & \emptyset & \mathbf{\Sigma}_x^{BB} & \emptyset & \emptyset \\ \emptyset & \mathbf{\Sigma}_{xd}^{BA} & \emptyset & \emptyset & \mathbf{\Sigma}_{xd}^{BB} & \emptyset \\ \emptyset & \emptyset & \mathbf{\Sigma}_{xa}^{BA} & \emptyset & \emptyset & \mathbf{\Sigma}_{xa}^{BB} \end{bmatrix}, \tag{4.10}$$

where $\mathbf{\Sigma}^{AA}$ and $\mathbf{\Sigma}^{BB}$ are covariance matrices estimated from vectors of static coefficients from stream A and B, respectively, $\mathbf{\Sigma}^{AB}$ is cross-covariance matrix between vectors of static coefficients from stream A and B and $\mathbf{\Sigma}^{BA}$ is transposed matrix $\mathbf{\Sigma}^{AB}$. The other symbols stay for similar covariance matrices and cross-covariance matrices for delta and acceleration coefficients.

4.5.2 PCA stabilization

In a highly dimensional feature space, such dimensions with very low variance can be usually found, in which the variance of data is comparable with available numerical

precision or with other type of noise introduced into the data. Almost no information important for discrimination between classes is usually contained in these dimensions. Moreover, such information is too noisy to be useful for recognition. The existence of such dimensions is potentially very dangerous for LDA and HLDA, where the criterion for the importance of a dimension is based on the separability of classes. According to this criterion, some low-variance dimensions can wrongly appear to be very important. PCA stabilization (PCA smoothing) can be used to overcome this problem. PCA transformation is used to perform dimensionality reduction in order to discard dimensions with very low variance. According to description given in section 2.5.1, basis of PCA transformation (columns of transformation matrix) are given by eigen vectors of global covariance matrix and each eigen value specifies the amount of variance in data preserved by projecting the data into the corresponding eigen vector. To perform dimensionality reduction, several eigen vectors corresponding to several lowest eigen values are not included to the transformation. In our experiments, such number of eigen vectors is omitted, that the amount of data variability in discarded dimensions[5] does not exceed specified limit ε. The limit ε is given by percentage of overall variance in the data (sum of all eigen values). The number of dimensions that have to be preserved is given by equation:

$$d = \arg\min_k \left\{ \frac{\sum_{i=1}^{k} |\lambda_i|}{\sum_{i=1}^{n} |\lambda_i|} > 1 - \frac{\varepsilon}{100} \right\}, \tag{4.11}$$

where n is dimensionality of feature vectors before PCA stabilization and λ_i are eigen values sorted in decreasing order. Note, that ε is set to 0.5% in most of our experiments. In the following text, we will refer to the feature space obtained by applying PCA stabilization as to *smoothed space*.

4.5.3 PCA stabilization preserving feature vector coefficient independency assumptions

One way of performing PCA stabilization is to preprocess all training and test data by projecting concatenated feature vectors into *smoothed space*. Using that approach, however, it is problematic to ensure that coefficient independency assumptions made on concatenated feature vectors (see section 4.5.1) will hold also in *smoothed space*. In order to fulfill this requirement, LDA or HLDA transformation and the necessary statistics are obtained in the following steps:

[5] Amount of data variability in discarded dimensions is given by the sum of absolute values of eigen values corresponding to omitted eigen vectors.

1. Global covariance $n \times n$ matrix $\hat{\mathbf{\Sigma}}_g^c$ is estimated from concatenated feature vectors of training set.

2. To obtain global covariance matrix $\hat{\mathbf{\Sigma}}_g^c$ in the form of 4.10, the proper coefficients of the matrix are set to zero.

3. Rows of PCA stabilization transformation $n \times d$ matrix, $\mathbf{S}$, are given by eigen vectors of covariance matrix $\hat{\mathbf{\Sigma}}_g^c$ corresponding to d highest eigen values, where d is given by equation 4.11. Because of the independency assumption imposed into covariance matrix $\hat{\mathbf{\Sigma}}_g^c$, each base vector (row) of transformation S effectively operates only on one set of feature coefficients that are assumed to be statistically dependent. In other words, the rows of S can be reordered to obtain matrix with many blocks of zeros similar to matrix 4.10.

4. All (global, class, across-class, within-class) covariance matrices $\hat{\mathbf{\Sigma}}_x^c$ (and other statistics[6]) required by LDA or HLDA are estimated from concatenated feature vectors of training set.

5. To obtain all covariance matrices $\hat{\mathbf{\Sigma}}_x^c$ in the form of 4.10, the proper coefficients of the matrices are set to zero.

6. All required covariance matrices $\hat{\mathbf{\Sigma}}_x^s$ for *smoothed space* are given as:

$$\hat{\mathbf{\Sigma}}_x^s = \mathbf{S}\hat{\mathbf{\Sigma}}_x^c\mathbf{S}^T. \tag{4.12}$$

   Because of the sparse character of matrices $\mathbf{S}$ and $\hat{\mathbf{\Sigma}}_x^c$, certain coefficients of resulting matrix $\hat{\mathbf{\Sigma}}_x^s$ will be zero too. This way, all independency assumption imposed into covariance matrices $\hat{\mathbf{\Sigma}}_x^c$ are projected also into *smoothed space* covariance matrices $\hat{\mathbf{\Sigma}}_x^s$.

7. Transformation matrix $\hat{\mathbf{A}}$ is derived by LDA or HLDA using covariance matrices $\hat{\mathbf{\Sigma}}_x^s$. This transformation can be used for postprocessing features that were projected into *smoothed space*.

8. Transformation matrix $\mathbf{A}$ for postprocessing concatenated feature vectors is given as:

$$\mathbf{A} = \hat{\mathbf{A}}\mathbf{S}. \tag{4.13}$$

[6]Mean vectors and Gaussian mixture occupation probabilities are required by the approach described in section 4.4.2. The extension of this procedure for mean vectors, which need to be treated in similar manner as covariance matrices, is straightforward and is not described here.

When the approach described in section 4.4.2 is used (HLDA transformation is re-estimated iteratively during the training of HMM models) steps 4 to 8 must be repeated in every iteration of HMM training.

4.5.4 Smoothed HLDA

HLDA requires the covariance matrix to be estimated for each class. The higher number of classes is used, the fewer feature vector examples are available for each class and class covariance matrix estimates become more noisy. LDA overcomes this problem by assuming that there is the same within-class covariance matrix for all classes. The within-class covariance matrix is computed as the weighted average of all class covariance matrices according to equation 2.20, which ensures its robust estimate. On the other hand, the assumption of the same covariance matrix for all classes is usually not fulfilled for real speech features, and therefore, the transformation derived using LDA is not the optimal one.

We propose a technique based on a combination of HLDA and LDA, 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. In experiments, this technique will be refereed as Smoothed HLDA (SHLDA). SHLDA differs from HLDA only in the way of class covariance matrices estimation. In the case of SHLDA, the estimate of a class covariance matrix is given by equation:

$$\check{\mathbf{\Sigma}}^{(j)} = \alpha\hat{\mathbf{\Sigma}}^{(j)} + (1-\alpha)\mathbf{\Sigma}_{wc}, \tag{4.14}$$

where $\check{\mathbf{\Sigma}}^{(j)}$ is "smoothed" estimate of covariance matrix of j^{th} class used by SHLDA, $\hat{\mathbf{\Sigma}}^{(j)}$ is estimate of ordinary covariance matrix of j^{th} class given by equation 2.21, $\mathbf{\Sigma}_{wc}$ is estimate of within-class covariance matrix given by equation 2.20 and α is smoothing factor, which is a value ranging from 0 to 1. Note that for α equal to 0, SHLDA becomes LDA and for α equal to 1, SHLDA becomes HLDA.

4.5.5 Clustered HLDA

We propose also an alternative modification of HLDA increasing its robustness, to which we will refer to as Clustered HLDA (CHLDA). The modification is based on an assumption that such clusters (sets of classes) can be found, that all classes belonging to one particular cluster have the same covariance matrix and differ only in mean

vectors. Instead of *class covariance matrices* $\hat{\mathbf{\Sigma}}^{(j)}$ and *class occupation counts*[7] N_j, which are the statistics used by ordinary HLDA, statistics used by CHLDA are *cluster within-class covariance matrices* $\hat{\mathbf{\Sigma}}^{C}_{cwc}$ and *cluster occupation counts* N^C. An estimate of *cluster within-class covariance matrix* for cluster C is given by equation:

$$\hat{\mathbf{\Sigma}}^{C}_{cwc} = \frac{\sum_{j \in C} N_c \hat{\mathbf{\Sigma}}^{(j)}}{N^C}. \tag{4.15}$$

Cluster occupation count is given as the sum of *class occupation counts* of all classes belonging to the cluster:

$$N^C = \sum_{j \in C} N_j. \tag{4.16}$$

In CHLDA, the issue is how to divide classes to clusters. For this purpose, a sophisticated clustering method can be used based, for example, on measuring similarities between feature distributions representing individual classes [62, 4]. Alternatively, an algorithm based on modified K-means clustering can be used that is looking for such clustering that locally maximizes likelihood of data for a model where covariance matrices of classes belonging to the same cluster are tied. Here *cluster within-class covariance matrices*, $\hat{\mathbf{\Sigma}}^{C}_{cwc}$, are estimated given the current clustering and then classes are reassigned to clusters using the formula:

$$C^{(j)} = \arg\max_{j} \left\{ -\log\left(\det\left(\hat{\mathbf{\Sigma}}^{C}_{cwc}\right)\right) - \mathrm{tr}\left(\hat{\mathbf{\Sigma}}^{j}\ \hat{\mathbf{\Sigma}}^{C\ -1}_{cwc}\right) \right\}, \tag{4.17}$$

were $C^{(j)}$ denotes the cluster to which class j is newly assigned. The whole procedure is repeated until no change in class assignment is observed. Since there will be many local maxima, the resulting clustering will be very dependent on the initial one. The clustering methods cited above can be used to find a good starting point.

However, in our experiment, a simple clustering is tested, where only two clusters are considered: classes (HMM states) representing non-speech parts of utterances and classes representing speech parts.

Note, that considering each particular class to form a separate cluster, CHLDA becomes HLDA. The other way around, making only one cluster consisting of all classes, CHLDA becomes LDA.

[7] Class occupation count is the number of frames (feature vectors) used to estimate the statistics of a particular class.

4.6 Feature combination experiments

In the following experiments, we will examine recognition systems using combined features, which are created as a combination of two base feature streams. A different combination method described above in this chapter is applied in each particular experiment. All the used combination methods are based on postprocessing of concatenated feature vectors. In the following text, we will say that an *experiment is based on* a particular postprocessing method (e.g. experiment based on LDA).

4.6.1 Experimental setup

The experimental setup was designed to be consistent with that one used for experiments with complementarity measures and ROVER combination (see chapter 3). The consistency of these two experimental setups will allow us:

- to verify, if the proposed complementarity measures are applicable also for system combination at the feature level.
- to directly compare the results obtained by system combination at the feature level and system combination using ROVER.

For training and testing, both clean speech data from TI-Digits database and data artificially corrupted by noise are used (same data as those in experiments with ROVER). The detailed description of training and test data can be found in section 3.4.5. In correspondence to experiments with ROVER, performance of each recognition system is represented by WER evaluated on *seen conditions test data* (see section 3.4.5). Except for the feature extraction part, recognizers are exactly the same as those described in section 3.4.5 (whole word continuous HMMs).

Feature stream for each recognizer is created by feature combination of pair of base feature streams. As base feature streams, features BSL, LPCC, DA1, DA4, B15, B30, ENG and POW were used. The detailed description of these features can be again found in section 3.4.5. Note, that NOE features are not used in the following experiments, since all coefficients in NOE feature vectors are absolutely identical to those in BSL and ENG features (only C0 or energy coefficient is missing). Therefore, no gain could be obtained by combining these features.

The feature combination consists of the following steps: First, each pair of corresponding base feature vectors is concatenated. Dimensionality of each base feature

vector is 45 (15 static coefficients, 15 delta and 15 acceleration coefficients). By default, all coefficients of any concatenated vector are scaled to have uniform variance according to equation 4.1. Each 90-dimensional concatenated feature vector is then decorrelated using a particular method such as PCA or HLDA and its dimensionality is reduced again to 45.

In all experiments, such statistical independency of concatenated feature vectors coefficients is implicitly assumed that imply the covariance matrices having the form 4.10 (see section 4.5.1). In addition, PCA stabilization is used in all experiments based on LDA, HLDA, SHLDA an CHLDA. The procedure described in section 4.5.3 is used for this purpose. In our experiments, only $\varepsilon = 0.5\%$ of data variability is removed by PCA stabilization. The main purpose of PCA stabilization is to remove the totally redundant dimensions.

As was already mentioned, BSL features, for example, differ from ENG feature vectors only in 3 coefficients. Therefore, in BSL-ENG concatenated feature vectors, there are 42 coefficients that are repeated twice. Performing PCA on such feature vector, 42 dimensions are found in which data has zero variability (there are 42 eigen values of global covariance matrix equal to zero). Using PCA stabilization, such zero variance dimensions are removed first and no special care must be taken of the repeated coefficients.

Removing more than 0.5% of data variability was not possible, because it led to concatenated feature vectors having less than required 45 coefficients for some particular combinations of base feature streams.

Any statistics required by PCA stabilization, PCA, LDA, HLDA, SHLDA and CHLDA are estimated from the same data used also for HMM training. HMM state labels define 180 classes in most of experiments based on LDA, HLDA, SHLDA and CHLDA. The labels were obtained using HMM state-level forced alignment performed on the clean speech. For this purpose, HMMs — with the same topology used also in all our recognition experiments — were trained using BSL features derived from clean speech. The obtained labels serve as a transcription for both clean speech and speech with artificially added noise, which is correct under the assumption that adding noise does not change alignment of speech frames to HMM states. In our experiments, we also assume that this alignment is the same for all the different base features and their combinations.

4.6.2 Size of required statistics

Using PCA stabilization, 90-dimensional concatenated feature vector is projected into less-dimensional *smoothed space*. Dimensionalities of these spaces, however, differ for combinations of different pairs of feature streams. The independency assumptions made for concatenated feature vectors are projected also into *smoothed* space. In other words, if certain coefficients of covariance matrices $\mathbf{\Sigma}_x$ estimated from concatenated features are forced to be zero, certain coefficients of corresponding covariance matrix $\hat{\mathbf{\Sigma}}_x$ in the smoothed space (see equation 4.12) will be zero too. Since covariance matrices $\hat{\mathbf{\Sigma}}_x$ are the necessary statistics for the following LDA or HLDA, the number of non-zero coefficients in these matrices is related to the complexity of the task and to the reliability of the estimated statistics.

For different combinations of base features, the number of covariance matrix non-zero coefficients together with the dimensionality of "smoothed" space are presented in table 4.1. Only upper triangular part of the full symmetric matrix is shown. Values in the diagonal correspond to the cases where concatenated feature vectors are created by concatenating the base features with themselves (each coefficient is in the feature vector twice). In these cases, PCA stabilization, removing exactly all the redundant dimensions, results in features having the same dimensionality (45) and the same number of non-zero coefficients in covariance matrices (675) as the original base features. Graphical representation of covariance matrix non-zero coefficient counts is given in figure 4.1. It can be seen that the combinations of DA4 or DA1 features with any other features require the highest size of the statistics (combinations with LPCC features follow), which makes the task much more complex in comparison with other possible feature pair combinations. On the other hand, according to findings from chapter 3, DA4 and LPCC features should combine very well with any other features.

4.6.3 How to read and compare experimental results

In each of the following experiments, where a particular feature combination method is used, WERs of 36 recognizers are evaluated. Each recognizer corresponds to one of 36 possible combinations of base feature stream pairs. Tables 4.3 to 4.14, each corresponding to one particular experiment, always include all these 36 WERs. Only upper triangular part of the full symmetric matrix is shown in the tables. Each value in the table diagonal corresponds to a case where particular base features are combined with themselves. In such a case, feature combination is equivalent to applying the

	POW	DA4	30B	ENG	BSL	15B	LPCC	DA1
POW	675/45	1850/72	1122/58	770/48	706/46	1083/57	1454/66	1909/73
DA4		675/45	1850/72	1795/71	1795/71	1973/75	2102/78	1907/73
30B			675/45	770/48	706/46	1083/57	1546/68	1909/73
ENG				675/45	675/45	1085/57	1409/65	1850/72
BSL					675/45	1009/55	1501/67	1850/72
15B						675/45	1546/68	1973/75
LPCC							675/45	2102/78
DA1								675/45

Table 4.1: Numbers of covariance matrix $\hat{\mathbf{\Sigma}}_x$ non-zero coefficients / dimensionality of feature vector in the *smoothed space* for different pairs of combined base features.

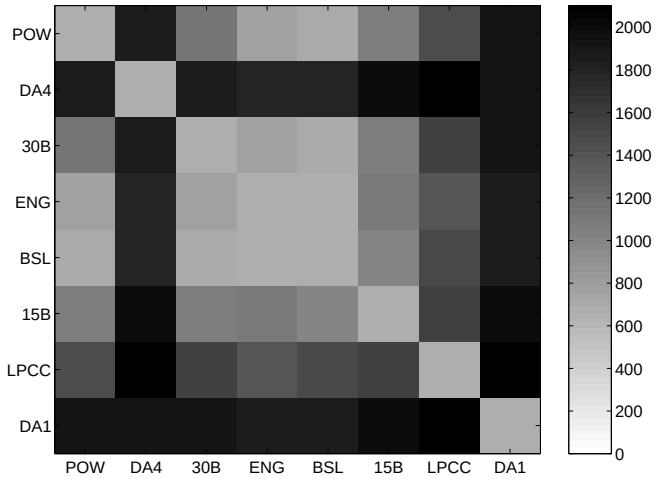


Figure 4.1: Numbers of covariance matrix $\hat{\mathbf{\Sigma}}_x$ non-zero coefficients in the *smoothed space* for different pairs of combined base features.

postprocessing method (PCA, LDA, ...) directly on the base features in order to decorrelate them without performing any dimensionality reduction. The values in the last row of each table are WERs of systems using only base features[8] (no combination, no postprocessing). We can compare these values with the values from the diagonal to examine the gain obtained only by additional decorrelation of base features using given feature combination method.

To compare results of different experiments, it is impractical to compare all individual values in the tables. Moreover, because of limited amount of test data, individual WERs are not very reliable. We will be rather interested in general trends seen in results of individual experiments. For example, we will want to know "what is the average performance of one combination method in comparison with another one" or "how beneficial is the combination of one particular base feature stream (e.g. DA4 or LPCC) with all other features using a given combination method". To make such general trends in results clearly visible, graphical representation of WER tables are shown in figures 4.2, 4.3, 4.4 and 4.5. The darker color of a field in the figure, the higher WER of corresponding combined system. Note that figure 4.2 can NOT be directly compared with the other three figures (the same range of colors represents different range of WERs). However, figures 4.3, 4.4 and 4.5 are directly comparable.

Table 4.2 is another source of information allowing for general comparison of experiments. Each line of the table corresponds to one experiment (one feature combination method). Each value in the column entitled *Average decorrelating system WER* is obtained by averaging all diagonal values from WER table for a given experiment. In fact, the value represents the average ability of a feature combination method to only properly decorrelate base features[9]. For comparison, average WER of systems using only base features is 3.11%. If *Average decorrelating system WER* is smaller than this value, an additional decorrelation using the corresponding combination method is in average helpful. Conversely, each value in the column entitled *Average combining system WER* represents the average ability of a feature combination method to combine a pair of different base features. The value is obtained by averaging all values from WER table that are out of diagonal.

[8] These WERs were already presented in table 3.2.

[9] Here, we believe that better decorrelation of features implies better recognition performance.

System combination method		Average decorrelating system WER	Average combining system WER
PCA	Default setting	3.70	3.35
	Derived only from clean speech	4.64	4.00
	Alternative coefficient scaling	4.05	3.72
	Derived only from non-silence	3.61	3.36
	No independency assumption	4.70	4.27
Classes given by HMM state labels	LDA ($\alpha = 0$)	**2.91**	2.87
	SHLDA $\alpha = 0.25$	3.08	2.82
	SHLDA $\alpha = 0.5$	3.03	2.82
	SHLDA $\alpha = 0.75$	3.04	2.80
	HLDA ($\alpha = 1$)	3.14	2.91
	CHLDA	2.96	2.78
classes given by mixture occupation probabilities	LDA	**2.91**	2.81
	SHLDA $\alpha = 0.25$	3.12	2.81
	SHLDA $\alpha = 0.5$	3.03	**2.77**
	SHLDA $\alpha = 0.75$	3.38	3.13
	HLDA	3.30	3.09

Table 4.2: For each feature combination method, two average WER are presented: *Average decorrelating system WER* is the average WER for systems where the given combination method is used only to decorrelate individual base features. This value is obtained by averaging values on the diagonal of proper WER table (tables 4.3 to 4.14). For comparison, average WER of systems using only base features is 3.11%. *Average combining system WER* is the average WER for systems where the given combination method is used to combine pairs of different base features. This value is obtained by averaging values out of the WER table diagonal.

	POW	DA4	30B	ENG	BSL	15B	LPCC	DA1
POW	3.37	3.32	3.39	3.34	3.43	3.64	3.11	3.36
DA4		3.59	3.24	3.33	3.25	3.05	3.15	3.18
30B			3.86	3.50	3.66	3.20	2.99	3.59
ENG				3.31	3.24	3.32	3.37	3.64
BSL					3.55	3.47	3.35	3.74
15B						3.34	3.27	3.51
LPCC							3.92	3.17
DA1								4.67
BASE	2.90	2.91	2.99	3.00	3.04	3.14	3.36	3.51

Table 4.3: WER of systems using feature combination based on PCA.

4.6.4 Experiments based on PCA

In the following experiments, PCA transformation is used for postprocessing concatenated feature vectors. WERs obtained in baseline PCA based experiment are presented in table 4.3. All default settings described in experimental setup are respected in this experiment. We can see that PCA has failed even for the task of decorrelation of base features (compare values from the table bottom line with the values from the diagonal). The decorrelation of any base features using PCA results in significant degradation in performance. In figure 4.2a, which is the graphical representation of table 4.3, bright rows and columns DA4 and LPCC indicate relatively lower WERs when these features participate in the feature combination. This is in agreement with our expectation that DA4 and LPCC features should combine well. Nevertheless, even the combined system with the lowest WER does not outperform the best system using base features POW.

PCA derived only on clean speech

Probably the most important factor responsible for PCA failure is the presence of noise in the data used for estimation of PCA transformation. In our experiments, all statistics required by PCA (LDA, HLDA, ...) are estimated from all the speech data used also for HMM training. This data consists of both clean utterances and utterances corrupted by noise. In the case of PCA, the importance of a dimension in the feature space is given only by the amount of data variability in that dimension. Therefore, a dimension with high variance caused by noise can be considered as important, even

	POW	DA4	30B	ENG	BSL	15B	LPCC	DA1
POW	3.37	3.32	3.39	3.34	3.43	3.64	3.11	3.36
DA4		3.59	3.24	3.33	3.25	3.05	3.15	3.18
30B			3.86	3.50	3.66	3.20	2.99	3.59
ENG				3.31	3.24	3.32	3.37	3.64
BSL					3.55	3.47	3.35	3.74
15B						3.34	3.27	3.51
LPCC							3.92	3.17
DA1								4.67
BASE	2.90	2.91	2.99	3.00	3.04	3.14	3.36	3.51

Table 4.4: WER of systems using feature combination based on PCA. Only feature vectors representing clean speech are used for estimation of PCA transformation.

thought no variance in that dimension is caused by speech. It can appear that a possible solution is to estimate the statistics only from features derived from clean speech. However, feature distributions for clean speech and noisy speech are very different and, therefore, directions of variance caused by clean speech and speech in noise are not the same. WERs of systems, where features are combined using PCA transformation estimated only on clean speech, are presented in table 4.4 and the graphical representation of the table is given in figure 4.2b. Results obtained in this experiment are even worse than those presented in table 4.3, where both clean and noisy speech were used. Therefore, in all other experiments both clean and noisy speech are used to estimate any statistics required by any combination method.

Alternative feature coefficient scaling

The importance of equalization of feature coefficient variances before performing PCA was discussed in section 4.3. In the previous experiments, all concatenated feature vector coefficients are scaled according to equation 4.1 to unity variance. We have proposed an alternative scaling given by equation 4.2, which preserves relative variances in the scope of each substream of concatenated feature stream. Here, we consider concatenated feature stream to consist of six substreams, each corresponding to one term of composite vector 4.9 (static, delta and acceleration coefficients from both original feature streams). The results obtained with this alternative scaling method are presented in table 4.5 and figure 4.2c. Comparing these results with those from table 4.3 (figure 4.2a), we can see consistent degradation in performance. Scaling all coefficients to unity variance is therefore used in all other experiments.

	POW	DA4	30B	ENG	BSL	15B	LPCC	DA1
POW	3.97	3.23	3.90	3.77	4.09	3.54	3.48	4.10
DA4		3.55	3.47	3.68	3.56	3.24	3.22	3.62
30B			4.19	4.08	4.36	3.70	3.57	4.14
ENG				3.79	3.80	3.67	3.38	4.27
BSL					3.88	3.49	3.73	4.02
15B						4.02	3.30	4.04
LPCC							4.26	3.67
DA1								4.77
BASE	2.90	2.91	2.99	3.00	3.04	3.14	3.36	3.51

Table 4.5: WER of systems using feature combination based on PCA. Alternative method of scaling concatenated feature vector coefficients given by equation 4.2 is used.

PCA derived only on non-silence parts of utterances

Another important problem discussed in section 4.3 is a selection of feature frames used for estimation of statistics. Often, majority of frames available for training represent non-speech parts of utterances. If all the frames are used for the estimation, the statistics are mainly given by these non-speech parts. In our previous experiments, all frames representing both speech and non-speech were used for estimation. Although, the ratio between the number of speech and non-speech frames is not critical in our data (715437 speech frames, 250155 non-speech frames), still, the number of non-speech frames is much higher than a number of frames representing any other individual speech event. In table 4.6 and figure 4.2d, we present the results of an experiment, in which non-speech frames at the boundaries of each utterance (99% of all non-speech frames) were not used for estimation of statistics. As can be seen, in average, only slight improvement was obtained in comparison with results from table 4.3 (figure 4.2a). No significant benefit of omitting non-speech frames was observed also for experiments with LDA and HLDA. Therefore, both speech and non-speech frames are used for the estimation in all other experiments.

PCA without independency assumption

It was declared in experimental setup section that the independency assumption leading to the covariance matrices of the form 4.10 is applied in all experiments. To justify this independency assumption, one PCA based experiment was carried out, in which

	POW	DA4	30B	ENG	BSL	15B	LPCC	DA1
POW	3.41	3.18	3.53	3.57	3.43	3.40	3.50	3.49
DA4		3.49	3.24	3.23	3.18	3.15	3.18	3.17
30B			3.59	3.53	3.56	3.34	3.31	3.45
ENG				3.56	3.44	3.34	3.61	3.39
BSL					3.44	3.20	3.32	3.37
15B						3.40	3.44	3.37
LPCC							4.00	3.29
DA1								4.01
BASE	2.90	2.91	2.99	3.00	3.04	3.14	3.36	3.51

Table 4.6: WER of systems using feature combination based on PCA. Feature frames corresponding to silence parts of utterances are not used for estimation of PCA transformation.

no such assumption is made. In this experiment, full covariance matrix is estimated and used to derive PCA transformations. The results are presented in table 4.7 and figure 4.2e. Significant and consistent degradation in performance is observed in comparison with results from table 4.3 (figure 4.2a). Similar degradation was observed in experiments where PCA was replaced by LDA or HLDA.

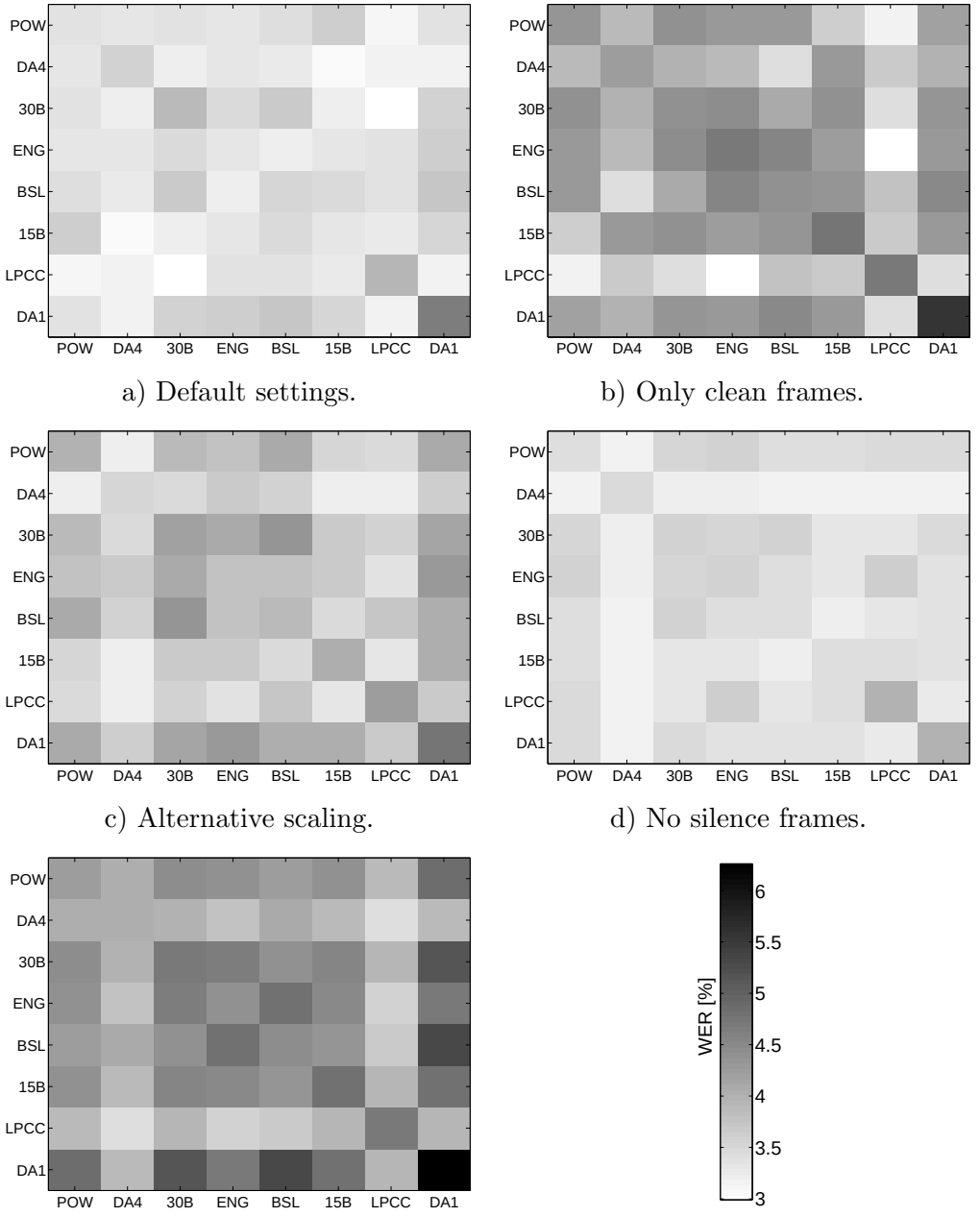
To conclude the previous experiments, note, that we had no success with any feature combinations when only PCA was used for postprocessing concatenated feature vectors.

4.6.5 Experiments based on LDA and HLDA with classes given by HMM state labels

In the following experiments, LDA, HLDA, SHLDA and CHLDA are used for postprocessing of concatenated feature vectors. Here, classes are given by state labels, which were described in section 4.6.1.

LDA

Table 4.8 presents WERs obtained in experiment based on LDA. We can see consistent improvement in performance when base features are only decorrelated using LDA (compare values from table bottom line with the values from the diagonal). However, values out of table diagonal do not indicate any consistent advantage of combining pairs of different features. Moreover, combination of DA4 features with any other



a) Default settings.

b) Only clean frames.

c) Alternative scaling.

d) No silence frames.

e) No independency assumption.

Figure 4.2: WER of systems using feature combination based on PCA. This figure can NOT be be directly compared with figures 4.3, 4.4 and 4.5; different colorbar is used for this particular figure.

	POW	DA4	30B	ENG	BSL	15B	LPCC	DA1
POW	3.37	3.32	3.39	3.34	3.43	3.64	3.11	3.36
DA4		3.59	3.24	3.33	3.25	3.05	3.15	3.18
30B			3.86	3.50	3.66	3.20	2.99	3.59
ENG				3.31	3.24	3.32	3.37	3.64
BSL					3.55	3.47	3.35	3.74
15B						3.34	3.27	3.51
LPCC							3.92	3.17
DA1								4.67
BASE	2.90	2.91	2.99	3.00	3.04	3.14	3.36	3.51

Table 4.7: WER of systems using feature combination based on PCA. No assumption is made on concatenated feature vector coefficients independency; full covariance matrix is used to derive PCA transformation.

features leads even to degradation in the performance, which is in contrast to our expectation that DA4 features should combine the best. This can be clearly seen in figure 4.3a, which is the graphical representation of table 4.8. All fields in the row (column) DA4 (except the field on the diagonal) are relatively dark, which corresponds to high WERs for combinations of DA4 features with any other features.

It will be shown that inability of LDA to correctly combine these features is caused by its wrong assumption that all classes have the same covariance matrices. Such assumption is relaxed in the following experiment based on HLDA.

HLDA

The results of experiment based on HLDA are shown in table 4.9 an figure 4.3b. In this experiment, the combination of DA4 or LPCC features with any other features is very beneficial, which is perfectly in agreement with findings from chapter 3, where complementarity measures were tested. In the cases where DA4 or LPCC features participate in the feature combination, WER of the combined system is always lower than WER of the better individual system. With one exception, WERs of these combined systems are also lower than WER of the best individual system POW. The best result was obtained for combination DA4-LPCC, where WER of 2.43% corresponds to 16% relative improvement with respect to system POW.

On the other hand, HLDA based systems combining features other than DA4 or LPCC provide often worse results than corresponding systems based on LDA. In

	POW	DA4	30B	ENG	BSL	15B	LPCC	DA1
POW	2.88	3.06	2.68	2.95	2.84	2.92	2.62	2.88
DA4		2.70	3.11	2.99	3.02	3.02	2.90	3.15
30B			2.72	2.82	3.00	2.71	2.64	2.88
ENG				2.84	2.83	2.94	2.70	2.72
BSL					2.77	2.86	2.87	2.90
15B						2.83	2.89	2.78
LPCC							3.13	2.61
DA1								3.43
BASE	2.90	2.91	2.99	3.00	3.04	3.14	3.36	3.51

Table 4.8: WER of systems using feature combination based on LDA. Classes are given by HMM state labels.

average, LDA outperforms HLDA in both the ability to decorrelate base features and the ability to combine different features (see table 4.2). A probable explanation for this HLDA behavior is the following:

As HLDA requires to estimate statistics of much larger size in comparison to LDA, and as we have only limited amount of data available for their estimation, the results obtained using HLDA are biased with an additional error caused by more noisy estimates. However, the ability of HLDA to combine complementary information from different feature streams seems to be much better in comparison to LDA. If two highly complementary feature streams are combined using HLDA, the error bias is negligible in comparison to the gain obtained by combination of complementary information. In such cases, HLDA is superior to LDA. In opposite, if two not much complementary feature streams are combined, the strength of HLDA is not employed and HLDA provides worse results than LDA because of the error bias. In such cases, success of HLDA is mainly given by the amount of necessary statistics[10].

Smoothed HLDA

SHLDA was proposed in section 4.5.4 as a modification of HLDA with more robust estimation of statistics. The robustness is achieved by smoothing estimates of class

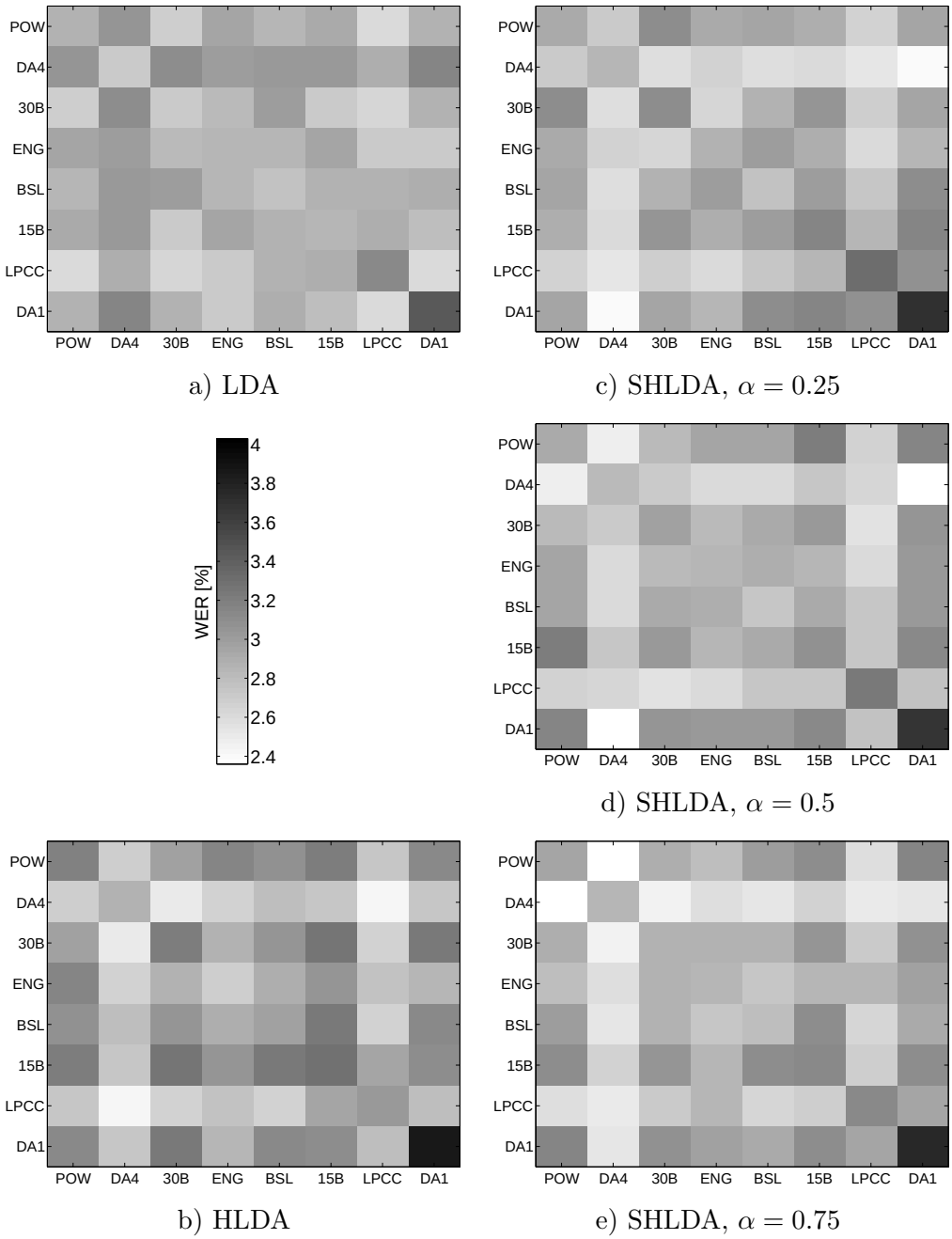
[10] As can be seen in figure 4.3b, HLDA often fails for combinations with 15B or DA1 features. These feature were not found to be so much complementary to most of other features (see figure 3.3 or 3.4), but the amounts of statistics required for combinations with these features are always high (see figure 4.1).

	POW	DA4	30B	ENG	BSL	15B	LPCC	DA1
POW	3.19	2.69	2.98	3.15	3.07	3.22	2.75	3.12
DA4		2.86	2.51	2.67	2.79	2.74	**2.43**	2.74
30B			3.20	2.87	3.06	3.27	2.65	3.24
ENG				2.68	2.89	3.06	2.77	2.85
BSL					2.98	3.24	2.66	3.12
15B						3.29	2.96	3.11
LPCC							3.02	2.78
DA1								3.87
BASE	2.90	2.91	2.99	3.00	3.04	3.14	3.36	3.51

Table 4.9: WER of systems using feature combination based on HLDA. Classes are given by HMM state labels.

covariance matrices according to equation 4.14. In fact, SHLDA can be perceived as a compromise between LDA and HLDA, where smoothing factor α (see equation 4.14) tunes SHLDA to be more similar to either LDA or HLDA. By setting α to 0, SHLDA becomes LDA, and the other way around, by setting α to 1, SHLDA becomes HLDA. In our experiments with SHLDA, values $\alpha = 0.25$, 0.5 and 0.75 were tested. Figures 4.3c, 4.3d and 4.3e show results obtained in these experiments, respectively. The best average performance of systems combining different features is obtained for $\alpha = 0.75$ (see table 4.2). For this case, results are presented also in the form of table 4.10. The ability of SHLDA to decorrelate base features is not as good as in the case of LDA (see table 4.2). However, the error bias discussed in the previous paragraph was visibly decreased and in average SHLDA outperforms both LDA and HLDA. Probably the most important fact about SHLDA is that, the advantage of using DA4 or LPCC features for the feature combination becomes even more prominent. We can also see that WERs of systems combined using SHLDA, which are shown in figure 4.3e, reasonably respect complementarity measures LBWER and DWER shown in figures 3.3 and 3.4[11].

[11] In figures 3.3 and 3.4, ignore row and column NOE corresponding to features not used in the feature combination experiments. Ignore also values on the diagonal, which must be principally much higher than values out of diagonal for the complementarity measures.



a) LDA

c) SHLDA, $\alpha = 0.25$

d) SHLDA, $\alpha = 0.5$

b) HLDA

e) SHLDA, $\alpha = 0.75$

Figure 4.3: WER of systems using feature combination based on LDA, HLDA and SHLDA. Classes are given by HMM state labels.

	POW	DA4	30B	ENG	BSL	15B	LPCC	DA1
POW	2.95	**2.36**	2.89	2.80	3.00	3.11	2.57	3.15
DA4		2.85	2.46	2.59	2.53	2.65	2.50	2.53
30B			2.86	2.87	2.87	3.06	2.72	3.08
ENG				2.85	2.73	2.85	2.85	2.97
BSL					2.80	3.10	2.63	2.92
15B						3.14	2.68	3.11
LPCC							3.13	2.95
DA1								3.76
BASE	2.90	2.91	2.99	3.00	3.04	3.14	3.36	3.51

Table 4.10: WER of systems using feature combination based on SHLDA for $\alpha = 0.75$. Classes are given by HMM state labels.

Clustered HLDA

CHLDA was proposed in section 4.5.5 as an alternative robust modification of HLDA. Here, the robustness is achieved by making clusters of classes having similar covariance matrices and computing LDA-like statistics for each such cluster (see section 4.5.5). In the following experiment, only two clusters are considered: classes (corresponding to HMM states) representing non-speech parts of utterances and classes representing speech parts. The results obtained in this experiment are shown in table 4.11 and figure 4.4. The ability of CHLDA to decorrelate base features is virtually the same as in the case of LDA (see table 4.2), which was the method giving so far the best results for this purpose. In average, CHLDA outperforms all previously described systems in its ability to combine different features streams. DA4 and LPCC features remain the features that combine the best with other features, however, results obtained for combinations with these features are not as good as those obtained with HLDA and SHLDA. For example, for combinations with DA4 features using SHLDA, $\alpha = 0.75$, WER ranges from 2.36 to 2.65. In the case of CHLDA, WER ranges from 2.50 to 2.79.

Note again, that considering only one cluster of classes, CHLDA becomes LDA. In our experiment, making only two clusters (speech, non-speech) CHLDA significantly outperforms LDA and in average also HLDA. Further improvement can be probably obtained using more sophisticated clustering method mentioned in section 4.5.5.

	POW	DA4	30B	ENG	BSL	15B	LPCC	DA1
POW	2.84	2.59	2.96	2.81	2.88	2.70	2.86	2.67
DA4		2.77	2.71	2.76	2.69	2.79	2.50	2.54
30B			2.93	2.92	2.89	2.88	2.67	3.02
ENG				2.72	2.85	2.92	2.66	2.83
BSL					2.77	2.84	2.74	2.89
15B						3.02	2.80	2.75
LPCC							2.99	2.62
DA1								3.60
BASE	2.90	2.91	2.99	3.00	3.04	3.14	3.36	3.51

Table 4.11: WER of systems using feature combination based on CHLDA. Classes are given by HMM state labels. Two clusters are considered: HMM states representing non-speech parts of utterances and states representing speech parts.

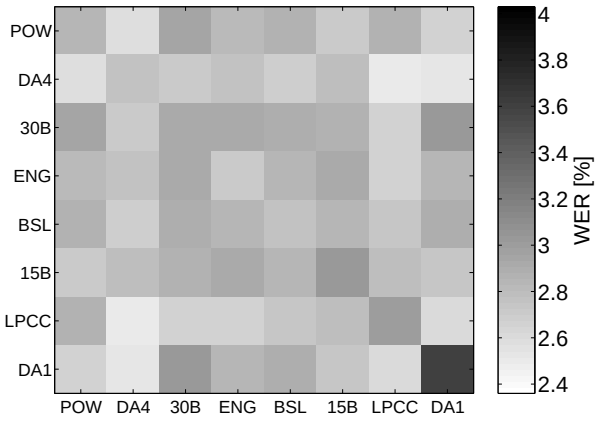


Figure 4.4: WER of systems using feature combination based on CHLDA. Classes are given by HMM state labels. Two clusters are considered: HMM states representing non-speech parts of utterances and states representing speech parts.

4.6.6 Experiments based on LDA and HLDA with classes given by mixture occupation probabilities

In the experiments described in the previous section, "hard" assignment of speech frames to classes was given by predefined HMM state labels. We will refer to the classes or the method used in those experiments using modifier *hard* (e.g. *hard* classes, *hard HLDA*). In the following experiments, the alternative approach of deriving LDA or HLDA transformation is used, that was described in section 4.4.2. In this approach, each class corresponds to one Gaussian mixture component of an HMM state and "soft" assignment of speech frames to the classes is given by the mixture component occupation probabilities. In this case, LDA or HLDA transformation matrix must be re-estimated after every iteration of HMM training according to the iterative algorithm described in section 4.4.2. We will use modifier *soft* to emphasize that a given method uses the mixture component related classes.

LDA

WERs obtained in the experiment based on *soft* LDA are presented in table 4.12 and figure 4.5a. Comparing these results with those obtained using *hard* LDA, we can see that, in both cases, the ability to decorrelate base features is about the same (see table 4.2). However, in average, *soft* LDA visibly outperforms *hard* LDA in its ability to combine different feature streams (see table 4.2). In contrast to *hard* LDA, *soft* LDA has NOT failed for the cases, where DA4 features participate in feature combination (fields in DA4 row in figure 4.5a are not as dark as the corresponding fields in figure 4.3a). On the other hand, again, no visible advantage of using DA4 or LPCC features for the feature combination is observed.

Note that although the number of *soft* classes is three times higher than the number of *hard* classes in our experiments (3 mixture components are used per state), the size of required statistics is the same for both *soft* LDA and *hard* LDA. In both cases only two covariance matrices are required - across-class and within-class.

HLDA

The results obtained in experiments with soft HLDA are shown in table 4.13 and figure 4.5b. In contrast to LDA, we have observed consistent degradation in performance when soft HLDA is used instead of hard HLDA (compare tables 4.13, 4.9 or figures 4.5b, 4.3b). We have already discussed the problem with the size of required

	POW	DA4	30B	ENG	BSL	15B	LPCC	DA1
POW	2.90	2.76	2.81	2.87	2.75	2.79	2.92	2.63
DA4		2.83	2.91	2.91	2.78	2.74	2.84	2.73
30B			2.64	2.83	2.75	2.90	2.73	2.97
ENG				2.58	2.86	2.84	2.94	2.74
BSL					2.92	2.88	2.73	2.82
15B						3.12	2.86	2.79
LPCC							2.97	2.62
DA1								3.35
BASE	2.90	2.91	2.99	3.00	3.04	3.14	3.36	3.51

Table 4.12: WER of systems using feature combination based on LDA. Soft assignment of speech frames to classes is given by Gaussian mixture occupation probabilities.

statistics for the case of hard HLDA (see section 4.6.5). In the case of soft HLDA, the size of required statistics is yet three times larger, which is the reason for the degradation in performance.

SHLDA

To overcome the problems related to the wrong assumption of shared covariance matrix in the case of LDA and noisy statistic estimates in the case of HLDA, soft SHLDA is tested again in three experiments for smoothing factor α equal to 0.25, 0.5 and 0.75. WERs of combined systems obtained in these experiments are presented in figures 4.5c, 4.5d and 4.5e. The best results were obtained for α equal to 0.5, which means that more smoothing of covariance matrices is needed in comparison to hard SHLDA with optimal $\alpha = 0.75$. Presumably, more smoothing is needed to compensate for more noisy statistic estimates caused by increase in the number of classes (by the factor of three). WERs obtained in experiment with soft SHLDA for $\alpha = 0.5$ are presented also in table 4.14. Note that this experiment is the one, for which the lowest average WER is obtained for combinations of different features (see table 4.2).

4.7 Discussion

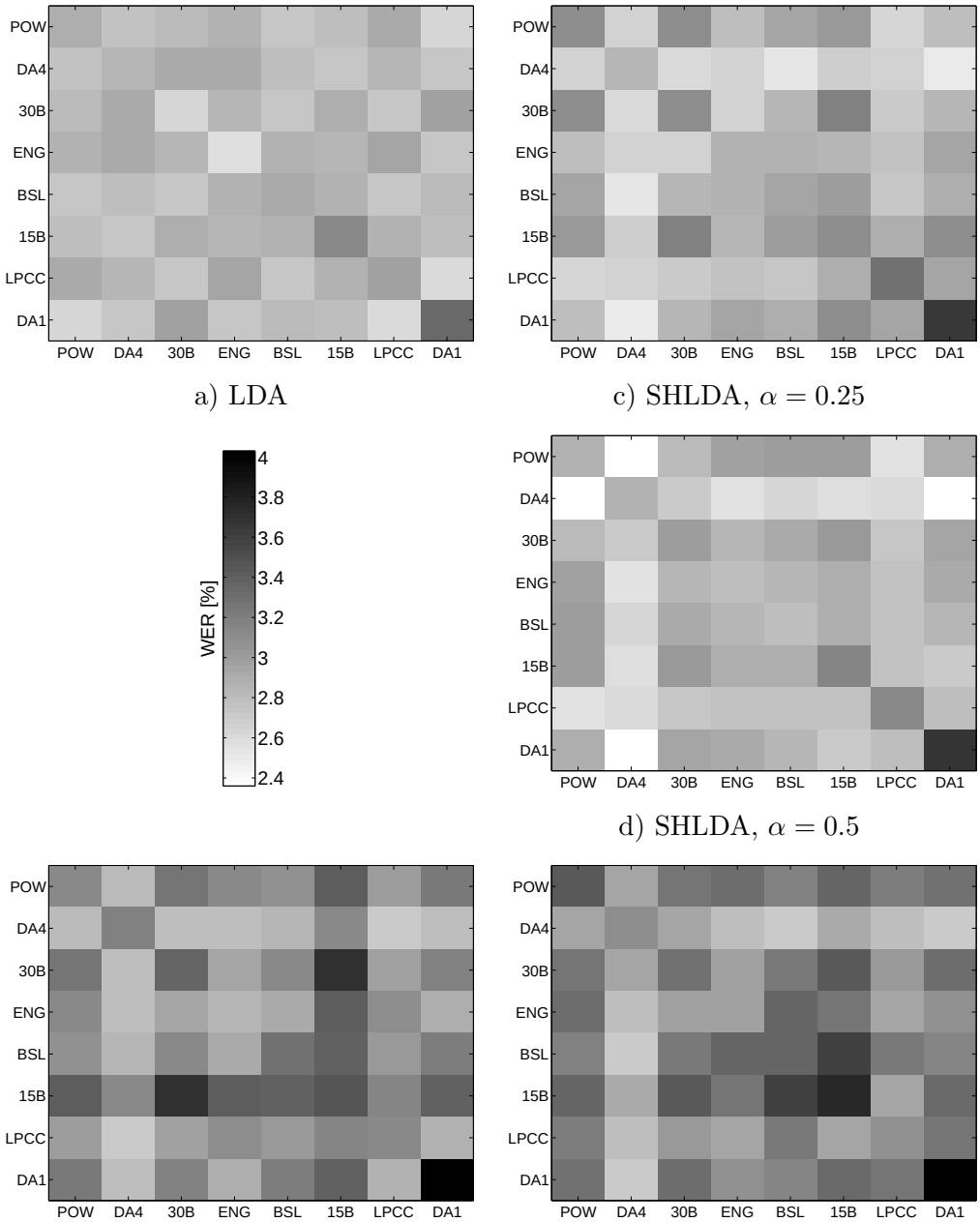
In our experiments, the size of combined feature vectors is always reduced to 45 coefficients. Note that 45 may not be the optimal number of dimensions for a combined feature vector and that optimal number of dimensions can be different for different

	POW	DA4	30B	ENG	BSL	15B	LPCC	DA1
POW	3.12	2.81	3.27	3.13	3.09	3.41	3.01	3.24
DA4		3.19	2.80	2.80	2.83	3.12	2.72	2.80
30B			3.36	2.96	3.12	3.70	2.98	3.19
ENG				2.83	2.91	3.41	3.10	2.90
BSL					3.28	3.40	3.03	3.22
15B						3.46	3.16	3.40
LPCC							3.14	2.88
DA1								4.03
BASE	2.90	2.91	2.99	3.00	3.04	3.14	3.36	3.51

Table 4.13: WER of systems using feature combination based on HLDA. Soft assignment of speech frames to classes is given by Gaussian mixture occupation probabilities.

	POW	DA4	30B	ENG	BSL	15B	LPCC	DA1
POW	2.87	2.38	2.81	2.97	2.99	2.99	2.55	2.90
DA4		2.87	2.70	2.55	2.64	2.59	2.60	**2.36**
30B			3.01	2.83	2.93	3.02	2.75	2.95
ENG				2.78	2.83	2.89	2.76	2.92
BSL					2.78	2.89	2.77	2.85
15B						3.16	2.76	2.71
LPCC							3.13	2.80
DA1								3.68
BASE	2.90	2.91	2.99	3.00	3.04	3.14	3.36	3.51

Table 4.14: WER of systems using feature combination based on SHLDA for $\alpha = 0.5$. Soft assignment of speech frames to classes is given by Gaussian mixture occupation probabilities.



a) LDA

c) SHLDA, $\alpha = 0.25$

d) SHLDA, $\alpha = 0.5$

b) HLDA

e) SHLDA, $\alpha = 0.75$

Figure 4.5: WER of systems using feature combination based on LDA, HLDA and SHLDA. Soft assignment of speech frames to classes is given by Gaussian mixture occupation probabilities.

pairs of base feature streams. For example, BSL features differ from ENG feature vectors only in 3 coefficients (energy vs. C0 and their delta and acceleration coefficients). By removing the redundant coefficients, the dimensionality of concatenated feature will be only 48. In contrast, BSL features differ from LPCC features in all coefficients. To preserve the important complementary information, probably more coefficients should be retained in BSL-LPCC combined feature vectors in comparison to BSL-ENG features. However, as mentioned above, the dimensionality 45 is used for combined features in all experiments for any pair of base features. The search for the optimal dimensionality was not conducted. Therefore, all recognizers have the same number of parameters, which makes their results more comparable.

In our experiments, base feature vectors are augmented with delta and acceleration coefficients before performing feature combination. Another approach, which can appear more reasonable, is to combine only static coefficients first and then augment the combined features with their dynamic coefficients. This is however not possible for our set of base features, since DA4 and DA1 features differ from the other features particularly in the length of window used for the computation of dynamic coefficients. We have shown that combination of features that differ in dynamic coefficients is especially beneficial.

PCA stabilization is used in the experiments based on LDA, HLDA, SHLDA and CHLDA mainly to discard totaly redundant dimensions and dangerous low-variance dimensions in the feature space. Another positive effect of PCA stabilization was found: Coefficients of feature vectors in the *smoothed space* can be sorted by their importance in the PCA sense (by eigen values). This was found to be beneficial for HLDA, where the optimization algorithm described in section 2.5.3 goes iteratively column-by-column through current estimate of HLDA transformation matrix and searches for its better solution. If identity matrix is used as the initial estimate of the transformation, which is the case of our experiments, the first "more important" coefficients in the "smoothed" feature vectors are taken into account in the first steps of the optimization process. This was found to speed up the whole optimization process. In opposite, when PCA stabilization was not used before HLDA, the optimization of HLDA transformation often ended up in a local minimum and did not provide any useful result.

4.8 Conclusions

In this chapter, we have presented methods of feature combination. These methods differ only in the way of postprocessing concatenated feature vectors, which was performed in order to decorrelate combined features and to reduce their dimensionality. The tested postprocessing technique were PCA LDA and HLDA. For the estimation of statistics required by these techniques, only a limited amount of training data was available. In order to overcome the data insufficiency problem when estimating statistics required by HLDA, new techniques were proposed (SHLDA and CHLDA), which use more robust estimation strategies.[12] Other techniques increasing robustness of estimation of statistics, namely PCA stabilization (see section 4.5.1) and the assumption of block diagonal covariance matrices (see section 4.5.2) were also tested.

PCA clearly failed as a postprocessing technique in our experiments. PCA, which is driven only by variances seen in different feature space dimensions, was unable to distinguish the important dimensions (where variance is given mainly by speech) from dimensions where variance is given by noise present in the training data. Mixed results were obtained in experiments based on LDA, which makes wrong assumption of equality of class covariance matrices, and HLDA, where class covariance matrices were estimated poorly. The best results were obtained using SHLDA and CHLDA, which make a compromise between both mentioned problems.

The very best result (WER of 2.36%) was obtained for combination DA4-POW using hard SHLDA with $\alpha = 0.75$ and for combination DA4-DA1 using soft SHLDA with $\alpha = 0.5$. Practically the same result was obtained in chapter 3, where WER of 2.38% was reported for the best system combination using ROVER. However, in the case of feature combination, only two systems were combined compared to eight systems used for the best ROVER combination. The comparison of WERs of individual system combinations is however not very useful. In our experiments, we could see differences of about $\pm 5\%$ relative for combinations of the same system pair using similar combination techniques. In the rest of this section, we will rather focus on general trends seen in the results. Using soft classes (see section 4.6.6), some improvement in performance can be obtained over the case of hard classes at the price of much higher

[12] Usually, there is much more training data available in real situations. However, in our experiments, the task is quite simple. For HLDA we distinguish 180 classes corresponding to HMM states (resp. 540 classes corresponding to mixture components). For continuous speech recognition, there can be thousands of classes corresponding to states of context-dependent phoneme models. The problem of robust estimation of class covariance matrices required by HLDA will be therefore again relevant.

complexity of the training algorithm.

One of objectives of this chapter was to show that complementarity measures proposed in chapter 3 are useful also for selection of systems that are to be combined by a technique different from ROVER. According to these measures, DA4 system was found to be the most complementary to all other systems. Inspecting results obtained using one of the most successful combination techniques, hard SHLDA with $\alpha = 0.75$ (see table 4.10 and figure 4.3e), combination with DA4 features turns out to be the most beneficial. For the cases where DA4 features participate in the combination, the WERs of combined systems ranges from 2.36% to 2.65%. All these combined systems therefore outperform the best individual system POW with WER of 2.90%. Similarly, LPCC were observed as the second set of features most useful for combination, which is again in agreement with findings based on complementarity measures. Similar trends in results were observed also for other successful combination techniques such as soft SHLDA and CHLDA.

Chapter 5

Combination based on Multi-stream HMM

5.1 Introduction

Two sets of experiments were described in the previous chapters, where recognition systems were combined at two very opposite positions. System combination experiments were carried out in order to test how the complementarity measures proposed in chapter 3 reflect the suitability of different systems for their combination. The first set of experiments was described in chapter 3, where systems were combined at the very end of the recognition chain using ROVER merging individual recognized word sequences. In contrast, the systems were combined at the very beginning (feature level) in experiments described in chapter 4. In both cases, a correlation between complementarity measures and the actual WER of a combined system was shown. In this section, additional experiments are described where systems are combined "in between" using Multi-stream HMM approach [65]. Again, the recognition performances for different combinations are compared with corresponding complementarity measures and a correlation between them is reported.

5.2 Multi-stream Hidden Markov Models

Multi-stream HMM allows to combine different feature streams at the level of modeling state-dependent feature vector distributions (state output distributions). Let $\mathbf{O}^1, \mathbf{O}^2, \ldots, \mathbf{O}^R$ be R feature streams that are synchronized in time (all streams have the same number of feature vectors, where corresponding vectors across streams rep-

resent the same part of speech). In the case of standard HMM, which was described in section 2.3, state distributions $b_s(\mathbf{o})$ are modeled using Gaussian mixture model (GMM) according to formula 2.4. In the case of Multi-stream HMM, R Gaussian mixture models, $b_s^1(\mathbf{o}^1), b_s^2(\mathbf{o}^2), \ldots, b_s^R(\mathbf{o}^R)$, are associated with each state to independently model the distribution of each individual stream. The state output distribution is then modeled as:

$$b_s(\mathbf{o}^1, \mathbf{o}^2, \ldots, \mathbf{o}^R) = \prod_{r=1}^{R} b_s^r(\mathbf{o}^r)^{\eta^r}, \tag{5.1}$$

where η^r is the weight of r^{th} stream. Setting the weights of all streams equal to one, equation 5.1 becomes ordinary evaluation of joint distribution over all the streams under the assumption that feature vectors from different streams are statistically independent. However, in our experiments, weights are set to sum up to one ($\sum_{r=1}^{R} \eta^r = 1$). In such case log likelihood $\log b_s(\mathbf{o}^1, \mathbf{o}^2, \ldots, \mathbf{o}^R)$ is calculated as a weighted average of log likelihoods that are produced by individual experts (stream-dependent Gaussian mixture models). With the exception of stream weight η^r, all parameters of Multi-stream HMM can be estimated in the ML framework (see section 2.3). The formulae for re-estimation of parameters of Multi-stream HMM can be found in [65].

5.3 Experimental Setup

The setup of experiments with Multi-stream HMMs was designed to make results comparable with those obtained in the previous experiments with ROVER (chapter 3) and with feature combination (chapter 4). Both training and testing is performed on the same clean and noisy speech data (extracted from TI-Digits database) that were used in the previous experiments (see section 3.4.5 for details). The performance of each recognition system is again evaluated on *seen conditions test data* (see section 3.4.5). The recognition systems (whole word continuous HMMs) follow the description from section 3.4.5 with the exception that always pair or triplet of feature streams form the input of each recognizer and information from the two or three streams is combined in Multi-stream HMM fashion. Equal weights η^r ($\frac{1}{2}$ or $\frac{1}{3}$) are given to all (two or three) streams. In the following experiments, two different sets of feature streams are combined. The description of the first feature stream set, which was previously referred

to as *different features*[1] can be found in section 3.4.5. The description of the second set, which was referred to as *missing bands MFCC*, can be found in section 3.5.5.

5.4 Results

In the first experiment, all possible pairs of feature streams taken from set of *different features* are combined using Multi-stream HMM approach. WERs of all such systems are shown in table 5.1. In our case, where stream weight η^r is 0.5 for both streams, Multi-stream HMM using the same features in both streams equals to ordinary (single-stream) HMM system. Values in the table diagonal are therefore WERs of systems using individual feature streams that were already reported in the last column of table 3.2. The best result (2.66% WER) was obtained for POW-DA4 combination, which corresponds to statistically significant 8.3% relative improvement over the best individual system POW. However, we will be again more interested in general trends in the results, which can be better seen in figure 5.1, which is the graphical representation of table 5.1. According to our complementarity measures, DA4 and LPCC were found to be two features most complementary to any other features from the set. The bright fields in figure 5.1 indicate that Multi-stream HMM combinations where DA4 or LPCC feature stream participate are generally very beneficial. Comparing figure 5.1 with figures 3.3 and 3.4 showing LBWER and DWER measures for pairs of systems using *different features*, we observe similar patterns[2] suggesting that higher complementarity estimated using our measures generally indicate better recognition performance of corresponding Multi-stream HMM system.

Similar correspondence can be observed between figure 5.2 showing performance of Multi-stream HMM on *missing bands MFCC* task and figure 3.6 showing LBWER for corresponding system pairs. In this task, the goal is to improve over BSL system (3.04% WER) "seeing" information from all Mel filter bank bands (see section 3.5.5). The best performance is obtained for combination of features M13-15 and M15-17. This combination reaches the goal with 2.86% WER, however, the improvement over BSL system is not statistically significant.

In the next experiment, Multi-stream combinations of three feature streams are examined. All such possible combinations of *different features* and *missing bands*

[1] Features W15 and W35, which were introduced in section 3.5.5, cannot be used in Multi-stream HMM experiments, as they are not synchronized in time with the other features.

[2] Ignore values on the diagonal, which must be principally much higher than values out of diagonal for the complementarity measures.

MFCC streams are shown in figures 5.3a and 5.3b, respectively. Each dot corresponds to one particular combination of three different streams. Axis Y represents WER of combined system, while axis X represents average WER of three systems using corresponding individual features. For the set of *different features* (figure 5.3a), no significant correlation can be observed between these two quantities, which means that lower WER obtained for individual features does not imply lower WER of combined system. Some correlation is observed for the case of *missing bands MFCC* streams, however, even stronger correlation will be shown between WER of combined system and our complementarity measure. Note that figures 5.3a and 5.3b can be compared with figures 3.8a and 3.8b showing similar plots for ROVER based combinations[3].

Figures 5.3a and 5.3b shows correlation between WER of combined system and ALBWER, which is the complementarity measure recognized as the most appropriate for combination of few systems (see section 3.6). For both *different features* and *missing bands MFCC*, much higher correlation is observed in comparison to figures 5.3a and 5.3b, which suggests that the complementarity measure can be used with advantage also for selection of features for Multi-stream based combination.

5.5 Discussion and conclusions

In this chapter, we have presented Multi-stream Hidden Markov Models as an alternative method combining complementary information from different feature streams. The main purpose of the experiments described in this chapter was to once more verify that the complementarity measures proposed in chapter 3 can be advantageously used by various combination methods to select systems suitable for combination.

In the first experiment, pairs of feature streams from the set of *different features* were combined using Multi-stream HMM. Consistently with the complementarity measures, DA4 an LPCC were again recognized as the features best suitable for combination with any other features. For example, for combinations of two streams where DA4 features participate, WER ranges between 2.66% and 2.81%. All these systems perform better than the best individual system POW with 2.90% WER, however, the performances are significantly worse in comparison to the best feature combination techniques described in chapter 4. For example, hard SHLDA with $\alpha = 0.75$ ranges

[3]Only 9 streams of *different features* were combined in Multi-stream based experiments compared to 11 systems used in ROVER based experiments. Therefore, compared to figure 3.8a, the fewer dots in figure 5.3a corresponds to the fewer possible combinations.

between 2.36% and 2.65% for combinations where DA4 features participate. *Average combining system WER* (ACSWER) was introduced in section 4.6.3 to allow for objective comparison of different feature combination methods. Given a combination method, WERs of all systems combining pairs of different features are averaged[4] to obtain ACSWER. Multi-stream HMM combination results in 2.87% ACSWER, which is worse than 2.77% ACSWER obtained for the best feature combination system: soft SHLDA with $\alpha = 0.5$.

For all feature combination techniques that were described in chapter 4, it is not trivial to combine larger number of feature streams because of the increasing size of required statistics. In the case of Multi-stream HMM combination, however, more than two feature streams can be easily combined to improve recognition performance for the price of an additional computation cost. While 2.66% WER was obtained for the best combination of two streams of *different features*, the best combination of three streams results in 2.55% WER . Note that the same recognition performance was obtained for the best ROVER combination of three systems.

In the experiments, where *missing bands MFCC* features were combined, only insignificant improvement over BSL system was obtained with the best Multi-stream HMM systems (2.86% and 2.82% WER for two and three stream combinations, respectively). Note that the best ROVER system for this task, which combines only three different systems, performs already significantly better (2.62% WER) than BSL system (3.04% WER). Therefore, it seems that the combination of feature streams, where complementarity is introduced by hiding part of information in each stream, is beneficial only for techniques performing a combination at a higher level.

Tuning stream weights η^r is another possibility where the recognition performance could be improved. For simplicity, fixed weight $\eta^r = 0.5$ is used for all streams in our experiments. However, the usual practice is to tune the weights on a held-out part of training data for the best recognition accuracy.

[4]NOE features were not used in feature combination experiments. Therefore, WER of systems where NOE features participate in combination are not included to the average even for Multi-stream HMM combination.

	POW	DA4	30B	ENG	BSL	15B	LPCC	DA1	NOE
POW	2.90	2.66	2.90	2.84	3.00	2.70	2.84	3.10	3.04
DA4		2.91	2.80	2.69	2.68	2.80	2.67	2.70	2.81
30B			2.99	2.93	3.08	3.09	2.70	3.15	3.08
ENG				3.00	2.84	2.87	2.85	2.85	2.96
BSL					3.04	2.84	2.76	3.12	3.00
15B						3.14	2.85	3.19	2.98
LPCC							3.36	2.99	3.12
DA1								3.51	3.40
NOE									3.58

Table 5.1: WER of multi-stream systems combining *different features*.

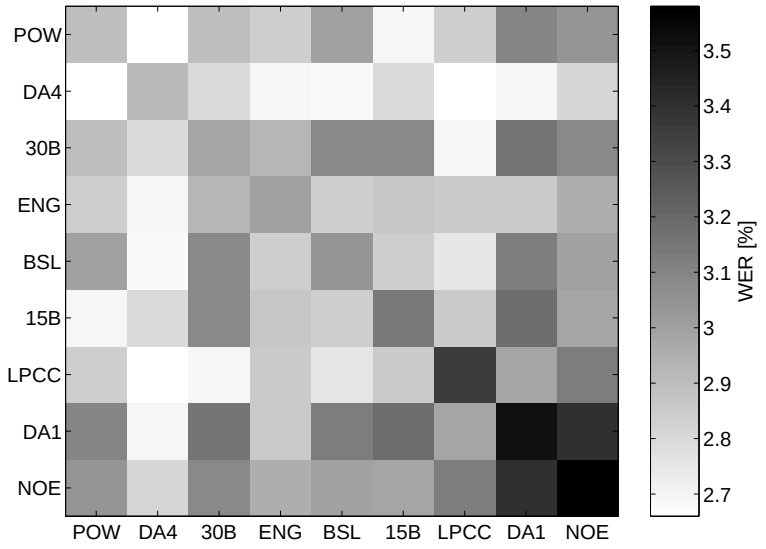


Figure 5.1: WER of multi-stream systems combining *different features*.

	$M_{1:3}$	$M_{3:5}$	$M_{5:7}$	$M_{7:9}$	$M_{9:11}$	$M_{11:13}$	$M_{13:15}$	$M_{15:17}$	$M_{17:19}$	$M_{19:21}$	$M_{21:23}$
M1-3	3.59	3.74	3.35	3.49	3.23	3.21	3.12	3.08	3.08	3.31	3.26
M3-5		4.41	3.78	3.51	3.31	3.24	3.06	3.29	3.27	3.40	3.29
M5-7			3.92	3.67	3.41	3.21	3.18	3.28	3.32	3.39	3.44
M7-9				3.93	3.38	3.25	3.15	3.20	3.23	3.40	3.37
M9-11					3.40	3.17	3.00	2.98	2.99	3.18	3.11
M11-13						3.30	2.94	3.03	3.00	3.06	3.08
M13-15							3.09	2.86	2.89	3.05	3.03
M15-17								3.25	3.02	3.12	3.27
M17-19									3.23	3.21	3.24
M19-21										3.71	3.51
M21-23											3.59

Table 5.2: WER of multi-stream systems combining *missing bands MFCC*.

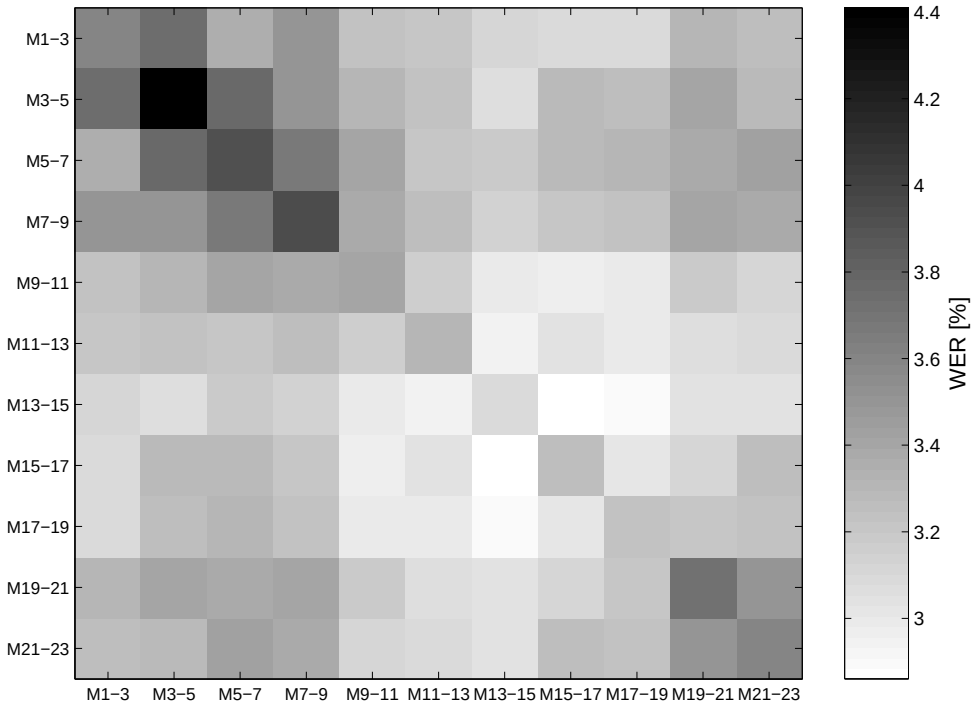
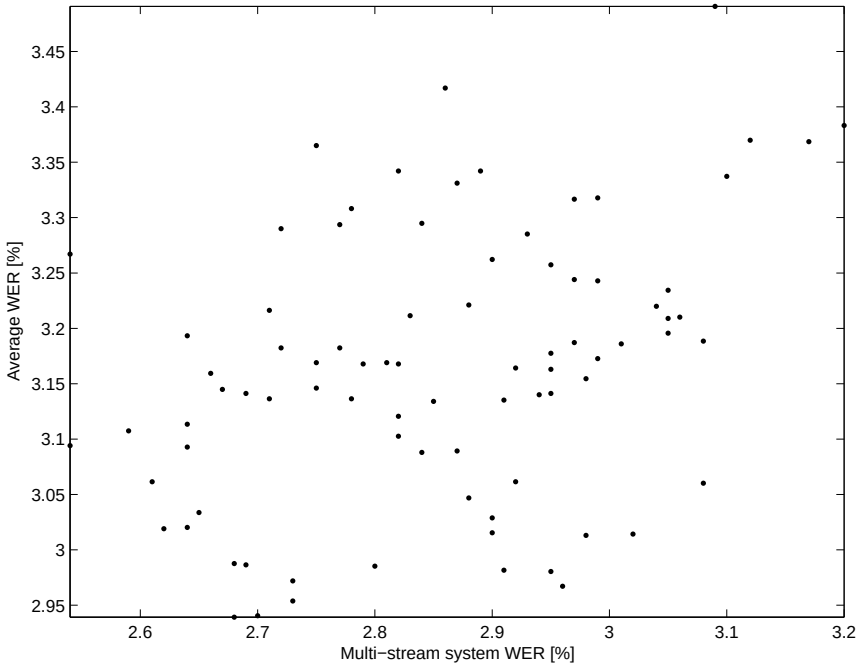
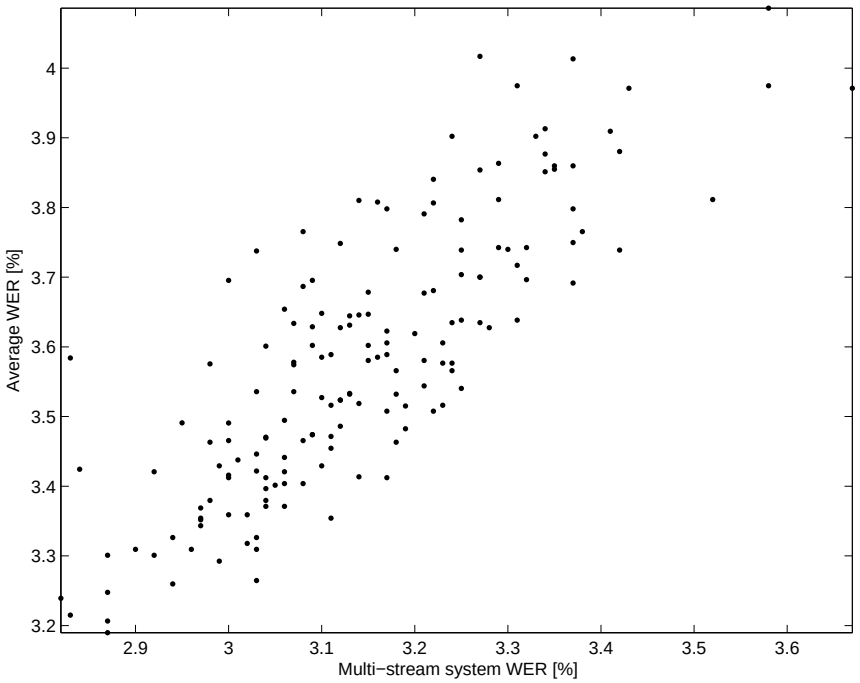


Figure 5.2: WER of multi-stream systems combining *missing bands MFCC*.

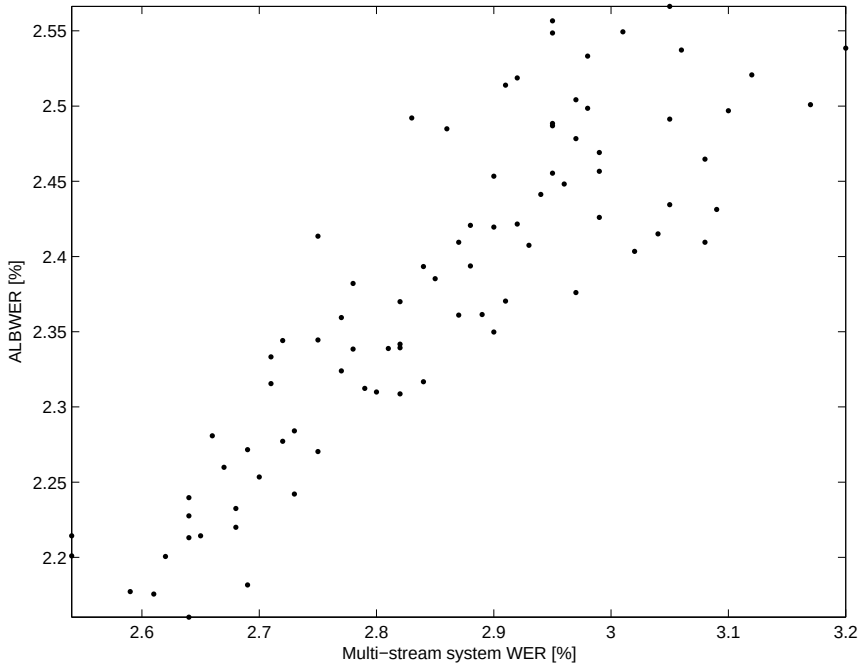


a) Different features

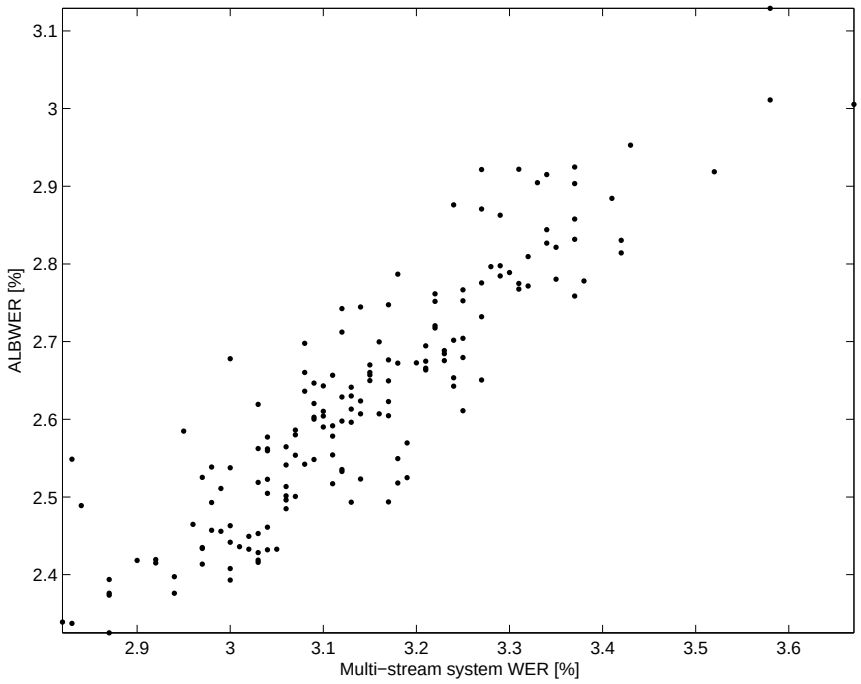


b) Missing band MFCC

Figure 5.3: Correlation between average WER and Multi-stream system WER.



a) Different features



b) Missing band MFCC

Figure 5.4: Correlation between ALBWER and Multi-stream system WER.

Chapter 6

Conclusion and future work

The combination of different speech recognition systems and techniques became a standard approach to improve recognition performance. Different combination techniques have been proposed in the past trying to utilize various sources of complementary information in speech. To benefit from the combination, first, the sources of complementary information must be identified. Complementary techniques have been typically searched by performing combinations of many different systems and their configurations. An apparatus allowing for more direct complementarity identification has been missing. The main aim of this work is to provide techniques allowing to measure such complementarity of recognition systems, which can be helpful in selection of systems or techniques whose combination is the most beneficial. The proposed measures of complementarity are based on comparison of individual recognition system outputs. Instead of exhaustive evaluation of all possible recognition system combinations, each individual system using a particular complementary technique is evaluated only once. The proposed complementarity measures can be efficiently calculated for any selected subset of individual systems to provide an estimate of suitability of these systems for combination.

Applicability of the complementarity measures for the system selection was verified in three sets of experiments using three different combination strategies. In the first set, output word sequences recognized by different systems were combined using ROVER. In the second set, complementary information was combined at the level of feature streams. Multi-stream HMM based combination was used for the final set of experiments. High correlation between the complementarity measures and the actual accuracies of combined systems was reported for all three combination strategies.

To allow for comparison of different combination strategies, the individual systems to be combined were taken from the set of *systems using different features*, where individual systems differed only in the feature extraction method. The best recognition performances, obtained with ROVER combining three and eight individual systems, were 2.55% and 2.38% WER, respectively, which corresponds to statistically significant improvement over the best individual system POW with 2.90% WER. Disadvantage of ROVER combination is its high computational complexity. The whole recognition process must be carried out once for each individual recognition system before the output word sequences can be combined using ROVER.

On the contrary, almost no additional computation cost is required during recognition in case of feature stream combination that was used in the second set of experiments. Here, corresponding feature vectors coming from different streams were concatenated and thereafter postprocessed in order to decorrelate the features and to reduce their dimensionality. Different postprocessing methods were tested, namely PCA, LDA, HLDA and newly proposed Smoothed HLDA (SHLDA) and Clustered HLDA (CHLDA), which can be seen as another original contribution of this work. Mixed results were obtained in experiments based on LDA, which is making wrong assumption about equality of class covariance matrices, and HLDA, where required class covariance matrices were estimated poorly on the limited amount of training data. The best results were obtained using SHLDA and CHLDA, where robustness of statistics estimation is increased. Using these feature combination methods, it is not trivial to combine more than few different feature streams, as the size of statistics required by postprocessing becomes prohibitive for highly dimensional concatenated feature vectors. Only pairs of feature streams were combined in our experiments. Features that combine the best with any other features from our set are DA4. For the cases where DA4 features participate in the combination, performance of hard SHLDA with $\alpha = 0.75$, which is one of the best feature combination methods, ranges from 2.36% to 2.65%. In all these cases, the best individual system POW with 2.90% WER is significantly outperformed. WER of 2.36% was the best result obtained in feature combination experiments at all. This result is virtually the same as the best result obtained with ROVER, where however eight systems were combined compared to only a pair of combined feature streams.

In contrast to ROVER, where each individual system is trained and evaluated only once and ROVER combination can be quickly tested for any subset of individual systems, in the case of feature combination, the whole recognition system must be

trained and evaluated for each tested pair (or set) of feature streams. Therefore, the application of complementarity measures allowing to quickly estimate which systems will combine well is even more beneficial.

In the last series of experiments, Multi-stream HMM approach was used to combine pairs of different feature streams. In these experiments, WER ranges from 2.66% to 2.81%, for systems where DA4 features participate in combination. All these systems again outperform the best individual system, however, the improvement is not statistically significant. Unlike the feature combination, Multi-stream HMM can be easily used to combine more than two feature streams. The best combination of three streams with 2.55% WER performs already equally well as the best ROVER combination of three systems.

Another set of feature streams was used in experiments with ROVER and Multi-stream HMM combination, where individual feature streams were derived from BSL features by hiding information about energies from three consecutive Mel filter bank bands. In each feature stream, different bands were leaved out. WER of the systems using individual features ranges from 3.09% to 4.42%, which is always worse than 3.04% WER of BSL system utilizing the information from all the bands. However, the best ROVER combination of eight such individual system results in 2.51% WER and significantly outperforms BSL system. This is an interesting example: hiding certain information at lower (feature) level in the recognition process and its recombination at higher (output sequence) level can result in higher recognition performance. In the case of Multi-stream HMM approach, where combination is performed on somewhat lower level in comparison to ROVER, the improvement over BSL system obtained in a similar experiment was not statistically significant.

We are aware that, during the work on this thesis, we just familiarized ourselves with the interesting problem of system combination and that there are many open issues in the theoretical development as well as in the experimental work. To cite just the ones that could be investigated in short- to medium-time:

- In ROVER-based measures of complementarity of systems and in the ROVER combination itself, we consider all output text tokens having equal confidence. Considerable work has however been done on computation of confidence measures [29, 63, 19]. If properly estimated, these measures should further improve the complementarity measures as well as the system combination.

- The complementarity measures are based on "hard" strings of recognized tokens. This approach could be naturally extended to recognition lattices.
- In this work, one-to-one matching of recognized tokens was used to derive complementarity measures. Ideally, these measures as well as the combination itself should use the information about context (probably not for digits recognition, but rather for phoneme-based approaches).
- For newly developed Smoothed HLDA (SHLDA), an automatic approach to determine the smoothing factor should be proposed, ideally based on the amount of available training data, number of classes, etc.
- For newly developed Clustered HLDA (CHLDA), the only setup of clusters used in our experiments is speech/silence. Finer clustering, based on hand-crafted sound classes and/or automatic clustering (see section 4.5.5) should be tested.
- Finally, the proposed techniques need to be evaluated on different speech recognition tasks, especially on a standard LVCSR experiment.

To conclude, I hope this thesis has brought some new insights into the problematic of system combination in speech recognition. I will be happy to work (alone or in team with colleagues and/or students) on the problems that only now — at the end of this thesis — begin to have clearer shape and that actually have arisen based on the theoretical and practical work described in the previous chapters. I also hope that this thesis will find readership in the speech research community and I am looking forward to reactions of colleagues that could bring some new light on the described problems.

Appendix A

Matlab implementation of HLDA

In this section, an implementation of routines performing Heteroscedastic Linear discriminant analysis is presented. The routines, which are implemented as a Matlab functions [39], were used in our feature combination experiments (see chapter 4) to decorrelate speech features and to reduce their dimensionality. Two functions are presented here: *hlda* and *shlda*.

Function *hlda* allows to derive HLDA transformation matrix from full covariance statistics of classes. The description of iterative optimization algorithm that is used for its implementation can be found in section 2.5.3. More detailed description together with derivation of formulae can be found in [23, 22]. The instructions to use *hlda* function and the description of its input/output parameters can be found directly in the source code.

Function *shlda* is an implementation of Smoothed HLDA, which was introduced in section 4.5.4. The function can be used in the same manner as function *hlda*. It comes with only one additional input parameter *ALPHA*, which is the factor of smoothing class covariance matrices. Clustered HLDA was also tested in our experiments as a modification of HLDA. Source code of this method is not included here, however, its implementation is straightforward.

```
function A = hlda(A, p, SIGMA_g, SIGMA, gamma, iters, initers)
%HLDA Heteroscedastic linear discriminant analysis
%   NewA = hlda(A, P, SIGMA_g, SIGMA, GAMMA, ITERS, INITTERS) performs
%   an iterative optimization of HLDA transformation matrix [2] and
%   returns its new estimate, NewA.
%
%   A        - initial guess of transform matrix
%              (e.g. eye(size(SIGMA_g)) or LDA transformation matrix)
%   P        - number of useful (wanted) dimensions
%   SIGMA_g  - global covariance matrix
%   SIGMA    - cell array of class covariance matrices
%   GAMMA    - vector of class occupancy counts (samples per class)
%   ITERS    - number of optimization iterations
%   INITERS  - number of optimization inner loop iterations (default 10)
%
%   Result is the maximum likelihood transformation for diagonal
%   covariance modeling of data in projected space (i.e., setting P
%   to the original data dimensionality results in MLLT [3]). Resulting
%   transformation is normalized to be volume preserving (i.e., full
%   covariance modeling in both the original and the rotated space lead to
%   the same likelihood of data). After each iteration of the optimization
%   process, likelihood of the data is reported for model where each class
%   is modeled in rotated space by Gaussian with diagonal covariance
%   matrix and where parameters of Gaussians are tied for nuisance
%   (unwanted) dimensions [1,2]. This likelihood is guaranteed to increase
%   (or at least not to decrease) in each iteration. To perform
%   dimensionality reduction, project the original data to only first P
%   columns of resulting transformation matrix, NewA(:,1:P)
%
%     See also COV
%
%   References
%
%   [1]  N. Kumar, Investigation of Silicon-Auditory Models and
%        Generalization of Linear Discriminant Analysis for Improved
%        Speech Recognition, Ph.D. Thesis, John Hopkins University,
%        Baltimore, USA, 1997
%
%   [2]  M.J.F. Gales, Semi-tied covariance matrices for hidden Markov
%        Models", IEEE Transaction Speech and Audio Processing,
%        vol. 7, pp. 272-281, 1999.
%
%   [3]  R. Gopinath, Maximum likelihood modeling with Gaussian
%        distributions for classification. In Proc. ICASSP,
%        vol. II, pp 661-664, Seattle, USA, May, 1998.
%
```

```
if nargin < 7
  initers = 10;
end

M = length(SIGMA);    % number of classes
d = size(SIGMA_g, 1); % original feature space dimension
tau = sum(gamma);     % total nuber of samples

disp(' ')
disp('Iteration    Likelihood')
disp('-------------------------')

for iter =1:iters,
  Q = tau * log(det(A')^2)  - tau * d * (log(2*pi)+1);
  for i =1:d,
    if i <= p
      G = zeros(d, d);
      for m = 1:M,
        sigma_i = A(:,i)' * SIGMA{m} * A(:,i);
        G = G + gamma(m) / sigma_i * SIGMA{m};
        Q = Q - gamma(m) * log(sigma_i);
      end
    else
      sigma_i = A(:,i)' * SIGMA_g * A(:,i);
      G = tau / sigma_i * SIGMA_g;
      Q = Q - tau * log(sigma_i);
    end
    invG{i} = inv(G);
  end

  Q = Q / 2;
  disp(sprintf('%5d        %.6g', iter-1, Q));

  for initer =1:initers,
    for i =1:d,
      C = (inv(A')*det(A'))';
      ci_invG = C(i,:) * invG{i};
      A(:,i) = (ci_invG * sqrt(tau / (ci_invG * C(i,:)')))';
    end
  end
end

% Normalize transformation to be volume preserving ( det(A) = 1 )
A=A/(det(A)^(1/d));
```

```
function A = shlda(A, P, ALPHA, SIGMA_g, SIGMA, GAMMA, ITERS, INITERS)
%SHLDA Smoothed heteroscedastic linear discriminant analysis
%   NewA = shlda(A, P, ALPHA, SIGMA_g, SIGMA, GAMMA, ITERS, INITTERS)
%   performs an iterative optimization of SHLDA transformation matrix
%   and returns its new estimate, NewA.
%
%   A        - initial guess of transformation matrix
%              (e.g. eye(size(SIGMA_g)) or LDA transformation matrix)
%   P        - number of useful (wanted) dimensions
%   ALPHA    - smoothing factor (value between 0.0 and 1.0)
%   SIGMA_g  - global covariance matrix
%   SIGMA    - cell array of class covariance matrices
%   GAMMA    - vector of class occupancy counts (samples per class)
%   ITERS    - number of optimization iterations
%   INITERS  - number of optimization inner loop iterations (default 10)
%
%   For ALPHA equal to 1.0, resulting trasformation is identical to that
%   obtained with function HLDA. For ALPHA equal to 0.0, resulting
%   trasformation correspons to LDA solution. For values between 0.0 and
%   1.0, resulting trasformation can be seen as a compromise between  LDA,
%   which is making wrong assumption of equality of class covariance
%   matrices, and HLDA, where required class covariance matrices may be
%   poorly estimated if the amount of training data is limited.
%
%     See also HLDA
%
%   References
%
%   [1]  L. Burget, Combination of Speech Features Using Smoothed
%        Heteroscedastic Linear Discriminant Analysis, In Proc. ICSLP,
%        Jeju Island, Korea, October 2004, accepted.

if nargin < 8,  INITERS = 10; end

% Estimation of within-class covariance matrix
WC = zeros(size(SIGMA_g));
for i=1:length(GAMMA)
  WC = WC + SIGMA{i} * GAMMA(i);
end
WC = WC / sum(GAMMA);

% Covariance matrices smoothing
for i=1:length(GAMMA)
  SIGMA{i} = SIGMA{i} * ALPHA + WC * (1-ALPHA);
end

A = hlda(A, P, SIGMA_g, SIGMA, GAMMA, ITERS, INITERS);
```

Appendix B

Description of developed software

During elaboration of the thesis, software tools have been developed that could be of interest for other researchers concerned about the speech recognition or general speech processing. The source code and more detailed description of the tools can be found at our web site (http://www.fit.vutbr.cz/speech/sw.html). Here, the most important tools are listed and their main features are described. Input and output data files (features, labels, HMM parameters) used by these tools are in well known HTK format. Certain tools actually serve as a substitution for HTK tools [65] coming with some new functionality. For the tools dealing with Hidden Makrov Models, only the usual choice of Gaussian mixture model with diagonal covariance matrices is supported for modeling state distributions. In addition, all these tools are able to perform linear transformation of input features with the possibility of changing feature dimensionality. The model is then applied on projected features. An interesting feature could be also the possibility to use different linear transformations for different gaussian components. This allows to implement advanced methods of covariance matrix modeling such as: STC (MLLT) [25, 23] or EMMLT [58].

SVite is a clone of HVite tool from HTK toolkit. SVite is an implementation of one-best Viterbi decoder using Beam-search algorithm. In contrast to HVite, SVite combines pronunciation dictionary and grammar description into single file containing description of expanded recognition network. This allows to perform various optimizations of the recognition network by an external tool before running the recognition. Instead of reading speech features and computing likelihoods using Gaussian mixture model, state output probabilities can be directly read from files, which, for example, allows to easily build a hybrid ANN-HMM

systems. Another useful feature of the tool is its ability to automatically decrease the pruning threshold and re-run the recognition of utterances, for which the threshold was too tight and the decoding process failed. In near future, we plan to implement N-best decoding and lattice generation. Possible extensions of the tool to make it usable for LVCSR applications are also of our interest.

SERest is a clone of HERest tool from HTK toolkit, which is an implementation of Baum-Welch algorithm for HMM training. An interesting extension with respect to HERest is that transcriptions of utterances do not have to be linear strings of phonemes. Utterance can be described by more complicated (recognition) network allowing to use multiple pronunciation variants and various filler models during training. In addition, a single recognition network can be used to describe all utterances, which, for example, allows for unsupervised model retraining (adaptation) on test data. Viterbi training is also implemented allowing to speed up the whole training process. Moreover, the tool offers the possibility of using information about timing of labels, which allows to use the tool also as a replacement for HRest and HInit tools from HTK toolkit.

SXStat is very similar to SERest. However, besides standard HMM parameters, it allows to re-estimate also the linear transformation mentioned above. This tool is an implementation of algorithm described in section 4.4.2. Using old model parameters and with help of Forward-Backward algorithm, the tool collects original feature space full covariance statistic for all Gaussian components. Then it calls an external routine to estimate linear transformation. In our experiments, matlab routines from appendix A are used for this purpose. Using the new transformation, SXStat updates all HMM parameters to model features in newly projected space. Note, that SXStat also allows to train models based on advanced methods of covariance matrix modeling mentioned above (STC, EMLLT).

SRover is an implementation of ROVER technique, which is typically used to combine word sequences recognized by different speech recognition systems (see section 3.3). The input to this tool is a set of HTK Master Label Files (MLF) representing transcriptions recognized by individual systems. The output is MLF representing combined transcription. The combination can be based on the simple majority voting (see section 3.3), however, various ways of utilizing possible information about word confidences are also implemented. The

key problem for ROVER technique is the alignment of individual recognized hypotheses. SRover implements two different approaches to perform this alignment. The first approach, which was described in section 3.3 completely ignores information about timing of words and performs the alignment to only (approximately) minimize Levenstein distance between individual word hypotheses. In the second approach, only the word timing is important. This *time mediated alignment* is purely driven by amount of time overlap of words from different hypotheses. Time mediated alignment turns out to be more effective in cases where all combined hypotheses have very high WER. Note that SRover allows to give different importance to individual input transcriptions in order to finely control the alignment. SRover can be also instructed not to perform the voting and to output only the aligned hypotheses. Then, for example, it is simple to write a script allowing for scoring recognition results (similar to HEResults HTK tool). Our scripts counting dependent and simultaneous errors (see section 3.4) are indeed based on this option.

Bibliography

[1] A. Adami, L. Burget, S. Dupont, H. Garudadri, H. Hermansky F. Grezl, P. Jain, S. Kajarekar, N. Morgan, and S. Sivadas. Qualcomm-ICSI-OGI features for ASR. In *Proc. International Conference on Spoken Language Processing*, Denver, Colorado, USA, September 2002. ISCA.

[2] S. Axelrod, V. Goel, R. A. Gopinath, P. A. Olsen, and K. Visweswariah. Subspace constrained Gaussian mixture models for speech recognition. *IEEE Transactions on Speech and Audio Processing*, September 2003, submited.

[3] S. Axelrod, R. Gopinath, and P. Olsen. Modeling with a subspace constraint on inverse covariances. In *Proc. International Conference on Spoken Language Processing*, Denver, Colorado, USA, September 2002. ISCA.

[4] L. R. Bahl, P. V. de Souza, P. S. Gopalkrishnan, D. Nahamoo, and M. A. Picheny. Context dependent modelling of phones in continuous speech using decision trees. In *DARPA Speech and Natural Language Processing Workshop*, pages 264–270, 1991.

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

[6] C. Benitez, L. Burget, B. Chen, S. Dupont, H. Hermansky H. Garudadri, P. Jain, S. Kajarekar, and S. Sivadas. Robust ASR front-end using spectral-based and discriminant features: experiments on the Aurora tasks. In *Proc. Eurospeech*, Aalborg, Denmark, September 2001. ISCA.

[7] J. Bilmes. Buried Markov models for speech recognition. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, Phoenix, Arizona, USA, March 1999.

[8] H. Bourlard, S. Dupont, and C. Ris. Multi-stream speech recognition. Technical Report IDIAP-RR 96-07, Dalle Molle Institute for Perceptive Artificial Intelligence, Martigny, Switzerland, December 1996.

[9] H. Bourlard and N. Morgan. *Connectionist Speech Recognition - A Hybrid Approach*. Kluwer Academic Publishers, 1994.

[10] 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.

[11] L. Burget, P. Motlíček, F. Grézl, and P. Jain. Distributed speech recognition. *Radioengineering*, 2002(4):12–16, 2002.

[12] L. Burget and J. Černocký. Recognition speech of with non-random attributes. In *Proc. International conference on Text Speech and Dialogue*, České Budějovice, Czech Republic, September 2003. Springer.

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

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

[15] 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.

[16] R. Duda and P. Hart. *Pattern Classification and Scene Analysis*. John Wiley & Sons, New York, 1973.

[17] D.P.W. Ellis and J.A. Bilmes. Using mutual information to design feature combinations. In *Proc. International Conference on Spoken Language Processing*, volume III, pages 79–82. ISCA, 2000.

[18] D.W.P. Ellis, R. Singh, and S. Sivadas. Tandem acoustic modeling in large-vocabulary recognition. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, Salt Lake City, Utah, USA, May 2001.

[19] G. Evermann and P.C. Woodland. Posterior probability decoding, confidence estimation and system combination. In *Speech Transcription Workshop*, College Park, 2000.

[20] J. G. Fiscus. A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (ROVER). In *IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 347–354, 1997.

[21] K. Fukunaga. *Introduction to statistical pattern recognition.* Academic press, Boston, USA, 2 edition, 1990.

[22] M.J.F. Gales. Maximum likelihood multiple projection schemes for hidden Markov models. Technical Report CUED/F-INFENG/TR.365, Cambridge University, UK, October 1999.

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

[24] B. Gold and N. Morgan. *Speech and Audio Signal Processing.* New York, 1999.

[25] R. Gopinath. Maximum likelihood modeling with Gaussian distributions for classification. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume II, pages 661–664, Seattle, Washington, USA, May 1998.

[26] F. Grézl, L. Burget, P. Jain, and J. Černocký. Improving TRAPS features using LDA. In *Proc. International Czech-Slovak Scientific Conference Radioelektronika*, Bratislava, Slovak Republic, May 2002. FEI STUBA.

[27] T. Hain, P.C. Woodland, T.R. Niesler, and E.W.D. Whittaker. The 1998 HTK system for transcription of conversational telephone speech. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 57–60, Phoenix, Arizona, USA, March 1999.

[28] D. Halberstadt and J. Glass. Heterogeneous measurements and multiple classifiers for speech recognition. In *Proc. International Conference on Spoken Language Processing*, pages 995–998, Sydney, Australia, November 1998. ISCA.

[29] T. J. Hazen, S. Seneff, and J. Polifroni. Recognition confidence scoring and its use in speech understanding systems. *Computer Speech & Language*, 16:49–67, 2002.

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

[31] H. Hermansky, D. Ellis, and S. Sharma. Tandem connectionist feature extraction for conventional HMM systems. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, Istanbul, Turkey, June 2000.

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

[33] H. Hermansky and N. Morgan. Rasta processing of speech. *IEEE Transactions on Speech and Audio Processing*, 2(4):578–589, 1994.

[34] H. Hermansky and S. Sharma. TRAPS - classifiers of temporal patterns. In *Proc. International Conference on Spoken Language Processing*, pages 1003–1006, Sydney, Australia, November 1998. ISCA.

[35] H. G. Hirsch and D. Pearce. The AURORA experimental framework for the performance evaluations of speech recognition systems under noisy conditions. In *Automatic Speech Recognition: Challenges for the Next Millennium*. ISCA ITRW ASR2000, Paris, France, 2000.

[36] X. Huang, A. Acero, F. Alleva, D. Beeferman, M. Hwang, and M. Mahajan. From CMU Sphinx-II to Microsoft Whisper: Making speech recognition usable. In *Advanced Topics in Speech Recognition*. Kluwer Academic Publishers, 1995.

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

[38] X. Huang, F. Alleva, H.-W. Hon, M.-Y. Hwang, K.-F. Lee, and R. Rosenfeld. The SPHINX-II speech recognition system: an overview. *Computer Speech & Language*, 7(2):137–148, 1992.

[39] B.R. Hunt, R.L. Lipsman, and J.M. Rosenberg. *A Guide to MATLAB: for Beginners and Experienced Users*. Cambridge University Press, 2001.

[40] M. J. Hunt. A statistical approach to metrics for word and syllable recognition. *Journal of the Acoustic Society of America*, 66(S1), S35(A), 1979.

[41] P. Jain, H. Hermansky, and B. Kingsbury. Distributed speech recognition using noise-robust MFCC and TRAPS estimated manner features. In *Proc. International Conference on Spoken Language Processing*, Denver, Colorado, USA, September 2002. ISCA.

[42] A. Janin, D. Ellis, and N. Morgan. Multi-stream speech recognition: Ready for prime time? In *Proc. Eurospeech*, pages 591–594, Budapest, Hungary, September 1999. ISCA.

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

[44] N. S. Kim and C. K. Un. Frame-correlated hidden Markov model based on extended logarithmic pool. *IEEE Transactions on Speech and Audio Processing*, 3(2):149–160, March 1997.

[45] K. Kirchhoff, G.A. Fink, and G. Sagerer. Conversational speech recognition using acoustic and articulatory input. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 3, pages 1435–1438, Istanbul, Turkey, June 2000.

[46] J. Kittler, M. Hatef, R. P.W. Duin, and J.Matas. On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3):226–239, March 1998.

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

[48] N. Kumar and A. G. Andreou. Heteroscedastic discriminant analysis and reduced rank HMMs for improved speech recognition. *Speech Communcation*, 26:283–297, 1998.

[49] R.G. Leonard. A database for speaker-independent digit recognition. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 3, San Diego, California, March 1984.

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

[51] J. Ming and F. Smith. Modelling of the interframe dependence in an HMM using conditional Gaussian mixtures. *Computer Speech & Language*, 10(4), October 1996.

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

[53] P. Motlíček and L. Burget. Application of Mel-scale filter bank for noise estimation in speech processing. In *Proc. International Czech-Slovak Scientific Conference Radioelektronika*, Bratislava, Slovak Republic, May 2002. FEI STUBA.

[54] P. Motlíček and L. Burget. Efficient noise estimation and its application for robust speech recognition. In *Proc. International conference on Text Speech and Dialogue*, pages 229–236, Brno, Czech Republic, September 2002. Springer.

[55] P. Motlíček and L. Burget. Noise estimation for efficient speech enhancement and robust speech recognition. In *Proc. International Conference on Spoken Language Processing*, pages 1033–1036, Denver, Colorado, USA, September 2002. ISCA.

[56] 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.

[57] 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.

[58] P. Olsen and R. A. Gopinath. Modeling inverse covariances by basis expansion. *IEEE Transactions on Speech and Audio Processing*, 12:37–46, January 2004.

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

[60] G. Saon, M. Padmanabhan, R. Gopinath, and S. Chen. Maximum likelihood discriminant feature spaces. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 2, pages 1129–1132, Istanbul, Turkey, June 2000.

[61] H. Schwenk and J.-L. Gauvain. Combining multiple speech recognizers using voting and language model information. In *Proc. International Conference on Spoken Language Processing*, volume II, pages 915–918. ISCA, 2000.

[62] R. Singh, B. Raj, and R. M. Stern. Automatic clustering and generation of contextual questions for tied states in hidden Markov models. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, Phoenix, Arizona, USA, March 1999.

[63] G. Williams and S. Renals. Confidence measures from local posterior probability estimates. *Computer Speech & Language*, 13:395–411, 1999.

[64] S. Young. Acoustic modeling for large vocabulary continuous speech recognition. In K. Ponting, editor, *Computational Models of Speech Pattern Processing*, pages 19–39. NATO ASI Series, Berlin, 1999.

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