UNIVERSITÄT DER BUNDESWEHR MÜNCHEN
Fakultät für Elektrotechnik und Informationstechnik

Exploration and Optimization of Noise Reduction Algorithms for Speech Recognition in Embedded Devices

Panji Setiawan

Vorsitzender des Promotionsausschusses:	Prof. Dr.-Ing. B. Lankl
1. Berichterstatter	Prof. Dr.-Ing. H. Höge
2. Berichterstatter	Prof. Dr.-Ing. T. Fingscheidt

Tag der Prüfung 10.3.2009

Mit der Promotion erlangter akademischer Grad:
Doktor-Ingenieur
(Dr.-Ing.)

München, den 24. April 2009

Summary

Environmental noise present in real-life applications substantially degrades the performance of speech recognition systems. An example is an in-car scenario where a speech recognition system has to support the man-machine interface. Several sources of noise coming from the engine, wipers, wheels etc., interact with speech. Special challenge is given in an open window scenario, where noise of traffic, park noise, etc., has to be regarded. The main goal of this thesis is to improve the performance of a speech recognition system based on a state-of-the-art hidden Markov model (HMM) using noise reduction methods. The performance is measured with respect to word error rate and with the method of mutual information.

The noise reduction methods are based on weighting rules. Least-squares weighting rules in the frequency domain have been developed to enable a continuous development based on the existing system and also to guarantee its low complexity and footprint for applications in embedded devices. The weighting rule parameters are optimized employing a multidimensional optimization task method of Monte Carlo followed by a compass search method. Root compression and cepstral smoothing methods have also been implemented to boost the recognition performance. The additional complexity and memory requirements of the proposed system are minimum.

The performance of the proposed system was compared to the European Telecommunications Standards Institute (ETSI) standardized system. The proposed system outperforms the ETSI system by up to 8.6 % relative increase in word accuracy and achieves up to 35.1 % relative increase in word accuracy compared to the existing baseline system on the ETSI Aurora 3 German task. A relative increase of up to 18 % in word accuracy over the existing baseline system is also obtained from the proposed weighting rules on large vocabulary databases.

An entropy-based feature vector analysis method has also been developed to assess the quality of feature vectors. The entropy estimation is based on the histogram approach. The method has the advantage to objectively asses the feature vector quality regardless of the acoustic modeling assumption used in the speech recognition system.

Acknowledgment

First, I would like to thank my supervisors Harald Höge and Tim Fingscheidt for making this work possible. I am grateful for the support they have given me over the years through their suggestions and criticism.

Throughout the years I have benefited greatly from interaction with other members of speech processing group at Siemens AG Corporate Technology and former members of signal processing group at Siemens AG COM Mobile Devices. There are too many people to mention individually, but I must thank Josef Bauer and Bernt Andrassy for providing me, in particular, with the baseline speech recognition system. I would like also to thank Sorel Stan and Christophe Beaugeant for their ideas and fruitful discussions during the early stage of the work. I have also enjoyed having valuable discussions with Martin Schönle, Virginie Gilg, Stefanie Aalburg, Bruno Tramblay, Sergey Astrov, Kai Steinert, Suhadi Suhadi, Mickael de Meuleneire, and Emmanuel Thepie Fapi.

Finally, I would like to thank my family, especially my parents, for all their support over the years.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Advances in speech recognition have reached the point where commercial systems can be deployed with the existing technology. This includes the deployment in embedded devices such as mobile phones and handheld computers. In general, impressive recognition performance can be achieved in the laboratory on very large vocabulary continuous speech recognition tasks. However, problems usually arise in many realistic conditions where humans expect speech input systems to behave as much as people would. The robustness issue remains the biggest challenge especially in noisy environments such as public places, cars, etc. The focus of this thesis is to improve recognition performance with respect to robustness focusing on embedded devices.

Speech recognition can be stated as a system which involves a process of mapping human speech to its corresponding sequence of linguistic units such as phonemes and words. Basically, a typical speech recognizer consists of two parts as shown in Figure 1.1, i.e., a front-end and a back-end. The front-end is responsible for the removal of noise which is affecting the original input speech and then extracting the speech features out of it. The back-end is performing the actual classification based on the extracted speech features. As shown in the figure, the front-end usually consists of several processing modules, i.e., spectral analysis, noise reduction, and feature extraction modules. The figure also depicts an environmental model of an additive noise which degrades the input speech.

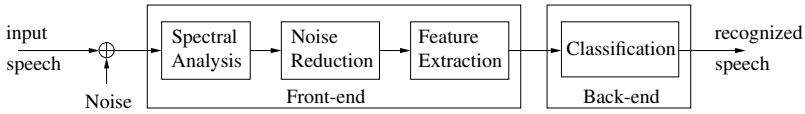


Figure 1.1: Typical modular structure of a speech recognizer having an additive noise as the environmental model.

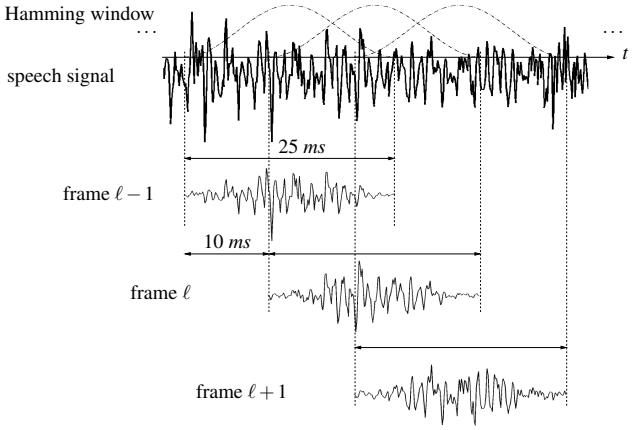


Figure 1.2: The speech signal is segmented into overlapping segments called *frames* having a length of 25 *ms* and frame shift of 10 *ms*. Feature vectors are calculated from these frames.

The spectral analysis module takes a short-time input speech segment and transforms it to the frequency domain. Noise is estimated and removed by the noise reduction module followed by the feature extraction module which converts each segment into an acoustic pattern called the feature vector which describes the short-time spectrum of the speech signal. This speech segment with a length of typically $10-30$ *ms* is usually referred to as a *frame*. It is obtained in an overlapping manner as depicted in Figure 1.2. It is shown in the figure that the frame shift determines the distance between consecutive frames. A Hamming windowing is applied to the frame as part of the spectral analysis processing which is transforming the time domain signal representation to its frequency domain one. For each frame ℓ a feature vector is calculated. The classification module assigns the sequence of feature vectors to the corresponding linguistic units. The speech recognizer is based on the state-of-the-art hidden Markov model (HMM) technology. Particular interest is put on the front-end part while the back-end will not be the focus of this thesis.

A particular focus of the thesis is the noise reduction module in the front-end which is aimed at reducing the effect of additive noise prior to the feature extraction processing. The disturbances caused by the presence of noise are giving a direct impact on the feature vectors which will cause problems to the classification module. This is shown in Figure 1.3 where the short-time power spectrum of a speech frame is depicted before and after the noise reduction. The presence of

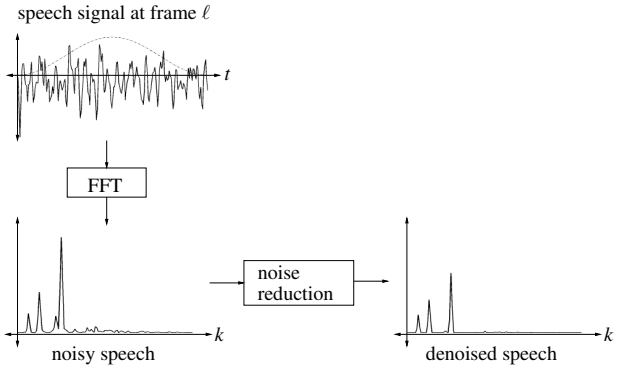


Figure 1.3: The short-time power spectrum of a noisy speech signal at frame ℓ and frequency bin k is fed into the noise reduction to produce the denoised power spectrum.

noise is resulting in distorted feature vectors and the distortion is properly described by the signal-to-noise-ratio (SNR) of the speech signal defined as the power ratio between the clean speech signal and the noise

$$\text{SNR} = 10 \cdot \log_{10}\left(\frac{P_{\text{clean speech}}}{P_{\text{noise}}}\right) \quad [\text{dB}],$$

where P denotes the average power. Lower SNR implies more distorted feature vectors. Noise reduction techniques aim at obtaining a good estimate of the power spectrum which is expected to match the power spectrum of the clean speech signal.

Many approaches have been developed to cope with the robustness issue in speech recognition. The robustness refers to the ability of the system to maintain minimum WER or good recognition accuracy even when the input speech is degraded due to the presence of noise. Noise reduction remains one of the basic approaches which is widely implemented in the front-end of a speech recognizer to deal with the presence of additive noise. It was initially developed in the context of speech enhancement focusing on the problem of enhancing a degraded speech signal. Its application is obvious in other areas such as speech coding.

Figure 1.4 shows the influence of the noise reduction. It depicts the speech recognition performance in word error rate (WER) based on the SNR of the input speech signal. The WER is defined as

$$WER = \frac{\text{\# of word errors}}{\text{\# of reference words}} \times 100 \quad [\%],$$

where word errors consist of deletion, substitution, and insertion errors. This performance mea-

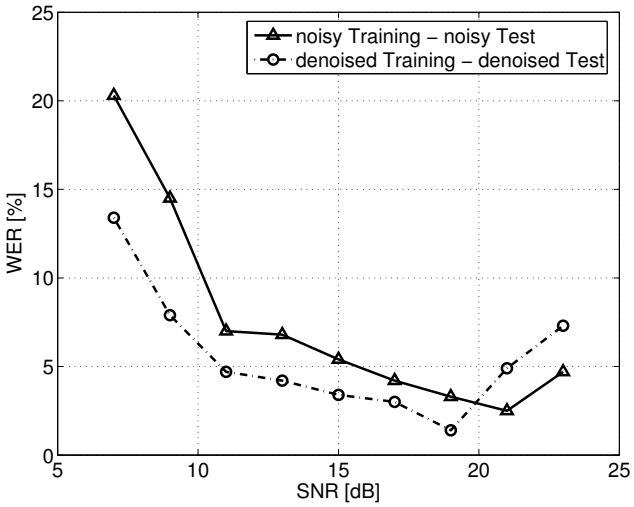


Figure 1.4: Typical performance of a speech recognizer measured in word error rate (WER), given the input speech utterances with different signal-to-noise-ratios (SNRs).

surement criterion is further described in Section 3.4 together with other performance measurement criteria, i.e., word accuracy and word recognition rate. The figure shows that the performance is deteriorating when the SNR of the input speech utterances is low. The use of a noise reduction technique significantly improves the performance by reducing the WER in the low SNR region. However, it also introduces distortions for the high SNR input speech utterances as observed in the figure. The distortions are caused by the *artifacts* produced by the noise reduction.

There exist various approaches in noise reduction which are operating either in the time or frequency domain. Particular emphasis is given for the frequency domain noise reduction approach widely known as the short-time spectral amplitude (STSA) estimator approach. This approach can be formulated as to find an estimate of the clean speech short-time spectral amplitude given the noisy speech short-time spectral amplitude. Important to note is that the noise reduction for speech enhancement needs further adjustment when applied in the speech recognition context.

In order to improve the performance of automatic speech recognition (ASR) systems, the concept of evaluation has been proven as very beneficial to push for progress such as the one organized by the Defense Advanced Research Projects Agency (DARPA) [Young and Chase

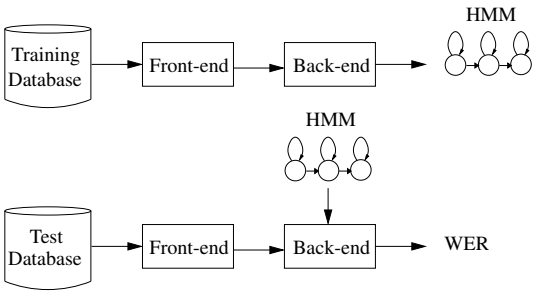


Figure 1.5: Evaluation framework used to test the performance of an ASR system.

1998]. The evaluation framework is defined by two speech databases, i.e., a training database and a test database as depicted in Figure 1.5. The training database is used to configure the back-end (the hidden Markov model (HMM) parameter training as described in Section 2.2) and the test database is used to determine the WER. The evaluation also defines a baseline ASR system and its performance. It is expected that any ASR system under test should achieve better performance than the baseline system.

In the area of robust speech recognition, recently the Aurora Working Group within the European Telecommunication Standards Institute (ETSI) has conducted an evaluation campaign yielding a robust speech recognition front-end called the ETSI advanced front-end (AFE) which has been defined and standardized in [ETSI STQ-Aurora 2003a]. It has been shown that the AFE is performing well when tested in various noisy conditions on several different languages [ETSI STQ-Aurora 2002]. The noise reduction part in the AFE is based on the STSA estimator approach and the back-end was built following the Aurora experimental framework [Hirsch and Pearce 2000] using the hidden Markov model toolkit (HTK) [Young et al. 2005].

The Aurora evaluation activities have attracted many publications, e.g., in [Fujimoto and Nakamura 2005, Wu et al. 2005, Droppo et al. 2001]. In this thesis, we are performing evaluations and developing several front-end techniques within the Aurora framework as well. As the baseline, an existing ASR system was used consisting of specific front-end and back-end, i.e., a system developed by Siemens AG, Corporate Technology [Astrov et al. 2003, Varga et al. 2002, Andrassy et al. 2001, Bauer 2001], henceforth called the *Siemens front-end* (SFE) and *Siemens back-end* (SBE), respectively. Note that SBE was used instead of the HTK back-end as specified by the Aurora framework. The evaluation is done on one of the evaluation databases specified in the Aurora framework, i.e., the Aurora 3 German which contains speech recorded in a car environment. The front-end development is aiming at achieving an advanced SFE which gives better performance than the baseline SFE.

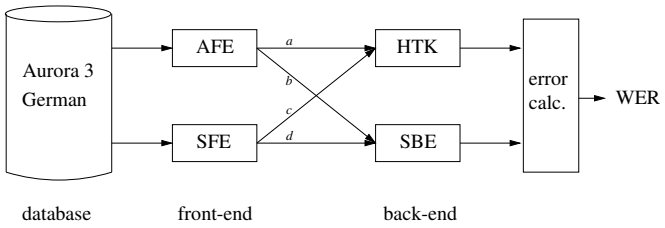


Figure 1.6: The Aurora evaluation framework showing all possible configurations between two different front-ends and back-ends.

Figure 1.6 depicts all possible configurations in performing the evaluation on the Aurora framework. The original AFE-HTK configuration is shown in the figure with an arrow label *a*. Initial benchmarking work done with the SFE within the Aurora framework was in fact employing the HTK back-end and it was showing lower performance achievement compared to the AFE [Andrassy et al. 2001]. This is shown in the figure having an arrow with the label *c*. In this thesis, the baseline SFE results are defined with the configuration as shown with the arrow label *d* in the figure. Further improvements are all investigated and evaluated employing this configuration. The AFE-SBE configuration as shown with the arrow label *b* serves for reference purposes. Achieving a better performance than the AFE is our particular interest.

In the AFE-SBE configuration, several modifications are introduced to the original AFE in order to make the front-end achieve its optimal performance when using the SBE instead of the HTK as the back-end. The original SFE has to be modified as well to match some parametric configurations used in the AFE. Those are done to ensure proper and correct performance benchmarking process using only the SBE as the common back-end. The results obtained after having the adjustments done in both front-ends show that the AFE is still showing superior performance than the baseline SFE. Based on the modified AFE results, the AFE-SBE system is used as the performance reference. The work in this thesis is directed to improve the baseline SFE focusing on the noise reduction.

We are mainly focusing on the development of noise reduction approaches within the speech enhancement area using a single microphone (single channel approach) for the following reasons:

- The trade-off between the cost and benefit of multi-channel processing implementation in embedded devices still need further considerations. Multi-channel processing certainly implies higher cost.
- Some derivatives of classical speech enhancement technique are still the best approaches available to date in speech enhancement context. These approaches are relatively simple

to implement and they usually outperform more elaborate techniques. Their application in speech recognition was shown in the ETSI advanced front-end [ETSI STQ-Aurora 2003a] mentioned previously as the standardized ETSI robust front-end for distributed speech recognition (DSR), which is based on the classical Wiener filtering noise reduction technique.

Another interesting focus of this thesis is the analysis of the feature vectors utilizing the theory of entropy. The quality of feature vectors in speech recognition is usually evaluated based on the resulting recognition performance. This leads to the practice where the algorithms in the front-end have to be tuned in a way that yields a better recognition performance without necessarily having a satisfactory explanation or analysis on the improvements being made. It is also sometimes the case where the front-end is optimally tuned to yield a better recognition performance on a sub-optimal back-end. Hence, an inadequate analysis on the feature vectors and the influence of *modeling errors* in the back-end are the two main problems we try to address in this thesis.

The motivation to use the theory of entropy is based on the fact that a relationship between the Bayes probability of error and entropy of the feature vectors exists as described in Section 5.1. Thus, an entropy measure can be used as a tool in judging the quality of feature vectors. It gives a first insight into the statistical properties of the feature vectors depending on the chosen noise reduction algorithm although it is generally applicable to any improvements made in the front-end. Since the Bayes probability of error is used as the performance measurement, it automatically excludes the influence of modeling errors in the back-end.

1.1 Objectives and Main Achievements

The goal of this thesis is to address the problem of reducing the WER under noisy conditions while still maintaining low computational load for applications in embedded devices. We restrict ourselves to achieving improvement in recognition performance through modifications introduced in the front-end. Car noise is of our particular interest. The improvement is evaluated based on the performance achieved with the baseline SFE after introducing some additional processing and finally compared to the performance of the state-of-the-art ETSI AFE. The evaluation is done on the Aurora 3 German digits database and other car databases. The SBE is used as the common back-end.

Main achievements contributed in this thesis are listed as follows:

- Further improvements developed on the noise reduction module, particularly in the weighting rule formulation, contribute to a higher recognition performance. This is further en-

hanced by integrating several pre-processing techniques in the feature extraction modules of the baseline SFE. The *new* SFE outperforms the AFE in the given task.

- The improvements are still suitable for embedded devices implementation which is a very important aspect for the deployment of a commercial speech recognition technology. It maintains high recognition accuracy while keeping the computational load low.
- Results on larger databases confirm the superiority of the proposed weighting rules.
- An alternative analysis on the feature vectors based on the mutual information or entropy approach. This presents a way of measuring the quality of the feature vectors without performing the classification task in the recognition phase.

1.2 Thesis Outline

The rest of the thesis is organized as follows:

Chapter 2 is describing the most important aspects when dealing with the stochastic speech recognition approach. This includes the use of the HMM technique, Viterbi algorithm, and some aspects regarding the HMM training. Some words on the mismatch problem are also mentioned.

Chapter 3 is presenting the speech recognition systems under evaluation. Two speech recognition systems are mainly used, i.e., the ETSI standardized front-end and the SFE. Only a single back-end is described, i.e., the SBE, which is used for both front-ends. This chapter also presents other front-end techniques which have been successfully applied to the SFE. The database description and the performance evaluation used throughout this thesis are presented in the subsequent sections. Finally, some words concerning the embedded system are given.

Chapter 4 is dedicated mainly to the state-of-the-art frequency domain formulations and our contributions in this area. Our contribution is described in the section dealing with the least-squares based weighting rule formulation in the frequency domain.

Chapter 5 deals with an alternative feature evaluation technique which is based on the maximization of mutual information. This chapter is presenting our contribution to the feature analysis technique.

Chapter 6 presents the evaluation of the front-end systems. The work is based on the digit task Aurora 3 German database. Various experiments are presented which show some gradual improvements obtained using various methods implemented in the SFE. This chapter highlights the superiority of our improved SFE as compared to the AFE.

Chapter 7 describes further improvement achieved when larger vocabulary databases are used. It focuses on the superiority of the proposed noise reduction techniques which show further improvement compared to the noise reduction in the baseline SFE.

Chapter 8 evaluates the proposed feature evaluation technique on the Aurora 3 German digits database.

Chapter 9 finally presents the conclusions and hints for future work.

Chapter 2

Stochastic Speech Recognition

Speech recognition can be regarded as a pattern recognition task, where the speech signal patterns have to be mapped on *words*. *Words* are speech classes defined by the corresponding linguistic units in the vocabulary, also called lexicon. The lexicon is denoted as $\mathbb{W}$ which is a set of words. The lexicon size simply tells about the amount of words stored in it. A particular word in the set $\mathbb{W}$ is denoted with W.

In practice, the feature extraction module converts the speech signal into a sequence of acoustic feature vectors $\mathbf{x}^M = \{\mathbf{x}(\ell)\}_{\ell=0}^{M-1} = \{\mathbf{x}(0),\ \cdots,\ \mathbf{x}(M-1)\}$ of length M where each $\mathbf{x}(\ell),\ \ell = 0 \cdots M-1$ is a d-dimensional instance of a multivariate random variable $\mathbf{X} = [\ X_1\ \cdots\ X_d\ ]^T$. There exist several approaches in designing a speech recognition system which can be classified as template-based, knowledge-based, stochastic, and connectionist approaches. The stochastic framework is the one considered in this thesis.

Depending on the given task, a speech recognition system can be categorized as *isolated word* recognition, *connected word* recognition, and *continuous* speech recognition. In isolated word recognition, the input speech utterance must have deliberate pauses between words which is then considered as taking an isolated word input. This is the easiest task in speech recognition since the task is merely formulated as finding the most likely word given a list of word references in the vocabulary. Connected word recognition allows the utterance to have multiple word inputs and the task is stated as to find the optimum sequence of word reference patterns that best matches the input utterance. The word inputs are limited to those stored in the vocabulary. Continuous speech recognition constitutes the most difficult task in speech recognition since the input utterance is considered to be natural where words are spoken continuously without pauses or other apparent division between words. This usually involves larger vocabulary containing unknown words, hesitations, etc. and thus makes the problem of finding the optimum sequence of word references more difficult.

The Bayes decision rule has been widely used as the fundamental approach to solve the problems in speech recognition which theoretically leads to minimum error rate [Fukunaga 1990]. In the case of isolated word recognition, it can be shown that given an observed sequence $\mathbf{x}^M$ the recognition problem is formulated as

$$\widehat{W} = \arg\max_W \; P(W|\mathbf{x}^M), \tag{2.1}$$

where it is assumed that the *a posteriori* probability $P(W|\mathbf{x}^M)$ given the feature vectors exists and is known. For the connected and continuous speech recognition cases, the recognition problem is formulated as finding the most likely word sequence. For simplicity, we will proceed with the isolated word recognition case. According to the Bayes rule of probability, (2.1) is conceptually equivalent to

$$\widehat{W} = \arg\max_W \; p(\mathbf{x}^M|W)P(W), \tag{2.2}$$

where the *a posteriori* probability is replaced by the conditional probability density $p(\mathbf{x}^M|W)$ and the *a priori* probability $P(W)$ which leads to the well-known maximum likelihood method. The probabilities are also known as the acoustic model and language model probabilities, respectively. Throughout this thesis we denote probability densities with $p(\cdot)$ and discrete probabilities with $P(\cdot)$.

Based on this framework, the goal is to find a parameter set Ω so that $P_\Omega(W|\mathbf{x}^M)$ can best approximate $P(W|\mathbf{x}^M)$. The acoustic model $p(\mathbf{x}^M|W)$ is approximated by $p_\Lambda(\mathbf{x}^M|W)$, where Λ denotes the parameter set of an assumed acoustic model which is represented by the hidden Markov model (HMM) in the stochastic framework. This offers a way to characterize the statistical properties of the acoustic feature vectors by assuming that the feature vectors are the outputs of a parametric random process. Tutorials on this statistical modeling and its application in ASR are given in, e.g., [Young et al. 2005, Rabiner 1989].

The language model $P(W)$ is approximated by $P_\Theta(W)$, where Θ denotes the parameter set of an assumed language model. Language modeling contributes to the scoring in the classifier according to grammar or semantics. It is independent of the acoustic feature vectors and is aiming at placing some constraints on how to build a certain sentence given the recognized words as stored in the vocabulary.

Based on the assumed acoustic and language models, the calculation of (2.2) is shown as

$$\begin{aligned} \widehat{W} &= \arg\max_W \; P_\Omega(W|\mathbf{x}^M) \\ &= \arg\max_W \; p_\Lambda(\mathbf{x}^M|W)P_\Theta(W). \end{aligned} \tag{2.3}$$

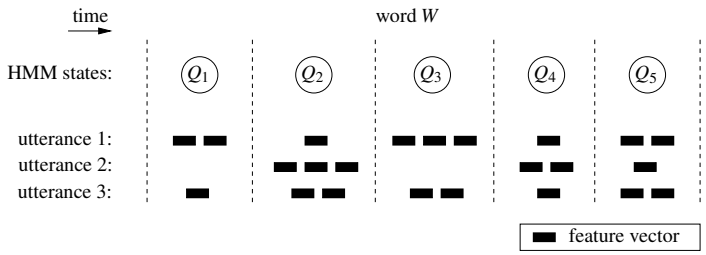


Figure 2.1: Illustration of different feature vector alignments to the HHM states for the same word W.

Thus, we can formulate the parameter set as $\Omega = \{\Lambda, \Theta\}$ which refers to the classification module of a speech recognizer.

In this context three research directions in speech recognition can be stated as

- Improving the feature vectors $\mathbf{x}^M$ aiming at achieving lower error rate.
- Improving the distributions $p_\Lambda(\mathbf{x}^M|W)$ and $P_\Theta(W)$ which is also aiming at achieving lower error rate.
- Improving the *search* problem as performed by the `arg max` function which is dedicated to achieving lower computational load and memory requirements.

An overview of the statistical decision and estimation in the statistical pattern recognition context is given, for example, in [Fukunaga 1990].

Acoustic modeling of speech using the HMM is following the assumption that speech can be broken down into states in which the speech signal is considered to be stationary. Another assumption is related to the observed feature vectors where each observation depends only on the current state and not on the previous observation. This implies the independence assumption of the feature vectors. Both assumptions are crude assumptions of the speech signal. The HMM itself is a stochastic finite-state machine which consists of a Markov chain of states and a probabilistic function assigned to each state.

A sequence of feature vectors $\mathbf{x}^M$ describing a speech utterance corresponds to a sequence of states $\psi^M = \{\psi(0),\ \cdots,\ \psi(M-1)\}$ where $\psi(\ell),\ \ell = 0 \cdots M-1$ denotes the state occupied at frame ℓ. Variations in uttering a particular word W yield different state sequences as illustrated in Figure 2.1. An example of a *visited* state sequence is given by the second utterance in the figure:

$$\psi^6 = \{Q_2,\ Q_2,\ Q_2,\ Q_4,\ Q_4,\ Q_5\},$$

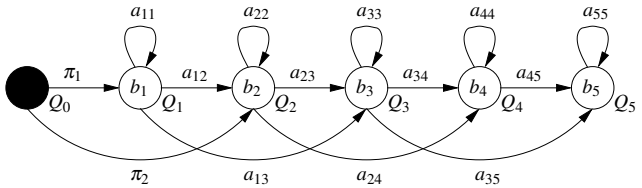


Figure 2.2: Illustration of 5-states Bakis topology with its initial, state transition, and emission probabilities.

with Q_j being the HMM state. The term *visited* is used since the figure depicts several utterances with different state sequences following an allowed state ordering $Q_1, \cdots, Q_5$ to describe the word W. This is actually the case in the training phase where an allowed state ordering of a particular word is known from the transcription and the feature vectors have to be aligned to the HMM states. The alignment process requires the knowledge of the *hidden* state sequence. In the recognition phase the state sequence is also not known or *hidden*. The task is to find the most likely state sequence given all possible word references or state orderings stored in the vocabulary. Note that each word is always identified with a particular state ordering.

Having sufficient utterance variations for a particular word W and grouping together all feature vectors which belong to the same state lead to an understanding that the states are statistically describing the speech. These states are emitting the feature vectors according to the *emission probability* function of each state denoted by b_i where the index i refers to the a particular state. It is shown in Figure 2.1, for example, that all six feature vectors assigned to the state Q_2 determine the emission probability b_2. It is also shown in the figure that the feature vectors are not always aligned to all available states as illustrated in the second utterance where it skips the states Q_1 and Q_3. This behavior is explained by the *transition probability* a_{ij} where the indices i and j denote the state origin and destination of the transition. The notation a_{22}, for example, denotes the transition probability of going from state Q_2 to Q_2.

The HMM topology suited in modeling speech is called the *left-to-right* Bakis topology as shown in Figure 2.2. This is due to the speech characteristics which are changing over time in a successive manner as previously depicted in Figure 2.1. As shown in the topology, a *skip* is allowed in the state sequence.

Finally, the construction of a word using HMM states is depicted in Figure 2.3 where the states are representing the basic structure of speech. The word W is divided into several subwords labeled with /word:0/ $\cdots$ /word:7/ and each subword consists of several segments, e.g., word:0.0 $\cdots$ word:0.2 for the subword /word:0/. A certain amount of HMM states is assigned to the segment. The figure shows that only one state is used to model a segment.

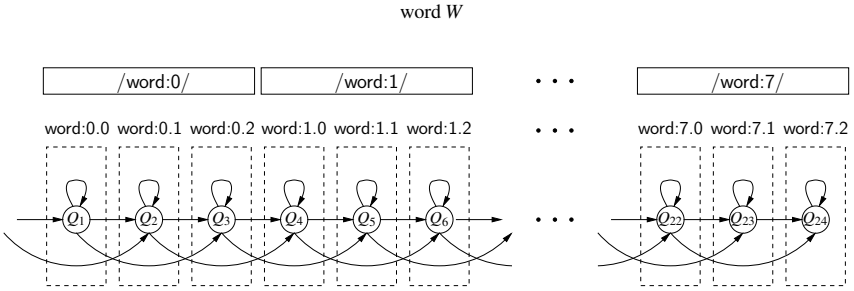


Figure 2.3: HMM modeling of speech showing the construction of a word based on subwords.

Based on the word reference unit, speech recognition can be classified either as *whole word* or *phoneme* based speech recognition. Whole word based speech recognition uses each word as its reference. An acoustic model is generated for each word stored in the vocabulary. This is suitable for small vocabulary tasks where the entries in the vocabulary are usually limited. The phoneme based speech recognition uses a subword such as a phoneme as its reference. This is practically suitable for large vocabulary tasks where storing the subwords greatly reduces the amount of storage and computational complexity required in comparison with the whole word modeling. It also implies that adding new words is not always related to an increase in the vocabulary size. The main issue is actually concerning the training of HMM parameters. Both speech recognition classes allow the implementation of such structure as depicted in Figure 2.3. Note that the subwords in the whole word speech recognition do not relate to a linguistic unit such as a phoneme.

2.1 Hidden Markov Model (HMM) Parameter Formulation

To summarize, the parameters describing the HMM can be categorized as

1. the number of states N_S. The states in the HMM are denoted as $\{Q_j\}_{j=1}^{N_S}$ and $s_j(\ell)$ denotes being in state Q_j at frame ℓ. The term N_S is used in connection with the modeling of a single word reference unit. The term N_Q denotes the total number of states used to model all words in the vocabulary.

2. the initial state distribution set $\Pi = \{\pi_i\}_{i=1}^{N_S}$, where

$$\pi_i = P\big(s_i(0)\big). \tag{2.4}$$

Figure 2.2 shows that the state sequence starts at Q_0 which is denoted with the black circle to indicate that the state is non-emitting. Thus, the frame index $\ell = 0$ starts either at Q_1 or Q_2.

3. the state transition probability set $A = \{a_{ij}\}_{i=1,j=1}^{N_S}$, where

$$a_{ij} = P\big(s_j(\ell+1)|s_i(\ell)\big). \tag{2.5}$$

Putting it in a matrix form of all permitted transitions, the matrix is certainly not full given the description in the figure. Furthermore, the probabilities must satisfy the following constraint:

$$\sum_{j=1}^{N_S} a_{ij} = 1. \tag{2.6}$$

4. the emission probability distribution set B which is a set of parameters describing the distribution. The distribution set itself is denoted with $\big\{b_i\big(\mathbf{x}(\ell)\big)\big\}_{i=1}^{N_S}$, where

$$b_i\big(\mathbf{x}(\ell)\big) = p\big(\mathbf{x}(\ell)|s_i(\ell)\big), \tag{2.7}$$

denoting the probability density function of observing $\mathbf{x}(\ell)$ when the system is in state Q_i at frame ℓ.

Depending on the data type of the output distribution the HMM may be a discrete or continuous density one. The discrete HMMs (DHMMs) are modeling the outputs as a discrete set and the continuous density HMMs (CDHMMs) are modeling a continuous set of observable outputs $\mathbb{X} = \mathbb{R}^d$. Feature vectors are usually modeled with the CDHMMs having a mixture of multivariate Gaussian distributions

$$b_i\big(\mathbf{x}(\ell)\big) = \sum_{r=1}^{R_i} c_{ir}\, \mathcal{N}\big(\mathbf{x}(\ell); \mu_{ir}, \Sigma_{ir}\big) = \sum_{r=1}^{R_i} c_{ir}\, b_{ir}\big(\mathbf{x}(\ell)\big), \tag{2.8}$$

where R_i denotes the amount of multivariate Gaussian densities assigned to the state Q_i

$$\mathcal{N}\big(\mathbf{x}(\ell); \mu_{ir}, \Sigma_{ir}\big) = \frac{1}{\sqrt{(2\pi)^d |\Sigma_{ir}|}}\, e^{-\frac{1}{2}(\mathbf{x}(\ell)-\mu_{ir})^T\, \Sigma_{ir}^{-1} (\mathbf{x}(\ell)-\mu_{ir})}, \tag{2.9}$$

with μ_{ir} and Σ_{ir} denoting the mean vector and covariance matrix, respectively. The mixture

coefficient c_{ir} must satisfy the following constraint:

$$\sum_{r=1}^{R_i} c_{ir} = 1.$$

For CDHMMs B consists of a set of means, variances, and mixture coefficients $B = \{c_{ir}, \mu_{ir}, \Sigma_{ir}\}$. This formulation is the one considered in this thesis.

The notation describing a HMM is thus formulated as

$$\Lambda = (\Pi,\, A,\, B).$$

2.2 The HMM Parameter Training

Obtaining the best model for each word is done in the training phase by solving an optimization problem aimed at estimating the HMM parameters based on a certain optimization criterion. There exist many criteria in this optimization problem, however, the maximum likelihood (ML) estimator turns out to be the most popular one and widely implemented to date. It is necessary to understand how the likelihood function is evaluated before proceeding to the maximum likelihood estimator. The problem is also known as the acoustic model evaluation and will be described in this section.

The evaluation of the acoustic model $p_\Lambda(\mathbf{x}^M|W)$ is obtained through the evaluation of the likelihood function $p(\mathbf{x}^M|\Lambda)$. The likelihood is defined as the probability that a model with particular parameter values assigns to the data that has actually been observed [Neal 1993]

$$\mathcal{L}(\Lambda|\mathbf{x}^M) = p(\mathbf{x}^M|\Lambda) = \prod_{\ell=0}^{M-1} p\big(\mathbf{x}(\ell)|\Lambda\big), \tag{2.10}$$

where $\mathcal{L}(\cdot)$ denotes the likelihood. Equation (2.10) is, however, not considering the existence of a state sequence. It is therefore not pointing out to a practical solution for our specific HMM problem.

Taking into account the underlying state sequence, the likelihood equation is determined by calculating the joint probability function of both the observation sequence and state sequence

given the model and then summing over all possible state sequences

$$\begin{aligned} p(\mathbf{x}^M|\Lambda) &= \sum_{\psi^M \in \Psi^M} p\big(\mathbf{x}^M, \psi^M|\Lambda\big) \\ &= \sum_{\psi^M \in \Psi^M} p\big(\mathbf{x}^M|\psi^M, \Lambda\big) P(\psi^M|\Lambda), \end{aligned} \tag{2.11}$$

where Ψ^M is the set of all possible state sequences of length M in the model. The probabilities $p\big(\mathbf{x}^M|\psi^M, \Lambda\big)$ and $P(\psi^M|\Lambda)$ are calculated as

$$\begin{aligned} p\big(\mathbf{x}^M|\psi^M, \Lambda\big) &= \prod_{\ell=0}^{M-1} b_{Q_\psi(\ell)}\big(\mathbf{x}(\ell)\big), \\ P(\psi^M|\Lambda) &= \pi_{Q_\psi(0)} \prod_{\ell=1}^{M-1} a_{Q_\psi(\ell-1)Q_\psi(\ell)}, \end{aligned}$$

where $Q_\psi(\ell)$ returns the index of the state Q at frame ℓ in the state sequence ψ. The likelihood of (2.11) is thus shown as

$$p(\mathbf{x}^M|\Lambda) = \sum_{\psi^M \in \Psi^M} \pi_{Q_\psi(0)} b_{Q_\psi(0)}\big(\mathbf{x}(0)\big) \prod_{\ell=1}^{M-1} a_{Q_\psi(\ell-1)Q_\psi(\ell)} b_{Q_\psi(\ell)}\big(\mathbf{x}(\ell)\big). \tag{2.12}$$

The solution of (2.12) is obviously computationally expensive. Fortunately, an efficient algorithm exists to solve the problem, i.e., the *forward-backward* algorithm (e.g., in [Rabiner and Juang 1993], Appendix A.1), originally known as the BCJR algorithm [Bahl et al. 1974]. The maximum likelihood optimization problem is then solved using the Baum-Welch re-estimation algorithm where the best HMM parameters are obtained in an iterative manner using the forward-backward algorithm. This algorithm is described briefly in Appendix A.2. The segmental *K*-means algorithm [Juang and Rabiner 1990] offers an alternative solution to the Baum-Welch re-estimation algorithm to estimate the HMM parameters.

The *forced* Viterbi alignment or the Viterbi training procedure is usually taken to replace the Baum-Welch re-estimation training procedure due to its simplicity and capability to deliver comparable performance. This training procedure is based on the Viterbi algorithm [Viterbi 1967] as described in the subsequent section. The term *forced* is used since the Viterbi algorithm is forced to deliver the best path calculation for a given state ordering. This can also be seen as a feature vector alignment to the states. In both training procedures, the allowed state ordering is known from the transcription. The Viterbi training is thus using the information obtained from the Viterbi best path calculation to estimate the HMM parameters while the Baum-Wlech

training is basically calculating the expectations of events.

The database used in the training phase is holding an important role in estimating the HMM parameters. It is necessary to train the parameters on the training database which is matching the target recognition environment.

2.3 The HMM recognition

In the recognition phase it is necessary to evaluate $p\left(\mathbf{x}^M|\Lambda\right)$ for each possible word as shown in (2.3). The evaluation of this term according to (2.12) is usually not employed since it needs high computational power. Therefore, in practice the task of finding the most likely state sequence associated with $\mathbf{x}^M$ is used so that (2.12) is formulated as

$$p(\mathbf{x}^M|\Lambda) = \max_{\psi^M \in \Psi^M} \pi_{Q_\psi(0)} b_{Q_\psi(0)}\left(\mathbf{x}(0)\right) \prod_{\ell=1}^{M-1} a_{Q_\psi(\ell-1)Q_\psi(\ell)} b_{Q_\psi(\ell)}\left(\mathbf{x}(\ell)\right). \tag{2.13}$$

The recognition problem in (2.3) is thus shown as

$$\widehat{W} = \arg\max_W \; p_\Lambda(\mathbf{x}^M, \psi^{M*}|W) P_\Theta(W), \tag{2.14}$$

where ψ^{M*} denotes the most likely state sequence of length M.

Several optimization criteria exist to obtain the *optimal* state sequence. One of them is by maximizing the expected number of correct individual states. This is done by computing an individually most likely state $Q(\ell)$ for each frame ℓ

$$Q(\ell) = \arg\max_{1\le i\le N_S} \gamma_i(\ell) \qquad 0 \le \ell \le M-1, \tag{2.15}$$

where

$$\gamma_i(\ell) = p\left(s_i(\ell)|\mathbf{x}^M, \Lambda\right). \tag{2.16}$$

Note that $Q(\ell)$ shall return the index of the state Q at frame ℓ. This criterion might exhibit a problem that the resulting state sequence might not be the valid state sequence and also that it neglects the probability of occurrence of sequences of states. The Viterbi algorithm is thus commonly used to solve the problem of finding the most likely state sequence.

To apply the Viterbi algorithm in the recognition phase, the following variable is defined:

$$\phi_i(\ell) = \max_{\psi^{\ell-1} \in \Psi^{\ell-1}} p\left(\mathbf{x}^\ell, s_i(\ell), \psi^{\ell-1}|\Lambda\right), \tag{2.17}$$

where $\Psi^{\ell-1}$ denotes the set of all partial paths of length $\ell-1$. This can be recursively calculated using

$$\phi_j(\ell+1) = \max_{1\le i\le N_S} \big[\phi_i(\ell)a_{ij}\big] b_j\big(\mathbf{x}(\ell+1)\big), \tag{2.18}$$

for $1 \le j \le N_S$ and initialized with

$$\phi_j(0) = \pi_j b_j\big(\mathbf{x}(0)\big). \tag{2.19}$$

Replacing the maximization procedure in (2.18) by the summation procedure yields the iterative *forward* algorithm to obtain the likelihood in (2.12). The total likelihood score of the most likely state sequence is given by

$$\text{Likelihood score } = \max_{1\le i\le N_S} \phi_i(M-1). \tag{2.20}$$

The forced Viterbi alignment as mentioned in the previous section is done by tracing back the most likely state sequence. It is done by introducing the following variable:

$$\vartheta_j(\ell+1) = \arg \max_{1\le i\le N_S} \big[\phi_i(\ell)a_{ij}\big], \tag{2.21}$$

for $1 \le j \le N_S$. The state sequence is thus obtained with

$$Q(\ell) = \vartheta_{Q(\ell+1)}(\ell+1), \tag{2.22}$$

where the backtracing is calculated for $\ell = M-2,\ M-3,\ \cdots,\ 0$. The index of the last occupied state $Q(M-1)$ of the most likely sequence is known from

$$Q(M-1) = \arg \max_{1\le i\le N_S} \phi_i(M-1). \tag{2.23}$$

An illustration of the Viterbi algorithm is given in Figure 2.4 using the second utterance depicted in Figure 2.1. The path of the most likely state sequence is shown with the solid line. All possible state sequences of the same length must be within the shaded area following the Bakis topology. For each new feature vector $\mathbf{x}(\ell+1)$ the variables $\phi_j(\ell+1)$ and $\vartheta_j(\ell+1)$ are recursively calculated for each dot in the shaded area using (2.18) and (2.21), respectively. The indices i and j in the equations must lie within the shaded area and for a particular index j, only the values of the index i within the set $i \in \{j,\ j-1,\ j-2\}$ are considered. Finally, the total likelihood score of the most likely state sequence in (2.20) is given as $\phi_{N_S}(M-1)$ and $Q(M-1)$ in (2.23) is set to Q_{N_S} to start the bakctracing process using (2.22).

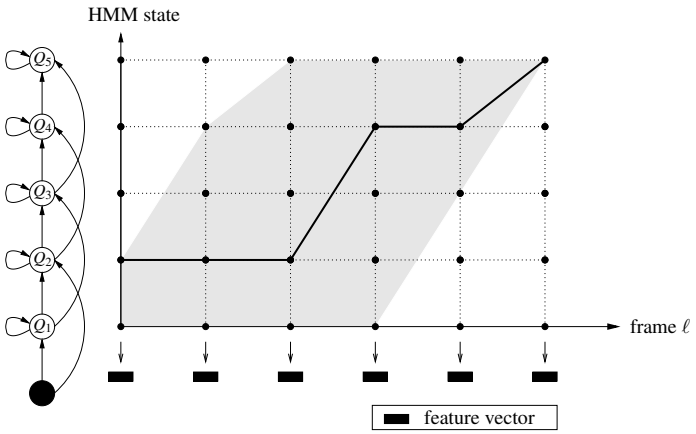


Figure 2.4: Illustration of the Viterbi algorithm yielding the most likely sequence.

2.4 The Problem of Mismatch

A noisy environment is usually characterized by the presence of one or more noise sources. These sources are usually categorized as additive and convolutive distortion. Examples of additive noise are the noise produced by machinery, background conversation, telephone ringing, etc. Sources of convolutive noise are, for example, room reverberation, microphone transducer, vocal tract characteristic of individual speaker, etc. There also exist other noise sources which are still difficult to quantify such as different speaking styles influenced by the environment known as the Lombard effect and psychological awareness of the speaker.

Developing a robust speech recognition system with a high level of recognition accuracy in a dynamically varying acoustical environment continues to be a challenging research topic. The various environmental noise contributes to the mismatch between the testing and training conditions. This is related to the feature vectors showing different distributions in both conditions. A significant number of studies has been dedicated to address the mismatch problem which is briefly summarized in [Furui and Lee 1995, Gong 1995, Juang 1991].

An example of the mismatch problem can be seen in Figure 2.5. The testing environment is noisy and two different HMMs were obtained, one being trained in a clean environment and the other in a noisy environment to simulate the mismatch and matched scenario, respectively. In this example, the type of noise is the same in the noisy training and testing environment. It is obvious that performing a recognition task in the noisy environment using a recognition system

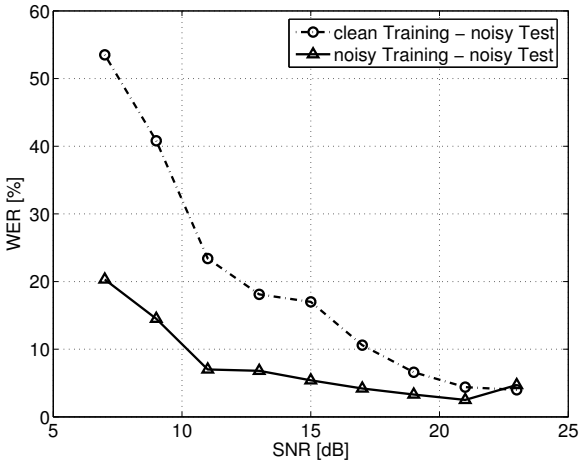


Figure 2.5: Effect of mismatch in the speech recognition system.

trained in the same noisy environment yields a much better result than the one trained in a clean environment. The advantage is clearly seen for almost all SNR values with the highest gain shown for the lowest SNR. This highlights the importance of having a matched scenario.

The use of a denoising technique in both scenario is certainly boosting the performance. An example of applying a denoising technique in a matched scenario was shown previously in Figure 1.4. In the case where the testing environment is not known in advance, an HMM trained in a clean environment is usually provided. A denoising technique is simply applied during the recognition phase to cope with the possible case of encountering a noisy environment. This is a sub-optimal solution but offers a possibility of deploying a recognition system in a wide range of environment. It is basically recommended that the denoising technique is also applied during the training to take into account possible artifacts introduced by the denoising technique which usually occur in the high SNR region. The decision is usually based on the additional cost and effort considerations due to the following facts:

- Replacing an existing HMM is usually not preferred.
- An HMM is usually provided by performing the training in a clean environment.
- Optimizing a particular denoising technique requires a retraining process which is obviously not preferred if the training data is large.

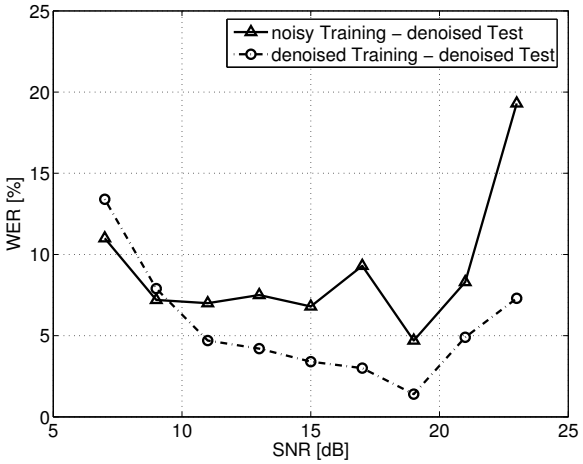


Figure 2.6: An alternative mismatch scenario for a speech recognition system operating in a low SNR environment.

- A rapid development of denoising techniques offers a wide range of possibility to update the system performance and simply optimizing the denoising technique in the testing environment given the clean HMM is more appropriate than using an HMM based on a different denoising technique.

In the case where the testing environment is known to be considerably noisy, it is sometimes preferable to adopt the matched scenario and an additional denoising technique applied only in the testing. This is a mismatch scenario but it proves to be a promising alternative scenario as shown in Figure 2.6 where it yields a better performance for SNR region at 9 dB and below compared to the matched scenario. This phenomenon is attributed to the limitation of the denoising technique in the low SNR region.

2.5 A Survey of Robustness in Speech Recognition

The effect of noise in speech recognition has been discussed in Section 2.4. Most of the research in robust speech recognition has been directed towards compensating the additive and convolutive effects as shown in the environmental model in Figure 2.7 [Stern et al. 1996]. In this thesis,

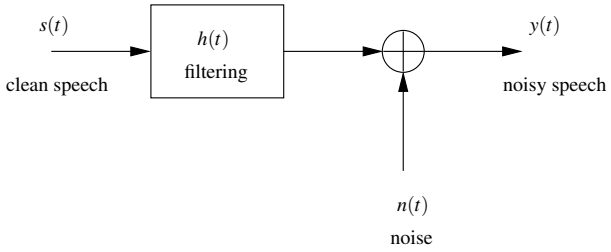


Figure 2.7: An environmental model with additive noise and convolutive distortion.

however, only the additive noise is being considered. We use an existing algorithm to handle the convolutive distortion.

In general, it is convenient to categorize research areas for robust speech recognition into the following:

- Speech enhancement, e.g., [Lim and Oppenheim 1979], which is aimed at improving perceptual aspects of degraded speech for a human listener. Among the important areas are the noise reduction techniques which are dealing with the statistically independent additive noise, and echo cancellation techniques, which are trying to reduce the effects of acoustic echo and of reverberation. Speech enhancement techniques are definitely applicable to speech recognition.

- Acoustic feature representations, which are focusing on basic feature extraction schemes. The robustness of the acoustic representations is usually evaluated when extracting the feature vectors from degraded speech. Besides the most common methods of the mel frequency cepstral coefficients (MFCC) [Davis and Mermelstein 1980] and the linear predictive coding (LPC) coefficients [Makhoul 1975], other feature extraction schemes which are based on the human auditory model have been developed such as perceptual linear predictive (PLP) analysis [Hermansky 1990], ensemble interval histogram (EIH) [Ghitza 1994], zero-crossings with peak-amplitudes (ZCPA) [Kim et al. 1999], etc. Dimensionality reduction techniques such as principal component analysis (PCA) or the Karhunen-Loève expansion (KLE) or eigenvector orthonormal expansion and linear discriminant analysis (LDA) could further contribute to the robustness issue.

- Feature compensation, which compensates the acoustic feature vectors for the environmental degradation. The compensation techniques may operate in any processing levels in the feature extraction scheme such as in the power spectrum domain, on the filter-bank

output, logarithm output, and cepstrum. Other techniques could also be mentioned such as microphone array processing, cepstral mean subtraction/normalization (CMS/N) [Furui 1981], first and second order time derivatives of cepstra [Furui 1986] (also called delta and delta-delta or dynamic feature vectors or regression coefficients), relative spectral (RASTA) [Hermansky et al. 1993], frequency smoothing based on auditory properties in [Höge 1984], quantile based histogram equalization [Hilger and Ney 2001], vector Taylor series (VTS) [Moreno et al. 1996], statistical linear approximation (SLA) [Kim 1998], interacting multiple model (IMM) [Kim 2002], etc.

- Acoustic model compensation, which seeks to minimize the mismatch problem through some processing in the acoustic model. Early development of acoustic modeling methods in speech recognizers was relying on an isolated word *template* modeling employing methods such as vector quantization (VQ) [Gray 1984, Linde et al. 1980], nearest neighbor (NN) pattern classification [Cover and Hart 1967], dynamic time warping (DTW) [Rabiner et al. 1978], etc. These template-based classifiers employ different distance measure criteria to increase robustness. Statistical modeling using the hidden Markov model has also been developed in the classifier to model the acoustic feature vectors. Several well known techniques for the HMM-based classifier are parallel model combination (PMC) [Gales and Young 1996], maximum likelihood linear regression (MLLR) [Gales 1997], discriminative training [Juang and Katagiri 1992, Bahl et al. 1988], etc. Another direction in the speech recognizing is also developed based on neural networks [Lippmann 1987].

We have started the work on speech recognition by initially developing a speaker verification system [Setiawan et al. 2003a], which has now been improved to cope with various noise conditions. The work on the speech recognition was started by closely following the front-end development within the ETSI Aurora working group, where we have initially reported the speech level variation problem as described in [Setiawan et al. 2003b]. The work on the robustness issue was initially pursued in several areas, such as microphone array as presented in [Höge et al. 2004, Setiawan 2004], interactive multiple model and vector Taylor series as given in [Fingscheidt et al. 2004, Setiawan et al. 2004]. Finally, we started the work on the noise reduction as described, for example, in [Fingscheidt et al. 2005b, Setiawan et al. 2005b] due to the considerations presented in Chapter 1. We have also managed to work on a noise-related system for the ITU-T speech coding G.729.1 silence compression scheme which was standardized in June 2008 as G.729.1 Annex C: DTX/CNG Scheme [Setiawan et al. 2008].

Chapter 3

Speech Recognition System Description and Evaluation

The speech recognition system being evaluated comprises of the MFCC based Siemens front-end (SFE) and the HMM based Siemens back-end (SBE). Improvements have been developed in the front-end to exceed the performance of the ETSI standardized advanced front-end (AFE). To be able to fairly compare the performance of both front-ends a single experimental framework has to be defined. The framework defines the back-end, database, and task. In this case the database and task are following the Aurora 3 experimental framework as defined by the ETSI Aurora working group while the SBE is used as the common back-end. Several adjustments have to be done for both front-ends. The SFE needs to be modified to match the general configuration in the AFE and the AFE needs to be adjusted to perform optimally with the back-end. In this chapter, the original SFE and SBE system configuration is presented together with the database and task. Adjustments made on the SFE and AFE are later described in Chapter 6.

3.1 ETSI Distributed Speech Recognition (DSR) Front-End

A robust speech recognition front-end for mobile devices has been developed and standardized by the Speech Processing, Transmission, and Quality Aspects (STQ)-Aurora Working Group within ETSI for a distributed speech recognition (DSR) framework [Pearce 2000]. In the proposed framework, the MFCC feature vectors and energy coefficient are extracted at the *terminal front-end* and transmitted over a data network to the *server front-end*, where some remaining front-end operations are performed in the remote server before being processed by the back-end recognizer. This scheme requires the feature vectors to be compressed and encoded in the terminal before

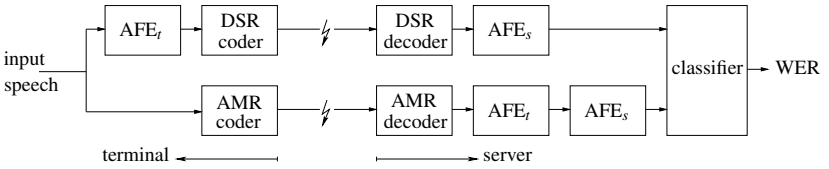


Figure 3.1: Speech processing architectures. The diagram block on the top shows the proposed DSR method with its AFE processing and the bottom one shows the widely spread speech processing architecture using AMR speech codec concatenated with the AFE. The subscripts *t* and *s* to AFE denote the terminal and server, respectively.

being sent. The decompression and decoding operations are thus performed in the server front-end which also calculates the derivative coefficients.

The first DSR front-end standard [ETSI STQ-Aurora 2000], was initially published in 2000 describing basic MFCC feature extraction and its feature compression and coding schemes. The second standard [ETSI STQ-Aurora 2003a], was initially published in 2002 adding noise robustness to the first standard. This is known as the advanced front-end (AFE). Extended front-end (XFE) [ETSI STQ-Aurora 2003c], and extended advanced front-end (XAFE) [ETSI STQ-Aurora 2003b], as the extensions of the previous standards were published in 2003 to support tonal language recognition and speech reconstruction capability [Sorin et al. 2004, Ramabadran et al. 2004].

The recognition performance of the first two front-ends alone is substantially better than their combination with the 3rd Generation Partnership Project (3GPP) Adaptive Multi-Rate (AMR) speech codec and the front-ends [ETSI STQ-Aurora 2001b; 2002], using a common experimental framework [Hirsch and Pearce 2000, Pearce and Hirsch 2000]. This has eventually pushed 3GPP to approve the latest ETSI standard DSR XAFE as an optional codec for Speech Enabled Services for release 6 in 2004. Figure 3.1 depicts the AFE being evaluated on the DSR and AMR speech processing architectures. It shows that the AFE in the proposed DSR architecture is divided into two parts, i.e., AFE$_t$ and AFE$_s$ referring to the terminal and server parts, respectively.

The robustness of the standardized front-ends is obviously shown in the AFE. The techniques described in the AFE [Macho et al. 2002] are mostly based on speech enhancement techniques. The AFE mainly consists of a frequency domain noise reduction scheme called two-stage mel-warped Wiener filter [Agarwal and Cheng 1999, Noé et al. 2001], and SNR-dependent waveform processing [Macho and Cheng 2001] to enhance the time domain signal representation. Blind equalization [Mauuary 1998] is applied to the MFCC feature vectors to reduce the convolutional effect. Frame dropping [de Veth et al. 2001] is finally used to reduce the insertion errors. Some additional configurations of the front-end are as follows:

- Supports three sampling rates, i.e., 8 kHz, 11 kHz, and 16 kHz.
- 25/10 ms frame length/shift.
- 23 normalized triangular mel filterbanks applied to the denoised power spectrum for 8 and 11 kHz processing. For 16 kHz processing, 3 additional filterbanks are used for the upper 4 kHz bandwidth.
- 12 MFCC and 1 energy feature as static coefficients. The energy feature is a weighted sum of log-energy and $c0$ coefficient.
- 39 dimensional feature vector, i.e., 13 static coefficients and their first and second order derivatives.

Note that while the AFE provides an additional module to process the 16 kHz utterances, the 11 kHz processing is conducted in the same way as the 8 kHz processing by previously performing a downsampling to 8 kHz.

3.2 Front-End and Back-End System Description

3.2.1 Siemens Front-End (SFE)

The front-end is based on the mel frequency cepstral coefficients (MFCC) feature extraction method and taking an input speech signal having a sampling frequency of 8 kHz [Andrassy et al. 2001]. The diagram block of the front-end is depicted in Figure 3.2. It mainly consists of the spectral analysis, noise reduction, cepstrum calculation, channel compensation, and feature processing. Each of the components is described as follows:

- Spectral analysis. This generally consists of the framing, pre-emphasis, windowing, and Fourier transform. The framing takes a frame length of 32 ms with a frame shift of 15 ms. Pre-emphasis is done using a first-order high pass filter with the coefficient 0.95. Hamming windowing is then applied to the frame before the $N_{FFT} = 256$ points FFT operation. The noisy speech spectral power $|Y_k(m)|^2$ is finally calculated for each frequency bin $k = 1 \cdots N_{FFT}/2 + 1$ and frame index m.
- Noise reduction. The additive noise is removed using the recursive least-squares (RLS) method following [Beaugeant et al. 2002] where the noise reduction filter is calculated as

$$G_k^{RLS}(m) = \frac{\sum_{\ell=0}^{m} \rho_Y^{m-\ell} \cdot |Y_k(\ell)|^2}{\sum_{\ell=0}^{m} \rho_Y^{m-\ell} \cdot |Y_k(\ell)|^2 + \alpha \cdot \sum_{\ell=0}^{m} \rho_N^{m-\ell} \cdot |N_k(\ell)|^2}, \tag{3.1}$$

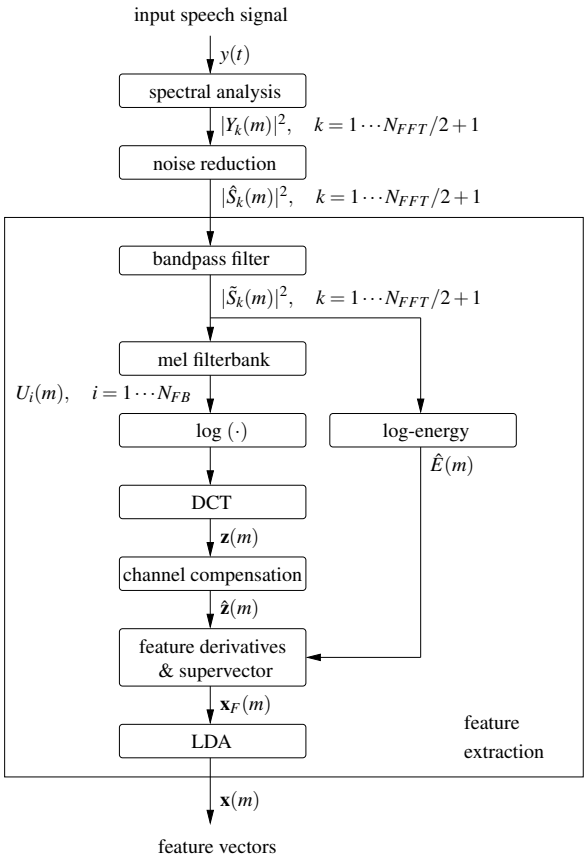


Figure 3.2: The diagram block of the baseline Siemens front-end (SFE).

to obtain the clean speech spectral power estimate $|\hat{S}_k(m)|^2$ using

$$|\hat{S}_k(m)|^2 = G_k^2(m) \cdot |Y_k(m)|^2. \tag{3.2}$$

The variable α denotes the overestimation factor while ρ_Y and ρ_N denote the smoothing coefficients for noisy speech and noise, respectively. The noise spectral power $|N_k(\ell)|^2$ is estimated by means of a noise power spectral density estimator as described in Section 4.1.1.

- Bandpass filter. The power spectrum within the frequency range from 180 to 4000 Hz is taken with frequencies below 180 Hz are set to zero.

- Log-energy. The normalized log-energy is calculated from the bandpass outputs by

$$\hat{E}(m) = E(m) - \overline{E(m)}, \tag{3.3}$$

with

$$\overline{E(m)} = \begin{cases} E(m) & \text{if } m = 0, \\ (1-0.9) \cdot \overline{E(m-1)} + 0.9 \cdot E(m-1) & \text{otherwise,} \end{cases} \tag{3.4}$$

where $\hat{E}(m)$ and $E(m) = 10 \cdot \log_{10} \sum_{k=1}^{N_{FFT}/2+1} |\tilde{S}_k(m)|^2$ denoting the normalized and unnormalized log-energy coefficients, respectively. Note that $|\tilde{S}_k(m)|^2$ refers to the estimate of the clean speech power spectrum after bandpass filtering.

- Mel filterbank. Instead of taking the power spectrum for the $N_{FB} = 15$ mel filterbank inputs as required by the classical MFCC computation, the spectral magnitudes are taken. The mel filterbank is depicted in Figure 3.3 having the center frequencies equidistantly spaced in the mel frequency domain with $\Delta_{\text{mel}} = 133.66$ mel and the filter width of $2 \cdot \Delta_{\text{mel}}$ [Siemens 2000].

- log $(\cdot)$. The 15 filterbank outputs $U_i(m)$, $i = 1 \cdots N_{FB}$, are subject to a scaled logarithm $c \times 10 \cdot \log_{10} U_i(m)$, where $c = 0.4$ and $N_{FB} = 15$.

- DCT. The discrete cosine transform (DCT) is taken resulting in a reduced number of coefficients to 12 which are known as the MFCC feature vectors $\mathbf{z}(m) = [\, z_1(m) \; \cdots \; z_{12}(m) \,]^T$. The DCT mainly aims at decorrelating the components of the vector.

- Channel compensation. The convolutive effect is handled by the maximum likelihood channel compensation (MLCC) technique [Hauenstein and Marschall 1995]. The effect

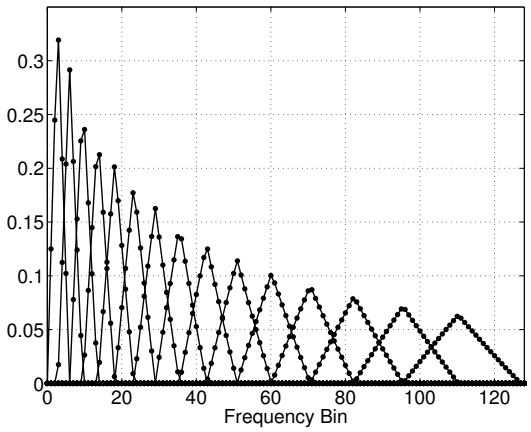


Figure 3.3: The 15 triangular-shaped mel filterbank as used in the Siemens front-end (SFE).

can be originated, for example, from the use of different channels (microphones, transmissions, etc.). This is categorized as a form of stationary distortion which gives an offset to the power spectrum (convolution on speech signal). The feature vectors observed in a particular dimension is assumed to be uncorrelated with the ones from its neighboring dimensions. Note that the MLCC is only applied to the MFCC feature vectors and not to the log-energy feature.

Using MLCC, a channel offset $\mu_{o_i}(m)$ is obtained recursively as

$$\mu_{o_i}(m) = \frac{1}{1+\alpha_i(m)} \left[\mu_{Z_i}(m) + \mu_i \cdot \alpha_i(m)\right] \qquad i \in [1,12], \tag{3.5}$$

where

$$\mu_{Z_i}(m) = \frac{m-1}{m}\,\mu_{Z_i}(m-1) + \frac{1}{m}\,z_i(m), \tag{3.6}$$

$$\alpha_i(m) = \frac{1}{m}\frac{\sigma_{o_i}^2}{\sigma_i^2}. \tag{3.7}$$

The terms μ_i, σ_i^2, and $\sigma_{o_i}^2$ are estimated from the training data which are utterances recorded on many different channels or training sessions. Utterances taken from a particular channel are assumed to have a similar convolutive effect, hence a feature mean and a feature

variance is calculated out of this channel. A set of feature means and variances is obtained based on the number of channels available in the training data. The terms μ_i and σ_i^2 are simply the mean and variance, respectively, of the feature means. The term $\sigma_{o_i}^2$ is estimated by taking the mean of feature variances.

As $m \rightarrow \infty$ the mean channel offset estimate is simply the mean $\mu_{Z_i}(m)$ which is exactly the CMS technique. The CMS is actually a ML estimate under the Gaussian assumption. It is also a MAP estimate under the Gaussian assumption given a uniform prior. The value of $\alpha_i(m)$ in (3.7) can be prevented to approach zero by setting $m = m_{max}$ for $m \geq m_{max}$. This could be advantageous to better adapt to a changing channel. Finally, the offset is simply subtracted from the original MFCC feature vectors

$$\hat{z}_i(m) = z_i(m) - \mu_{o_i}(m) \qquad i \in [1,12].$$

- Feature derivatives and supervector. First and second-order feature derivatives are calculated for both the MFCC feature vectors and log-energy feature by

$$\Delta_i(m) = v_i(m) - v_i(m-3), \tag{3.8}$$
$$\Delta\Delta_i(m) = \Delta_i(m+3) - \Delta_i(m), \tag{3.9}$$

where

$$v_i(m) = \begin{cases} \hat{z}_i(m) & i \in [1,12], \\ \\ \hat{E}(m) & i = 13. \end{cases} \tag{3.10}$$

At this stage, frame m of the speech signal has been converted into a single feature vector $\mathbf{x}_{F_1}(m) = [\ v_1(m)\ \cdots\ v_{13}(m)\ \Delta_1(m)\ \cdots\ \Delta_{13}(m)\ \Delta\Delta_1(m)\ \cdots\ \Delta\Delta_{13}(m)\ ]^T$, where the log-energy feature is simply appended to be the 13^{th} component of the feature vector.

After calculating the feature derivatives, two consecutive feature vectors are concatenated to yield a single supervector

$$\mathbf{x}_F(m) = [\ \mathbf{x}_{F_1}(m)\ \mathbf{x}_{F_1}(m-1)\ ]^T.$$

This results in the supervector having 78 components.

- LDA processing. The aim is to preserve the discriminant power between several classes while reducing the dimensionality of the feature vectors. The LDA is applied on the supervector following [Haeb-Umbach and Ney 1992] to produce a $d = 24$ dimensional feature vector $\mathbf{x}(m) = [\ x_1(m)\ \cdots\ x_{24}(m)\ ]^T$. In general, the LDA aims at reducing the dimensionality of the feature vector to achieve a memory efficient solution and to be compatible

with the implemented HMM modeling using diagonal covariance matrices having the same variances [Varga et al. 2002].

3.2.2 Siemens Back-End (SBE)

Like most automatic speech recognition systems, we are using a state of the art speaker independent HMM-based phoneme and whole word recognizer [Astrov et al. 2003, Varga et al. 2002, Bauer 2001]. Moreover, the following specifications are used to deal with the digit task problem in the Aurora 3 framework:

- Whole word and continuous density HMM modeling.
- Each digit is modeled with eight subwords having 3 segments per subword. The *silence* is modeled with a single segment. An additional `[nib]` model is introduced being a two-subword model having 3 segments per subword to model the residual noise.
- Each segment is having one state with different numbers of Gaussian mixture densities assigned to it.
- A total of 271 segments and 1104 Gaussian mixture densities are available; the diagonal covariance matrix shares a single common variance.

The realization of the whole word digit HMM modeling using a phoneme recognizer is done by partitioning the digit into several *subwords*. Each subword is constructed from several segments. In our case, the subword having three segments representing the initial sound, middle sound, and final sound of a phoneme. A certain amount of HMM states is assigned to each segment depending on the task but for the Aurora 3 task it is only one HMM state assigned to each segment.

3.3 Front-End Module Extensions

Additional root compression and cepstral smoothing methods as presented in this section have been implemented to boost the recognition performance of the SFE. Both methods only require a small amount of additional complexity and memory requirements.

3.3.1 Root-Cepstral Coefficients

It has been indicated in [Alexandre and Lockwood 1993] that the logarithmic deconvolution is not necessarily the optimal scheme for speech analysis. Several functions exist as an alternative to the log $(\cdot)$ operator, such as

$$f(\cdot) = \begin{cases} \frac{1}{\gamma}\left[(\cdot)^{\gamma} - 1\right] & \gamma \neq 0, \\ \log(\cdot) & \gamma = 0, \end{cases} \tag{3.11}$$

which is known as the generalized logarithmic function [Kobayashi and Imai 1984], and a direct root function [Lim 1979]:

$$f(\cdot) = (\cdot)^{\gamma}, \tag{3.12}$$

where γ is a real number with $-1 < \gamma < 1$.

It has been shown that an optimal value of γ has the following advantages over the logarithmic operation:

- Better estimation of the pole-zero model of the vocal tract impulse response which is represented by the full cepstral coefficients calculated on compressed spectral coefficients in the linear frequency domain.
- More robust cepstral representation in the presence of background noise, especially for the high-order cepstral indices.
- Bigger effect resulting from the first-order pre-emphasis operation.

All of these lead to a more robust representation of acoustic feature vectors in the cepstral domain. The improvements were also reported even when a noise reduction technique is used. Please note that initial development of the root function was in the linear frequency domain.

The use of mel-based root-cepstral coefficients for speech recognition, where the root function is applied after the mel frequency warping, was initially reported in [Lockwood and Boudy 1992] and showed significant robustness in car noise environments. Later experiments conducted on the Aurora 2 corpus also showed improvements in several noisy conditions [Yapanel et al. 2001]. Analysis in the cepstrum domain is presented in [Sarikaya and Hansen 2001].

In this thesis, the root function or compression in (3.12) is used to compute the mel-based root-cepstral coefficients as an alternative to the mel-based log-cepstral coefficients widely known as the MFCC.

3.3.2 Cepstral Smoothing

Feature normalization and smoothing techniques are done to further reduce the mismatch between training and testing conditions. It has been shown in [Chen et al. 2005; 2002] that the techniques allow improvements for ASR systems especially for small vocabulary tasks. The mean and variance feature normalization are performed using MLCC and LDA techniques, respectively [Varga et al. 2002]. In this thesis, we additionally apply the smoothing using the following causal auto-regressive moving average (ARMA) low-pass filter:

$$\hat{x}_i(m) = \frac{\sum_{\ell=1}^{L} \hat{x}_i(m-\ell) + \sum_{\ell=0}^{L} x_i(m-\ell)}{2L+1}, \tag{3.13}$$

where $\hat{x}_i(m)$ and $x_i(m)$ denoting the filtered and unfiltered cepstral coefficients for each cepstral dimension i and L denotes the length of the filter. The smoothing is performed on the cepstral coefficients after the DCT before the MLCC and LDA, hence we refer to this as the cepstral smoothing technique. Several different values of $L = 1 \cdots 6$ are investigated in this thesis.

3.4 Performance Evaluation

The performance of the approaches is measured in *word accuracy* (ACC),

$$\text{ACC} = \frac{N-D-S-I}{N} \times\ 100\%, \tag{3.14}$$

and *word recognition rate* (WRR),

$$\text{WRR} = \frac{N-D-S}{N} \times\ 100\%. \tag{3.15}$$

The symbol N denotes the total number of reference words and the errors D, S, I denote the total number of words being deleted, substituted, and inserted, respectively. The widely used *word error rate* (WER) is defined as $1 - ACC$. The relative improvement of a certain performance measurement value p [%] over a reference value q [%] is calculated as

$$\text{relative [ACC / WRR]} = \frac{p-q}{100-q} \times 100\%. \tag{3.16}$$

3.5 Databases and Tasks

Car noise databases are of particular interest in this thesis. The fact that a database was recorded in a real car environment and not using an artificially added noise signal is also preferred. This is necessary since it simulates the real conditions of the target deployment where the speech production is affected by the surrounding environment (e.g., the Lombard effect).

3.5.1 Aurora 3 German

Table 3.1: Recognition task using the Aurora 3 German database.

Sampling Rate	Task	Test Sentences/Words	Lexicon Entries
8 kHz	digits	well matched: 897/5009	11
		medium mismatch: 241/1366	11
		high mismatch: 394/2162	11

Aurora 3 German digits database [ETSI STQ-Aurora 2001a], as a subset of the 16 kHz German SpeechDat-Car (SDC) database [Moreno et al. 2000], is primarily used to evaluate the performance of the approaches. Downsampling and other adjustments on the original database were done by the Aurora working group to create the Aurora 3 version. The database was recorded in a real car environment and divided into three different training and test cases, i.e., well matched, medium mismatch, and high mismatch as shown in Table 3.1. The overall weighted performance is computed with the following weights: 0.4, 0.35, and 0.25 for well matched, medium mismatch, and high mismatch, respectively.

Sources of mismatch are different microphone types and driving conditions. A close-talking microphone and hands-free microphone types were used to record the utterances simultaneously. The driving conditions are high speed good road, low speed rough road, stopped with motor running, and town traffic. In addition to that, the car was placed in various configurations: climate control on/off, left front window open/closed, right front window open/closed, rear window open/closed, sunroof open/closed, windshield wipers on/off.

3.5.2 SpeechDat-Car Spanish

The SpeechDat-Car database collection [Moreno et al. 2000] aims at developing large-scale speech resources for a wide range of languages and for in-car applications (voice dialing, car

Table 3.2: Recognition task using the SpeechDat-Car Spanish database.

Sampling Rate	Language	Task	Test Sentences/Words	Lexicon Entries
11.025 *kHz*	Spanish	commands	858/858	248
		city names	508/508	232

accessories control, etc.). The recordings were taken using five different microphones, one being a close-talk microphone, three being hands-free microphones, and another one being located at the far-end location of the GSM communications system. The first four microphones were recording wideband audio signals directly in the car sampled at 16 *kHz* and the last one was recording GSM signals transmitted from the car sampled at 8 *kHz*. The location of the three hands-free microphones are: one on the ceiling of the car near the A-pillar, one on the ceiling of the car behind the sunvisor that is in front of the speaker, and one on the ceiling of the car over the mid-console (near the rear-view mirror).

Car environments are defined with the following conditions: car stopped with motor running, car in town traffic, car in town traffic with noisy conditions, car moving at a low speed with rough road conditions, car moving at a low speed with rough road conditions and with noisy conditions, car moving at a high speed with good road conditions, car moving at a high speed with good road conditions and with audio equipment on. Only outputs of the hands-free microphones are used for the testing and they were downsampled to 11.025 *kHz*. The testing set is described in Table 3.2.

3.5.3 SPEECON Spanish

Table 3.3: Recognition task using the SPEECON Spanish database.

Sampling Rate	Language	Task	Test Sentences/Words	Lexicon Entries
11.025 *kHz*	Spanish	appl. words	1555/1555	535
		city names	1484/1484	5159
		appl. words	1507/1507	484

Speech-Driven Interfaces for Consumer Devices (SPEECON) [Siemund et al. 2000, Iskra et al. 2002], is a project focusing on collecting linguistic data for automatic speech recognizer training. It is a shared-cost project funded by the European Commission under Human Language Technologies which is part of the Information Society Technologies programme. The recording in the car environment was done using three different microphones sampled at 16 *kHz*. Two microphones, one simulating the mobile phone position and the other simulating the hands-free

position (below the chin of the speaker), are identified as close distance microphones. Another medium distance microphone is installed within the range of 0.5 - 1 m.

Stationary noise such as car engine and instantaneous kind of noise such as the one produced by the wipers are expected in the recordings. SPEECON differs from its SpeechDat predecessors in that it contains spontaneous speech recordings and an extensive number of application specific commands. The original database was then downsampled to 11.025 kHz and two microphone outputs, i.e., from hands-free and medium distance, are used for testing. The testing set is shown in Table 3.3.

3.6 System Requirements in Embedded Devices

Implementation of a certain application in embedded devices requires careful treatment since the application must fulfill the system requirements which are usually limited and heavily depending on the embedded devices themself. Several aspects need to be considered are the amount of memory the application uses (the *footprint*), the amount of processing power in million instructions per second (MIPS) it needs, and the need of support peripheral, e.g., for data transfer and controlling.

Instead of using the complex instruction set computing (CISC) processor type usually deployed in the non-embedded computing devices, current processors for embedded devices are of reduced instruction set computing (RISC) type. These processors are generally based on certain architectures such as the ARM and the microprocessor without interlocked pipeline stages (MIPS) architectures. Here we regard a concrete case of an ARM based processor as used in the mobile phones. The following system requirements for a specific type of mobile phone are presented as a realistic reference for the embedded devices system requirements:

- Processor: ARM9 which is capable of supporting up to 300 MIPS.
- Memory: 64 - 128 MByte NAND Flash and 64 MByte SDRAM.

System requirements for basic operating mode of the mobile phone already requires around 42 MByte NAND Flash and 14 MByte SDRAM. This includes the program codes, the constants or static data and the local variables for modem, operating system, device layer, multimedia codecs, etc. Note that the user data stored in NAND Flash need to be considered as well. The rest of the memory is available for any additional application.

There is another aspect in the deployment of a speech recognition system, i.e., the processing architecture. The most common architectures are the existing mobile voice telephony which

is a widely-implemented architecture, and the recently proposed distributed speech recognition architecture. Both are mainly designed for the use in the mobile wireless world. The main difference lies in the data being transmitted through the network. The first transmits a parameterized representation of speech employing a speech codec and the latter transmits the extracted feature vectors from the front-end. Further discussion about the pursued DSR standards was presented in Section 3.1.

As mentioned earlier, current speech recognition systems may use different processing architecture. Regardless of the architecture, usually it is only the front-end complexity and footprint which are taken into account since the back-end which requires a much higher processing power and memory is located at a server in the network and not at the mobile terminal. It is also possible to divide the front-end into two parts to be located at the terminal and the server sides which allows further reduction of the complexity and memory requirements at the terminal. Below is the minimal system configuration which represents the optimized version of the baseline front-end:

- 22 MIPS processing power.
- 15 KByte in ROM for HMM data per language.
- 40 KByte code size and 20 KByte heap size.

It is obvious that there is still a considerable processing power and memory space available for an additional processing in the baseline front-end. However, it is always better to keep the complexity and computational load of the whole front-end as low as possible in case of higher performance is desired.

Chapter 4

Frequency Domain Noise Reduction

In this chapter an overview of single-channel speech enhancement approaches based on the short-time spectral amplitude (STSA) estimation is presented. The approaches are restricted to those which aim at enhancing a speech signal degraded by statistically independent additive noise, also widely known as *noise reduction* approaches. This setup constitutes one of the most difficult situations of speech enhancement, since no reference signal to the noise is assumed available, and the clean speech cannot be preprocessed prior to being affected by noise.

The choice of short-time spectral amplitude estimation is motivated by the fact that some of the approaches are relatively simple to implement and they usually outperform more elaborate approaches. The approach is suitable for embedded devices such as in hands-free telephony which require low computational load and still maintain an excellent performance. This also means that the discussion is limited to frequency domain algorithms instead of time domain ones. Although there exists an equivalence between time and frequency domain algorithms, they differ from the implementation point of view [Gustafsson 1999].

Speech signal is usually evaluated based on two criteria, i.e., speech quality and speech intelligibility. The speech quality is a subjective measure which reflects on the way the signal is perceived by the listeners. It can be expressed in terms of how pleasant the signal sounds. Speech intelligibility is an objective measure of the amount of information which can be extracted by listeners from the given signal, whether the signal is clean or noisy. There exist a subjective and an objective test method to assess both the speech quality and speech intelligibility although more attention has been given to the speech quality.

The subjective test method for speech quality is usually measured in terms of mean opinion score (MOS) and conducted following ITU-T Recommendation P.800 [ITU-T P.800 1996] and P.835 [ITU-T P.835 2003], where the latter is more suitable for speech enhancement purpose.

The corresponding objective test method is usually performed by means of ITU Recommendation P.862 [ITU-T P.862 2001] and P.862.2 [ITU-T P.862.2 2007], segmental SNR, log likelihood ratio (LLR), Itakura-Saito distance measure, etc., although they are not necessarily suitable for speech enhancement purpose [Hu and Loizou 2008, Grundlehner et al. 2005]. Several approaches have been proposed to subjectively assess speech intelligibility, such as the articulation index (AI) and speech transmission index (STI). The corresponding objective test method is usually utilizing the approaches used to assess the speech quality. Recent work reveals that an automatic speech recognition experiment offers a reliable way to objectively assess speech intelligibility [Liu et al. 2006].

Speech signal is non-stationary, however, it can be considered as a quasi-stationary signal on a short-time basis. This usually implies an analysis of speech with an analysis window of 10 - 30 *ms* length. The short-time spectral analysis/synthesis is performed using the short-time Fourier transform (STFT) [Portnoff 1980]. Based on this short-time analysis framework, the problem of speech degraded with statistically independent additive noise is formulated as

$$y(t) = s(t) + n(t), \tag{4.1}$$

where $y(t)$, $s(t)$, and $n(t)$ denote the time domain short-time noisy speech, clean speech, and noise signals, respectively. Furthermore, taking the Fourier transform of (4.1) yields

$$Y_k(m) = |Y_k(m)| \cdot e^{\theta_{Y_k}(m)} = S_k(m) + N_k(m), \tag{4.2}$$

where the complex variables $Y_k(m)$, $S_k(m)$, and $N_k(m)$ denote the short-time Fourier transform of $y(t)$, $s(t)$, and $n(t)$, respectively and k denotes a particular frequency bin. The magnitude and phase of $Y_k(m)$ is denoted by $|Y_k(m)|$ and $\theta_{Y_k}(m)$, respectively. It is assumed that an appropriate window function has been applied to obtain the short-time equation in (4.2).

In the STSA context, a collection of speech samples taken in a short-time analysis is generally referred to as a *frame*. The term *frame length* denotes the amount of speech samples taken for the short-time analysis and *frame shift* denotes the difference between the first sample of the current frame and the first sample of the previous frame. Both terms are usually expressed in milliseconds (*ms*). When the frame shift is half of the frame length for example, then it has 50 % overlap between consecutive frames. It is common to do the short-time analysis/synthesis in an overlapping manner. The *overlap-add* [Stockham, Jr. 1966] method is generally employed.

The objective of speech enhancement approaches based on the short-time spectral amplitude is to obtain $\hat{S}_k(m)$ as an estimate of $S_k(m)$. This is usually done by finding a weighting rule $G_k(m)$

$$\hat{S}_k(m) = G_k(m) \cdot Y_k(m). \tag{4.3}$$

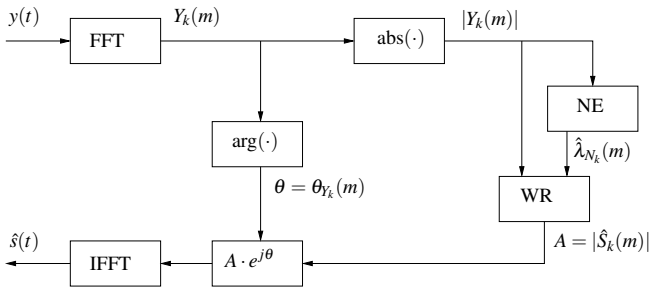


Figure 4.1: General structure of noise reduction scheme. The noise estimation (NE) block estimates the noise PSD $\lambda_{N_k}(m)$. The noise reduction weighting rule (WR) block calculates $G_k(m)$ to yield the clean speech PSD estimate $|\hat{S}_k(m)|$. The FFT/IFFT block refers to the STFT operation.

In the problem formulation, the weighting rule $G_k(m)$ is not constrained to a real-valued variable. However, the final formulation of $G_k(m)$ often results in a non-negative real-valued variable. Nevertheless, we keep the notation $G_k(m)$ as a complex variable.

Principally, the clean speech short-time spectral magnitude estimate is more important than the short-time spectral phase estimate for speech intelligibility and quality [Lim and Oppenheim 1979]. It has been shown in [Wang and Lim 1982] through a sequence of experiments, that the short-time phase is not important for speech reconstruction when combined with an independently estimated short-time spectral magnitude. The phase of the noisy speech $\theta_{Y_k}(m)$ is often used in combination with the estimated clean speech short-time spectral magnitude $|\hat{S}_k(m)|$. It is also common to assume that there is no statistical dependency between spectral components.

Figure 4.1 shows the general structure of a noise reduction scheme. The noise power spectral density (PSD) is denoted with $\lambda_{N_k}(m)$ and is defined below with other PSDs

$$\lambda_{Y_k}(m) \triangleq E\{|Y_k(m)|^2\}, \tag{4.4}$$

$$\lambda_{S_k}(m) \triangleq E\{|S_k(m)|^2\}, \tag{4.5}$$

$$\lambda_{N_k}(m) \triangleq E\{|N_k(m)|^2\}, \tag{4.6}$$

where $E\{\cdot\}$ denotes the ensemble average or the expectation operator. The cross PSD between the above spectral components is written as $\lambda_{Y_kS_k}(m)$ for example, to denote the cross PSD between the noisy and clean speech spectral components

$$\lambda_{Y_kS_k}(m) \triangleq E\{Y_k^*(m) \cdot S_k(m)\}. \tag{4.7}$$

As described above, the problem in noise reduction can be categorized into spectral analysis/synthesis, computation of a weighting rule, and an estimation of noise PSD. Spectral analysis is focusing on finding the noisy speech PSD estimate which is usually computed with the modified periodogram method by applying a window function and FFT to the speech frame. Taking a particular window function, averaging and smoothing the modified periodogram over time or frequency are often done to reduce the bias and variance of the estimate. The estimation of noise PSD given the noisy speech PSD is beyond the scope of this thesis but several state-of-the-art techniques are presented in Section 4.1. The actual focus in this thesis is restricted to the weighting rule computation and the noise terms $E\{|N_k(m)|^r\},\ r \in \mathbb{N}$ occurred in the formulation should always be calculated based on the estimated noise PSD $\hat{\lambda}_{N_k}(m)$.

4.1 Noise Estimation Techniques

Noise estimation techniques in speech enhancement are actually aimed at the estimation of the noise PSD $\lambda_{N_k}(m)$. Early noise estimation techniques are based on voice activity detection to indicate the speech and non-speech part of an utterance and simply doing some averaging and/or smoothing on the non-speech part to obtain the best estimate of the noise PSD. It is motivated by the assumption that the background noise is stationary during the non-speech part. A more sophisticated noise estimator such as minimum statistics [Martin 2001; 1994] which is not using any VAD information. It just has the drawback that a higher amount of static memory is required. Both noise estimators are used in this thesis.

4.1.1 Three-State Voice Activity Driven Noise PSD Estimation

An energy-based noise estimation technique as shown in Figure 4.2 is used. The algorithm employs a widely used smoothing technique based on a VAD-like decision logic. It takes the smoothed noisy observation $\overline{|Y_k(m)|^2}$ as the input, where

$$\overline{|Y_k(m)|^2} = (1-\varepsilon_Y)\cdot\overline{|Y_k(m-1)|^2} + \varepsilon_Y \cdot |Y_k(m)|^2, \tag{4.8}$$

and compares $\overline{|Y_k(m)|^2}$ with an adaptive control parameter $\Theta_k(m)$ to determine whether the noise estimate needs an update.

We assume that it is not necessary to update the noise PSD estimate if the term $\overline{|Y_k(m)|^2}$ is greater than $2\cdot\Theta_k(m)$, otherwise a new noise PSD estimate is calculated for the current frame m. The parameter $\varepsilon_{up} = 0.03125$ and 0.125 for 8 and $11.025\ kHz$ sampling frequency, respectively, is chosen to slowly adapt to the environment considering that speech might be present

if $\left(\overline{|Y_k(m)|^2} \leq 2 \cdot \Theta_k(m)\right)$,
  if $\left(\overline{|Y_k(m)|^2} \geq \hat{\lambda}_{N_k}(m-1)\right)$,
    $\hat{\lambda}_{N_k}(m) = (1-\varepsilon_{up}) \cdot \hat{\lambda}_{N_k}(m-1) + \varepsilon_{up} \cdot \overline{|Y_k(m)|^2}$;
  else
    $\hat{\lambda}_{N_k}(m) = (1-\varepsilon_{dn}) \cdot \overline{|Y_k(m)|^2} + \varepsilon_{dn} \cdot \hat{\lambda}_{N_k}(m-1)$;
  end
end

Figure 4.2: Energy-based noise PSD estimation technique

if $\overline{|Y_k(m)|^2} \geq \hat{\lambda}_{N_k}(m-1)$, while $\varepsilon_{dn} = 0.25$ and 0.5 for 8 and 11.025 kHz sampling frequency, respectively, is chosen to quickly adapt to the environment assuming that at this step the current frame is noise or silence.

The control parameter $\Theta_k(m)$ is also adapted on a frame basis as depicted in Figure 4.3. If $\overline{|Y_k(m)|^2}$ is greater than $2 \cdot \Theta_k(m)$, the value $\Theta_k(m)$ should always be increased through a multiplication with a step-size constant $\Delta = 1.03$. This ensures that $\Theta_k(m)$ adapts quickly whenever it is assumed that the current frame is a noise frame and adapts slowly using the step-size parameter if it is assumed that speech is present.

if $\left(\overline{|Y_k(m)|^2} < \Theta_k(m)\right)$,
  $\Theta_k(m+1) = \overline{|Y_k(m)|^2}$;
else
  if $\left(\overline{|Y_k(m)|^2} > 2 \cdot \Theta_k(m)\right)$, $\Theta_k(m+1) = \Theta_k(m) \cdot \Delta$; **end**
end

Figure 4.3: Control parameter adaptation for noise estimation

4.1.2 Minimum Statistics Noise PSD Estimation

The minimum statistics noise PSD estimation technique was originally proposed in [Martin 1994] and later improved in [Martin 2001]. This technique basically tracks the minimum value of the smoothed noisy power spectra within a finite window and multiplied by a constant that compensates the estimate for a possible bias. This technique performs very well in speech enhancement and is implemented without any modifications. It is computationally more expensive than the previous technique.

As stated in [Martin 2001], it is based on the observation that even during speech activity a short term PSD estimate of the noisy signal frequently decays to values which are representative

of the noise power level. Basic approach of the minimum statistics is implemented by tracking the minimum value of smoothed periodogram within a sliding window of approximately 1.5 *sec*. For a sampling frequency of $f_s = 8000\ Hz$ the periodogram is obtained from a 50 % frame overlap using a $N_{FFT} = 256$ point FFT after applying a particular window $h(\mu)$ to the frame satisfying $\sum_{\mu=0}^{N_{FFT}-1} h^2(\mu) = 1$. It is then recursively smoothed over time using

$$\overline{|Y_k(m)|^2} = \varepsilon_Y \cdot \overline{|Y_k(m-1)|^2} + (1-\varepsilon_Y) \cdot |Y_k(m)|^2, \tag{4.9}$$

where ε_Y denotes a smoothing coefficient. The constant ε_Y is obtained by equating the variance of smoothed periodogram $\overline{|Y_k(m)|^2}$ to the variance of a moving average estimator within a window span of 0.2 *sec* assuming statistically independent periodograms. This yields a smoothing constant $\varepsilon_Y \approx 0.85$.

This basic approach has been observed to provide a rough estimate of noise PSD which has the following drawbacks:

- Having a constant ε_Y close to 1 increases the spectral bias resulting in the widening of spectral peaks during speech activity. This leads to inaccurate noise estimates.
- Choosing a small value of ε_Y results in a larger variance of noise estimates.
- The noise estimates were shown to be biased towards lower values.
- In the case of increasing noise power, the tracking lags behind.

These problems are addressed with the introduction of an optimal time varying smoothing, error monitoring, and a bias correction factor. The problem of delay and computational complexity inherent in this method is particularly handled through the implementation of minimum search which is not discussed in detail in this section.

The optimal smoothing procedure considers the following first order smoothing equation

$$\overline{|Y_k(m)|^2} = \varepsilon_{Y_k}(m) \cdot \overline{|Y_k(m-1)|^2} + (1-\varepsilon_{Y_k}(m)) \cdot |Y_k(m)|^2, \tag{4.10}$$

where $\varepsilon_{Y_k}(m)$ denotes a time and frequency dependent smoothing parameter. The optimal value of $\varepsilon_{Y_k}(m)$ is obtained by solving

$$\begin{aligned} \hat{\varepsilon}_{Y_k}(m) &= \arg\min_{\varepsilon_{Y_k}(m)} E\left\{ \left[\overline{|Y_k(m)|^2} - \lambda_{N_k}(m) \right]^2 \middle| \overline{|Y_k(m-1)|^2} \right\} \\ &= \left[1 + \left(\frac{\overline{|Y_k(m-1)|^2}}{\lambda_{N_k}(m)} - 1 \right)^2 \right]^{-1}. \end{aligned} \tag{4.11}$$

In a practical implementation its maximum value is limited to $\varepsilon_{\max} = 0.96$ and $\lambda_{N_k}(m)$ is replaced by $\hat{\lambda}_{N_k}(m-1)$. To guarantee a reliable operation under all circumstances, an error monitoring procedure is formulated where the optimal smoothing factor is multiplied by a time varying constant factor $\varepsilon_c(m)$ and an SNR-dependent value $\varepsilon_{\min}$ is set as the lower limit.

4.2 State-of-the-Art STSA Estimators

4.2.1 Spectral Subtraction

The idea of spectral subtraction is formulated as an approach of estimating the clean speech magnitude spectrum by subtracting the knowledge contained in the noise spectrum from that contained in the noisy speech spectrum. The knowledge was initially gained from the magnitude spectrum but later developments have utilized a generalization by taking an exponent r to the magnitude spectrum. This idea has also been applied to other weighting rules such as Wiener filtering.

The *magnitude spectral subtraction* (MSS) [Boll 1979] is the simplest form of the spectral subtraction family. The weighting rule is formulated based on (4.2) where the magnitude estimator $|\hat{S}_k(m)|$ is approximated by

$$|\hat{S}_k(m)| = |Y_k(m)| - E\{|N_k(m)|\}. \tag{4.12}$$

Please note that the noise magnitude spectrum $|N_k(m)|$ is replaced with its expected value. The magnitude spectral subtraction weighting rule is therefore formulated as

$$G_k^{MSS}(m) = \frac{|Y_k(m)| - E\{|N_k(m)|\}}{|Y_k(m)|}. \tag{4.13}$$

The next variant is referred to as *power spectral subtraction* (PSS) [Lim and Oppenheim 1979] or *correlation subtraction* method [Lim 1978]. The noisy PSD estimate is given as

$$|Y_k(m)|^2 = |S_k(m)|^2 + |N_k(m)|^2 + S_k(m) \cdot N_k^*(m) + S_k^*(m) \cdot N_k(m), \tag{4.14}$$

with $[\cdot]^*$ denoting the complex conjugate operator. This can also be written in terms of a true noisy speech PSD estimate as

$$|Y_k(m)|^2 = \lambda_{S_k}(m) + \lambda_{N_k}(m) + \lambda_{N_k S_k}(m) + \lambda_{S_k N_k}(m). \tag{4.15}$$

Assuming $n(n)$ is zero mean and uncorrelated with $s(n)$, where $n(n)$ and $s(n)$ denote the discrete

samples of time domain short-time continuous noise and clean speech signal of $n(t)$ and $s(t)$, respectively, then the terms $\lambda_{N_k S_k}(m)$ and $\lambda_{S_k N_k}(m)$ are zero. The corresponding weighting rule is

$$G_k^{PSS}(m) = \left(\frac{|Y_k(m)|^2 - \lambda_{N_k}(m)}{|Y_k(m)|^2} \right)^{1/2}. \tag{4.16}$$

A major problem in the implementation of spectral subtraction methods is the appearance of a *new* kind of noise which is called *musical noise* [Berouti et al. 1979]. This type of noise is characterized by randomly spaced narrow spectral peaks. It sounds like the sum of tone generators with random fundamental frequencies which are turned on and off at a rate of about 20 *ms* [Boll 1979]. It usually occurs in the absence of speech activity as a result of having a lower noise estimate.

Several improvements have been proposed to enhance the original spectral subtraction method. They are formulated in the *generalized spectral subtraction* (GSS) weighting rule [Lim and Oppenheim 1979, Berouti et al. 1979, Lim 1978] as depicted below

$$\begin{aligned} G_k^{GSS}(m) &= \left(\frac{|Y_k(m)|^r - \alpha \cdot E\{|N_k(m)|^r\}}{|Y_k(m)|^r} \right)^{1/r} \\ &= \left(\frac{D_k(m)}{|Y_k(m)|^r} \right)^{1/r}, \end{aligned} \tag{4.17}$$

with

$$D_k(m) = |Y_k(m)|^r - \alpha \cdot E\{|N_k(m)|^r\},$$

where r is a constant factor which is adding a degree of freedom to the weighting rule and α is an *overestimation* factor [Berouti et al. 1979]. Spectral flooring is introduced to prevent the resulting subtraction to take negative values and to introduce a noise-masking effect. It is implemented as follows:

$$D_k(m) = \begin{cases} D_k(m) & \text{if } D_k(m) \geq \beta \cdot E\{|N_k(m)|^r\}, \\ \beta \cdot E\{|N_k(m)|^r\} & \text{otherwise,} \end{cases} \tag{4.18}$$

where β denotes the *spectral floor* parameter [Berouti et al. 1979]. The flooring could also be applied as a minimum gain/attenuation $G_{\min}$

$$G_k(m) = \max\{G_k(m),\, G_{\min}\}. \tag{4.19}$$

The floor value is usually determined by subjective criteria to yield noise naturalness during

periods with speech being absent [Yang 1993].

Another approach to enhance the spectral subtraction is to make the overestimation factor α dependent on some segmental signal-to-noise ratio (SNR) definition. Linear dependency was proposed in [Berouti et al. 1979]

$$\alpha_k(m) = \alpha_0 - \frac{\mathrm{SNR}_k(m)}{s}, \tag{4.20}$$

where $\alpha_k(m)$ is the SNR-dependent overestimation factor, α_0 is the desired overestimation value for $\mathrm{SNR}_k(m) = 0$ dB, $\mathrm{SNR}_k(m)$ is the segmental SNR, and $1/s$ is the slope of the line with $s > 0$.

The *nonlinear spectral subtractor* (NSS) [Lockwood and Boudy 1992] is using a more elaborate variant of an SNR-dependent overestimation factor. It is formulated based on the magnitude spectral subtraction method where $r = 1$ in (4.17)

$$G_k^{NSS}(m) = \frac{\overline{|Y_k(m)|} - \Phi\Big(\bar{v}_k(m), \alpha_k(m)\Big)}{\overline{|Y_k(m)|}}, \tag{4.21}$$

where the smoothed noisy speech and noise magnitude estimates are defined as

$$\overline{|Y_k(m)|} = \varepsilon_Y \cdot \overline{|Y_k(m-1)|} + (1-\varepsilon_Y) \cdot |Y_k(m)|, \tag{4.22}$$

$$\overline{|N_k(m)|} = \varepsilon_N \cdot \overline{|N_k(m-1)|} + (1-\varepsilon_N) \cdot |N_k(m)|, \tag{4.23}$$

with $0.1 \leq \varepsilon_Y \leq 0.5$ and $0.5 \leq \varepsilon_N \leq 0.9$. The frequency- and time-dependent overestimation factor is calculated with

$$\alpha_k(m) = \max_{m-40 \leq \ell \leq m} \{\, |N_k(\ell)| \,\}, \tag{4.24}$$

and the biased estimate of the SNR is computed by

$$\bar{v}_k(m) = \frac{\overline{|Y_k(m)|}}{\overline{|N_k(m)|}}. \tag{4.25}$$

The nonlinear function $\Phi\Big(\bar{v}_k(m), \alpha_k(m)\Big)$ is defined by

$$\Phi\Big(\bar{v}_k(m), \alpha_k(m)\Big) = \frac{\alpha_k(m)}{1 + c \cdot \bar{v}_k(m)}, \tag{4.26}$$

where c denotes a scaling factor depending on the variation range of $\bar{v}_k(m)$. This nonlinear

function was implemented with the upper and lower limits

$$\overline{|N_k(m)|} \leq \Phi\Big(\bar{v}_k(m), \alpha_k(m)\Big) \leq 3 \cdot \overline{|N_k(m)|}. \tag{4.27}$$

The idea of using this function is to apply a minimum subtraction factor in high SNR regions and subtract more noise in low SNR regions, until a spectral floor is reached.

Please note that the nonlinear spectral subtraction formulation differs from the generalized spectral subtraction in a way that the short-term magnitude estimate of noisy speech $|Y_k(m)|$ is replaced with a smoothed estimate $\overline{|Y_k(m)|}$, as depicted in (4.22), and the noise term is modeled with the nonlinear function $\Phi(\bar{v}_k(m), \alpha_k(m))$.

A variant of the generalized spectral subtraction method was proposed in [Sim et al. 1998]. It is based on the *parametric formulation of the generalized spectral subtraction* (PGSS), where the weighting rule is defined as

$$G_k^{PGSS}(m) = \left(\frac{a_{k,r}(m) \cdot |Y_k(m)|^r - b_{k,r}(m) \cdot E\{|N_k(m)|^r\}}{|Y_k(m)|^r} \right)^{1/r}, \tag{4.28}$$

where $a_{k,r}(m)$ and $b_{k,r}(m)$ are the parameters in the parametric formulation.

Both parameters are obtained by minimizing the mean-square error (MSE) of the following real-valued error function with respect to $a_{k,r}(m)$ and $b_{k,r}(m)$:

$$e_{k,r}(m) = |S_k(m)|^r - |\hat{S}_k(m)|^r, \tag{4.29}$$

with

$$|S_k(m)|^r = |Y_k(m)|^r - |N_k(m)|^r, \tag{4.30}$$

$$|\hat{S}_k(m)|^r = [\, G_k^{PGSS}(m) \cdot |Y_k(m)| \,]^r. \tag{4.31}$$

Each individual clean speech and noise spectral component is assumed to be a statistically independent zero-mean complex Gaussian random variable with time-varying variance. The *unconstrained* parametric generalized spectral subtraction yields

$$a_{k,r}(m) = \frac{\xi_k^r(m)}{1 + \xi_k^r(m)}, \tag{4.32}$$

$$b_{k,r}(m) = \frac{\xi_k^r(m) \cdot [1 - \xi_k^{-r/2}(m)]}{1 + \xi_k^r(m)}, \tag{4.33}$$

where the *a priori* SNR $\xi_k(m)$ is defined and estimated in [Ephraim and Malah 1984] without

the generalized power exponent r

$$\xi_k(m) = \frac{\lambda_{S_k}(m)}{\lambda_{N_k}(m)}. \tag{4.34}$$

Its estimate is following the *decision-directed* approach

$$\hat{\xi}_k(m) = (1-\varepsilon)\cdot \max\{v_k(m)-1,\ 0\} + \varepsilon \cdot \frac{|\hat{S}_k(m-1)|^2}{\alpha \cdot \lambda_{N_k}(m-1)}, \tag{4.35}$$

where ε denotes the smoothing constant and $v_k(m)$ the *a posteriori* SNR

$$v_k(m) = \frac{|Y_k(m)|^2}{\alpha \cdot \lambda_{N_k}(m)}. \tag{4.36}$$

Optimizing the error function of (4.29) with the constraint $a_{k,r}(m) = b_{k,r}(m)$ gives the following formulation:

$$a_{k,r}(m) = \frac{\xi_k^r(m)}{\xi_k^r(m) + \beta_r}, \tag{4.37}$$

where

$$\beta_r = \frac{\Gamma(r+1) - \Gamma^2(r/2+1)}{\Gamma(r+1)}, \tag{4.38}$$

with $\Gamma(\cdot)$ denoting the gamma function. The values of β_r for $r = 1,\ 2,$ and 3 are $0.2146,\ 0.5,$ and 0.7055, respectively. This is called the *constrained* parametric generalized spectral subtraction where applying the constraint was found to give a good noise suppression performance. A spectral flooring for this formulation is implemented as

$$|\overline{S}_k(m)| = \begin{cases} |\hat{S}_k(m)| & \text{if } |\hat{S}_k(m)| \geq \beta \cdot |Y_k(m)|, \\ 0.5 \cdot \left[\ \beta \cdot |Y_k(m)| + |\overline{S}_k(m-1)|\ \right] & \text{otherwise,} \end{cases} \tag{4.39}$$

where β denotes the spectral floor parameter, with a value, e.g., $\beta = 0.05 \cdots 0.2$.

The term $\xi_k^r(m)$ in both parametric formulations is approximated by taking $\hat{\xi}_k(m)$ in (4.35) to the power of r. The noise term $E\{|N_k(m)|^r\}$ is approximated during nonspeech intervals as

$$E\{|N_k(m)|^r\} \approx (1-\varepsilon_N)\cdot |N_k(m)|^r + \varepsilon_N \cdot |N_k(m-1)|^r, \tag{4.40}$$

where $\varepsilon_N = 0.90$ denotes the smoothing constant.

The decision-directed approach to estimate the *a priori* SNR has been reported to be the major factor which drastically reduces the musical noise in combination with the Ephraim and Malah noise reduction method [Cappé 1994]. A similar method has been applied to enhance the

power spectral subtraction method [Scalart and Vieira Filho 1996]

$$G_k^{PSS}(m) = \left(\frac{\hat{\xi}_k(m)}{1+\hat{\xi}_k(m)} \right)^{1/2}. \tag{4.41}$$

Please note that the *a priori* SNR estimate used in (4.41) is similar to (4.35) with the term $\lambda_{N_k}(m-1)$ is replaced by the current frame $\lambda_{N_k}(m)$.

4.2.2 Wiener Filtering

Wiener filtering is derived from the optimal filter theory where the filter $G_k^W(m)$ is obtained by minimizing the mean-square error between the clean speech and estimated signal waveforms. The filter is called the *noncausal Wiener filter* [Papoulis 1991, Lim and Oppenheim 1979, van Trees 1968] and can be applied in the time and frequency domain. In the STSA context, the real-valued filter coefficients are obtained as

$$G_k^W(m) = \frac{\lambda_{S_k}(m)}{\lambda_{S_k}(m)+\lambda_{N_k}(m)}, \tag{4.42}$$

which is calculated on the short-time basis where the stationary condition is assumed and therefore indicates the use of a weighting rule.

The estimate of $\lambda_{S_k}(m)$ may be obtained by first estimating $\lambda_{Y_k}(m)$ through locally averaging or smoothing $|Y_k(m)|^2$ and then subtracting the estimated $\lambda_{N_k}(m)$ from $\lambda_{Y_k}(m)$

$$G_k^W(m) = \frac{\lambda_{Y_k}(m)-\lambda_{N_k}(m)}{\lambda_{Y_k}(m)}. \tag{4.43}$$

Other spectral subtraction methods are also applicable which directly use the raw noisy speech periodogram $|Y_k(m)|^2$. There also exist other ways to estimate $\lambda_{S_k}(m)$ as described in this section.

A generalization of Wiener filtering is given for some constants α and r which is called the short-time *parametric Wiener filtering* [Lim and Oppenheim 1979]

$$G_k^W(m) = \left(\frac{\lambda_{S_k}(m)}{\lambda_{S_k}(m)+\alpha\cdot\lambda_{N_k}(m)} \right)^r. \tag{4.44}$$

Please note that the use of Wiener filtering is done by first calculating the filter $G_k^W(m)$ and then performing

$$\hat{S}_k(m) = G_k^W(m)\cdot Y_k(m). \tag{4.45}$$

The *implicit Wiener filtering* is formulated based on the parametric Wiener filtering which leads to the formulation of spectral subtraction and maximum likelihood amplitude estimator [McAulay and Malpass 1980]. The implicit relationship is obtained using the following approximation:

$$\hat{\lambda}_{S_k}(m) = |\hat{S}_k(m)|^2, \tag{4.46}$$

to yield

$$|\hat{S}_k(m)| = \left(\frac{|\hat{S}_k(m)|^2}{|\hat{S}_k(m)|^2 + \alpha \cdot \lambda_{N_k}(m)} \right)^r \cdot |Y_k(m)|. \tag{4.47}$$

Taking $r = 1/2$ and solving for $|\hat{S}_k(m)|$ we obtain the power spectral subtraction method

$$|\hat{S}_k(m)| = \left(|Y_k(m)|^2 - \alpha \cdot \lambda_{N_k}(m) \right)^{1/2}, \tag{4.48}$$

and with $r = 1$ and $\alpha = 1/4$, the maximum likelihood amplitude estimator is obtained [McAulay and Malpass 1980]

$$|\hat{S}_k(m)| = 0.5 \cdot |Y_k(m)| + 0.5 \cdot \left(|Y_k(m)|^2 - \lambda_{N_k}(m) \right)^{1/2}. \tag{4.49}$$

An iterative procedure to estimate the term $\lambda_{S_k}(m)$ in (4.42) leads to the formulation of the *iterative Wiener filtering* [Lim and Oppenheim 1979]. Denoting the estimate of $\lambda_{S_k}(m)$ at iteration i as $\lambda_{S_k}^{(i)}(m)$, the following calculations are repeated until a convergence criterion is reached:

$$G_k^i(m) = \frac{\lambda_{S_k}^{(i)}(m)}{\lambda_{S_k}^{(i)}(m) + \lambda_{N_k}(m)}, \tag{4.50}$$

$$\lambda_{S_k}^{(i+1)}(m) = \left[\, G_k^i(m) \cdot |Y_k(m)| \, \right]^2. \tag{4.51}$$

An alternative approach is to previously estimate $\lambda_{S_k}^{(i)}(m)$ in (4.50) with the speech model parameters [Lim and Oppenheim 1978]

$$\lambda_{S_k}^{(i)}(m) = \frac{[g^i(m)]^2}{\left| 1 - \sum_{\ell=1}^{L} a_\ell^i(m) \cdot e^{-j\omega_k \ell} \right|^2}, \tag{4.52}$$

where $a_\ell^i(m)$ are the short-time all-pole model parameters of length L of the speech waveform and $[g^i(m)]^2$ corresponds to the gain in the excitation at iteration i. This procedure is based on the maximum *a posteriori* (MAP) estimation of the all-pole speech model parameters in additive white Gaussian noise (AWGN). The estimate of $a_\ell^i(m)$ is obtained using the correlation method

of linear prediction analysis [Makhoul 1975] and the gain $g^i(m)$ is calculated using Parseval's theorem

$$\frac{1}{2\pi}\int_{-\pi}^{\pi}\frac{[g^i(m)]^2}{\left|1-\sum_{\ell=1}^{L}a_\ell^i(m)\cdot e^{-j\omega\ell}\right|^2}\cdot d\omega=\frac{1}{N}\sum_{n_m=0}^{N_m-1}y^2(n_m)-\sigma_N^2(m), \tag{4.53}$$

with $\sigma_N^2(m)$ denoting the noise energy and n_m denotes the discrete time index for speech samples at frame m. Improvements in this area leads to the formulation of the *constrained iterative Wiener filtering* [Hansen and Clements 1991].

For the specific use in speech recognition where clean templates or HMMs are available, the *hypothesized Wiener filtering* [Berstein and Shallom 1991] is formulated. This method simply use the Wiener filtering in (4.42) with the clean speech PSD obtained from the clean templates or HMM.

Using the decision-directed approach in the Wiener filtering yields the *a priori* SNR based weighting rule $G_k^{W_{prio}}(m)$ as defined in [Scalart and Vieira Filho 1996]

$$G_k^{W_{prio}}(m)=\frac{\xi_k(m)}{\xi_k(m)+1}, \tag{4.54}$$

where the *a priori* SNR estimate $\hat{\xi}_k(m)$ is calculated following (4.35). This weighting rule is showing good results and is easy to implement. It is therefore used as a reference. In our implementation, the *a priori* SNR estimate in (4.54) has a minimum value of 0.01 and $\lambda_{N_k}(m-1)$ in (4.35) is being replaced by $\lambda_{N_k}(m)$. Choosing $\varepsilon=0$ in (4.35), approximately the well known *a posteriori* SNR based Wiener filtering follows.

4.2.3 Gaussian Model and Ephraim-Malah Estimator

In this section we show that some of the approaches presented earlier can be derived under a reasonable statistical model assumption. The widely used assumptions can be briefly stated as follows:

- The consecutive spectral components of the noisy speech or the observations are assumed as statistically independent.
- The spectral components of the clean speech signal and of the noise process at any given frame are assumed statistically independent zero-mean Gaussian random variables. The real and imaginary parts of each spectral component are also assumed to be statistically independent identically distributed random variables.

- The variance of each spectral component is time-varying due to speech and possibly noise non-stationarity.

As we will see later in this section, the Gaussian assumption has actually been used to design some of the noise reduction techniques mentioned previously.

The above assumptions are challenged with the following facts and/or ideas:

- The use of overlapping analysis frames makes the consecutive observations statistically dependent.
- The clean speech and noise spectral components may be better modeled with Gamma [Martin 2002] or Laplace distributions [Martin and Breithaupt 2003].
- The statistical independency between spectral components is justified when the analysis frame length approaches infinity, since the normalized correlation between the coefficients approaches zero in this condition. A short-time spectral analysis will consequently introduce correlation to the coefficients. The use of windowing such as Hanning or Hamming reduces the correlation between widely separated spectral components at the expense of increasing the correlation between adjacent spectral components.

Nevertheless, the Gaussian assumption will be used in the following.

The power spectral subtraction method is derived based on the maximum likelihood variance estimator of the assumed Gaussian statistical model based on L observations and the variances are assumed to be slowly varying. The latter implies that the spectral component variances are considered constant during the L observations. The likelihood function of the joint conditional probability density function (PDF) of the observations $Y_k^L = \{\, Y_k(m),\, Y_k(m-1),\, \cdots,\, Y_k(m-L+1) \,\}$ given the clean speech and noise variances, λ_{S_k} and λ_{N_k}, respectively, is given as

$$p\,(\, Y_k^L \mid \lambda_{S_k}, \lambda_{N_k}\,) = \frac{L}{\pi \cdot (\lambda_{S_k} + \lambda_{N_k})} \cdot \prod_{\ell=0}^{L-1} \exp\left[- \frac{|Y_k(m-\ell)|^2}{\lambda_{S_k} + \lambda_{N_k}} \right], \tag{4.55}$$

which is a joint Gaussian distribution with the variance of each observation denoted by $\lambda_{Y_k} = \lambda_{S_k} + \lambda_{N_k}$. The clean speech variance estimator $\hat{\lambda}_{S_k}$ is easily obtained by maximizing the natural logarithm of the likelihood function with respect to the parameter λ_{S_k}

$$\hat{\lambda}_{S_k} = \begin{cases} \frac{1}{L} \sum_{\ell=0}^{L-1} |Y_k(m-\ell)|^2 - \lambda_{N_k} & \text{if nonnegative,} \\ \\ 0 & \text{otherwise,} \end{cases} \tag{4.56}$$

where λ_{N_k} is the noise variance estimated given L observations. The derivation given in [McAulay and Malpass 1980] was simply assuming $L = 1$.

The Wiener filtering is obtained by directly minimizing the mean square-error or by minimizing the conditional mean $E\{S_k^M \mid Y_k^M\}$ under the Gaussian assumption. In further development under the same statistical model assumption, it is of interest to estimate directly the clean speech spectral amplitude since from a perceptual point of view the phase is of low importance in short-term spectral analysis context. This leads to the development of the maximum likelihood clean speech spectral amplitude estimator [McAulay and Malpass 1980] under the assumption of Gaussian noise and a deterministic clean speech waveform of unknown magnitude A_k and phase θ_{S_k}

$$S_k = A_k \exp\,(j\theta_{S_k}). \tag{4.57}$$

For simplicity we drop the frame index m and replace the magnitude notation $|S_k|$ with A_k.

The likelihood function is formulated as

$$p(\,Y_k \mid A_k, \theta_{S_k}\,) = \frac{1}{\pi \cdot \lambda_{N_k}} \cdot \exp\left[- \frac{|Y_k|^2 - 2A_k \cdot \mathrm{Re}\,\{e^{-j\theta_{S_k}} Y_k\} + A_k^2}{\lambda_{N_k}} \right], \tag{4.58}$$

where the phase is assumed to have a uniform distribution

$$p\,(\theta_{S_k}) = \begin{cases} \frac{1}{2\pi} & \text{if } \theta_{S_k} \in [-\pi, \pi), \\ 0 & \text{otherwise.} \end{cases} \tag{4.59}$$

Taking the expected value of the likelihood function with respect to the phase and maximizing the function with respect to the amplitude yields

$$\hat{A}_k = \frac{1}{2} \cdot \left[|Y_k| + \sqrt{|Y_k|^2 - \lambda_{N_k}} \right]. \tag{4.60}$$

The final weighting rule is formulated by simply appending the observation phase

$$G_k^{MLAE}(m) = \frac{1}{2} + \frac{1}{2} \sqrt{\frac{|Y_k(m)|^2 - \lambda_{N_k}(m)}{|Y_k(m)|^2}}\,. \tag{4.61}$$

The frame index m is directly appended to the weighting rule.

The extension of the clean speech spectral amplitude estimator when using the Gaussian assumption for the clean speech spectral components instead of a deterministic waveform is derived in [Ephraim and Malah 1984; 1983]. In a particular bin k, the observed spectral component

$Y_k = R_k \exp(j\vartheta_k)$ is equal to the sum of the clean speech spectral component $S_k = A_k \exp(j\theta_k)$ and the noise N_k. The minimum mean square-error (MMSE) estimator $\hat{A}_k$ is calculated as

$$\begin{aligned}\hat{A}_k &= E\{A_k|\, Y_k\} \\ &= \frac{\int_0^{\infty}\int_0^{2\pi} a_k\, p(Y_k|\, a_k,\theta_k)\, p(a_k,\theta_k)\, d\theta_k\, da_k}{\int_0^{\infty}\int_0^{2\pi} p(Y_k|\, a_k,\theta_k)\, p(a_k,\theta_k)\, d\theta_k\, da_k},\end{aligned} \tag{4.62}$$

where

$$p(a_k) = \begin{cases} \frac{2a_k}{\lambda_{S_k}} \exp\left(-\frac{a_k^2}{\lambda_{S_k}}\right) & \text{if } a_k \in [0,\infty), \\ 0 & \text{otherwise,} \end{cases} \tag{4.63}$$

and the phase has a uniform distribution as given in (4.59). The joint and conditional distributions are

$$p(a_k,\theta_k) = \frac{a_k}{\pi\lambda_{S_k}} \exp\left(-\frac{a_k^2}{\lambda_{S_k}}\right), \tag{4.64}$$

$$p(Y_k\,|a_k,\theta_k) = \frac{1}{\pi\lambda_{N_k}} \exp\left(-\frac{|Y_k - a_k e^{j\theta_k}|^2}{\lambda_{N_k}}\right). \tag{4.65}$$

The MMSE spectral amplitude estimator $\hat{A}_k$ is thus given by [Ephraim and Malah 1984; 1983]

$$\hat{A}_k = \frac{\sqrt{v_k}}{\gamma_k} \cdot \Gamma(1.5) \cdot M(-0.5,1,-v_k) \cdot R_k, \tag{4.66}$$

where $\Gamma(\cdot)$ denotes the gamma function $M(a,b,z)$ is the confluent hypergeometric function and v_k is defined in terms of *a priori* SNR ξ_k and *a posteriori* SNR γ_k as

$$v_k = \frac{\xi_k}{1+\xi_k}\, \gamma_k. \tag{4.67}$$

Combining the spectral amplitude estimator with the optimal phase estimator, which is the observed phase ϑ_k, and substituting the following equations:

$$M(-0.5,1,-z) = \exp\left(-\frac{z}{2}\right) \cdot \left[(1+z)\cdot I_0\left(\frac{z}{2}\right) + z \cdot I_1\left(\frac{z}{2}\right)\right], \tag{4.68}$$

$$\Gamma(1.5) = \frac{\sqrt{\pi}}{2}, \tag{4.69}$$

where $I_0(\cdot)$ and $I_1(\cdot)$ denote the modified Bessel functions of zero and first order, respectively, the so-called spectral amplitude weighting rule is shown as

$$G_k^{SA}(m) = \frac{\sqrt{\pi \cdot \nu_k(m)}}{2 \cdot \gamma_k(m)} \cdot \left[\left(1+\nu_k(m)\right) \cdot I_0\left(\frac{\nu_k(m)}{2}\right) + \right. \\ \left. + \nu_k(m) \cdot I_1\left(\frac{\nu_k(m)}{2}\right) \right] \cdot \exp\left(-\frac{\nu_k(m)}{2}\right). \tag{4.70}$$

Note that the weighting rule requires the computation of exponential and Bessel functions. Alternative solutions to efficiently compute the weighting rule are given in [Wolfe and Godsill 2001].

Improvement in this algorithm has been derived in [Ephraim and Malah 1985] where the estimator $\hat{A}_k$ is calculated to minimize the distortion measure

$$E\{(\log A_k - \log \hat{A}_k)^2\}, \tag{4.71}$$

given the noisy observations. The resulting so-called log-spectral amplitude estimator yields the weighting rule

$$G_k^{LSA}(m) = \frac{\xi_k}{1+\xi_k} \cdot \exp\left\{\frac{1}{2}\int_{\nu_k}^{\infty} \frac{e^{-t}}{t}\, dt\right\}. \tag{4.72}$$

In the newly derived weighting rule, the computation of the integral is done numerically. The decision-directed approach for the *a priori* SNR estimate of both weighting rules is shown to be the dominant factor. Modifications of the estimator have been proposed in [Cohen 2004a;b, Fingscheidt et al. 2005a] which utilize relaxed statistical model assumptions for the speech spectral variance. For speech recognition as shown in [Gemello et al. 2004], an SNR-dependent overestimation factor and spectral flooring in the decision-directed estimator are proposed.

4.2.4 Least-Squares Amplitude Estimator

The method of least-squares is a deterministic approach where statistical assumptions are not used to derive the estimator. However, statistical assumptions are usually introduced after obtaining the estimator. Applying this method in the STSA context yields the formulation of the least-squares amplitude estimator [Beaugeant and Scalart 2001]. This offers an alternative to the stochastic approaches as presented in Section 4.2.3. In general, the weighting rule is formulated as

$$G_k^{LS}(m) = \frac{\sum_{\ell=0}^{m} w(\ell) Y_k^*(\ell) S_k(\ell)}{\sum_{\ell=0}^{m} w(\ell) Y_k^*(\ell) Y_k(\ell)}, \tag{4.73}$$

where $w(\ell)$ denotes a weighting series. An exponential forgetting factor is used as the weighting series

$$w(\ell) = \rho^{m-\ell} \qquad 0 \leq \ell \leq m, \tag{4.74}$$

for $0 < \rho < 1$ which converges for $m \to \infty$ to $\frac{1}{1-\rho}$. Assuming that the weighted average of scalar products between the following speech and noise terms is approximately zero

$$\sum_{\ell=0}^{m} w(\ell) S_k^*(\ell) N_k(\ell) = \sum_{\ell=0}^{m} w(\ell) N_k^*(\ell) S_k(\ell) \approx 0, \tag{4.75}$$

which is justified based on the statistical independency assumption that the weighted average is an estimate of the expectation operator $E\{S_k^*(\ell)N_k(\ell)\} = E\{N_k^*(\ell)S_k(\ell)\} \approx 0$, and taking the exponential forgetting factor yields

$$G_k^{LS}(m) = \frac{\sum_{\ell=0}^{m} \rho^{m-\ell} \cdot |S_k(\ell)|^2}{\sum_{\ell=0}^{m} \rho^{m-\ell} \cdot |S_k(\ell)|^2 + \alpha \cdot \sum_{\ell=0}^{m} \rho^{m-\ell} \cdot |N_k(\ell)|^2}. \tag{4.76}$$

Noise overestimation factor α is assigned to the noise summation term.

The exponential forgetting factor weighting series can be implemented recursively

$$\begin{aligned} E_{U_k}(m) &= \sum_{\ell=0}^{m} \rho_U^{m-\ell} \cdot |U_k(\ell)|^2 \\ &= \rho_U \cdot \sum_{\ell=0}^{m-1} \rho_U^{m-\ell-1} \cdot |U_k(\ell)|^2 + |U_k(m)|^2 \\ &= \rho_U \cdot E_{U_k}(m-1) + |U_k(m)|^2, \end{aligned} \tag{4.77}$$

for $U \in \{S,N\}$. The weighting rule in (4.76) can therefore be written as

$$G_k^{LS}(m) = \frac{E_{S_k}(m)}{E_{S_k}(m) + \alpha \cdot E_{N_k}(m)}. \tag{4.78}$$

Recursive averaging as shown in (4.77) is actually neglecting the normalization factor $(1-\rho)$ if compared to the widely used smoothing equation. It can be shown that by taking the normalization factor into consideration, we get the following recursive calculation:

$$\begin{aligned} E_{U_k}(m) &= (1-\rho_U) \cdot \sum_{\ell=m}^{m} \rho_U^{m-\ell} \cdot |U_k(\ell)|^2 \\ &= (1-\rho_U) \cdot \Big[\rho_U \cdot \sum_{\ell=m}^{m-1} \rho_U^{m-\ell-1} \cdot |U_k(\ell)|^2 + |U_k(m)|^2 \Big] \\ &= \rho_U \cdot E_{U_k}(m-1) + (1-\rho_U) \cdot |U_k(m)|^2, \end{aligned} \tag{4.79}$$

which is the widely used smoothing method with a smoothing factor ρ.

Following [Beaugeant et al. 2002], the *recursive* least-squares weighting rule in the STSA context is defined as in (3.1)

$$G_k^{RLS_{post}}(m) = \frac{E_{Y_k}(m)}{E_{Y_k}(m) + \alpha \cdot E_{N_k}(m)}, \tag{4.80}$$

with the following PSD estimates:

$$E_{Y_k}(m) = \rho_Y \cdot E_{Y_k}(m-1) + |Y_k(m)|^2, \tag{4.81}$$

$$E_{N_k}(m) = \rho_N \cdot E_{N_k}(m-1) + \lambda_{N_k}(m), \tag{4.82}$$

This weighting rule is formulated by simply replacing the clean speech energy term in (4.76) with the noisy speech one $|S_k(\ell)|^2 = |Y_k(\ell)|^2$ and assigning two different weighting constants ρ_Y and ρ_N for noisy speech and noise PSDs, respectively. The term *recursive* was derived due to the recursive calculation of the energy summation using the exponential forgetting factor. The performance of the weighting rule has been previously reported in [Andrassy et al. 2001, Aalburg et al. 2002].

4.2.5 Two-Stage Mel-Warped Wiener Filter

Initially conducted speech recognition experiments indicate that the noise reduction scheme in the AFE, i.e., two-stage mel-warped Wiener filter, provides the biggest improvement among other techniques. The scheme can be described as follows:

- Hanning windowing.
- Obtain the power spectrum after applying the fast Fourier transform (FFT).
- Smooth the power spectrum by averaging two adjacent frequency bins. This reduces the amount of frequency bins to $N_{FFT}/4+1$. The smoothed power spectrum at frequency bin k and frame m is denoted as $|Y_k(m)|^2$, for $k = 0 \cdots N_{FFT}/4$.
- Calculate the Wiener filter coefficient $G_{k,2}(m)$. For each stage, the Wiener filter coefficient is computed in two steps. This is mainly done to enhance the *a priori* SNR estimate as described in the following:

$$|\hat{S}_{k,1}(m)| = G_{k,1}(m) \cdot \sqrt{\frac{|Y_k(m)|^2 + |Y_k(m-1)|^2}{2}}\ , \tag{4.83}$$

where the first stage weighting rule $G_{k,1}(m)$ is defined as

$$G_{k,1}(m) = \frac{\sqrt{\xi_{k,1}(m)}}{1+\sqrt{\xi_{k,1}(m)}}. \tag{4.84}$$

The square root of the *a priori* SNR $\sqrt{\xi_{k,1}(m)}$ is estimated using the idea of decision-directed approach with $\varepsilon = 0.98$

$$\begin{aligned}\sqrt{\xi_{k,1}(m)} &= \varepsilon \cdot \sqrt{\frac{|\hat{S}_{k,2}(m-1)|^2}{\lambda_{N_k}(m)}} + \\ &+ (1-\varepsilon)\cdot \max\left\{ \sqrt{\frac{|Y_k(m)|^2 + |Y_k(m-1)|^2}{2\cdot\lambda_{N_k}(m)} - 1},\ 0 \right\}.\end{aligned} \tag{4.85}$$

The denoised signal $|\hat{S}_{k,1}(m)|$ is then used to estimate the final *a priori* SNR $\xi_{k,2}(m)$

$$\xi_{k,2}(m) = \max\left\{ \frac{|\hat{S}_{k,1}(m)|^2}{\lambda_{N_k}(m)},\ -22 \text{ dB} \right\}. \tag{4.86}$$

The final weighting rule $G_{k,2}(m)$ is defined as

$$G_{k,2}(m) = \frac{\sqrt{\xi_{k,2}(m)}}{1+\sqrt{\xi_{k,2}(m)}}, \tag{4.87}$$

which is also used to compute $|\hat{S}_{k,2}(m)|$ in (4.85)

$$\hat{S}_{k,2}(m)| = G_{k,2}(m)\cdot|Y_k(m)|. \tag{4.88}$$

It is necessary to note that $|\hat{S}_{k,2}(m-1)|$ is initialized to zero.

- Smooth and transform the coefficients $G_{k,2}(m)$ to the mel frequency scale using triangular-shaped frequency windows. Figure 4.4 shows the 25 triangular-shaped frequency windows applied to $N_{FFT}/4+1$ frequency bins for $N_{FFT} = 256$.

- In the second stage, adjust the aggressiveness of the noise reduction by applying a single constant factor to the coefficients. The constant factor is calculated based on the speech/noise frame knowledge.

- Transform the coefficients to the time domain using the mel-warped inverse discrete cosine transform (IDCT) to obtain its impulse response.

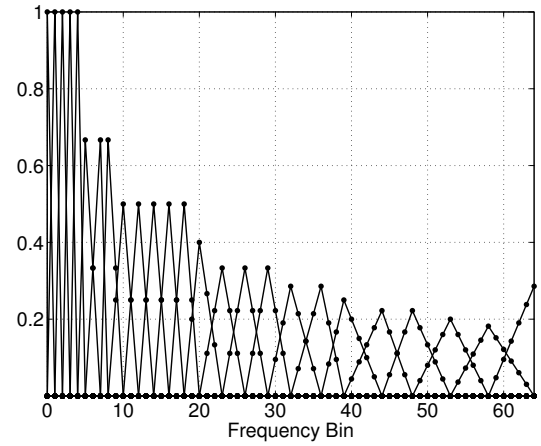


Figure 4.4: Triangular-shaped mel filterbank as used in the ETSI advanced front-end (AFE).

- Truncate the impulse response, Hanning windowing, and filter the input signal. At the end of the first stage, the filtered input signal is used as the input signal for the second stage.
- At the end of the second stage, perform a notch filtering to remove the DC offset.

It is obvious that there exist redundancies in this two-stage mel-warped Wiener filter algorithm. These redundancies result in a higher computational load. The main factor is the filtering process itself, where the filter coefficient is computed in the frequency domain and applied in the time domain using convolution method. This has to be done twice which corresponds to two-stage filtering. This has been observed and an improved version of the algorithm is proposed in [Li et al. 2004] and called the two-stage mel-warped filterbank Wiener filtering.

4.3 Least-Squares Based Weighting Rules

There are two ways of interpreting the least-squares weighting rule. The least-squares weighting rule formulation in (4.76) may be related to the Wiener filtering formulation in (4.42) given the PSD estimates as

$$\hat{\lambda}_{S_k}(m) = E_{S_k}(m) = \sum_{\ell=0}^{m} w(\ell) S_k^*(\ell) S_k(\ell) = \sum_{\ell=0}^{m} w(\ell) |S_k(\ell)|^2,$$

and

$$\hat{\lambda}_{N_k}(m) = E_{N_k}(m) = \sum_{\ell=0}^{m} w(\ell) N_k^*(\ell) N_k(\ell) = \sum_{\ell=0}^{m} w(\ell) |N_k(\ell)|^2.$$

In this context, the choice of a weighting series $w(\ell)$ is usually directed towards achieving an unbiased and consistent estimate of the PSD. This problem is well defined in the spectral analysis context dealing with the modified periodogram.

Another way of interpreting the weighting rule is shown by taking more suitable PSD estimates instead of the modified periodogram ones and applying the weighting series to these estimates

$$E_{S_k}(m) = \sum_{\ell=0}^{m} w(\ell) \hat{\lambda}_{S_k}(\ell),$$
$$E_{N_k}(m) = \sum_{\ell=0}^{m} w(\ell) \hat{\lambda}_{N_k}(\ell).$$

In this context, the weighting series simply assigns a degree of importance to the data since a higher weight implies higher importance or contribution. The choice of a weighting series is focused on the error function formulation of the least-squares method.

4.3.1 Batch Least-Squares Formulation in the Frequency Domain

Let's now formulate a weighting rule derived using the weighted least-squares error criterion in the frequency domain. The cost function is defined as

$$J_{LS} = \sum_{\ell=\mu}^{m} w(\ell) \cdot |e_k(\ell)|^2, \tag{4.89}$$

where w_ℓ gives a weight for the error at a particular frame ℓ, defined as

$$E_k(\ell) = S_k(\ell) - \hat{S}_k(\ell). \tag{4.90}$$

m denotes the most recent frame where a noisy speech signal $Y_k(m)$, is known. At time instant m, a set of clean speech estimates $\hat{S}_k(\ell)$, for $\mu \leq \ell \leq m$, are obtained with

$$\hat{S}_k(\ell) = G_k(m) \cdot Y_k(\ell), \tag{4.91}$$

where $G_k(m)$ denotes the weighting rule calculated based on all available noisy observation frames starting from frame μ until current frame m.

Using matrix notations, the cost function J_{LS} in (4.89) is defined as

$$J_{LS} = \underline{\mathrm{E}}_m^H \, \mathbf{W}_m \, \underline{\mathrm{E}}_m, \tag{4.92}$$

where

$$\underline{\mathrm{E}}_m = [\, E_k^*(\mu) \;\; E_k^*(\mu+1) \;\; \cdots \;\; E_k^*(m) \,]^{\,H}, \tag{4.93}$$

and $\mathbf{W}_m$ is a real-valued diagonal matrix of size $(m-\mu+1)-by-(m-\mu+1)$ with elements

$$W(i,j) = \begin{cases} w(\mu+i-1) & \text{if } i = j, \\ \\ 0 & \text{if } i \neq j, \end{cases}$$

for $1 \leq i, j \leq m-\mu+1$. The notations $(\cdot)^*$ and $(\cdot)^H$ denote the conjugate and conjugate transpose, respectively. Using (4.90), (4.91) and (4.93), the error $\underline{\mathrm{E}}_m$ can be written as

$$\underline{\mathrm{E}}_m = \underline{\mathrm{S}}_m - \hat{\underline{\mathrm{S}}}_m = \underline{\mathrm{S}}_m - G_k(m) \cdot \underline{\mathrm{Y}}_m, \tag{4.94}$$

where

$$\underline{\mathrm{S}}_m = [\, S_k^*(\mu) \;\; S_k^*(\mu+1) \;\; \cdots \;\; S_k^*(m) \,]^{\,H}, \tag{4.95}$$

$$\hat{\underline{\mathrm{S}}}_m = [\, \hat{S}_k^*(\mu) \;\; \hat{S}_k^*(\mu+1) \;\; \cdots \;\; \hat{S}_k^*(m) \,]^{\,H}, \tag{4.96}$$

$$\underline{\mathrm{Y}}_m = [\, Y_k^*(\mu) \;\; Y_k^*(\mu+1) \;\; \cdots \;\; Y_k^*(m) \,]^{\,H}. \tag{4.97}$$

Using (4.94), we can write (4.92) as

$$\begin{aligned} J_{LS} &= [\underline{\mathrm{S}}_m - G_k(m) \cdot \underline{\mathrm{Y}}_m]^H \, \mathbf{W}_m \, [\underline{\mathrm{S}}_m - G_k(m) \cdot \underline{\mathrm{Y}}_m] \\ &= \underline{\mathrm{S}}_m^H \mathbf{W}_m \underline{\mathrm{S}}_m - G_k^*(m) \, \underline{\mathrm{Y}}_m^H \mathbf{W}_m \underline{\mathrm{S}}_m - \\ &\quad - \underline{\mathrm{S}}_m^H \mathbf{W}_m \underline{\mathrm{Y}}_m \, G_k(m) + G_k^*(m) \, \underline{\mathrm{Y}}_m^H \mathbf{W}_m \underline{\mathrm{Y}}_m \, G_k(m). \end{aligned} \tag{4.98}$$

Minimizing (4.98) with respect to $G_k^*(m)$ yields [Haykin 2002]

$$-\underline{\mathrm{Y}}_m^H \mathbf{W}_m \underline{\mathrm{S}}_m + \underline{\mathrm{Y}}_m^H \mathbf{W}_m \underline{\mathrm{Y}}_m \, G_k(m) = 0. \tag{4.99}$$

Solving for $G_k(m)$, we obtain the optimal estimate

$$G_k(m) = [\, \underline{\mathrm{Y}}_m^H \mathbf{W}_m \underline{\mathrm{Y}}_m \,]^{-1} \, \underline{\mathrm{Y}}_m^H \mathbf{W}_m \underline{\mathrm{S}}_m. \tag{4.100}$$

Consequently, at current frame m the clean speech estimate $\hat{S}_k(m)$ is obtained using $G_k(m)$, which is based on $m-\mu+1$ noisy observations. With a new noisy observation $Y_k(m+1)$, $G_k(m+1)$ is recomputed using old and new noisy observations and $\hat{S}_k(m+1)$ is estimated. The past estimate $G_k(m)$ is discarded. The method is therefore called the *batch* least-squares method and is identical to the formulation in (4.73) by fixing the frame μ to the first processed frame. As stated previously, the use of the *recursive* term on this formulation is only possible through the use of a weighting series such as the exponential forgetting factor.

An alternative formulation of this batch least-squares method is proposed by taking into consideration the additive noise assumption and applying it to the term $\underline{\mathrm{Y}}_m^H \mathbf{W}_m \underline{\mathrm{S}}_m$ in (4.100) where

$$\underline{\mathrm{S}}_m = \underline{\mathrm{Y}}_m - \underline{\mathrm{N}}_m, \tag{4.101}$$

yielding

$$\begin{aligned} \underline{\mathrm{Y}}_m^H \mathbf{W}_m \underline{\mathrm{S}}_m &= \underline{\mathrm{Y}}_m^H \mathbf{W}_m \left[\, \underline{\mathrm{Y}}_m - \underline{\mathrm{N}}_m \, \right] \\ &= \underline{\mathrm{Y}}_m^H \mathbf{W}_m \underline{\mathrm{Y}}_m - \underline{\mathrm{Y}}_m^H \mathbf{W}_m \underline{\mathrm{N}}_m \\ &= \underline{\mathrm{Y}}_m^H \mathbf{W}_m \underline{\mathrm{Y}}_m - \underline{\mathrm{S}}_m^H \mathbf{W}_m \underline{\mathrm{N}}_m - \underline{\mathrm{N}}_m^H \mathbf{W}_m \underline{\mathrm{N}}_m. \end{aligned} \tag{4.102}$$

Using the same approximation for the term $\underline{\mathrm{S}}_m^H \mathbf{W}_m \underline{\mathrm{N}}_m$ as in (4.75) we get

$$G_k(m) = \left[\, \underline{\mathrm{Y}}_m^H \mathbf{W}_m \underline{\mathrm{Y}}_m \, \right]^{-1} \left[\, \underline{\mathrm{Y}}_m^H \mathbf{W}_m \underline{\mathrm{Y}}_m - \underline{\mathrm{N}}_m^H \mathbf{W}_m \underline{\mathrm{N}}_m \, \right]. \tag{4.103}$$

Based on the formulation and by taking the noise overestimation factor α and the energy estimates $E_{Y_k}(m)$ and $E_{N_k}(m)$ as in (4.81) and (4.82), respectively, the spectral subtraction based *recursive* least-squares weighting rule is defined as [Setiawan et al. 2005a]

$$G_k^{RLS_{post-sub}}(m) = \frac{E_{Y_k}(m) - \alpha \cdot E_{N_k}(m)}{E_{Y_k}(m)}. \tag{4.104}$$

Once again, the term *recursive* is used to indicate the role of the forgetting factor weighting series. This formulation does not require an additional complexity and memory requirements.

Another weighting rule is also proposed in this *batch* least-squares formulation by taking the state-of-the-art *recursive* least-squares of [Beaugeant et al. 2002] as in (4.80) and performing another recursive estimation by simply replacing $E_{Y_k}(m-1)$ in (4.81) with

$$E_{\hat{S}_k}(m-1) = \rho_S \cdot E_{\hat{S}_k}(m-2) + |\hat{S}_k(m-1)|^2, \tag{4.105}$$

yielding the *a priori recursive* least-squares [Setiawan et al. 2005c]

$$G_k^{RLS_{prior}}(m) = \frac{\rho_Y \cdot \rho_S \cdot E_{\hat{S}_k}(m-2) + \rho_Y \cdot |\hat{S}_k(m-1)|^2 + |Y_k(m)|^2}{\rho_Y \cdot \rho_S \cdot E_{\hat{S}_k}(m-2) + \rho_Y \cdot |\hat{S}_k(m-1)|^2 + |Y_k(m)|^2 + \alpha \cdot E_{N_k}(m)}. \tag{4.106}$$

This is motivated by our aim to combine the weighting rule formulation of (4.76) with (4.80). Note that the past estimate of the clean speech energy term is taken into consideration which is a key feature of this weighting rule. The proposed formulation only adds a small amount of additional complexity and memory requirements.

4.3.2 Recursive Gain Least-Squares

In this section, a general recursive solution of the *batch* least-squares is presented. The weighting at current frame m, is defined by utilizing the exponential forgetting factor weighting series

$$w(\ell) = \rho^{m-\ell} \qquad 0 \le \ell \le m, \tag{4.107}$$

for $0 < \rho < 1$ and μ was set to 0. The *batch* least-squares is defined in (4.100) as

$$G_k(m) = [\, \underline{Y}_m^H \mathbf{W}_m \underline{Y}_m \,]^{-1}\, \underline{Y}_m^H \mathbf{W}_m \underline{S}_m. \tag{4.108}$$

Defining the inverse term as Q_m and exploiting the diagonal nature of $\mathbf{W}_m$, we can write

$$\begin{aligned} Q_m &= [\, \underline{Y}_m^H \mathbf{W}_m \underline{Y}_m \,]^{-1} \\ &= \left[\sum_{\ell=0}^{m} w(\ell)\, |Y_k(\ell)|^2 \right]^{-1}. \end{aligned} \tag{4.109}$$

Taking the forgetting factor in (4.107) yields

$$\begin{aligned} Q_m &= \left[\sum_{\ell=0}^{m} \rho^{m-\ell}\, |Y_k(\ell)|^2 \right]^{-1} \\ &= \left[\rho \sum_{\ell=0}^{m-1} \rho^{m-\ell-1}\, |Y_k(\ell)|^2 + |Y_k(m)|^2 \right]^{-1} \\ &= \left[\rho\, Q_{m-1}^{-1} + |Y_k(m)|^2 \right]^{-1}. \end{aligned} \tag{4.110}$$

Using the *matrix inversion lemma* [Haykin 2002]

$$[\,\mathbf{B}^{-1}+\mathbf{C}\mathbf{D}^{-1}\mathbf{C}^{H}\,]^{-1}=\mathbf{B}-\mathbf{B}\mathbf{C}\,[\,\mathbf{D}+\mathbf{C}^{H}\mathbf{B}\mathbf{C}\,]^{-1}\mathbf{C}^{H}\mathbf{B}, \tag{4.111}$$

and making the following substitutions:

$$\begin{aligned}\mathbf{B} &= \rho^{-1}\,Q_{m-1},\\ \mathbf{C} &= Y_k^*(m),\\ \mathbf{D} &= 1,\end{aligned} \tag{4.112}$$

we obtain the recursive form of Q_m

$$\begin{aligned}Q_m &= \rho^{-1}\,Q_{m-1}-\frac{\rho^{-1}\,Q_{m-1}\,Y_k^*(m)}{1+\rho^{-1}\,Q_{m-1}\,|Y_k(m)|^2}\,\rho^{-1}\,Y_k(m)\,Q_{m-1}\\ &= \rho^{-1}\,[\,1-K_{k,m}\,Y_k(m)\,]\,Q_{m-1},\end{aligned} \tag{4.113}$$

where the gain vector $K_{k,m}$ is defined as

$$K_{k,m}=\frac{\rho^{-1}\,Q_{m-1}\,Y_k^*(m)}{1+\rho^{-1}\,Q_{m-1}\,|Y_k(m)|^2}. \tag{4.114}$$

Following similar formulation as in (4.110), we can write the non-inverse term in (4.108) as

$$\underline{\mathbf{Y}}_m^H\mathbf{W}_m\underline{\mathbf{S}}_m=\rho\,\underline{\mathbf{Y}}_{m-1}^H\,\mathbf{W}_{m-1}\underline{\mathbf{S}}_{m-1}+Y_k^*(m)\,S_k(m). \tag{4.115}$$

Finally, using (4.108), (4.110), and (4.115), we obtain the recursive equation for $G_k(m)$

$$\begin{aligned}G_k(m) &= \rho^{-1}\,[\,1-K_{k,m}\,Y_k(m)\,]\,Q_{m-1}\cdot\,[\,\rho\,\underline{\mathbf{Y}}_{m-1}^H\mathbf{W}_{m-1}\underline{\mathbf{S}}_{m-1}+Y_k^*(m)\,S_k(m)\,]\\ &= [\,1-K_{k,m}\,Y_k(m)\,]\,G_k(m-1)\;+\;\rho^{-1}\,[\,1-K_{k,m}\,Y_k(m)\,]\,Q_{m-1}\,Y_k^*(m)\,S_k(m)\\ &= G_k(m)^{(1)}+G_k(m)^{(2)},\end{aligned} \tag{4.116}$$

where the second factor of (4.116), denoted by $G_k(m)^{(2)}$, can be written as

$$\begin{aligned}G_k(m)^{(2)} &= \rho^{-1}\,Q_{m-1}\,Y_k^*(m)\,S_k(m)-\rho^{-1}\,K_{k,m}\,Y_k(m)\,Q_{m-1}\,Y_k^*(m)\,S_k(m)\\ &= K_{k,m}\,S_k(m).\end{aligned} \tag{4.117}$$

Thus, the so-called recursive gain least-squares (RGLS) is formulated as

$$G_k^{RGLS}(m)=G_k^{RGLS}(m-1)+K_{k,m}\,r_{k,m}, \tag{4.118}$$

where the residual $r_{k,m}$ is defined as

$$\begin{aligned} r_{k,m} &= S_k(m) - Y_k(m)\, G_k(m-1) \\ &= \sqrt{|Y_k(m)|^2 - |N_k(m)|^2 - Y_k^*(m)N_k(m) - N_k^*(m)Y_k(m)}\ \cdot\ e^{j\theta_{S_k}(m)} - \\ &\quad - Y_k(m)\, G_k(m-1). \end{aligned} \tag{4.119}$$

The above residual formulation requires the knowledge of clean speech and noise phases which is not available. A modification is therefore proposed and the final weighting rule is implemented with the following residual formulation [Setiawan et al. 2005a]:

$$r_{k,m} = \sqrt{E_{Y_k}(m) - \alpha \cdot E_{N_k}(m)}\ \cdot\ e^{j\theta_{Y_k}(m)} - Y_k(m)\, G_k(m-1), \tag{4.120}$$

where the energy estimates $E_{Y_k}(m)$ and $E_{N_k}(m)$ are calculated as in (4.81) and (4.82), respectively. The proposed formulation only adds a small amount of additional complexity and memory requirements.

4.4 Parameter Optimization: A Multidimensional Optimization Task

All the weighting rules presented previously have several parameters which are subject to an optimization task for a particular application. For speech enhancement applications for example, a proper subjective listening test is sufficient in order to determine a suitable set of parameters. This is usually not the case when applied to speech recognition applications due to the practices such as employing a different length of analysis window and overlap in the spectral analysis and the fact that a recognizer does not have the same properties as a human ear.

The parameters used in the weighting rules are generally listed as the overestimation factor, *floor*, and some specific PSD estimation coefficients. Based on our investigation, applying a different overestimation factor for a particular segmental SNR level is boosting the recognition performance of a small vocabulary task. Setting a *floor* to a weighting rule can be done explicitly as

$$G = \max\{G,\, G_{\min}\},$$

where $G_{\min}$ indicates the *floor*. Other approaches are directly setting the floor to the segmental SNR term used in the weighting rule. The specific PSD estimation coefficients vary from one weighting rule to another. An *a priori* SNR based Wiener filtering requires only one coefficient, i.e., the smoothing coefficient for its *decision directed* approach while the least-squares weighting

rules use two different smoothing coefficients for the speech and noise PSDs.

The problem of finding an optimal parameter set for a specific recognition task is therefore formulated as a non-linear multidimensional optimization task. Assuming that $f_\theta(\cdot)$ is a recognition function with a parameter set θ, the goal is to find an optimum parameter set $\hat{\theta}$ which minimizes the recognition function. Depending on a given recognition task, this could be a time-consuming process and there is no guarantee that an optimum parameter set for a particular task is an optimum set for another task. In other words, a different recognition task can be viewed as having a different recognition function with a different global optimum.

Since this multidimensional optimization task can not be solved analytically, a numerical method is commonly used to solve the problem, e.g., in [Press et al. 1992]. Following the methods implemented in [Freeman et al. 2001, Grumm 1999], we can generally categorize the optimization task into *single-shot* and *scatter-shot* methods. The single-shot method investigates the behavior of the function around a given starting point and performs a movement based on a guess of the best direction to move. This method is relatively quick but depends heavily on the given starting point since it assumes that the global optimum is located near the given point. Some well-known single-shot algorithms are the Levenberg-Marquardt, Nelder-Mead simplex, and Powell. The scatter-shot method attempts to search the entire parameter space for a better optimum than the given initial point. This method requires many function evaluations which makes it a slow process. Some of the algorithms within this scope are the grid, Monte Carlo, and simulated annealing. It is also recommended to perform the optimization by starting it with a scatter-shot method and refining the search with a single-shot method. The algorithms categorized in the single-shot method are also known in the context of *direct search* [Kolda et al. 2003, Lewis et al. 2000].

For our particular purpose to minimize the recognition function, we are basically using the Monte Carlo algorithm to search for a possible optimum within the range of the parameter space and then refining the search based on one of the classical direct search algorithms, i.e., the *compass search*. In the Monte Carlo algorithm, all parameters are randomly varied to find other possible local optimums. It is often the case that the initial point is chosen based on the parameters yielding the best informal subjective listening test. Finally, three best local optimums are selected and the compass search is performed on the three local optimums. The compass search is implemented as follows:

1. Based on a given starting point, it randomly generates a set of directions to move, one parameter at a time. Note that a single parameter can only move in one-dimensional plane.

2. After evaluating the function on this set, the directions are ordered so that the point will move towards the most promising direction first. This set of directions serves as the *initial*

reference set of directions during the search.

3. If indeed an optimum is observed in a particular direction, the point is immediately moved and the reference set is updated where the current direction is considered as the most promising one.

4. If an optimum is not observed, the algorithm evaluates the function using an *extended set* of directions where it simply contains the opposite directions of the original one and is reversely ordered.

5. If an optimum is observed using the extended set, the algorithm simply repeats steps 3 and 4, otherwise it terminates the search.

6. If there are more than one point yielding equal optimum after terminating the search, a new search is initiated on each of those points following step 3-5 using the initial reference set of directions.

Compared to the classical compass search, our algorithm generates a reference set of directions which is reducing the amount of function evaluations and is accelerating the movement by first evaluating the most promising direction. In addition to that, the algorithm is using a fixed step size for the parameter grid by taking into account the function sensitivity with respect to a particular parameter. The optimized weighting rule parameters obtained from this optimization task are the ones used in the experiments presented in this thesis.

Chapter 5

The Concept of Entropy for Feature Vector Analysis

To evaluate the effectiveness of the front-end algorithms automatic speech recognition experiments are performed. Feature vectors leading to higher recognition rate under given test conditions are preferred. As stated previously in Chapter 2, the recognition rate depends not only on the properties of the feature vectors $\mathbf{x}^M$ but also on the acoustic and language models, $p_\Lambda(\mathbf{x}^M|W)$ and $P_\Theta(W)$, respectively.

The goal in this chapter is to derive a method to investigate the quality of the feature vectors without using the HMM modeling of the recognizer to avoid the influence of the models in the analysis of the feature vectors. An approach into this direction is the concept of entropy which describes the information contained in the feature vectors with respect to the speech units to be recognized. As speech units we use states as used in the HMM approach and regard the information contained in the feature vectors to recognize states.

In the following we only regard the isolated word recognition tasks, where an isolated word W is to be recognized. An extension to the continuous speech recognition leads not to more insight to our problem, as we regard only the information contained in the feature vectors with respect to the states. We assume that we have initially segmented the speech signal into a sequence of correctly recognized states by means of forced Viterbi training using HMM.

One way to assess the quality of feature vectors is by measuring the state error rate using the following maximum likelihood state recognizer:

$$\widehat{Q}_{ML} = \arg\max_{Q_j} \; p(\mathbf{x}|Q_j). \tag{5.1}$$

This approach is motivated by the assumption that a high recognition rate on the state level leads to a high recognition rate on the word level. Feature vectors leading to a higher recognition rate for the states contain more information. However, experiments have shown that minimizing the error rate on the state level does not always correspond to minimizing the error on the word or sentence level.

Instead of using the state error rate concept given by (5.1), we will apply the concept of entropy in order to assess the quality of the feature vectors for recognition. In this approach the recognition is modeled as a coding problem. A source sends a chain of symbols ψ^M (the input alphabet) which is transmitted through a channel. The channel emits the feature vectors $\mathbf{x}^M$ (the output alphabet). The concept of entropy gives an answer to the question of how much information is contained in the sequence of feature vectors $\mathbf{x}^M$ in order to reconstruct the transmitted chain ψ^M.

To proceed with this concept we need to define some entropy terms related to the feature vector random variable $\mathbf{X} = [\, X_1 \ \cdots \ X_d \,]^T$. To reconstruct a given state Q_j without errors the information

$$H(Q) = -\sum_{j=1}^{N_Q} P(Q_j) \text{ ld } P(Q_j), \tag{5.2}$$

is needed, where $H(Q)$ denotes the entropy of the state and ld denotes the logarithm to the base 2. The Shannon's conditional entropy [Shannon 1948] or *equivocation* which is defined as

$$H(Q|\mathbf{X}) = -\sum_{j=1}^{N_Q} P(Q_j|\mathbf{X}) \text{ ld } P(Q_j|\mathbf{X}), \tag{5.3}$$

describes the information missing to reconstruct the state from the feature vector. It can also be written as

$$\begin{aligned} H(Q|\mathbf{X}) &= H(\mathbf{X},Q) - H(\mathbf{X}) \\ &= H(Q) - [H(\mathbf{X}) - H(\mathbf{X}|Q)], \end{aligned} \tag{5.4}$$

If $H(Q|\mathbf{X})$ equals 0, we have an error free *recognition channel*. It is shown in (5.4) that $H(\mathbf{X}) - H(\mathbf{X}|Q)$ represents the amount of information contained in the feature vectors to recognize the states. The quantity is called the mutual information

$$I(\mathbf{X};Q) = H(\mathbf{X}) - H(\mathbf{X}|Q) \geq 0. \tag{5.5}$$

We have $I(\mathbf{X};Q) = 0$ if $\mathbf{X}$ contains no information about Q and $I(\mathbf{X};Q) = H(Q)$ if $\mathbf{X}$ contains all

information about Q. Due to the relation

$$\begin{aligned} H(Q|\mathbf{X}) &= H(Q) - I(\mathbf{X};Q) \geq 0, \\ H(Q) &\geq I(\mathbf{X};Q) \geq 0, \end{aligned} \tag{5.6}$$

we can state that minimizing the uncertainty $H(Q|\mathbf{X})$ to reconstruct the sequence of states is equivalent to maximizing the mutual information.

Let us also define the minimal probability of error P_e in estimating the state Q given an observed feature vector $\mathbf{X} = \mathbf{x}$

$$P_e(Q|\mathbf{x}) = 1 - \widehat{Q}_{MAP}, \tag{5.7}$$

where

$$\widehat{Q}_{MAP} = \arg\max_{Q_j} \; p(Q_j|\mathbf{x}), \tag{5.8}$$

is the maximum *a posteriori* estimator (MAP). The expected minimal error probability is shown as the Bayes probability of error or Bayes risk P_B

$$\begin{aligned} P_B &= E_{\mathbb{X}}\Big[\, P_e(Q|\mathbf{x}) \,\Big] \\ &= \int_{\mathbb{X}} \Big[\, 1 - \arg\max_{Q_j} \; p(Q_j|\mathbf{x}) \,\Big] \; p(\mathbf{x}) \; d\mathbf{x}, \end{aligned} \tag{5.9}$$

where $\mathbb{X}$ denotes the set of all possible outcomes of the random variable $\mathbf{X}$ which can be decomposed into its dimensional subsets $\mathbb{X} = [\, \mathbb{X}_1 \; \cdots \; \mathbb{X}_d \,]^T$.

As will be discussed in the next section, it will be interesting to see that the Bayes probability of error P_B is upper and lower bounded by the uncertainty measure $H(Q|\mathbf{X})$. The goal of minimizing the uncertainty measure can, therefore, also be seen as minimizing the probability of wrongly recognizing the state Q given a feature vector $\mathbf{x}$. In this case, having an accurate estimate of $H(Q|\mathbf{X})$ is crucial and it is done by estimating $H(Q)$ and $I(\mathbf{X};Q)$ which will be presented in Sections 5.2 and 5.3, respectively.

5.1 Uncertainty Bounds of the Bayes Probability of Error

The relation between the Bayes probability of error and Shannon's conditional entropy is presented in this section. It is known that there is no one-to-one relation between the Bayes probability of error and the conditional entropy. The relation exists in the formulation of the upper and lower bounds of the probability of error in terms of the entropy. While the Fano bound [Fano 1961] is shown as a tight upper bound of the probability of error, the lower bound inequality has

been the subject of several works in the area of information theory. The Fano bound is shown as

$$H(Q|\mathbf{X}) \leq H(P_B) + P_B \text{ ld } (N_Q - 1), \tag{5.10}$$

where $H(P_B) = -P_B \text{ ld } P_B - (1-P_B) \text{ ld } (1-P_B)$ is the binary entropy function.

In this thesis we are showing three different lower bounds for the probability of error, which have been independently proposed. The first bound is the Chu and Chueh bound [Chu and Chueh 1966], which proposed to lower bound the probability of error by a constant factor of 2

$$H(Q|\mathbf{X}) \geq 2 \cdot P_B. \tag{5.11}$$

An alternative proof of this bound was proposed in [Hellman and Raviv 1970]. The second bound is called the Höge bound [Höge 1999], which was proposing to lower bound the probability of error by another constant factor, i.e., ln 2

$$H(Q|\mathbf{X}) \geq \frac{1}{\ln 2} \cdot P_B. \tag{5.12}$$

The proof is rewritten in Appendix B. The third bound is called the Golić bound and was originally proposed in [Golić 1987]. This bound was formulated in a general form to accommodate not only the conditional entropy as formulated in (5.3), but also other concave information measures such as the Vajda's average conditional quadratic entropy. Using the conditional entropy, the bound is shown as

$$H(Q|\mathbf{X}) \geq j(j+1) \cdot \left(P_B - \frac{j-1}{j}\right) \cdot \text{ld } \left(\frac{j+1}{j}\right) + \text{ld } j \qquad \frac{j-1}{j} \leq P_B \leq \frac{j}{j+1}, \tag{5.13}$$

where $1 \leq j \leq N_Q - 1$. This bound has also been derived in [Feder and Merhav 1994] using the specific case of conditional entropy. All of the lower bounds shown above are valid for $P_B \in [0,\ 1 - 1/N_Q]$ and $N_Q \geq 2$. The error bounds are depicted in Figure 5.1. The Fano and Golić bounds provide a tight upper and lower bound of the Bayes probability of error in terms of the conditional entropy.

5.2 Estimating $H(Q)$

As shown in (5.6), the first step to obtain the conditional entropy $H(Q|\mathbf{X})$ is by estimating the value of $H(Q)$. The state entropy is a discrete entropy and is estimated according to the following. Given a fixed number of N_Q different states in the set Q and $P(Q_j)$ as the probability of being in

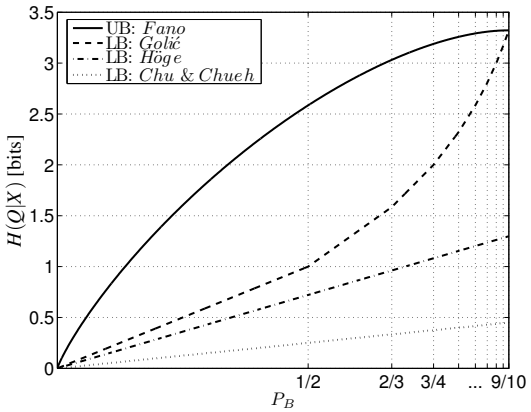


Figure 5.1: An upper bound (UB) and several lower bounds (LB) of Bayes probability of error P_B with $N_Q = 10$.

the state Q_j, the entropy $H(Q)$ is calculated as

$$H(Q) = -\sum_{j=1}^{N_Q} P(Q_j) \text{ ld } P(Q_j),$$

where the probability $P(Q_j)$ has the properties

$$0 \le P(Q_j) \le 1 \qquad \text{and} \qquad \sum_{j=1}^{N_Q} P(Q_j) = 1. \tag{5.14}$$

It can be shown that

$$H(Q) \le -\sum_{j=1}^{N_Q} \frac{1}{N_Q} \text{ ld } \frac{1}{N_Q} = \text{ld } N_Q, \tag{5.15}$$

i.e., the entropy is maximized if the states are uniformly distributed.

The discrete probabilities $P(Q_j),\ j = 1 \cdots N_Q$ are estimated from a finite set of N observed

occurrences of all states. With the definitions

$$\begin{aligned} n_j &= \text{number of occurrences of a state } Q_j, \\ N &= \sum_{j=1}^{N_Q} n_j, \end{aligned}$$

we define

$$P(Q_j) \approx \widetilde{P}_j = \frac{n_j}{N}, \tag{5.16}$$

$$\widetilde{H}(Q) = -\sum_{j=1}^{N_Q} \widetilde{P}_j \text{ ld } \widetilde{P}_j. \tag{5.17}$$

If the occurrence process of the states in Q is stationary both the approximated probabilities $\widetilde{P}_j$ and the approximated entropies $\widetilde{H}(Q)$ converge to their true probabilities and entropies, $P(Q_j)$ and $H(Q)$, respectively, provided that N goes to infinity. Since the approximated probabilities have the properties described by (5.14) they have the properties of the true probabilities. Hence, (5.15) also holds

$$\widetilde{H}(Q) \leq \text{ld } N_Q.$$

5.3 Approximations to the Mutual Information $I(\mathbf{X};Q)$

The second step of the conditional entropy calculation is done by estimating the mutual information $I(\mathbf{X};Q)$. According to (5.5) the mutual information is defined by

$$I(\mathbf{X};Q) = H(\mathbf{X}) - H(\mathbf{X}|Q), \tag{5.18}$$

where

$$H(\mathbf{X}) = -\int_{\mathbb{X}} p(\mathbf{x}) \text{ ld } p(\mathbf{x})\, d\mathbf{x}, \tag{5.19}$$

and

$$\begin{aligned} H(\mathbf{X}|Q) &= -\int_{\mathbb{X}} \sum_{j=1}^{N_Q} p(\mathbf{x}, Q_j) \text{ ld } p(\mathbf{x}|Q_j)\, d\mathbf{x} \\ &= \sum_{j=1}^{N_Q} P(Q_j)\, H(\mathbf{X}|Q_j), \end{aligned} \tag{5.20}$$

with

$$H(\mathbf{X}|Q_j) = -\int_{\mathbb{X}} p(\mathbf{x}|Q_j)\ \mathrm{ld}\ p(\mathbf{x}|Q_j)\ d\mathbf{x}. \tag{5.21}$$

In order to determine $I(\mathbf{X};Q)$ we have to handle the distributions $P(Q_j)$, $p(\mathbf{x}|Q_j)$, and $p(\mathbf{x})$. The approximation to $P(Q_j)$ is following (5.16) and $p(\mathbf{x}|Q_j)$ is discussed in Section 5.4 where the entropy $H(\mathbf{X}|Q_j)$ is calculated. Finally, $p(\mathbf{x})$ is approximated in Section 5.5 where the entropy $H(\mathbf{X})$ is estimated.

Since the feature vectors are continuous random variables, in general their statistical properties are described by (continuous) density functions $p(\mathbf{x})$ and $p(\mathbf{x}|Q_j)$. In practice the distributions have to be approximated either by discrete distributions (histograms) or by continuous distributions, e.g., a Gaussian mixture, where the parameters *mean vector* and *covariance matrix* have to be determined empirically. Both approximations are based on a set of observed pairs $\{(\mathbf{x},Q_j)\}$.

In this thesis $p(\mathbf{x}|Q_j)$ is assumed to be a monomodal Gaussian distribution, which leads to $p(\mathbf{x})$ having a Gaussian mixture distribution

$$p(\mathbf{x}) \sim \sum_{j=1}^{N_Q} p(\mathbf{x},Q_j) = \sum_{j=1}^{N_Q} P(Q_j)p(\mathbf{x} \mid Q_j). \tag{5.22}$$

It is worth noting that the distribution $p(\mathbf{x}|Q_j)$ in the HMM analysis is in fact usually modeled by a Gaussian mixture. Based on the assumption, there exists an analytical solution for $H(\mathbf{X}|Q)$ and not for $H(\mathbf{X})$ due to the multimodal properties of $p(\mathbf{x})$.

As mentioned previously, the *histogram* method to solve the entropy $H(\mathbf{X})$ was proposed in [Moddemeijer 1989] where it assumes that the observed pairs $\{(\mathbf{x},Q_j)\}$ are independent. In case of the dependent pair observations, the entropy estimation is discussed in [Moddemeijer 1999]. This method is suitable to determine an entropy value where no assumption is made on the underlying distribution of the observed pairs. When a certain density function is assumed on the observed pairs, such as the Gaussian mixture for $p(\mathbf{x})$ as proposed in this thesis, it might be better to estimate the parameters of the density function and determine its entropy based on the sampled values of the density function. When this alternative method is used, the final entropy has to take into account the bias introduced by the partition width as described in Appendix C.3.

In order to compare the performance of both methods, an experiment involving the one-dimensional data of a random variable X generated by a Gaussian distribution was conducted. The true entropy value of this density is analytically solvable as shown in Appendix C.4. In addition to that, depending on the total number of observations N_X available, the entropy is estimated following the method in [Moddemeijer 1989] and our proposed alternative method. In

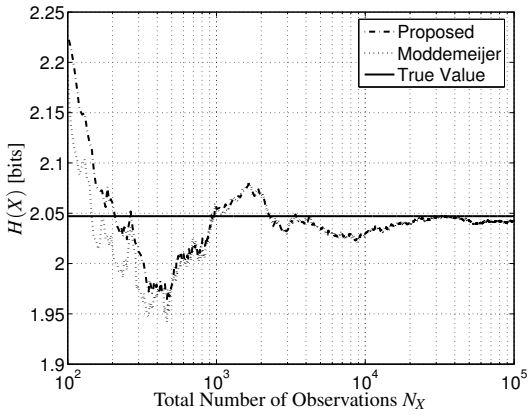


Figure 5.2: A comparison of two *histogram* methods to solve for $H(X)$ when the underlying density function, i.e., Gaussian in this example, is known.

our proposed method, the parameters describing the underlying Gaussian density are the sample mean and variance.

As shown in Figure 5.2, both methods are performing equally good when at least around 600 data is available. When only between 200 - 600 data is available, our proposed method is better than [Moddemeijer 1989] and the performance is reversed when only fewer data is available, i.e., below 180. The overall mean squared error is shown as 0.57 and 0.63 bits for the proposed method and the method in [Moddemeijer 1989], respectively. Throughout this thesis, we are going to use the proposed method to calculate an entropy of an arbitrary multimodal density data X by assuming an underlying Gaussian mixture, unless otherwise noted. In Section 5.5 we will elaborate more on this topic especially when dealing with the multidimensional data $\mathbf{X}$.

5.4 Approximation to $H(\mathbf{X}|Q)$

We assume that the d-dimensional multivariate conditional density $p(x_1, \cdots, x_d|Q_j)$ is monomodal Gaussian. The conditional density is defined as

$$p(\mathbf{x}|Q_j) = p(x_1, \cdots, x_d|Q_j) \sim \mathcal{N}(\mu_{\mathbf{X}|Q_j}, \Sigma_{\mathbf{X}|Q_j}). \tag{5.23}$$

The entropy of a variable having a monomodal Gaussian distribution is described in Appendix C.4. Following the multivariate derivation shown in (C.5) we get

$$H(\mathbf{X}|Q_j) = \frac{d}{2} \ln (2\pi e) + \frac{1}{2} \ln |\Sigma_{\mathbf{X}|Q_j}|, \tag{5.24}$$

with $|\Sigma_{\mathbf{X}|Q_j}|$ being the matrix determinant of the conditional covariance matrix $\Sigma_{\mathbf{X}|Q_j}$.

Furthermore, from (5.20) we obtain the expression

$$H(\mathbf{X}|Q) = \frac{d}{2} \ln (2\pi e) + \frac{1}{2} \sum_{j=1}^{N_Q} P(Q_j) \ln |\Sigma_{\mathbf{X}|Q_j}|. \tag{5.25}$$

In the case where the feature vector $\mathbf{X}$ is statistically independent, the covariance matrix $\Sigma_{\mathbf{X}|Q_j}$ is

$$\Sigma_{\mathbf{X}|Q_j} = \begin{pmatrix} \sigma^2_{X_1|Q_j} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \sigma^2_{X_d|Q_j} \end{pmatrix}.$$

Hence, we obtain

$$H(\mathbf{X}|Q) = \frac{d}{2} \ln (2\pi e) + \sum_{i=1}^{d} \sum_{j=1}^{N_Q} P(Q_j) \ln \sigma_{X_i|Q_j}. \tag{5.26}$$

The result depicted in (5.26) can also be derived using the differential entropy calculation as shown in Appendix C.2. Using (5.20) and (C.3) we obtain

$$\begin{aligned} H(\mathbf{X}|Q) &= \sum_{j=1}^{N_Q} P(Q_j) \sum_{i=1}^{d} H(X_i|Q_j) \\ &= \sum_{i=1}^{d} \sum_{j=1}^{N_Q} P(Q_j)\, H(X_i|Q_j) \\ &= \sum_{i=1}^{d} H(X_i|Q), \end{aligned} \tag{5.27}$$

where we have used

$$H(X_i|Q) = \sum_{j=1}^{N_Q} P(Q_j) H(X_i|Q_j).$$

Using the monomodal Gaussian approximation for $H(X_i|Q_j)$ as in (C.4) gives exactly (5.26)

$$\begin{aligned} H(X_i|Q) &= \sum_{j=1}^{N_Q} P(Q_j) \left[\frac{1}{2} \ln (2\pi e) + \ln \sigma_{X_i|Q_j} \right] \\ &= \frac{1}{2} \ln (2\pi e) + \sum_{j=1}^{N_Q} P(Q_j) \ln \sigma_{X_i|Q_j}. \end{aligned}$$

5.5 Approximation to $H(\mathbf{X})$

Determining the entropy $H(\mathbf{X})$ given the multimodal distribution of $p(\mathbf{x})$ as defined in (5.22) is not an easy task. The difficulty is caused by the multiple integral operation due to the high dimensionality of the feature vector $\mathbf{X}$ which is in the order of $20 \cdots 40$. Since we are relying on the underlying density to estimate the entropy as described at the end of Section 5.3, it is mandatory to somehow solve the problem of multidimensionality.

In the following we will regard an approximation of $p(\mathbf{x})$ where the integral operation can be treated. The multidimensional density function $p(\mathbf{x})$ can be written in the form

$$p(\mathbf{x}) = p(x_1, \cdots, x_d) = p(x_d) \prod_{i=1}^{d-1} p(x_i|x_{i+1}, \cdots, x_d),$$

leading to

$$\begin{aligned} H(\mathbf{X}) &= -\int_{\mathbb{X}} p(\mathbf{x}) \text{ ld } \left[p(x_d) \prod_{i=1}^{d-1} p(x_i|x_{i+1}, \cdots, x_d) \right] d\mathbf{x} \\ &= -\int_{\mathbb{X}_d} p(x_d) \text{ ld } p(x_d)\, dx_d - \sum_{i=1}^{d-1} \int_{\mathbb{X}_1} \cdots \int_{\mathbb{X}_d} p(\mathbf{x}) \text{ ld } p(x_i|x_{i+1}, \cdots, x_d)\, dx_1 \ldots dx_d \\ &= H(X_d) + \sum_{i=1}^{d-1} H(X_i|X_{i+1}, \cdots, X_d). \end{aligned} \tag{5.28}$$

This representation decomposes $H(\mathbf{X})$ into a sum of marginal dimensional entropies, where each entropy in $H(X_i|X_{i+1}, \cdots, X_d)$ is dedicated to a specific dimension i. In order to reduce the dimension of $p(x_i|x_{i+1}, \cdots, x_d)$ we consider the monogram, bigram, and n-gram approximations.

5.5.1 Monogram Approximation

The following approximation is used:

$$p(x_i|x_{i+1},\ \cdots,\ x_d) \approx p(x_i), \tag{5.29}$$

which gives

$$\begin{aligned} p(\mathbf{x}) &= p(x_d)\prod_{i=1}^{d-1} p(x_i), \\ H_1(\mathbf{X}) &= \sum_{i=1}^{d} H(X_i), \end{aligned} \tag{5.30}$$

where $H_1(\mathbf{X})$ denotes the monogram multimodal approximation of $H(\mathbf{X})$. The mutual information is calculated as

$$\begin{aligned} I_1(\mathbf{X};Q) &= H_1(\mathbf{X}) - H(\mathbf{X}|Q) \\ &= \sum_{i=1}^{d} H(X_i) - \sum_{i=1}^{d} H(X_i|Q) \\ &= \sum_{i=1}^{d} I_1(X_i;Q), \end{aligned} \tag{5.31}$$

where

$$\begin{aligned} I_1(X_i;Q) &= H(X_i) - H(X_i|Q) \\ &= H(X_i) - \frac{1}{2}\ \ln\ (2\pi e) - \sum_{j=1}^{N_Q} P(Q_j)\ \ln\ \sigma_{X_i|Q_j}. \end{aligned} \tag{5.32}$$

$H(X_i)$ is calculated using the distribution $p(x_i)$

$$H(X_i) = -\int_{\mathbb{X}_i} p(x_i)\ \mathrm{ld}\ p(x_i)\, dx_i. \tag{5.33}$$

As shown later in Section 5.6.1 several approximations of $p(x_i)$ are investigated. The value of $H(X_i|Q)$ is estimated following the calculation presented in Section 5.4.

5.5.2 Bigram Approximation

The following approximation is used:

$$p(x_i|x_{i+1},\ \cdots,\ x_d) \approx p(x_i|x_{i+1}). \tag{5.34}$$

Hence

$$\begin{aligned} p(\mathbf{x}) &= p(x_d)\prod_{i=1}^{d-1} p(x_i|x_{i+1}), \\ H_2(\mathbf{X}) &= H(X_d)+\sum_{i=1}^{d-1} H(X_i|X_{i+1}), \end{aligned} \tag{5.35}$$

where $H_2(\mathbf{X})$ denotes the bigram multimodal approximation of $H(\mathbf{X})$.

The bigram multimodal mutual information $I_2(\mathbf{X};Q)$ is calculated as

$$\begin{aligned} I_2(\mathbf{X};Q) &= H_2(\mathbf{X}) - H(\mathbf{X}|Q) \\ &= H(X_d)+\sum_{i=1}^{d-1} H(X_i|X_{i+1}) - \sum_{i=1}^{d} H(X_i|Q) \\ &= \sum_{i=1}^{d} I_2(X_i;Q), \end{aligned} \tag{5.36}$$

where

$$\begin{aligned} I_2(X_i;Q) &= \begin{cases} H(X_i|X_{i+1}) - H(X_i|Q) & i \in [1,d) \\ I_1(X_d;Q) & i = d \end{cases} \\ &= \begin{cases} H(X_i|X_{i+1}) - \frac{1}{2}\ \ln\ (2\pi e) - \sum_{j=1}^{N_Q} P(Q_j)\ \ln\ \sigma_{X_i|Q_j} & i \in [1,d), \\ I_1(X_d;Q) & i = d. \end{cases} \end{aligned}$$

The main issue is the calculation of $H(X_i|X_{i+1})$

$$H(X_i|X_{i+1}) = -\int_{\mathbb{X}_i}\int_{\mathbb{X}_{i+1}} p(x_i,x_{i+1})\ \mathrm{ld}\ p(x_i|x_{i+1})\ dx_i\ dx_{i+1}. \tag{5.37}$$

We define $p(x_i, x_{i+1})$ as a two-dimensional marginal distribution of $p(\mathbf{x})$

$$p(x_i, x_{i+1}) = \int_{\mathbb{X}_1} \cdots \int_{\mathbb{X}_{i-1}} \int_{\mathbb{X}_{i+2}} \cdots \int_{\mathbb{X}_d} p(\mathbf{x})\, dx_1 \ldots dx_{i-1}\, dx_{i+2} \ldots dx_d, \tag{5.38}$$

leading to

$$p(x_i|x_{i+1}) = \frac{p(x_i, x_{i+1})}{p(x_{i+1})}. \tag{5.39}$$

Following the same derivation as in (C.6) which relates $p(x_i)$ to $p(x_i|Q_j)$ and the Gaussian assumption of $p(x_i|Q_j)$ we get

$$\begin{aligned} p(x_i, x_{i+1}) &= \sum_{j=1}^{N_Q} P(Q_j) \prod_{i=0}^{1} \mathcal{N}(\mu_{X_{i+i}|Q_j}, \sigma^2_{X_{i+i}|Q_j}), & (5.40)\\ p(x_{i+1}) &= \sum_{j=1}^{N_Q} P(Q_j)\, \mathcal{N}(\mu_{X_{i+1}|Q_j}, \sigma^2_{X_{i+1}|Q_j}). & (5.41) \end{aligned}$$

5.5.3 n-gram Approximation

The following approximation is used:

$$p(x_i|x_{i+1},\ \cdots,\ x_d) \approx p(x_i|x_{i+1},\ \cdots,\ x_{i+n-1}). \tag{5.42}$$

Hence,

$$\begin{aligned} p(\mathbf{x}) &= p(x_d) \prod_{i=1}^{d-1} p(x_i|x_{i+1},\ \cdots,\ x_{i+n-1}),\\ H_n(\mathbf{X}) &= H(X_d) + H(X_{d-1}|X_d) + \cdots + \sum_{i=1}^{d-n+1} H(X_i|X_{i+1}, \cdots, X_{i+n-1}), \end{aligned} \tag{5.43}$$

where

$$H(X_i|X_{i+1},\ \cdots,\ X_{i+n-1}) =$$

$$-\int_{\mathbb{X}_i} \cdots \int_{\mathbb{X}_{i+n-1}} p(x_i,\ \cdots,\ x_d)\ \mathrm{ld}\ p(x_i|x_{i+1},\ \cdots,\ x_{i+n-1})\, dx_i \ldots dx_{i+n-1},$$

and $H_n(\mathbf{X})$ denotes the n-gram multimodal approximation of $H(\mathbf{X})$.

The n-gram multimodal mutual information $I_n(\mathbf{X};Q)$ is calculated as

$$\begin{aligned} I_n(\mathbf{X};Q) &= H_n(\mathbf{X}) - H(\mathbf{X}|Q) \\ &= H(X_d) + H(X_{d-1}|X_d) + \cdots + \sum_{i=1}^{d-n+1} H(X_i|X_{i+1},\ \cdots,\ X_{i+n-1}) - \sum_{i=1}^{d} H(X_i|Q) \\ &= \sum_{i=1}^{d} I_n(X_i;Q), \end{aligned} \tag{5.44}$$

where

$$I_n(X_i;Q) = \begin{cases} H(X_i|X_{i+1},\ \cdots,\ X_{i+n-1}) - H(X_i|Q) & i \in [1,\ d-n+2), \\ I_2^{(d+1-i)}(X_i;Q) & i \in [d-n+2,\ d]. \end{cases}$$

In general, the relation $H_1(\mathbf{X}) \geq H_2(\mathbf{X}) \geq H_n(\mathbf{X})$ holds [Papoulis 1991], i.e., more accurate modeling leads to lower entropy.

5.5.4 Monomodal Gaussian Approximation of $H(\mathbf{X})$

As a special case, we assume that $p(\mathbf{x})$ has a monomodal Gaussian distribution neglecting (5.22)

$$p(\mathbf{x}) \sim \mathcal{N}(\mu_{\mathbf{X}}, \Sigma_{\mathbf{X}}). \tag{5.45}$$

Using (C.5) we get

$$\begin{aligned} H_G(\mathbf{X}) &= \frac{d}{2}\ \ln\ (2\pi e) + \frac{1}{2}\ \ln\ |\Sigma_{\mathbf{X}}| \\ &= \frac{d}{2}\ \ln\ (2\pi e) + \frac{1}{2}\sum_{j=1}^{N_Q} P(Q_j)\ \ln\ |\Sigma_{\mathbf{X}}|, \end{aligned}$$

where $H_G(\mathbf{X})$ denotes the monomodal approximation of $H(\mathbf{X})$. Using $H(\mathbf{X}|Q)$ as in (5.25) the mutual information is expressed by

$$\begin{aligned} I_G(\mathbf{X};Q) &= H_G(\mathbf{X}) - H(\mathbf{X}|Q) \\ &= \frac{1}{2}\sum_{j=1}^{N_Q} P(Q_j)\ \ln\ \frac{|\Sigma_{\mathbf{X}}|}{|\Sigma_{\mathbf{X}|Q_j}|}. \end{aligned} \tag{5.46}$$

$I_G(\mathbf{X};Q)$ is the monomodal Gaussian approximation of $I(\mathbf{X};Q)$. In the case where the covariance matrices of $p(\mathbf{x})$ and $p(\mathbf{x}|Q_j)$ are diagonal, $p(\mathbf{x})$ can be factorized as

$$p(\mathbf{x}) = \prod_{i=1}^{d} p(x_i), \tag{5.47}$$

with $p(x_i) \sim \mathcal{N}(\mu_{X_i}, \sigma_{X_i}^2)$.

Applying a similar derivation as in (5.27) it can be shown that

$$H_G(\mathbf{X}) = \sum_{i=1}^{d} H(X_i),$$

with

$$\begin{aligned} H_G(X_i) &= \frac{1}{2} \ln\ (2\pi e) + \ln\ \sigma_{X_i} \\ &= \frac{1}{2} \ln\ (2\pi e) + \sum_{j=1}^{N_Q} P(Q_j) \ln\ \sigma_{X_i}. \end{aligned}$$

Defining

$$I_G(X_i;Q) = H_G(X_i) - H(X_i|Q),$$

we get

$$\begin{aligned} I_G(\mathbf{X};Q) &= H_G(\mathbf{X}) - H(\mathbf{X}|Q) \\ &= \sum_{i=1}^{d} I_G(X_i;Q), \end{aligned} \tag{5.48}$$

where

$$I_G(X_i;Q) = \sum_{j=1}^{N_Q} P(Q_j) \ln \frac{\sigma_{X_i}}{\sigma_{X_i|Q_j}}. \tag{5.49}$$

This result shows, that the mutual information $I_G(\mathbf{X};Q)$ can be determined by the sum of the mutual informations $I_G(X_i;Q)$ corresponding to each dimension i. To evaluate (5.49) the value σ_{X_i} has to be determined. Empirically calculating σ_{X_i} will result in the same variance calculation assuming multimodal distribution of $p(\mathbf{x})$ in (5.22).

5.6 Sample Cases and Analysis

5.6.1 Monogram Approximation One-Dimensional Example

To get a first insight into the calculation of the mutual information we start with a one-dimensional feature x. In the one-dimensional case the monogram approximation leads to the exact solution provided the correct distributions are used. As a classification task we assume that we have to distinguish 2 states Q_1 and Q_2 with equal probability $P(Q_j),\ j = 1,\ 2$. As described in Section 5.2 $H(Q)$ is given by

$$H(Q) = \text{ld } N_Q = 1 \text{ bit}, \tag{5.50}$$

where $P(Q_1) = P(Q_2) = 0.5$ and $N_Q = 2$. Furthermore, $I(X;Q)$ is bounded by

$$I(X;Q) = H(X) - H(X|Q) \leq H(Q) = 1 \text{ bit}. \tag{5.51}$$

According to the assumptions described in Section 5.3, we define the distributions $p(x|Q_1)$ and $p(x|Q_2)$ as monomodal Gaussian distributed

$$p(x|Q_j) = \mathcal{N}(\mu_{X|Q_j}, \sigma^2_{X|Q_j}),$$

having $\mu_{X|Q_1} = 15,\ \mu_{X|Q_2} = -15$, and $\sigma^2_{X|Q_1} = \sigma^2_{X|Q_2} = 100$. Consequently, the distribution of $p(x)$ is given by

$$p(x) = \sum_{j=1}^{2} P(Q_j) \cdot \mathcal{N}(\mu_{X|Q_j}, \sigma^2_{X|Q_j}),$$

according to (5.22). Furthermore, as treated in Section 5.5.4 $p(x)$ can be approximated by a monomodal Gaussian distribution $p(x) \sim \mathcal{N}(\mu_X, \sigma^2_X)$, where the mean value and variance are shown according to (C.7) and (C.8), respectively, as

$$\mu_X = \sum_{j=1}^{2} \frac{1}{2}\, \mu_{X|Q_j} = 0, \tag{5.52}$$

$$\sigma^2_X = \sum_{j=1}^{2} \frac{1}{2} \Big[\, \sigma^2_{X|Q_j} + (\mu_X - \mu_{X|Q_j})^2 \,\Big]. \tag{5.53}$$

Figure 5.3 shows both distributions. It is obvious that the monomodal Gaussian distribution differs from the Gaussian mixture distribution. In the overlapping region the information given by x is insufficient to reconstruct the states Q_1 and Q_2, i.e., in these regions a maximum likelihood classifier will make errors. The monomodal Gaussian approximation of the mutual information

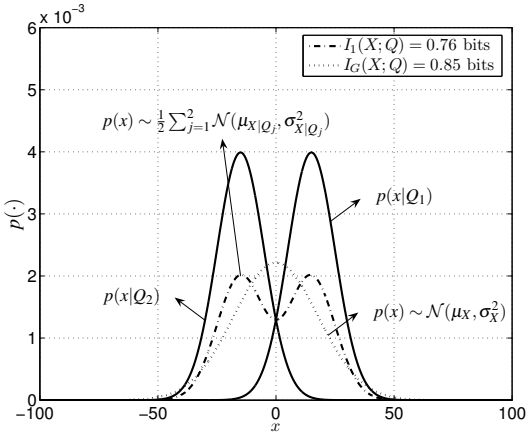


Figure 5.3: Monomodal Gaussian and bimodal Gaussian mixture distributions of a one-dimensional two-class classification task.

is shown in (5.49) as

$$I_G(X;Q) = \sum_{j=1}^{2} P(Q_j)\ \ln\ \frac{\sigma_X}{\sigma_{X|Q_j}}. \tag{5.54}$$

The calculation of Gaussian mixture mutual information is done following the monogram approach given by (5.32)

$$I_1(X;Q) = H_1(X) - H(X|Q), \tag{5.55}$$

with

$$\begin{aligned} H(X|Q) &= \frac{1}{2}\ \ln\ (2\pi e) + \sum_{j=1}^{2} P(Q_j)\ \ln\ \sigma_{X|Q_j} \\ &= \frac{1}{2}\ \ln\ (2\pi e) + \frac{1}{2}\ \ln\ (\sigma_{X|Q_1} \cdot \sigma_{X|Q_2}), \\ H_1(X) &= -\int_{\mathbb{X}} p(x)\ \mathrm{ld}\ p(x)\, dx, \end{aligned}$$

where $p(x)$ is a bimodal Gaussian distribution

$$p(x) \sim \frac{1}{2}\sum_{j=1}^{2}\mathcal{N}(\mu_{X|Q_j}, \sigma^2_{X|Q_j}).$$

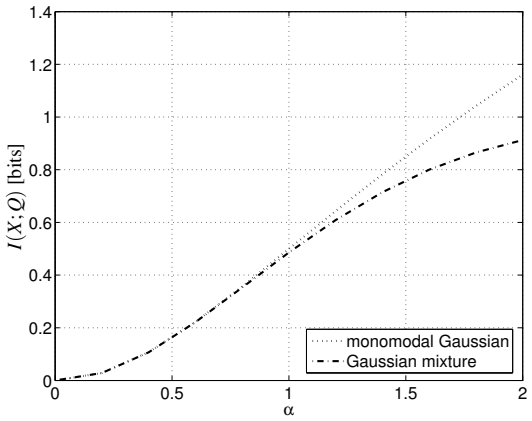


Figure 5.4: Mutual information of the monomodal Gaussian and bimodal Gaussian mixture for a specific case of one-dimensional two-class classification task.

Table 5.1 shows the mutual information calculation for both approaches with the means and variances as used in Figure 5.3. This result shows that the monomodal approximation yields a higher mutual information. Nevertheless, the monomodal result does not violate (5.6), because the value of $I(X;Q)$ is still below $H(Q) = 1$ bit. According to (5.4) the entropy $H(Q|X)$ takes the value 0.24 bit.

Now we investigate a specific case of distributions as shown previously with the properties

$$\sigma^2_{X|Q_j} = \sigma^2; \qquad \mu_{X|Q_2} = -\mu_{X|Q_1}; \qquad |\mu_{X|Q_j}| = \mu. \tag{5.56}$$

In this case (5.52) and (5.53) of the monomodal Gaussian approximation simplify to

$$\mu_X = 0; \qquad \sigma^2_X = \sigma^2 + \mu^2, \tag{5.57}$$

leading to the monomodal approximation

$$\begin{aligned} I(X;Q) &= \ln \frac{\sigma_X}{\sigma} = \ln \frac{\sqrt{\sigma^2+\mu^2}}{\sigma} \\ &= \ln \sqrt{1+\alpha^2}, \end{aligned} \tag{5.58}$$

where $\alpha = \mu/\sigma$. Figure 5.4 plots the relation between α and the corresponding mutual informa-

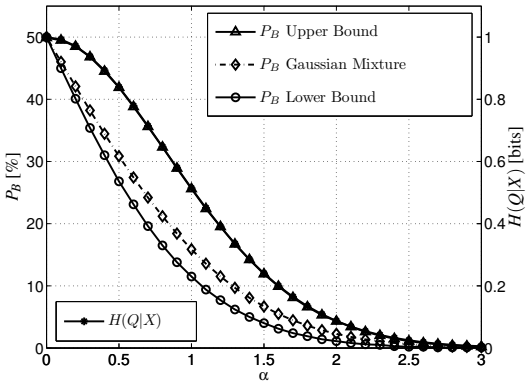


Figure 5.5: Upper and lower bounds of the probability of error using the Gaussian mixture approximation.

tion. The monomodal Gaussian mutual information is plotted following (5.58) and the Gaussian mixture mutual information is following (5.55). As shown in the figure, for small values of α ($\alpha < 1$) the mutual information of the monomodal Gaussian approximation is still giving the correct value as shown by the Gaussian mixture curve. It starts to give higher values for α higher than 1. The distributions shown in Figure 5.3 present the case of $\alpha = 1.5$, where the monomodal Gaussian approximation differs significantly from the Gaussian mixture.

Table 5.1: Mutual information of the monomodal Gaussian and bimodal Gaussian mixture approximations of a one-dimensional two-class classification task.

$p(x)$	$I(X;Q)$ [bits]	$H(X\|Q)$ [bits]	$H(X)$ [bits]
Gaussian mixture	0.76	5.369	6.1290
Monomodal Gaussian	0.85	5.369	6.2192

Basen on the Gaussian mixture plot in Figure 5.4, a set of upper and lower bounds of the probability of error can be plotted as depicted in Figure 5.5. By varying the value of α, the corresponding probability of error using the Gaussian mixture assumption can be obtained. Both upper and lower bounds of the probability of error are derived from its entropy value where we have used the Fano entropy upper bound to depict the lower bound of the probability of error and the Golić entropy lower bound to depict the upper bound of the probability of error. The figure shows that the probability of error in the Gaussian mixture classification task lies between the upper and lower bounds of the probability of error. The conditional entropy $H(Q|X)$ is also shown in the figure where it coincides with the upper bound of the probability of error which

implies that the upper bound curve of the probability of error or the Golić entropy lower bound curve has a linear curve in the probability of error vs. conditional entropy plot given a two-class classification task.

5.6.2 Bigram Approximation of a 2-dimensional Example

Now we consider an example of a distribution of a two-dimensional feature vector $\mathbf{X} = [X_1\ X_2]^T$. The two states task of Section 5.6.1 is adopted where the states Q_1 and Q_2 with equal probability have to be distinguished. The distributions $p(\mathbf{x}|Q_1)$ and $p(\mathbf{x}|Q_2)$ are monomodal Gaussian densities with diagonal covariance matrices

$$p(\mathbf{x}|Q_1) \sim \mathcal{N}(\mu_{\mathbf{X}|Q_1}, \Sigma_{\mathbf{X}|Q_1}); \qquad \mu_{\mathbf{X}|Q_1} = [-15\ 15]^T; \qquad \Sigma_{\mathbf{X}|Q_1} = \begin{pmatrix} 100 & 0 \\ 0 & 100 \end{pmatrix},$$

$$p(\mathbf{x}|Q_2) \sim \mathcal{N}(\mu_{\mathbf{X}|Q_2}, \Sigma_{\mathbf{X}|Q_2}); \qquad \mu_{\mathbf{X}|Q_2} = [15\ -15]^T; \qquad \Sigma_{\mathbf{X}|Q_2} = \begin{pmatrix} 100 & 0 \\ 0 & 100 \end{pmatrix}.$$

Given the equally distributed states Q_1 and Q_2 we have the following distribution:

$$p(\mathbf{x}) \sim \sum_{j=1}^{2} P(Q_j)\, \mathcal{N}(\mu_{\mathbf{X}|Q_j}, \Sigma_{\mathbf{X}|Q_j}); \qquad P(Q_j) = \frac{1}{2} \qquad j = 1,\ 2.$$

Figure 5.6 shows the distribution of $p(\mathbf{x})$ and Figure 5.7 depicts its monogram approximation which is calculated as

$$p(\mathbf{x}) = \prod_{i=1}^{2} p(x_i),$$

where

$$p(x_i) \sim \sum_{j=1}^{2} P(Q_j)\, \mathcal{N}(\mu_{X_i|Q_j}, \sigma^2_{X_i|Q_j}).$$

As shown in the figures, the difference between the two approximations are evident. The calculation of mutual information is done following (5.36)

$$I_2(\mathbf{X}; Q) = \sum_{i=1}^{2} I_2(X_i; Q),$$

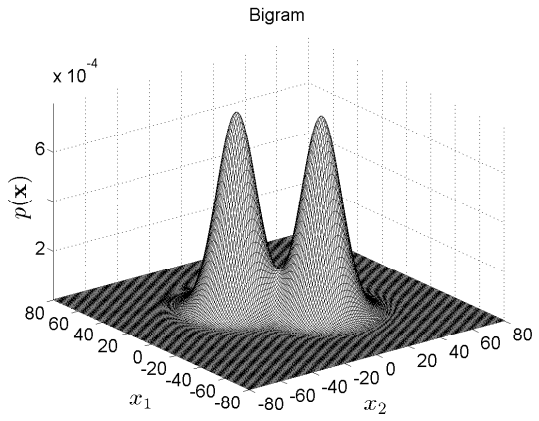


Figure 5.6: The distribution of $p(\mathbf{x})$ with the bigram approach in a two-dimensional two-class classification task.

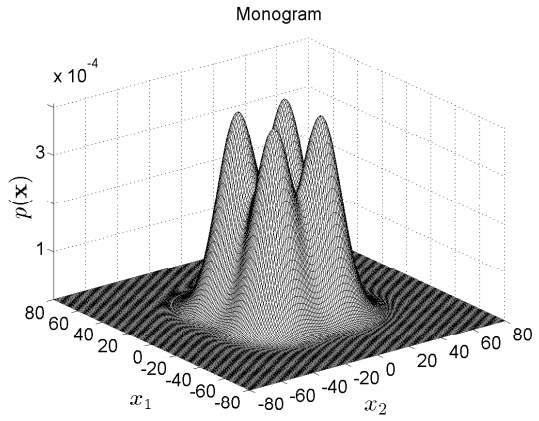


Figure 5.7: The distribution of $p(\mathbf{x})$ with the monogram approach in a two-dimensional two-class classification task.

where

$$\begin{aligned} I_2(X_1;Q) &= H(X_1|X_2) - H(X_1|Q), \\ I_2(X_2;Q) &= H(X_2) - H(X_2|Q), \end{aligned}$$

and

$$\begin{aligned} H(X_1|X_2) &= -\int_{\mathbb{X}_1}\int_{\mathbb{X}_2} p(x_1,x_2)\ \mathrm{ld}\ p(x_1|x_2)\, dx_1 dx_2, \\ H(X_2) &= -\int_{\mathbb{X}_2} p(x_2)\ \mathrm{ld}\ p(x_2)\, dx_2, \\ H(X_i|Q) &= \frac{1}{2}\ln\ (2\pi e) + \sum_{j=1}^{2} P(Q_j)\ \ln\ \sigma_{X_i|Q_j}. \end{aligned}$$

We also treat the monomodal monogram approximation $H_G(\mathbf{X})$ with the monomodal distribution on the marginals $p(x_i)$, where the resulting $I(\mathbf{X};Q)$ is given by (5.49). For the values of the variances and means as used in the example we obtain $H(\mathbf{X}) = 10.7380$ bits and the mutual information as shown in Table 5.2. The monomodal bigram approximation is giving the same result as the monomodal monogram one given the fact that the joint probability $p(x_1,x_2)$ is a statistically independent bivariate Gaussian distribution. It can also be seen that the bimodal monogram approximation leads to a rather wrong value of $I(\mathbf{X};Q)$ because it should be upperbounded by $H(Q) = 1$.

Table 5.2: Mutual information of the two-dimensional monogram and bigram approximations.

	$H(\mathbf{X})$ [bits]	$I(\mathbf{X};Q)$ [bits]
Monomodal - Monogram	10.7380	0
Bimodal - Monogram	12.2580	1.5200
Bimodal - Bigram	11.6723	0.9343

5.7 Influence of Noise on the Feature Vectors

In this section we are going to analyze the influence of noise on the feature vectors. We assume that we have access to the speech signal with and without noise. We denote the clean speech feature vector random variable with $\mathbf{X}$ and the noisy speech feature vector random variable with $\mathbf{X}_E$. The distortion feature vector random variable $\mathbf{E}$ includes the distortions caused by the noise on the speech signal and by the artifacts of noise reduction algorithms. We assume that the

distortions are additive and stationary defined by the given mean vector and covariance matrix $\mu_{\mathbf{E}}$ and $\Sigma_{\mathbf{E}}$, respectively,

$$\mathbf{X}_E = \mathbf{X} + \mathbf{E}.$$

Furthermore, we assume that the distortion feature vector $\mathbf{E}$ is statistically independent from $\mathbf{X}$

$$p(\mathbf{x}, \mathbf{e}) = p(\mathbf{x}) \cdot p(\mathbf{e}).$$

Denoting the mean vectors and covariance matrices of the noisy speech and clean speech vectors with $\mu_{\mathbf{X}_E}, \Sigma_{\mathbf{X}_E}$ and $\mu_{\mathbf{X}}, \Sigma_{\mathbf{X}}$, respectively, we get the relation

$$\mu_{\mathbf{X}_E} = \mu_{\mathbf{X}} + \mu_{\mathbf{E}}, \tag{5.59}$$

$$\Sigma_{\mathbf{X}_E} = \Sigma_{\mathbf{X}} + \Sigma_{\mathbf{E}}. \tag{5.60}$$

As the distortion vector $\mathbf{E}$ is also present on $\mathbf{X}_E$ in each state Q_j, (5.59) and (5.60) also hold on the state level

$$\mu_{\mathbf{X}_E|Q_j} = \mu_{\mathbf{X}|Q_j} + \mu_{\mathbf{E}},$$

$$\Sigma_{\mathbf{X}_E|Q_j} = \Sigma_{\mathbf{X}|Q_j} + \Sigma_{\mathbf{E}}.$$

We also assume that the density functions $p(\mathbf{e})$ and $p(\mathbf{x}|Q_j)$ are monomodal Gaussians leading to a multimodal Gaussian $p(\mathbf{x}_E)$

$$p(\mathbf{x}_E|Q_j) \sim \mathcal{N}(\mu_{\mathbf{X}_E|Q_j}, \Sigma_{\mathbf{X}_E|Q_j}),$$

$$p(\mathbf{x}_E) \sim \sum_{j=1}^{N_Q} p(Q_j) \cdot p(\mathbf{x}_E|Q_j). \tag{5.61}$$

Given (5.61) we can determine $I(\mathbf{X}_E, Q)$ for the monomodal and multimodal approximation. In the following, the monomodal approximation is discussed.

Following the monomodal derivation as in (5.46), we get

$$I(\mathbf{X}_E; Q) = \frac{1}{2} \sum_{j=1}^{N_Q} P(Q_j) \ \ln \ \frac{|\Sigma_{\mathbf{X}} + \Sigma_{\mathbf{E}}|}{|\Sigma_{\mathbf{X}|Q_j} + \Sigma_{\mathbf{E}}|}. \tag{5.62}$$

For diagonal covariance matrices we have

$$\begin{aligned} I(\mathbf{X}_E;Q) &= \frac{1}{2}\sum_{j=1}^{N_Q} P(Q_j)\ \ln\ \prod_{i=1}^{d}\frac{\sigma_{X_i}^2+\sigma_{E_i}^2}{\sigma_{X_i|Q_j}^2+\sigma_{E_i}^2} \\ &= \sum_{i=1}^{d} I(X_{E_i};Q), \end{aligned} \tag{5.63}$$

where

$$I(X_{E_i};Q) = \frac{1}{2}\sum_{j=1}^{N_Q} P(Q_j)\ \ln\ \left(\gamma_{i,j}^2 \cdot \frac{1+SNR_i}{\gamma_{i,j}^2+SNR_i}\right), \tag{5.64}$$

with

$$\gamma_{i,j}^2 = \frac{\sigma_{X_i}^2}{\sigma_{X_i|Q_j}^2}; \qquad SNR_i = \frac{\sigma_{X_i}^2}{\sigma_{E_i}^2}.$$

As shown in (5.64), the distortion influences the mutual information via the SNR_i.

Furthermore, we can define the loss of mutual information $I(R_i,Q)$ as

$$I(X_{E_i};Q) = I(X_i;Q) - I(R_i,Q), \tag{5.65}$$

where

$$\begin{aligned} I(X_i;Q) &= \frac{1}{2}\sum_{j=1}^{N_Q} P(Q_j)\ \ln\ \gamma_{i,j}^2, \\ I(R_i;Q) &= \frac{1}{2}\sum_{j=1}^{N_Q} P(Q_j)\ \ln\ \frac{\gamma_{i,j}^2+SNR_i}{1+SNR_i}. \end{aligned} \tag{5.66}$$

Since $\gamma_{i,j}^2 \geq 1$ we have

$$\frac{\gamma_{i,j}^2+SNR_i}{1+SNR_i} \geq 1,$$

and thus $I(R_i;Q) \geq 0$.

The influence of distortion E_i in the loss $I(R_i;Q)$ is determined by the values of SNR_i. For large values of SNR_i, i.e., for small distortion variances, the loss is small. If SNR_i approaches 0 as for high distortion case, $I(R_i;Q)$ approaches the value of $I(X_i;Q)$ and the mutual information $I(X_{E_i};Q)$ approaches 0. This results imply that noise reduction methods should maximize the values of SNR_i. Since the loss also depends on $\gamma_{i,j}^2$, maximizing SNR_i should also be done especially for the dimension i where $\gamma_{i,j}^2$ has large values.

Chapter 6

Front-End Optimization and Evaluation on the Aurora 3 German Digits Database

The first two experiments presented in this chapter are dealing with the baseline SFE and AFE using a common back-end, i.e., the SBE, as described in Section 3.2.2. The remainder of the experiments are presenting the experiments with our proposed least-squares weighting rules and some additional front-end processing. The performance analysis is done in terms of word accuracy. The performance in word recognition rate is also tabulated to specifically observe the effect of insertion errors during the recognition phase. A new training procedure is always conducted to obtain each result tabulated in all experiments.

6.1 Experiment I: AFE and SFE Experimental Setups on the SBE

6.1.1 ETSI Advanced Front-End

Aim: To create an experimental setup for the AFE which is running on the SBE and use the performance results as the reference to be achieved.

The only available recognition results of the AFE are those evaluated with the back-end built using the hidden Markov model toolkit (HTK) specified by the ETSI working group as described in [ETSI STQ-Aurora 2002]. The toolkit is available at no cost from the Cambridge University

Engineering Department (CUED) [Young et al. 2005]. The HTK recognizer framework is described in [Hirsch and Pearce 2000]. Some adjustments have to be done to be able to obtain the performance of the AFE on the SBE instead of the HTK. This is necessary since any further developments to improve the SFE are subject to evaluation using the SBE as the target back-end. In this thesis, we perform the following investigations and finally introduce necessary modifications to allow the evaluation using the SBE:

- The features produced by the AFE need to be scaled to a range of 8 bit representation since the SBE is only taking 8 bit range inputs. First of all, the features are normalized using the z-normalization method and the scaling factor $c = 35$ is applied to the normalized features:

$$x_i' = c \times \frac{x_i - \mu_{X_i}}{\sigma_{X_i}}, \tag{6.1}$$

  for $i = 1 \cdots d$ where x_i', x_i, μ_{X_i}, and σ_{X_i} denote the transformed feature, original feature, mean and standard deviation of the original feature, respectively, and d is the dimension of the feature vector $\mathbf{x}$.

- The use of LDA is investigated since it may increase the AFE performance. The LDA is currently running on the SFE and it is considerably bringing improvement.

- Frame-dropping in the AFE also needs further investigation since this operation may result in some performance degradation. This is due to the fact that the LDA reduces the information contained in the feature vectors by reducing the dimensionality of the feature vector from two consecutive frames. In this case, frame-dropping might have a contradictory effect on the performance.

Tables 6.1 and 6.2 show the performance in word accuracy and word recognition rate, respectively. It is obvious that the LDA needs to be done for the scaled and normalized features since the results showed that it significantly increases the performance of the front-end with and without frame-dropping. It is also shown that the frame-dropping method is not optimal in the presence of LDA. It decreases the effect of LDA processing as shown in the tables, where the LDA without frame-dropping is more than 18 % relatively better in word accuracy compared to the LDA with frame-dropping. The performance of the AFE used as the baseline is the one with the 91.9 % word accuracy. Given the excellent performance of the AFE, it is a challenging task to exceed its performance on the same framework.

Table 6.1: Word accuracy performance of the AFE.

	with frame-dropping		without frame-dropping	
	with LDA	without LDA	with LDA	without LDA
High Mismatch	88.3 %	78.0 %	92.0 %	78.1 %
Medium Mismatch	87.8 %	74.7 %	89.2 %	77.5 %
Well Matched	93.2 %	80.1 %	94.3 %	78.5 %
Weighted Average	90.1 %	77.7 %	91.9 %	78.1 %

Table 6.2: Word recognition rate performance of the AFE.

	with frame-dropping		without frame-dropping	
	with LDA	without LDA	with LDA	without LDA
High Mismatch	89.0 %	80.5 %	92.6 %	81.3 %
Medium Mismatch	88.5 %	78.8 %	90.0 %	80.8 %
Well Matched	94.3 %	83.6 %	95.6 %	83.3 %
Weighted Average	90.9 %	81.1 %	92.9 %	81.9 %

Result: The performance of the baseline AFE on the SBE is obtained with the following additional processing:

- Scaled and normalized features.
- LDA technique.
- No frame-dropping.

It achieves 91.9 % word accuracy.

6.1.2 Siemens Front-End

Aim: To adjust the basic SFE configuration following the AFE in order to achieve the baseline SFE performance.

One of the differences between the AFE and the current SFE configuration lies in the frame length/shift values. The AFE uses 25/10 *ms* configuration while the SFE uses 32/15 *ms*. The SFE configuration is modified following the AFE configuration and the scaling and normalization operation as in (6.1) is performed on the features produced after channel compensation. The LDA is already part of the SFE and the frame-dropping is switched-off as done for the AFE. The new configuration of the SFE is taken as the baseline and any further improvements on the SFE

Table 6.3: Word accuracy performance of the SFE with different frame length/shift.

	32/15 *ms*	25/10 *ms*
High Mismatch	86.1 %	85.8 %
Medium Mismatch	84.4 %	85.1 %
Well Matched	93.0 %	93.5 %
Weighted Average	88.3 %	88.6 %

Table 6.4: Word recognition rate performance of the SFE with different frame length/shift.

	32/15 *ms*	25/10 *ms*
High Mismatch	86.4 %	86.2 %
Medium Mismatch	84.8 %	85.5 %
Well Matched	94.1 %	95.0 %
Weighted Average	88.9 %	89.5 %

is now comparable to the AFE.

Word accuracy and word recognition rate performance of the SFE are shown in Tables 6.3 and 6.4, respectively. The new 25/10 *ms* frame length/shift configuration is better than the original 32/15 *ms* of the SFE. This is due to the fact that shorter frame shift implies that more features are generated for the same speech utterance. And this turns out to yield a significant improvement on the SFE. The new frame length/shift is adopted in the SFE. If we compare the baseline SFE using the new 25/10 *ms* frame length/shift with the AFE, it is obvious that the AFE is far better than the SFE. The room for improvement is still widely open given the fact that the AFE is currently more than 28 % relatively better in word accuracy than the SFE.

Result: The baseline SFE performance is obtained with the 25/10 *ms* frame length/shift and by employing the following additional processing:

- Scaled and normalized features.
- No frame-dropping.

It achieves 88.6 % word accuracy.

6.2 Experiment II: Investigations on the AFE components

Since the AFE has been developed by many experts in the related area, it is of our interest to initiate the direction of our research towards the components in the AFE which bring the biggest

improvement. In this section, major AFE components are separately tested on both the AFE and SFE baseline frameworks. The major AFE components are identified as

- Noise reduction, which is the two-stage mel-warped Wiener filtering method and the DC-offset removal. This could be further broken down into the first stage mel-warped Wiener filtering with the DC-offset removal (NR1) and using both stage of the Wiener filtering with the DC-offset removal (NR2)
- SNR-dependent waveform processing (SWP).

Experiments are conducted where the above components are tested on the blind equalization (BE) and maximum likelihood channel compensation techniques as both of them are acting to compensate the channel effect.

6.2.1 Effects of the AFE Components Combined with the Blind Equalization (BE) technique

Aim: To identify the main component in AFE which shows the biggest improvement in combination with the blind equalization technique.

Table 6.5: Word accuracy performance having the AFE components with the BE.

	NR1+BE	NR2+BE	NR2+SWP+BE (baseline AFE)
High Mismatch	88.1 %	89.6 %	92.0 %
Medium Mismatch	87.6 %	90.8 %	89.2 %
Well Matched	93.5 %	94.3 %	94.3 %
Weighted Average	90.1 %	91.9 %	91.9 %

Table 6.6: Word recognition rate performance having the AFE components with the BE.

	NR1+BE	NR2+BE	NR2+SWP+BE (baseline AFE)
High Mismatch	88.8 %	90.0 %	92.6 %
Medium Mismatch	88.5 %	91.5 %	90.0 %
Well Matched	94.8 %	95.7 %	95.6 %
Weighted Average	91.1 %	92.8 %	92.9 %

The blind equalization technique is already used in the AFE framework. Therefore, we conducted the experiments exactly in this framework. Tables 6.5 and 6.6 show the results for the

various system configurations in word accuracy and word recognition rate, respectively. First of all, it is obvious that a performance increase is observed in all cases with the additional second stage noise reduction technique. Note that NR1 denotes the first stage of the noise reduction and NR2 denotes both stages of the noise reduction. Secondly, the improvement gained from the SNR-dependent waveform processing over the two-stage noise reduction does not apply in the medium mismatch case showing an absolute performance drop of 1.6 % in word accuracy. The improvement is also not observed for the well matched case but it gains around 23 % relative increase in word accuracy for high mismatch case.

Result: In combination with the blind equalization technique we conclude:

- The second stage of the noise reduction part in the AFE consistently improves the performance of the first stage.
- The SNR-dependent waveform processing gives significant improvement over the noise reduction part only in the high mismatch case. It shows a contrary effect in the medium mismatch case.

6.2.2 Effects of the AFE Components Combined with the Maximum Likelihood Channel Compensation (MLCC) technique

Aim: To identify the main component in AFE which shows the biggest improvement in combination with the maximum likelihood channel compensation technique.

When using the maximum likelihood channel compensation technique, the blind equalization technique should be switched-off in the AFE framework. After introducing the modifications to the AFE framework we conducted similar experiments as done previously. Tables 6.7 and 6.8 show the results for the various system configurations in word accuracy and word recognition rate, respectively.

It again shows that the second stage noise reduction part is consistently improving the performance of the first stage. The SNR-dependent waveform processing is now showing a consistent performance increase in all cases with a significant performance gain still observed in high mismatch case over the two-stage noise reduction part, i.e., 11.5 % relative performance increase in word accuracy.

Table 6.7: Word accuracy performance having the AFE components with the MLCC.

	NR1+MLCC	NR2+MLCC	NR2+SWP+MLCC
High Mismatch	87.7 %	88.7 %	90.0 %
Medium Mismatch	86.8 %	89.1 %	89.3 %
Well Matched	93.7 %	94.1 %	94.1 %
Weighted Average	89.8 %	91.0 %	91.4 %

Table 6.8: Word recognition rate performance having the AFE components with the MLCC.

	NR1+MLCC	NR2+MLCC	NR2+SWP+MLCC
High Mismatch	88.1 %	88.9 %	90.3 %
Medium Mismatch	87.3 %	90.0 %	90.3 %
Well Matched	95.1 %	95.4 %	95.2 %
Weighted Average	90.6 %	91.9 %	92.3 %

Result:

- In combination with the maximum likelihood channel compensation technique we conclude:
  - The second stage of the noise reduction part in the AFE consistently improves the performance of the first stage.
  - The SNR-dependent waveform processing is significantly increasing the performance in the high mismatch case over the two-stage noise reduction part.
- In addition to that, blind equalization is better than maximum likelihood channel compensation. This could be explained by the fact that the maximum likelihood channel compensation was not optimized for the AFE.

6.3 Experiment III: Weighting Rule Evaluations

As shown in previous experiments, the noise reduction component in the AFE is the part which is consistently showing improvement. The SNR-dependent waveform processing is generally showing improvement except for the medium mismatch case in combination with the blind equalization technique. A good choice of a channel compensation technique is also necessary to further improve the front-end performance as shown in the experiments between MLCC and BE techniques.

However, in this section we focus on the noise reduction part by testing the proposed least-squares based weighting rule formulations. There are two different noise PSD estimation techniques used:

- the three-state voice activity driven noise PSD estimator described in Section 4.1.1, and
- the minimum statistics noise PSD estimator described in Section 4.1.2.

The *a priori* and *a posteriori* SNR based Wiener filtering as described in Section 4.2.2 are used for the purpose of performance comparison with the MMSE based weighting rules. MLCC is used as the channel compensation technique in order to minimize the modifications introduced in the SFE.

We briefly restated all the evaluated weighting rules:

- *a posteriori* and *a priori* SNR based Wiener filtering,
- *a posteriori recursive* least-squares (used in the baseline SFE), *a priori recursive* least-squares, spectral subtraction based *recursive* least-squares, and recursive gain least-squares as described in Section 4.2.4.

6.3.1 Using the Three-State Voice Activity Driven Noise PSD Estimator

Aim: To evaluate the proposed least-squares based weighting rules on the SFE using the three-state voice activity driven noise PSD estimator.

Tables 6.9 and 6.10 show the performance of the weighting rules using the three-state voice activity driven noise PSD estimator in word accuracy and word recognition rate, respectively. It can be seen that all weighting rules are equal to or better than the *a posteriori* recursive least-squares approach in word accuracy. The spectral subtraction based *recursive* least-squares weighting rule is better than the advanced *a priori* SNR based Wiener filtering. It achieves 5.2 % relative improvement in word accuracy. Both weighting rules are basically showing equal performance in word recognition rate, but the advanced *a priori* SNR based Wiener filtering yields more insertion errors.

Table 6.9: Word accuracy performance of the SFE having the weighting rules and the three-state voice activity driven noise PSD estimator.

	Wiener		Least-Squares			RGLS
	post.	prior.	post.	spect-sub	prior.	
High Mismatch	85.8 %	88.7 %	85.8 %	88.8 %	86.7 %	88.1 %
Medium Mismatch	85.0 %	87.3 %	85.1 %	88.3 %	84.8 %	87.0 %
Well Matched	93.6 %	94.0 %	93.5 %	94.2 %	93.8 %	93.9 %
Weighted Average	88.6 %	90.3 %	88.6 %	90.8 %	88.9 %	90.0 %

Table 6.10: Word recognition rate performance of the SFE having the weighting rules and the three-state voice activity driven noise PSD estimator.

	Wiener		Least-Squares			RGLS
	post.	prior.	post.	spect-sub	prior.	
High Mismatch	86.4 %	89.9 %	86.2 %	89.5 %	87.0 %	89.0 %
Medium Mismatch	85.6 %	88.7 %	85.5 %	89.0 %	85.1 %	87.7 %
Well Matched	95.0 %	96.0 %	95.0 %	96.0 %	95.0 %	95.7 %
Weighted Average	89.6 %	91.9 %	89.5 %	91.9 %	89.5 %	91.2 %

Result:

- All weighting rules achieve better or equal performance than the baseline weighting rule in the SFE.
- The spectral subtraction based *recursive* least-squares weighting rule is superior than the advanced *a priori* SNR based Wiener filtering.

6.3.2 Using the Minimum Statistics Noise PSD Estimator

Aim: To evaluate the proposed least-squares based weighting rules on the SFE using the minimum statistics noise PSD estimator.

Tables 6.11 and 6.12 show the performance of the weighting rules using the minimum statistics noise PSD estimator in word accuracy and word recognition rate, respectively. It is shown that all weighting rules are better than the *a posteriori* recursive least-squares and the spectral subtraction based *recursive* least-squares weighting rule is again better than the advanced *a priori* SNR based Wiener filtering. It achieves 3.3 % relative improvement in word accuracy.

Table 6.11: Word accuracy performance of the SFE having the weighting rules and the minimum statistics noise PSD estimator.

	Wiener		Least-Squares			RGLS
	post.	prior.	post.	spect-sub	prior.	
High Mismatch	88.7 %	89.1 %	86.5 %	89.5 %	86.5 %	89.2 %
Medium Mismatch	87.7 %	88.4 %	84.4 %	88.8 %	84.5 %	87.7 %
Well Matched	94.2 %	94.4 %	93.7 %	94.6 %	93.9 %	94.5 %
Weighted Average	90.6 %	91.0 %	88.6 %	91.3 %	88.8 %	90.8 %

Table 6.12: Word recognition rate performance of the SFE having the weighting rules and the minimum statistics noise PSD estimator.

	Wiener		Least-Squares			RGLS
	post.	prior.	post.	spect-sub	prior.	
High Mismatch	89.5 %	90.2 %	86.9 %	90.5 %	86.8 %	90.7 %
Medium Mismatch	88.4 %	89.2 %	85.2 %	89.7 %	84.9 %	88.8 %
Well Matched	95.8 %	96.1 %	94.9 %	96.1 %	95.1 %	96.2 %
Weighted Average	91.6 %	92.2 %	89.5 %	92.5 %	89.5 %	92.2 %

Comparing with the previous experiment using the three-state voice activity driven, minimum statistics brings considerable improvements to the Wiener filtering weighting rules. It shows 17.5 % and 7.2 % relative improvements in word accuracy for the *a posteriori* and *a priori* SNR based Wiener filtering, respectively. This is attributed to the smoother noise PSD estimates produced by the minimum statistics. This effect is not shown in all least-squares based weighting rules due to the smoothing effect already introduced in the weighting rule formulations. Improvements are only observed for the spectral subtraction based *recursive* least-squares and recursive gain least-squares having 5.4 % and 8.0% relative improvements in word accuracy, respectively.

Result:

- All weighting rules outperform the weighting rule in the baseline SFE.
- Using the minimum statistics noise PSD estimator, the spectral subtraction based *recursive* least-squares achieves better results than the advanced *a priori* SNR based Wiener filtering.
- The Wiener filtering weighting rules take the advantage of having smoother noise PSD estimates from the minimum statistics.

6.4 Experiment IV: Root-Cepstral Coefficients

The root-cepstral coefficients technique is used instead of the commonly used mel filter cepstral coefficients. A direct root function as shown in (3.12) has been used instead of the logarithmic function. Experiments have been conducted to find the best value for the root. For both types of noise PSD estimators, the root value of $\gamma = 0.1$ was used.

6.4.1 Using the Three-State Voice Activity Driven Noise PSD Estimator

Aim: To evaluate the root-cepstral coefficients in the SFE using the three-state voice activity driven noise PSD estimator.

Table 6.13: Word accuracy performance of the SFE having the weighting rules, three-state voice activity driven, and $\gamma = 0.1$ for the root value.

	Wiener		Least-Squares			RGLS
	post.	prior.	post.	spect-sub	prior.	
High Mismatch	85.5 %	90.4 %	86.1 %	90.5 %	86.0 %	89.5 %
Medium Mismatch	85.2 %	88.1 %	85.4 %	89.2 %	85.3 %	88.9 %
Well Matched	94.2 %	95.0 %	94.3 %	94.8 %	93.5 %	94.8 %
Weighted Average	88.9 %	91.4 %	89.1 %	91.8 %	88.8 %	91.4 %

Table 6.14: Word recognition rate performance of the SFE having the weighting rules, three-state voice activity driven, and $\gamma = 0.1$ for the root value.

	Wiener		Least-Squares			RGLS
	post.	prior.	post.	spect-sub	prior.	
High Mismatch	85.8 %	91.0 %	86.3 %	91.1 %	86.4 %	90.4 %
Medium Mismatch	85.5 %	89.2 %	85.7 %	89.7 %	85.6 %	89.5 %
Well Matched	95.2 %	96.1 %	95.2 %	96.1 %	94.4 %	96.1 %
Weighted Average	89.5 %	92.4 %	89.7 %	92.6 %	89.3 %	92.4 %

Tables 6.13 and 6.14 show the results of the root-cepstral coefficients using the three-state voice activity driven noise PSD estimator in word accuracy and word recognition rate, respectively. In comparison to Tables 6.9 it can be shown that the use of an optimal root value increases the performance of most types of weighting rules. The best improvements are achieved by the spectral subtraction based *recursive* least-squares, the recursive gain least squares, and the *a priori* SNR based Wiener filtering with the relative improvements in word accuracy of 10.9 %, 14.0

%, and 11.3 %, respectively. The spectral subtraction based *recursive* least-squares weighting rule is better than the advanced *a priori* SNR based Wiener filtering with 4.7 % relative improvement in word accuracy. The RGLS is now showing comparable performance with the advanced *a priori* SNR based Wiener filtering. The performance of the spectral subtraction based *recursive* least-squares is comparable to the AFE.

Result:

- The root-cepstral coefficients with the three-state voice activity driven noise PSD estimator increases the performance of the front-end for most types of weighting rules.
- The spectral subtraction based *recursive* least-squares weighting rule and the AFE is comparable.

6.4.2 Using the Minimum Statistics Noise PSD Estimator

Aim: To evaluate the root-cepstral coefficients in the SFE using the minimum statistics noise PSD estimator.

Tables 6.15 and 6.16 show the results of the root-cepstral coefficients using the minimum statistics noise PSD estimator in word accuracy and word recognition rate, respectively. It is shown that the use of an optimal root value with the minimum statistics increases the performance of all types of weighting rules. The relative increase of performance in word accuracy compared to the one without the root-cepstral coefficients as shown in Table 6.11 is given as 8.5 %, 8.9 %, 6.1 %, 5.7 %, 3.6 %, and 10.9 % for the *a posteriori* SNR based Wiener filtering, *a priori* SNR based Wiener filtering, *A posteriori recursive* least-squares, spectral subtraction based *recursive* least-squares, *a priori recursive* least-squares, and recursive gain least squares, respectively.

The performance of the *a priori* SNR based Wiener filtering, the spectral subtraction based *recursive* least-squares, and the recursive gain least-squares are comparable to the AFE. Although the three weighting rules are having equal performance in word accuracy, in terms of word recognition rate the *a priori* SNR based Wiener filtering is still below both least-squares weighting rules. Compared to the results using the three-state voice activity driven, the top performance is comparable in word accuracy but the values in word recognition rate is overall better with minimum statistics.

Table 6.15: Word accuracy performance of the SFE having the weighting rules, minimum statistics, and $\gamma = 0.1$ for the root value.

	Wiener		Least-Squares			RGLS
	post.	prior.	post.	spect-sub	prior.	
High Mismatch	89.8 %	90.4 %	87.3 %	90.6 %	87.2 %	90.7 %
Medium Mismatch	88.4 %	89.0 %	85.4 %	88.5 %	85.1 %	88.7 %
Well Matched	95.0 %	95.0 %	93.9 %	95.5 %	93.9 %	95.1 %
Weighted Average	91.4 %	91.8 %	89.3 %	91.8 %	89.2 %	91.8 %

Table 6.16: Word recognition rate performance of the SFE having the weighting rules, minimum statistics, and $\gamma = 0.1$ for the root value.

	Wiener		Least-Squares			RGLS
	post.	prior.	post.	spect-sub	prior.	
High Mismatch	90.2 %	91.2 %	87.4 %	91.4 %	87.5 %	91.8 %
Medium Mismatch	88.9 %	89.6 %	85.9 %	89.4 %	85.5 %	89.5 %
Well Matched	95.9 %	96.1 %	94.8 %	96.6 %	94.9 %	96.3 %
Weighted Average	92.0 %	92.6 %	89.8 %	92.8 %	89.8 %	92.8 %

Result:

- Root-cepstral coefficients technique with the minimum statistics noise PSD estimator increases the performance of the front-end for all types of weighting rules.
- The performance of the *a priori* SNR based Wiener filtering, the spectral subtraction based *recursive* least-squares, and the recursive gain least-squares are comparable to the AFE.

6.5 Experiment V: Cepstral Smoothing

Cepstral smoothing as described in Section 3.3.2 is applied in combination with the proposed weighting rules and the root-cepstral coefficients. The performance effect of this smoothing technique is expected to be *additive*. Experiments were conducted with the purpose of finding the optimal smoothing coefficient L. The choice of L is limited from 1 to 6 to avoid having an oversmoothed cepstral coefficients. The parameters of the weighting rules and the root value are not modified.

6.5.1 Using the Three-State Voice Activity Driven Noise PSD Estimator

Aim: Finding the optimal smoothing coefficient L for the cepstral smoothing technique in the SFE using the three-state voice activity driven noise PSD estimator.

Table 6.17: Word accuracy performance of the SFE having the weighting rules, three-state voice activity driven, $\gamma = 0.1$, and several values of L.

		$L \rightarrow$	1	2	3	4	5	6
Wiener	post.	High Mismatch	87.0 %	87.2 %	87.7 %	89.1 %	88.8 %	88.8 %
		Medium Mismatch	85.8 %	86.2 %	86.2 %	86.7 %	86.3 %	86.2 %
		Well Matched	94.3 %	94.0 %	94.1 %	94.5 %	94.6 %	94.4 %
		Weighted Average	**89.5 %**	**89.6 %**	**89.7 %**	**90.4 %**	**90.2 %**	**90.1 %**
	prior.	High Mismatch	90.2 %	90.0 %	90.5 %	90.7 %	90.5 %	90.5 %
		Medium Mismatch	88.4 %	88.4 %	88.4 %	88.3 %	88.7 %	89.2 %
		Well Matched	94.6 %	94.9 %	94.6 %	94.9 %	94.6 %	94.9 %
		Weighted Average	**91.3 %**	**91.4 %**	**91.4 %**	**91.5 %**	**91.5 %**	**91.8 %**
Least-Squares	post.	High Mismatch	86.6 %	87.1 %	87.6 %	87.1 %	87.5 %	87 %
		Medium Mismatch	85.3 %	86.5 %	85.4 %	85.4 %	86.3 %	84.7 %
		Well Matched	94.1 %	94.4 %	94.1 %	94 %	94.2 %	93.7 %
		Weighted Average	**89.1 %**	**89.8 %**	**89.4 %**	**89.3 %**	**89.8 %**	**88.9 %**
	spect-sub.	High Mismatch	90.8 %	90.5 %	91.3 %	90.8 %	91.5 %	91.5 %
		Medium Mismatch	89.2 %	89.5 %	89.7 %	89.8 %	89.5 %	89.8 %
		Well Matched	95 %	95.2 %	95.3 %	95.2 %	95 %	94.9 %
		Weighted Average	**91.9 %**	**92.0 %**	**92.3 %**	**92.2 %**	**92.2 %**	**92.3 %**
	prior.	High Mismatch	86.4 %	87.3 %	86.8 %	88 %	87.7 %	87.1 %
		Medium Mismatch	85.7 %	85.3 %	85.6 %	85.8 %	85.1 %	84.6 %
		Well Matched	93.9 %	94.1 %	93.8 %	94 %	93.8 %	93.8 %
		Weighted Average	**89.2 %**	**89.3 %**	**89.2 %**	**89.6 %**	**89.2 %**	**88.9 %**
RGLS		High Mismatch	90.1 %	91.0 %	90.4 %	91.2 %	91 %	91.4 %
		Medium Mismatch	89.2 %	89.2 %	89.6 %	89.9 %	89.9 %	89 %
		Well Matched	94.8 %	95.1 %	95.0 %	95.0 %	95.2 %	95.0 %
		Weighted Average	**91.7 %**	**92.0 %**	**92.0 %**	**92.3 %**	**92.3 %**	**92.0 %**

Tables 6.17 and 6.18 tabulate the results in word accuracy and word recognition rate, respectively. It is shown that the optimal smoothing coefficients varies between 3 and 6. Taking the best performance achieved with a certain smoothing coefficient L in the cepstral smoothing technique, an increase in performance is observed for all weighting rules as compared to Table 6.13. They are given in word accuracy as 13.5 %, 4.7 %, 6.4 %, 6.1 %, 7.1 %, and 10.5 % for the *a posteriori* SNR based Wiener filtering, *a priori* SNR based Wiener filtering, *a posteriori recursive* least-squares, spectral subtraction based *recursive* least-squares, *a priori recursive*

Table 6.18: Word recognition rate performance of the SFE having the weighting rules, three-state voice activity driven, $\gamma = 0.1$, and several values of L.

		$L \rightarrow$	1	2	3	4	5	6
Wiener	post.	High Mismatch	87.5 %	87.7 %	88.1 %	89.4 %	89.4 %	89.5 %
		Medium Mismatch	86.2 %	86.7 %	87.2 %	88 %	87.8 %	87.8 %
		Well Matched	95.4 %	95.2 %	95.2 %	95.6 %	95.6 %	95.5 %
		Weighted Average	**90.2 %**	**90.4 %**	**90.6 %**	**91.4 %**	**91.3 %**	**91.3 %**
	prior.	High Mismatch	90.8 %	90.7 %	91.4 %	91.2 %	91.3 %	91.3 %
		Medium Mismatch	89.5 %	89.5 %	89.2 %	89.2 %	89.7 %	90.3 %
		Well Matched	96 %	96.1 %	95.9 %	96.1 %	95.7 %	96 %
		Weighted Average	**92.4 %**	**92.4 %**	**92.4 %**	**92.5 %**	**92.5 %**	**92.8 %**
Least-Squares	post.	High Mismatch	86.8 %	87.3 %	87.7 %	87.2 %	87.7 %	87.2 %
		Medium Mismatch	85.6 %	86.7 %	85.8 %	85.8 %	86.6 %	85.2 %
		Well Matched	94.9 %	95.3 %	94.9 %	95 %	95 %	94.6 %
		Weighted Average	**89.6 %**	**90.3 %**	**89.9 %**	**89.8 %**	**90.2 %**	**89.5 %**
	spect-sub.	High Mismatch	91.5 %	91.3 %	92 %	91.4 %	92 %	92.3 %
		Medium Mismatch	89.9 %	90.3 %	90.4 %	90.4 %	90.2 %	90.6 %
		Well Matched	96.3 %	96.4 %	96.5 %	96.4 %	96 %	96 %
		Weighted Average	**92.9 %**	**93.0 %**	**93.2 %**	**93.1 %**	**93.0 %**	**93.2 %**
	prior.	High Mismatch	86.6 %	87.4 %	87 %	88.2 %	88 %	87.4 %
		Medium Mismatch	86.2 %	85.7 %	86 %	86.1 %	85.5 %	85.3 %
		Well Matched	94.8 %	94.8 %	94.6 %	94.7 %	94.6 %	94.6 %
		Weighted Average	**89.7 %**	**89.8 %**	**89.7 %**	**90.1 %**	**89.8 %**	**89.5 %**
RGLS		High Mismatch	90.9 %	91.8 %	91.3 %	92.3 %	92.2 %	92.2 %
		Medium Mismatch	90.1 %	90.2 %	90.3 %	90.8 %	90.9 %	90.2 %
		Well Matched	96.1 %	96.3 %	96.3 %	96.3 %	96.4 %	96.2 %
		Weighted Average	**92.7 %**	**93.0 %**	**93.0 %**	**93.4 %**	**93.4 %**	**93.1 %**

least-squares, and recursive gain least squares, respectively. Both the spectral subtraction based *recursive* least-squares with $L = 3$ and recursive gain least squares with $L = 4$ achieve 6.1 % relative improvement in word accuracy compared to the *a priori* SNR based Wiener filtering. Both least-squares weighting rules are now outperforming the AFE while the *a priori* SNR based Wiener filtering is still comparable to the AFE.

Result:

- Cepstral smoothing with the three-state voice activity driven improves the front-end performance in all of the cases.
- The spectral subtraction based *recursive* least-squares and recursive gain least squares outperform the AFE.

6.5.2 Using Minimum Statistics Noise PSD Estimator

Aim: Finding the optimal smoothing coefficient L for the cepstral smoothing technique in the SFE using the minimum statistics noise PSD estimator.

Table 6.19: Word accuracy performance of the SFE having the weighting rules, minimum statistics, $\gamma = 0.1$, and several values of L.

		$L \rightarrow$	1	2	3	4	5	6
Wiener	post.	High Mismatch	90.2 %	90.1 %	91 %	90.6 %	90.8 %	91 %
		Medium Mismatch	88.7 %	88.9 %	89.8 %	89.6 %	89.3 %	88.7 %
		Well Matched	95.2 %	95 %	95.1 %	95.2 %	95 %	94.7 %
		Weighted Average	**91.7 %**	**91.6 %**	**92.2 %**	**92.1 %**	**92.0 %**	**91.7 %**
	prior.	High Mismatch	90.4 %	90.7 %	90.5 %	90.6 %	90.6 %	90.5 %
		Medium Mismatch	88.9 %	89.7 %	89.5 %	90.3 %	90 %	89.7 %
		Well Matched	95.2 %	95.4 %	95.2 %	95.2 %	95.1 %	95 %
		Weighted Average	**91.8 %**	**92.2 %**	**92.0 %**	**92.3 %**	**92.2 %**	**92.0 %**
Least-Squares	post.	High Mismatch	88 %	87.9 %	88.2 %	88.1 %	88.3 %	87.8 %
		Medium Mismatch	86.2 %	85.7 %	86.2 %	85.9 %	85.5 %	85.4 %
		Well Matched	94.3 %	94.1 %	94.3 %	94.5 %	94.1 %	94.3 %
		Weighted Average	**89.9 %**	**89.6 %**	**89.9 %**	**89.9 %**	**89.6 %**	**89.6 %**
	spect-sub.	High Mismatch	91 %	91.1 %	92 %	91.6 %	91.1 %	91.2 %
		Medium Mismatch	89.4 %	89.2 %	89.8 %	89.8 %	88.7 %	89.1 %
		Well Matched	95.3 %	94.9 %	95.4 %	95.4 %	94.9 %	95.4 %
		Weighted Average	**92.2 %**	**92.0 %**	**92.6 %**	**92.5 %**	**91.8 %**	**92.1 %**
	prior.	High Mismatch	88.2 %	87.7 %	87.6 %	87.8 %	88.1 %	87.7 %
		Medium Mismatch	85.7 %	85.3 %	85.6 %	84.2 %	85.8 %	85.1 %
		Well Matched	93.7 %	94.2 %	94.4 %	94.4 %	93.6 %	93.8 %
		Weighted Average	**89.5 %**	**89.5 %**	**89.6 %**	**89.2 %**	**89.5 %**	**89.2 %**
RGLS		High Mismatch	91 %	91.4 %	91.6 %	91.5 %	91.3 %	91.1 %
		Medium Mismatch	89.2 %	89.7 %	89.9 %	89.8 %	89.4 %	89.3 %
		Well Matched	95.3 %	95 %	95.2 %	95 %	95.1 %	94.9 %
		Weighted Average	**92.1 %**	**92.2 %**	**92.4 %**	**92.3 %**	**92.2 %**	**92.0 %**

Tables 6.19 and 6.20 tabulate the results in word accuracy and word recognition rate, respectively. It is shown that the optimal smoothing coefficients are mostly $L = 3$ except the *a priori* SNR based Wiener filtering where $L = 4$. Taking the best performance in word accuracy achieved with the cepstral smoothing technique, an increase in performance is observed for all weighting rules as compared to Table 6.15. They are given as 9.3 %, 6.1 %, 5.6 %, 9.8 %, 3.7 %, and 7.3 % for the *a posteriori* SNR based Wiener filtering, *a priori* SNR based Wiener filtering, *a posteriori recursive* least-squares, spectral subtraction based *recursive* least-squares, *a priori*

Table 6.20: Word recognition rate performance of the SFE having the weighting rules, minimum statistics, $\gamma = 0.1$, and several values of L.

		$L \rightarrow$	1	2	3	4	5	6
Wiener	post.	High Mismatch	90.5 %	90.6 %	91.6 %	91.5 %	92 %	91.8 %
		Medium Mismatch	89.6 %	89.6 %	90.5 %	90.4 %	90.2 %	89.8 %
		Well Matched	96 %	96.1 %	96.2 %	96.3 %	96 %	95.8 %
		Weighted Average	**92.4 %**	**92.5 %**	**93.1 %**	**93.0 %**	**93.0 %**	**92.7 %**
	prior.	High Mismatch	90.9 %	91.3 %	91 %	91.4 %	91.4 %	91.3 %
		Medium Mismatch	89.7 %	90.2 %	90.1 %	90.7 %	90.7 %	90.4 %
		Well Matched	96.3 %	96.4 %	96.2 %	96.3 %	96.1 %	96 %
		Weighted Average	**92.6 %**	**93.0 %**	**92.8 %**	**93.1 %**	**93.0 %**	**92.9 %**
Least-Squares	post.	High Mismatch	88.1 %	88.2 %	88.3 %	88.3 %	88.4 %	88.1 %
		Medium Mismatch	86.5 %	85.9 %	86.7 %	86.4 %	85.9 %	85.7 %
		Well Matched	95.2 %	95 %	95.2 %	95.3 %	94.9 %	95 %
		Weighted Average	**90.4 %**	**90.1 %**	**90.5 %**	**90.4 %**	**90.1 %**	**90.0 %**
	spect-sub.	High Mismatch	91.9 %	92.2 %	92.9 %	92.4 %	92.1 %	92.1 %
		Medium Mismatch	90.6 %	90.4 %	91.2 %	91.2 %	90.5 %	90.2 %
		Well Matched	96.3 %	96 %	96.4 %	96.3 %	96 %	96.5 %
		Weighted Average	**93.2 %**	**93.1 %**	**93.7 %**	**93.5 %**	**93.1 %**	**93.2 %**
	prior.	High Mismatch	88.3 %	87.8 %	87.8 %	88 %	88.2 %	87.8 %
		Medium Mismatch	86 %	85.7 %	86.1 %	84.7 %	86.1 %	85.4 %
		Well Matched	94.6 %	95.1 %	95 %	95.1 %	94.4 %	94.6 %
		Weighted Average	**90.0 %**	**90.0 %**	**90.1 %**	**89.7 %**	**89.9 %**	**89.7 %**
RGLS		High Mismatch	91.9 %	92.8 %	92.7 %	92.7 %	92.5 %	92.1 %
		Medium Mismatch	90.1 %	90.7 %	91.5 %	91.2 %	90.6 %	91.1 %
		Well Matched	96.6 %	96.2 %	96.5 %	96.5 %	96.2 %	96.1 %
		Weighted Average	**93.2 %**	**93.4 %**	**93.8 %**	**93.7 %**	**93.3 %**	**93.4 %**

recursive least-squares, and recursive gain least squares, respectively. The spectral subtraction based *recursive* least-squares is superior to the *a priori* SNR based Wiener filtering showing 3.9 % relative improvement in word accuracy. Three weighting rules are now outperforming the AFE, i.e., the *a priori* SNR based Wiener filtering, spectral subtraction based *recursive* least-squares, and recursive gain least squares.

The use of minimum statistics instead of the three-state voice activity driven noise PSD estimator drastically improves the performance of the *a posteriori* SNR based Wiener filtering by showing 18.8 % relative increase in performance in word accuracy. Minimum statistics are basically improving the performance of the front-end using most of the weighting rules where only the *a priori recursive* least-squares is staying with the same performance in both word accuracy and word recognition rate.

Result:

- Cepstral smoothing with the minimum statistics improves the front-end performance in all of the cases.
- The *a priori* SNR based Wiener filtering, spectral subtraction based *recursive* least-squares, and recursive gain least squares outperform the AFE but the best performance is still shown by the spectral subtraction based *recursive* least-squares.
- The *a posteriori* SNR based Wiener filtering highly benefits from the minimum statistics, root-cepstral coefficients, and cepstral smoothing.

Chapter 7

Noise Reduction Evaluation on the SPEECON and SpeechDat-Car Spanish

Experiments conducted on a small vocabulary database such as the Aurora 3 German digits as presented in Chapter 6 have shown that the proposed least-squares weighting rules, i.e., the recursive gain least squares and spectral subtraction based *recursive* least-squares, are good candidates to replace the baseline weighting rule, i.e., the *a posteriori recursive* least-squares. The performance of spectral subtraction based *recursive* least-squares in particular, is really encouraging since it consistently outperforms the advanced *a priori* SNR based Wiener filtering.

In this chapter, we are going to evaluate the performance of both proposed least-squares weighting rules on a large vocabulary database, i.e., the SpeechDat-Car and SPEECON databases. Both databases are described in Sections 3.5.2 and 3.5.3, respectively. It is shown in the description that the tasks have been extended to include commands, city names, and application specific words in Spanish instead of only digits as in the Aurora 3 German. In addition to that, a new sampling rate of 11.025 kHz is used. These new tasks constitute a more challenging test framework for the proposed weighting rules. As for the noise PSD estimation, the three-state voice activity driven noise PSD estimator is used due to its lower memory footprint and complexity compared to the minimum statistics one.

The baseline *a posteriori recursive* least-squares and *a priori* SNR based Wiener filtering are used to benchmark the performance of the proposed least-squares weighting rules. The advanced front-end (AFE) is not included in the benchmarking since we are focusing on the weighting rule improvement and also the fact that the AFE is not equipped with a real modification to process the 11.025 kHz speech utterances. The 11.025 kHz processing of the AFE is done by downsampling the utterances to 8 kHz prior to feature extraction.

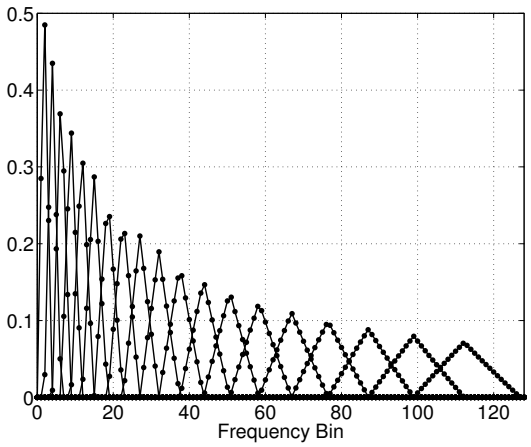


Figure 7.1: The 19 triangular-shaped mel filterbank as used in the 11.025 kHz Siemens front-end (SFE).

7.1 System Optimization for the 11.025 kHz Database

Instead of performing an isolated word recognition like the one for the Aurora 3 task, a continuous speech recognition is applied to cope with the given task. The training setup was based on an HMM trained on a Spanish database having 20000 Gaussian densities. A training list having 20000 entries was used to update the original HMM. The adjustments affect not only the back-end, but also the front-end which are tabulated in Table 7.1.

Table 7.1: Front-end adjustment for the 11.025 kHz task.

	8 kHz	11.025 kHz
Frame length/shift	32/15 ms	23.22/14.966 ms
Mel filter	$N_{FB} = 15$, $\Delta_{\text{mel}} = 133.66$ mel (see Fig. 3.3)	$N_{FB} = 19$, $\Delta_{\text{mel}} = 122.63$ mel (see Fig. 7.1)
Processed freq. range	180 - 4000 Hz	173 - 4996 Hz

As shown in Figure 7.1, more triangular-shaped mel filterbanks are produced to account for the increased information due to a higher sampling frequency.

7.2 Performance Evaluation

Five different test sets were defined. The recognition results in word accuracy are tabulated for each test set and the improvement is measured based on the relative increase in the *average* word accuracy. This implies that an average over all test sets in word accuracy was first obtained and a relative increase in the mean word accuracy was finally calculated. Equal weights are assigned for each test set. Details on the individual improvement are depicted in Table 7.2 in word accuracy and also shown in [Höge et al. 2008].

Table 7.2: Word accuracy results with noise reduction methods.

Algorithm →			*a posteriori* RLS	*a priori* SNR Wiener	recursive gain least-squares	spectral subtraction RLS
Database ↓			Word Accuracy			
SpeechDat Car ES	commands	Channel 2	93.0 %	92.8 %	92.0 %	93.0 %
		Channel 3	92.0 %	92.5 %	92.4 %	92.3 %
		Channel 4	88.9 %	88.6 %	89.9 %	90.3 %
	city names	Channel 2	87.8 %	88.0 %	90.6 %	90.0 %
		Channel 3	88.4 %	89.2 %	90.7 %	91.1 %
		Channel 4	87.6 %	88.8 %	89.2 %	89.0 %
Speecon Car ES	application specific words	Channel 2	85.4 %	88.0 %	88.2 %	88.8 %
		Channel 3	87.2 %	90.2 %	89.6 %	90.0 %
	city names	Channel 2	79.8 %	82.7 %	82.3 %	82.7 %
		Channel 3	81.8 %	85.6 %	85.8 %	85.8 %
Speecon Adult ES	application specific words	Channel 2	92.2 %	95.0 %	94.2 %	94.3 %
		Channel 3	89.5 %	92.3 %	92.6 %	92.6 %
Mean Word Accuracy			87.8 %	89.5 %	89.8 %	90.0 %
Relative Word Accuracy Improvement			0.0 %	13.9 %	16.4 %	18.0 %

The word recognition rate performance is not tabulated due to the given task requirement where the recognizer is expected to deliver a hypothesized word for each given utterance. This implies that only the substitution errors are present and counted. The results are shown in the table clearly show that the proposed noise reductions, i.e., the recursive gain least-squares and spectral subtraction based *recursive* least-squares, outperform the baseline noise reduction used in the baseline SFE, i.e., the *a posteriori* recursive least-squares, showing 16.4 % and 18 % relative increase in word accuracy. They also outperform the advanced *a priori* SNR Wiener filtering where it can only manage to gain 13.9 % relative improvement in word accuracy. This achievement confirms the robustness of our proposed least-squares based weighting rules.

Chapter 8

Evaluation of the Entropy Concept on the Aurora 3 German Digits Database

In this Chapter we will evaluate the concept of entropy as described previously in Chapter 5. We are using the well matched German digits data set in the Aurora 3 Task as defined by the ETSI STQ Aurora standardization body [ETSI STQ-Aurora 2001a] and briefly presented in Section 3.5.1. As states Q we regard speech states or segments of the digits as provided by the whole word HMM modeling. The number of states N_Q is 270 and the amount of feature vectors in the training set is $N = 407978$ feature vectors. Each feature vector is aligned to a particular state by means of the forced Viterbi algorithm which gives us a set of observation pairs $\{(\mathbf{x}, Q_j)\}$.

The histogram of the state size or the number of feature vectors in a state is shown in Figure 8.1. It shows that at least around 500 feature vectors describe a state Q_j and most of the states are having 1200 - 2000 feature vectors.

8.1 Determination of $H(Q)$

First of all, we need to determine the entropy of $H(Q)$ since it is the upper bound of the mutual information $I(\mathbf{X};Q)$ as shown in (5.6). The distribution $P(Q)$ is given by estimating the probabilities $P(Q_j)$ as described in Section 5.2. Based on these probabilities the entropy $H(Q)$ is given by the expression

$$H(Q) = -\sum_{j=1}^{N_Q} P(Q_j) \text{ ld } P(Q_j). \tag{8.1}$$

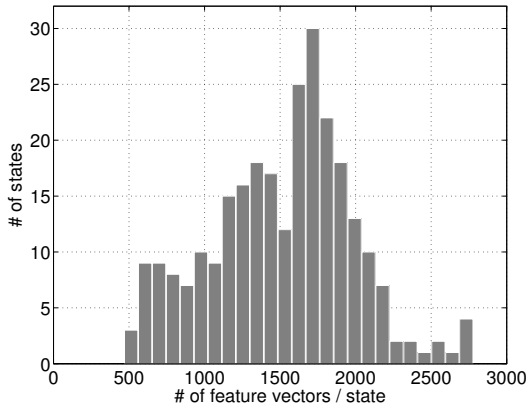


Figure 8.1: The histogram of state size (number of feature vectors in a state) from the training set.

In the case that all states are equally probable, $p(Q_j) = 1/N_Q$, the entropy $H(Q)$ is given by

$$H(Q) = \text{ld } N_Q. \tag{8.2}$$

Regarding the case of $N_Q = 270$ and assuming equal probability for each state Q_j, the entropy takes the value $H(Q) = \text{ld } 270 = 8.0768$ bits. This means that we must gain from each feature vector $\mathbf{x}$ an average of 8.0768 bits in order to reconstruct the sequences ψ^M of states without errors. However, this high amount of information is needed only for each pair $\left(\mathbf{x}, Q_j\right)$ if we disregard the context in time as described in Appendix D. This context embraces the fact that several feature vectors belong to one state and that the states are statistically dependent due to the morpho-syntactic and semantic constraints given by a language.

Based on the observed feature vectors, the value of $H(Q)$ equals 8.0025 bits. This result shows that the distribution of the states Q_j is nearly uniformly distributed. This experimental value of $H(Q)$ must be lower than 8.0768 bits due to the fact that the entropy is maximized under a uniform distribution.

8.2 Monogram Approximation

In this section we investigate the case of feature vectors having the dimension d and an arbitrary number N_Q of states. As previously stated, the monomodal Gaussian assumption is used for $p(\mathbf{x}|Q_j)$ and with diagonal covariance matrix according to (5.28) leading to

$$H(X_i|Q) = \frac{1}{2} \ln (2\pi e) + \sum_{j=1}^{N_Q} P(Q_j) \ln \sigma_{X_i|Q_j}. \tag{8.3}$$

Following (5.31) the monogram approximation of $H(\mathbf{X})$ is given by

$$I_1(\mathbf{X};Q) = \sum_{i=1}^{d} I_1(X_i;Q),$$

where

$$\begin{aligned} I_1(X_i;Q) &= H(X_i) - \frac{1}{2} \ln (2\pi e) - \sum_{j=1}^{N_Q} P(Q_j) \ln \sigma_{X_i|Q_j}, \\ H(X_i) &= -\int_{\mathbb{X}_i} p(x_i) \operatorname{ld} p(x_i)\, dx_i. \end{aligned} \tag{8.4}$$

In the following, we describe both approximations to $p(x_i)$, i.e., the monomodal Gaussian and the Gaussian mixture.

8.2.1 Monogram Approximation - Monomodal Gaussian

First we regard the monomodal approximation for $p(x_i)$ where the mutual information is formulated in (5.49) as

$$I_G(X_i;Q) = \sum_{j=1}^{N_Q} P(Q_j) \ln \frac{\sigma_{X_i}}{\sigma_{X_i|Q_j}}. \tag{8.5}$$

If we simply look at the above formulation, it is somehow doubtful since only the variances σ_{X_i} and $\sigma_{X_i|Q_j}$ are considered in the calculation and not the mean values. Both variances can be easily obtained from the observed feature vectors and the question regarding the mean values is still left unanswered.

The answer to the question above is given by carefully examining the variance σ_{X_i}. Following

an analogy to the variance formulation in (5.53), we get

$$\sigma_{X_i}^2 = \sum_{j=1}^{N_Q} P(Q_j) \left[\sigma_{X_i|Q_j}^2 + (\mu_{X_i} - \mu_{X_i|Q_j})^2 \right], \tag{8.6}$$

with

$$\mu_{X_i} = \sum_{j=1}^{N_Q} P(Q_j)\, \mu_{X_i|Q_j}.$$

The mean values μ_{X_i} and $\mu_{X_i|Q_j}$ do not explicitly appear in the formulation of mutual information in (8.5) but as it is shown in (8.6), the variance σ_{X_i} is basically obtained by taking the mean values into account.

Furthermore, let's define the following ratio:

$$\begin{aligned} \frac{\sigma_{X_i}^2}{\sigma_{X_i|Q_j}^2} &= \frac{\sum_{r=1}^{N_Q} P(Q_r) \left[\sigma_{X_i|Q_r}^2 + (\mu_{X_i} - \mu_{X_i|Q_r})^2 \right]}{\sigma_{X_i|Q_j}^2} \\ &= \alpha_{i,j}^2 + \beta_{i,j}^2, \end{aligned} \tag{8.7}$$

where

$$\begin{aligned} \alpha_{i,j}^2 &= \frac{\sum_{r=1}^{N_Q} P(Q_r)(\mu_{X_i} - \mu_{X_i|Q_r})^2}{\sigma_{X_i|Q_j}^2}, \\ \beta_{i,j}^2 &= \frac{\sum_{r=1}^{N_Q} P(Q_r)\, \sigma_{X_i|Q_r}^2}{\sigma_{X_i|Q_j}^2}. \end{aligned}$$

The mutual information is thus formulated as

$$I_G(X_i;Q) = \sum_{j=1}^{N_Q} P(Q_j)\, \ln \sqrt{\alpha_{i,j}^2 + \beta_{i,j}^2}. \tag{8.8}$$

In order to get a better understanding about the variables $\alpha_{i,j}^2$ and $\beta_{i,j}^2$, we shall refer to Figure 8.2 where the distributions of $\alpha_{i,j}^2$ and $\beta_{i,j}^2$ are depicted for $i = 1 \cdots d$ and $j = 1 \cdots N_Q$.

It is shown that the distribution of $\beta_{i,j}^2$ is centered around the value of 1 which implies that the variances of $p(x_i|Q_j)$ are approximately equal. However, as we further observe the distribution in each dimension, this actually only applies to higher dimension indices and not to the lower dimension ones as shown in Figures 8.3 and 8.4 for the dimension $i = 1$ and 30, respectively. For lower dimension indices, the distributions of $\alpha_{i,j}^2$ and $\beta_{i,j}^2$ are not conforming the distribution

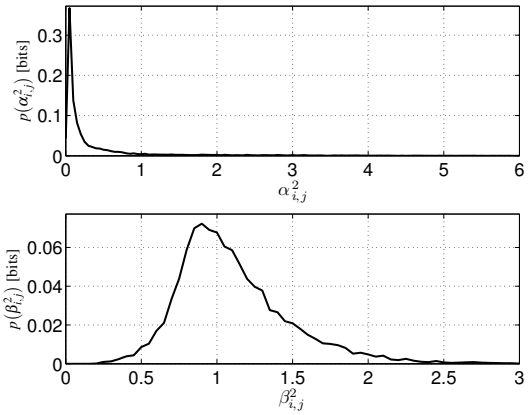


Figure 8.2: The distribution of $\alpha^2_{i,j}$ and $\beta^2_{i,j}$ for the 39-dimensional monomodal Gaussian approximation.

depicted in Figure 8.2 while for higher dimension indices they are showing a similarity.

Now if we assume that the variances $\sigma^2_{X_i|Q_j}$ are exactly equal for all states Q_j, denoted by σ^2_i, we get

$$\frac{\sigma^2_{X_i}}{\sigma^2_{X_i|Q_j}} \approx \alpha^2_{i,j} + 1,$$

and the mutual information is shown as

$$I_G(X_i;Q) \approx \sum_{j=1}^{N_Q} P(Q_j)\ \ln\ \sqrt{\alpha^2_{i,j}+1}. \tag{8.9}$$

Furthermore, if the following conditions hold:

$$\begin{aligned} P(Q_j) &= \frac{1}{N_Q}, \\ \alpha_{i,j} &= \alpha_i = \frac{\Delta_{\mu_i}}{\sigma_i}, \end{aligned}$$

where the states are uniformly distributed and the absolute distances Δ_{μ_i} between the *global mean* μ_{X_i} and the *local means* $\mu_{X_i|Q_j}$ are exactly the same, we get the relation

$$I_G(X_i;Q) \approx I_{G_A}(X_i;Q) = \ln\ \sqrt{\alpha^2_i+1}, \tag{8.10}$$

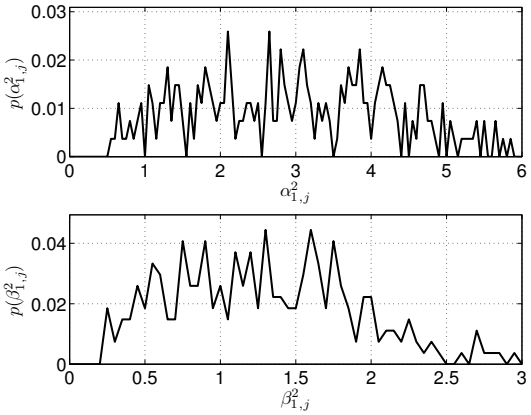


Figure 8.3: The distribution of $\alpha^2_{1,j}$ and $\beta^2_{1,j}$ for the first dimension of the 39-dimensional monomodal Gaussian approximation.

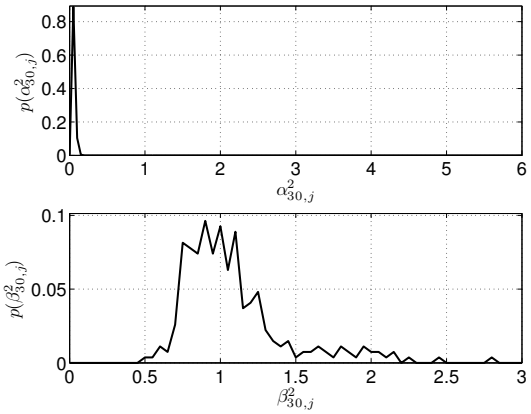


Figure 8.4: The distribution of $\alpha^2_{30,j}$ and $\beta^2_{30,j}$ for the 30-th dimension of the 39-dimensional monomodal Gaussian approximation.

where $I_{G_A}(X_i;Q)$ denotes the first constrained-approximation to $I_G(X_i;Q)$.

The formulation in (8.10) is in fact similar to (5.58) where we have presented an example of a two-state classification task. The behavior of the parameter α was depicted in Figure 5.4 and it was shown that this approximation is good for small values of α, i.e., $\alpha < 1$. Since the distribution of $\alpha_{i,j}^2$ is indicating small values of $\alpha_{i,j}^2$ for high dimension indices as described earlier, we could conclude that the above monomodal approximation should be quite good except for several low dimension indices.

Now we are proceeding with the investigation whether the formulation above, i.e., $I_{G_A}(X_i;Q)$, already gives a good approximation to (8.8). In this case, we keep the formulation in (8.10) and assume that

$$\beta_{i,j} = \beta_i,$$

which implies that the state variances are not equal as assumed previously. The value of β_i is given by

$$\beta_i^2 = \sum_{j=1}^{N_Q} P(Q_j)\ \beta_{i,j}^2.$$

Based on the above assumption we get

$$I_G(X_i;Q) \approx I_{G_B}(X_i;Q) = \ln\ \sqrt{\alpha_i^2 + \beta_i^2}, \tag{8.11}$$

where α_i^2 is now calculated as

$$\alpha_i^2 = \sum_{j=1}^{N_Q} P(Q_j)\ \alpha_{i,j}^2. \tag{8.12}$$

$I_{G_B}(X_i;Q)$ denotes the second constrained-approximation of $I_G(X_i;Q)$.

Figure 8.5 depicts the values for both approximations of $I_G(X_i;Q)$, i.e., $I_{G_A}(X_i;Q)$ and $I_{G_B}(X_i;Q)$. As shown in the figure, $I_{G_A}(X_i;Q)$ yields a better approximation to $I_G(X_i;Q)$ than $I_{G_B}(X_i;Q)$. This result indicates that β_i^2 can be neglected in the calculation. Nevertheless, the total mutual information $I_G(\mathbf{X};Q) = 8.03$ bits is already exceeding the upper bound of $H(Q) = 8.0025$ bits.

8.2.2 Monogram Approximation - Multimodal

The aim of this section is to determine the entropy $H(\mathbf{X})$ using the monogram approximation based on the observed feature vector distribution which is best described by its multimodality. The shape of the distribution can be described by a histogram or by means of a Gaussian mixture. The latter is taken into account due to the assumption that the distribution $p(\mathbf{x}|Q_j)$ which is

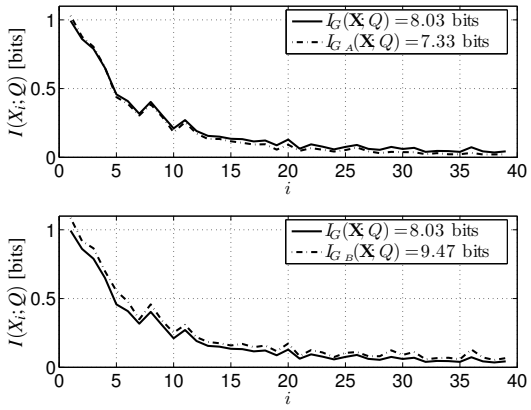


Figure 8.5: Comparison of two formulations of the mutual information calculation based on the monomodal Gaussian approximation.

making up the distribution $p(\mathbf{x})$ is assumed to be a Gaussian distribution.

Taking into account the feature component independency, we should be able to compare the distribution obtained with three different approaches:

- Measured histogram.
- Monomodal Gaussian: $p(x_i) \sim \mathcal{N}(\mu_{X_i}, \sigma^2_{X_i})$.
- Gaussian mixture: $p(x_i) \sim \sum_{j=1}^{N_Q} P(Q_j) \cdot \mathcal{N}(\mu_{X_i}, \sigma^2_{X_i})$.

Figure 8.6 depicts the three distributions for a low index $i = 1$ and a high index $i = 30$. The multimodal Gaussian mixture approximation leads to a more accurate approximation than the monomodal. However, the measured distribution is still more peaky than the multimodal approximation for feature components with high dimensional index i. This gives a hint that the approximation of $p(x_i|Q_j)$ as monomodal Gaussian distribution could be improved by a more peaky distribution.

The Kullback-Leibler distance [Kullback and Leibler 1951] is a distance measure which can be used to measure the distance between the measured histogram and its approximations which are the monomodal Gaussian and Gaussian mixture. It is defined as

$$D_{KL}(A||B) = \sum_x A(x) \text{ ld } \frac{A(x)}{B(x)}, \tag{8.13}$$

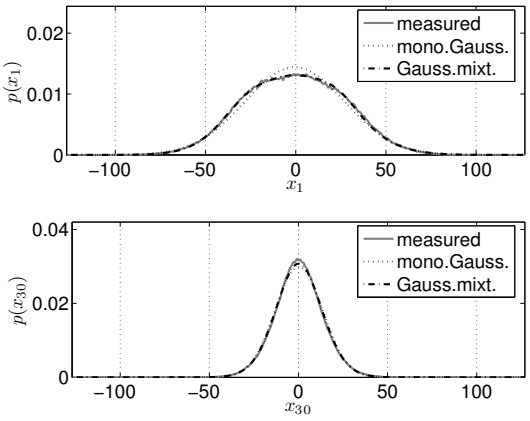


Figure 8.6: Comparison between the measured distribution, monomodal Gaussian, and multimodal Gaussian mixture distributions for the dimensions $i = 1$ and $i = 30$ of the feature vectors.

where $A(x)$ and $B(x)$ denote the measured histogram and the approximations, respectively. The distances from the measured histogram to the monomodal Gaussian approximation as depicted in Figure 8.6 are 0.0057 and 0.0046 bits for the top and the bottom figures, respectively. The distances from the measured histogram to the multimodal Gaussian mixture approximation are 0.0017 and 0.0013 bits, respectively. This implies that the multimodal approximation is superior to the monomodal one in all dimensions and a better approximation is achieved for the lower dimensional indices.

The calculation of the monogram mutual information $I_1(X_i;Q)$ is shown as

$$I_1(X_i;Q) = H(X_i) - H(X_i|Q),$$

where $H(X_i|Q)$ is obtained using (8.3) and $H(X_i)$ is calculated by assuming a Gaussian mixture. In this particular case, we present some results of the entropy $H(X_i)$ calculated based on the measured histogram. This method is basically suffering from the histogram accurateness where the entropy value is not unique depending on the histogram precision. Nevertheless, we are showing the results for a comparison purpose only.

Figure 8.7 depicts the mutual information in the case where $H(X_i)$ is calculated based on the measured histogram of $p(x_i)$. As shown in the figure $H(X_i|Q)$ is quite constant for all dimensions i. This result demonstrates that the variances of $\sigma^2_{X_i|Q_j}$ are approximately equal as shown previously in Figure 8.2 with the distribution of $\beta^2_{i,j}$. Furthermore, it also shows that the entropy $H(X_i)$

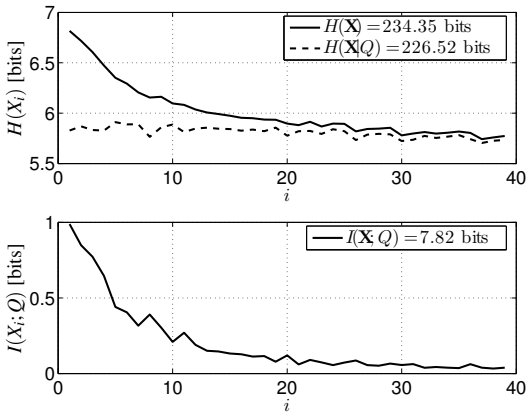


Figure 8.7: Entropy and mutual information calculated based on the measured histogram of $p(\mathbf{x})$.

decreases to the value of $H(X_i|Q)$ as i is higher which finally leads to a decrease of the mutual information $I(X_i;Q)$. Figure 8.8 shows the deviation $\Delta H(X_i)$ from the measured distribution $H(X_i)$ as depicted in Figure 8.7 given the monomodal Gaussian and Gaussian mixture approaches. As the figure shows, major differences are only observed for low values of the dimension index i. This is consistent with the distribution shown in Figure 8.2.

Table 8.1 shows the values of mutual information $I(\mathbf{X};Q)$. Three values from the monomodal approximation are given. The others are obtained with the Gaussian mixture and the measured distribution of $\mathbf{x}$. The measured mutual information leads to the lowest value. A more accurate approximation of $p(\mathbf{X}|Q_j)$ (e.g., a more peaky distribution) could further reduce the difference between the measured and multimodal approximation. The monomodal approximations $I_{G_B}(\mathbf{X};Q)$ and $I_G(\mathbf{X};Q)$ are not good as they exceed the upper bound $H(Q) = 8.0025$ bits.

Table 8.1: Mutual information obtained based on several approximations of $p(\mathbf{x})$.

$I_{G_A}(\mathbf{X};Q)$	7.33 bits
$I_{G_B}(\mathbf{X};Q)$	9.47 bits
$I_G(\mathbf{X};Q)$	8.03 bits
$I_1(\mathbf{X};Q)$ - Gauss.mixt.	7.92 bits
$I_1(\mathbf{X};Q)$ - measured	7.82 bits

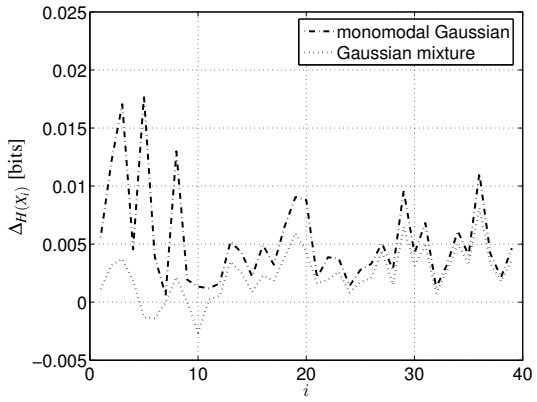


Figure 8.8: The mutual information difference between both the monomodal Gaussian and Gaussian mixture approximations and the measured distribution.

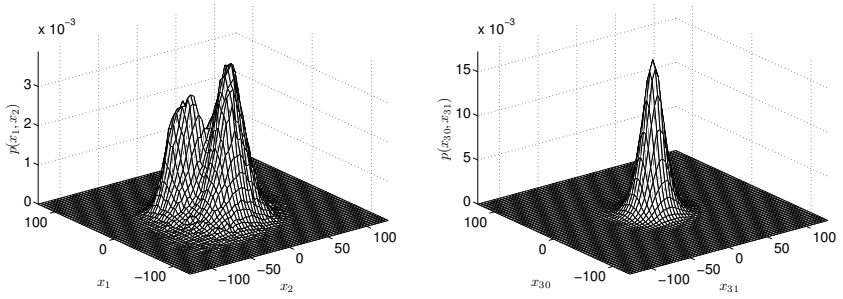
8.3 Bigram Approximation

In this bigram approximation, we still assume the monomodal Gaussian distribution for $p(\mathbf{x}|Q_j)$. To determine the entropy $H_2(X)$ as defined by (5.35) different methods can be applied depending on how the joint probability $p(x_i, x_{i+1})$ is determined. In the following we again consider three cases:

- Measured histogram.
- Monomodal approximation.
- Two-dimensional bimodal approximation.

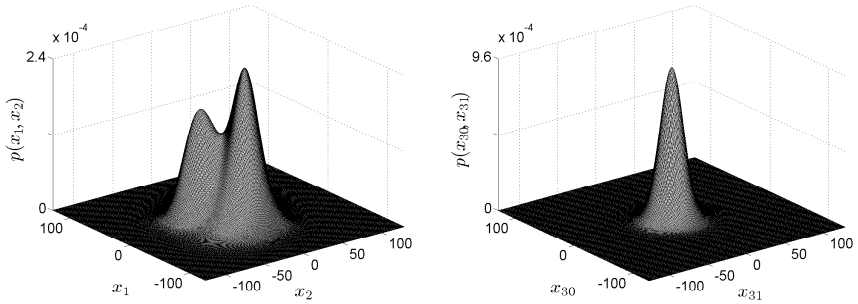
The measured histogram distribution is shown in Figure 8.9 and the bimodal approximation is depicted in Figure 8.10. As shown in the figure, the measured histogram depicts the bimodal property which can also be described by a two-dimensional bimodal function. The monomodal approximation will certainly fail in modeling the bimodality of the lower dimensional indices. We have used the sample means and variances of the observed feature vectors as the input parameters of the Gaussian mixture to produce the bimodal function.

Based on the shown distributions, the entropy $H(\mathbf{X})$ can be calculated. We again emphasize that the result obtained from the measured histogram distribution is not stable and used for comparison purpose as also intended in the previous section. Table 8.2 tabulates the values of



(a) $p(x_1,x_2)$ (b) $p(x_{30},x_{31})$

Figure 8.9: Bigram measured distribution.



(a) $p(x_1,x_2)$ (b) $p(x_{30},x_{31})$

Figure 8.10: Bigram multimodal Gaussian mixture approximation.

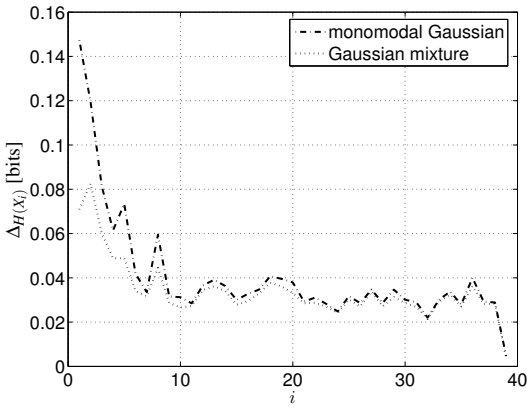


Figure 8.11: The mutual information difference between both the monomodal Gaussian and Gaussian mixture approximations and the measured distribution using the bigram approach.

$H(\mathbf{X})$ and $I(\mathbf{X},Q)$. For an informational purpose, the difference between the monomodal and the measured histogram distribution approximation and also between the bimodal Gaussian mixture and the measured histogram distribution is shown in Figure 8.11. As again shown in the figure, major differences are observed for lower dimension indices i.

Table 8.2: Mutual information of the bigram method.

	$H(\mathbf{X})$ [bits]	$I(\mathbf{X};Q)$ [bits]
Monomodal Gaussian	234.5539	8.0298
Bimodal Gaussian mixture	234.2957	7.7717
Measured distribution	232.9701	6.4461

Comparing the bigram values shown in the table with the monogram ones in Table 8.1 we conclude that:

- The monomodal approximation used in the bigram method gives worse value than in the monogram one. This is easily explained through its failure to model the bimodal property of the joint distribution as discussed previously.
- A slight improvement is observed using the Gaussian mixture approximation. This implies that the bigram modeling to capture the dependency properties of the feature vectors is indeed showing an improvement for the entropy estimation.

According to the results, we are proposing to use the Gaussian mixture assumption to determine the mutual information. The use of monogram approximation is sufficient to gain an insight into the data under analysis. In this monogram context, one of the proposed monomodal Gaussian approximation is considerable.

8.4 Analysis on the Influence of Noise in the Feature Vectors

As we have discussed in previous sections, the use of monomodal Gaussian assumption is somehow justified for the feature vectors obtained from the Aurora 3 German database. This has been particularly highlighted in Section 8.2.1 where the ratio of α is still within an acceptable range. This motivates us to start an analysis on the influence of noise in the feature vectors using the monomodal Gaussian assumption. The theoretical explanation has been derived in Section 5.7 and now we are going to show that the method of mutual information can be applied to assess the performance of weighting rules in particular and any future front-end processing improvements.

In order to carefully perform the analysis on the noise influence on the Aurora 3 German database, the noisy speech *hands-free* and clean speech *close talk* utterances have to be synchronized. The synchronization was done by estimating an integer time delay occurred during the recording and aligning the speech samples after removing the time delay. The forced Viterbi algorithm was then performed on the clean speech utterances to segment the clean speech and *silence* feature vectors. Finally, the information regarding these segmentations were used in the *hands-free* utterances to obtain the noisy speech, denoised speech, and noise feature vectors.

The mutual informations of the clean speech, denoised speech, and noisy speech feature vectors are shown in Figure 8.12. The loss of mutual information $I(R_i, Q)$ is also depicted. The figure shows that a denoising technique has improved the feature vectors on, for example, the first 7 feature components except the fourth feature component. Although there is no one-to-one relationship between the *gained* bits and the relative word accuracy improvement, it nevertheless gives us a hint that the denoising technique has created a better representation of the feature vectors. Future improvements can then be directed to obtain a much better improvement on all feature components.

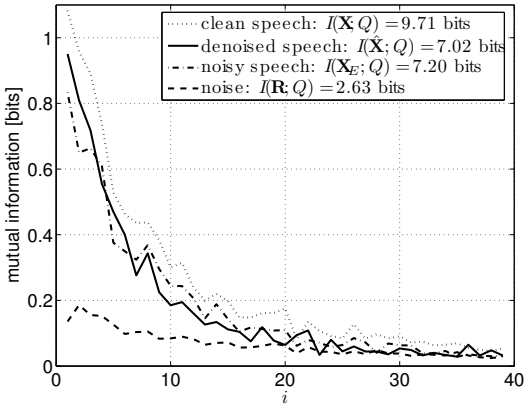


Figure 8.12: The mutual informations of the clean speech, denoised speech, and noisy speech feature vectors on the Aurora 3 German database. The loss of mutual information $I(R_i, Q)$ is also depicted in the figure.

Chapter 9

Conclusions and Future Directions

In this closing chapter we would like to highlight our main achievements and propose the directions for future development as described in the following:

- Two proposed least-squares weighting rules, i.e., the recursive gain least squares and spectral subtraction based *recursive* least-squares, have consistently outperformed the baseline noise reduction method used in the SFE on both small and large vocabulary databases. A relative improvement in word accuracy of up to 23.7 % is observed on the Aurora 3 German database and up to 18 % on SPEECON and SpeechDat-Car Spanish databases.

- The proposed method of multidimensional weighting rule parameter optimization has proven reliable to obtain a possible global optimum. Although there is no guarantee that a global maximum has always been reached, the method can at least be used to objectively optimize all the weighting rules under consideration. However, we believe that an improvement on the Monte Carlo and *direct search* method might be necessary in order to obtain a more accurate global optimum.

- Applying additional root compression and cepstral smoothing algorithms increases the SFE performance by up to 32.5 % relative increase in word accuracy when using the three-state voice activity driven noise PSD estimator and 35.1 % using the minimum statistics on the Aurora 3 German digits task.

- Applying root compression and cepstral smoothing algorithms increases the SFE performance to finally outperform the state-of-the-art front-end processing, i.e., the ETSI advanced front-end, on the Aurora 3 German digits task by a relative increase in word accuracy of up to 4.9 % when using the three-state voice activity driven noise PSD estimator

and 8.6 % using the minimum statistics. This fact gives a hint on a possible future improvement in the area of feature extraction where the mentioned algorithms can be explored to deliver a more elaborate method.

- The additional complexity and memory requirements of the proposed weighting rules are neglectable or minimum since they are having a similar structure as the baseline weighting rule. Root compression and cepstral smoothing algorithms only require a small amount of additional complexity and memory requirements.

- A conceptual direction in the analysis of feature vectors based on the mutual information has been proposed to asses the feature vector quality regardless of the acoustic modeling assumption used in the speech recognition system. This is done by utilizing the relation between the Bayes probability of error and the conditional entropy of the state given the feature vector.

- We have proposed to use the *histogram* approach to evaluate the mutual information by assuming the existence of an underlying density function. We are basically proposing the use of a Gaussian mixture model as the assumed underlying density. Our method simply estimates the parameters of the Gaussian mixture and performs a sampling on the function. We believe that there is at least a need to improve the estimation of the Gaussian mixture parameters or even explore another method of mutual information or entropy estimation, such as in [Nilsson and Kleijn 2007].

- We have also proposed a monomodal Gaussian feature vector analysis in the presence of noise. Based on this coarse formulation, we are able to identify the effect on the feature vectors when applying a certain weighting rule. The method can be used to initially explore possible improvements based on future developments on the front-end. A better estimation on the modeling technique is recommended and finally it might be a way to establish a relationship between the word accuracy measurement (not the Bayes probability of error) and the entropy bounds.

Appendix A

HMM Parameter Estimation

A.1 The Forward-Backward Algorithm

This algorithm offers an efficient method to compute the likelihood equation in (2.12). It is done by introducing two new variables, the forward probability $\alpha_j(\ell)$ and the backward probability $\beta_i(\ell)$, defined as

$$\alpha_j(\ell) = p(\mathbf{x}(0),\ \cdots,\ \mathbf{x}(\ell),\ s_j(\ell)|\Lambda), \tag{A.1}$$

$$\beta_i(\ell) = p(\mathbf{x}(\ell+1),\ \cdots,\ \mathbf{x}(M-1)|s_i(\ell),\ \Lambda). \tag{A.2}$$

Based on these definitions it is possible to obtain an iterative procedure for $\alpha_j(\ell)$ and $\beta_j(\ell)$. First of all, $\alpha_j(\ell)$ is calculated as

$$\alpha_j(0) = \pi_j b_j\big(\mathbf{x}(0)\big) \qquad \text{for } j \in [1,\ N_Q], \tag{A.3}$$

and for $1 \leq \ell \leq M-1$ and $1 \leq j \leq N_Q$

$$\alpha_j(\ell) = \left[\sum_{i=1}^{N_Q} \alpha_i(\ell-1)a_{ij}\right] b_j\big(\mathbf{x}(\ell)\big). \tag{A.4}$$

The calculation of $\beta_i(\ell)$ is shown as

$$\beta_i(M-1) = 1 \qquad 1 \leq i \leq N_Q, \tag{A.5}$$

and for $0 \leq \ell \leq M-2$ with $1 \leq i \leq N_Q$

$$\beta_i(\ell) = \sum_{j=1}^{N_Q} a_{ij} b_j\big(\mathbf{x}(\ell+1)\big)\beta_j(\ell+1). \tag{A.6}$$

The likelihood equation in (2.12) is thus computed as

$$\begin{aligned} p(\mathbf{x}^M|\Lambda) &= \sum_{j=1}^{N_Q} \alpha_j(M-1) = \sum_{i=1}^{N_Q} \pi_i b_i\big(\mathbf{x}(0)\big)\beta_i(0) && \text{(A.7)} \\ &= \sum_{j=1}^{N_Q} \alpha_j(\ell)\beta_j(\ell). && \text{(A.8)} \end{aligned}$$

A.2 The Baum-Welch Algorithm

A multi-dimensional optimization problem to estimate the HMM parameters based on the maximum likelihood criterion is described in this section. The aim is to find an estimate of the parameters $\hat{\Lambda}$ which maximizes the likelihood function. This is shown as

$$\hat{\Lambda} = \arg\max_{\Lambda} \mathcal{L}(\Lambda|\mathbf{x}^M), \tag{A.9}$$

where an iterative technique, the expectation-maximization (EM) [Dempster et al. 1977], is employed to solve the problem. For the purpose of the HMM parameters training, it is referred to as the Baum-Welch re-estimation Algorithm [Baum et al. 1970].

The algorithm is assuming the existence of additional *hidden* data in the likelihood function. In the HMM context the hidden data is the underlying state sequence ψ. The original likelihood function $\mathcal{L}(\Lambda|\mathbf{x}^M)$ is thus referred to as the incomplete-data likelihood function whereas the complete-data likelihood function is defined as $\mathcal{L}(\Lambda|\mathbf{x}^M, \psi)$. The first step in the iterative procedure is to evaluate an auxiliary function $\mathcal{Q}(\Lambda, \Lambda^{(i-1)})$ defined as the expectation of the complete-data log-likelihood with respect to the unknown data given the observed data and the current parameter estimate

$$\begin{aligned} \mathcal{Q}(\Lambda, \Lambda^{(i-1)}) &= E\left\{ \log \mathcal{L}(\Lambda|\mathbf{x}^M, \psi) \;\middle|\; \mathbf{x}^M, \Lambda^{(i-1)} \right\} \\ &= \sum_{\psi \in \Psi} \log\Big(p(\mathbf{x}^M, \psi|\Lambda)\Big) p(\mathbf{x}^M, \psi|\Lambda^{(i-1)}), && \text{(A.10)} \end{aligned}$$

where $\Lambda^{(i-1)}$ is the current parameter estimate. This is known as the E-step of the algorithm whereas the M-step is to maximize the auxiliary function defined in the first step

$$\Lambda^{(i)} = \arg\max_{\Lambda} \mathcal{Q}(\Lambda, \Lambda^{(i-1)}). \tag{A.11}$$

Evaluating both steps for each iteration is guaranteed to increase the log-likelihood and converge to a local maximum of the likelihood function.

The derivation of the re-estimation procedure from the $\mathcal{Q}$ function for CDHMMs having multivariate distributions is described in [Liporace 1982]. It was also shown that mixture distribution are treated as a special case. The results are briefly discussed here by first defining two variables

$$\gamma_i(\ell) = p\big(s_i(\ell)|\mathbf{x}^M, \Lambda\big), \tag{A.12}$$
$$\xi_{ij}(\ell) = p\big(s_i(\ell), s_j(\ell+1)|\mathbf{x}^M, \Lambda\big). \tag{A.13}$$

The first variable $\gamma_i(\ell)$ can be expanded as

$$\gamma_i(\ell) = \frac{p\big(\mathbf{x}^M, s_i(\ell)|\Lambda\big)}{p(\mathbf{x}^M|\Lambda)} = \frac{\alpha_i(\ell)\beta_i(\ell)}{\sum_{j=1}^{N_Q} \alpha_j(\ell)\beta_j(\ell)}, \tag{A.14}$$

and the second variable $\xi_{ij}(\ell)$ is expanded as

$$\xi_{ij}(\ell) = \frac{p\big(s_i(\ell), s_j(\ell+1), \mathbf{x}^M|\Lambda\big)}{p(\mathbf{x}^M|\Lambda)} = \frac{\alpha_i(\ell)a_{ij}b_j(\ell+1)\beta_j(\ell+1)}{\sum_{j=1}^{N_Q} \alpha_j(\ell)\beta_j(\ell)}. \tag{A.15}$$

It can be shown that both variables are related by

$$\gamma_i(\ell) = \sum_{j=1}^{N_Q} \xi_{ij}(\ell). \tag{A.16}$$

Using the above formulations, a re-estimation procedure to estimate the HMM parameters is shown as

$$\hat{\pi}_i = \gamma_i(0), \tag{A.17}$$
$$\hat{a}_{ij} = \frac{\sum_{\ell=0}^{M-2} \xi_{ij}(\ell)}{\sum_{\ell=0}^{M-2} \gamma_i(\ell)}. \tag{A.18}$$

While (A.17) is formulated employing the definition in (A.12), the formulation of (A.18) can be interpreted in terms of relative frequencies as the expected number of transitions from state Q_i to

state Q_j divided by the expected number of transitions from state Q_i as (A.16) implies.

For the EM derivation of a multivariate Gaussian mixture distribution, the formulation of (2.8) without the index i is considered. The re-estimation procedure can be shown as

$$\hat{c}_r = \frac{1}{M}\sum_{\ell=0}^{M-1} p\big(r|\mathbf{x}(\ell),\Theta^{(i-1)}\big), \tag{A.19}$$

$$\hat{\mu}_r = \frac{\sum_{\ell=0}^{M-1} p\big(r|\mathbf{x}(\ell),\Theta^{(i-1)}\big)\mathbf{x}(\ell)}{\sum_{\ell=0}^{M-1} p\big(r|\mathbf{x}(\ell),\Theta^{(i-1)}\big)}, \tag{A.20}$$

$$\hat{\Sigma}_r = \frac{\sum_{\ell=0}^{M-1} p\big(r|\mathbf{x}(\ell),\Theta^{(i-1)}\big)\big(\mathbf{x}(\ell)-\hat{\mu}_r\big)\big(\mathbf{x}(\ell)-\hat{\mu}_r\big)^T}{\sum_{\ell=0}^{M-1} p\big(r|\mathbf{x}(\ell),\Theta^{(i-1)}\big)}, \tag{A.21}$$

where

$$p\big(r|\mathbf{x}(\ell),\Theta^{(i-1)}\big) = \frac{c_r\, b_r\big(\mathbf{x}(\ell)\big)}{b(\mathbf{x}(\ell))}, \tag{A.22}$$

with $\Theta^{i-1} = \Big(\{c_r\},\{\mu_r\},\{\Sigma_r\}\Big)$. In order to formulate the above re-estimation procedures in the CDHMMs context, a new variable that is defined which is taking account the knowledge about the state i

$$\gamma_{ir}(\ell) = p\big(s_{ir}(\ell)|\mathbf{x}^M,\Lambda\big) = \gamma_i(\ell)\frac{c_r\, b_r\big(\mathbf{x}(\ell)\big)}{b(\mathbf{x}(\ell))}, \tag{A.23}$$

where $s_{ir}(\ell)$ denotes as being in the mixture component r of state Q_i at frame ℓ. Knowing the re-estimation procedure for the Gaussian mixture shown previously, the following procedure is formulated:

$$\hat{c}_{ir} = \frac{\sum_{\ell=0}^{M-1}\gamma_{ir}(\ell)}{\sum_{\ell=0}^{M-1}\gamma_i(\ell)}, \tag{A.24}$$

$$\hat{\mu}_{ir} = \frac{\sum_{\ell=0}^{M-1}\gamma_{ir}(\ell)\mathbf{x}(\ell)}{\sum_{\ell=0}^{M-1}\gamma_{ir}(\ell)}, \tag{A.25}$$

$$\hat{\Sigma}_{ir} = \frac{\sum_{\ell=0}^{M-1}\gamma_{ir}(\ell)\big(\mathbf{x}(\ell)-\hat{\mu}_{ir}\big)\big(\mathbf{x}(\ell)-\hat{\mu}_{ir}\big)^T}{\sum_{\ell=0}^{M-1}\gamma_{ir}(\ell)}, \tag{A.26}$$

where $\hat{c}_{ir}$ is interpreted as the expected number of times being in the mixture component r of state Q_i divided by the expected number of times being in the state Q_i and $\hat{\mu}_{ir}$, $\hat{\Sigma}_{ir}$ follow directly from $\hat{\mu}_r$, $\hat{\Sigma}_r$.

The specific problem with the left-to-right Bakis topology is that the amount of observations generated by any state in a single observation sequence is not sufficient to estimate the parameters. In this case, multiple observation sequences are needed [Rabiner and Juang 1993]. Given

a number of $\mathcal{V}$ observation sequences with the index v being the v-th observation sequence of length M_v, the re-estimation procedure can be intuitively shown in terms of relative frequencies as

$$\hat{\pi}_i = \frac{\sum_{v=1}^{\mathcal{V}} \gamma_i^{(v)}(0)}{\mathcal{V}}, \tag{A.27}$$

$$\hat{a}_{ij} = \frac{\sum_{v=1}^{\mathcal{V}} \sum_{\ell=0}^{M_v-2} \xi_{ij}^{(v)}(\ell)}{\sum_{v=1}^{\mathcal{V}} \sum_{\ell=0}^{M_v-2} \gamma_i^{(v)}(\ell)}, \tag{A.28}$$

$$\hat{c}_{ir} = \frac{\sum_{v=1}^{\mathcal{V}} \sum_{\ell=0}^{M_v-1} \gamma_{ir}^{(v)}(\ell)}{\sum_{v=1}^{\mathcal{V}} \sum_{\ell=0}^{M_v-1} \gamma_i^{(v)}(\ell)}, \tag{A.29}$$

$$\hat{\mu}_{ir} = \frac{\sum_{v=1}^{\mathcal{V}} \sum_{\ell=0}^{M_v-1} \gamma_{ir}^{(v)}(\ell) \mathbf{x}^{(v)}(\ell)}{\sum_{v=1}^{\mathcal{V}} \sum_{\ell=0}^{M_v-1} \gamma_{ir}^{(v)}(\ell)}, \tag{A.30}$$

$$\hat{\Sigma}_{ir} = \frac{\sum_{v=1}^{\mathcal{V}} \sum_{\ell=0}^{M_v-1} \gamma_{ir}^{(v)}(\ell) \left(\mathbf{x}^{(v)}(\ell) - \hat{\mu}_{ir}\right) \left(\mathbf{x}^{(v)}(\ell) - \hat{\mu}_{ir}\right)^T}{\sum_{v=1}^{\mathcal{V}} \sum_{\ell=0}^{M_v-1} \gamma_{ir}^{(v)}(\ell)}, \tag{A.31}$$

with $(\cdot)^{(v)}$ denotes belonging to the v-th observation sequence and M_v denotes the length of the v-th observation sequence.

Appendix B

A Lower Bound on the Bayes Probability of Error

In this Section we rewrite the lower bound of Bayes probability of error P_B based on [Höge 1999]. The probability of error is formulated as

$$\begin{aligned} P_B &= E_{\mathbb{X}}\Big[\, P_e(Q|\mathbf{x}) \,\Big] \\ &= \int_{\mathbb{X}} \Big[\, 1 - \arg\max_{Q_j}\ p(Q_j|\mathbf{x}) \,\Big]\ p(\mathbf{x})\ d\mathbf{x}. \end{aligned} \tag{B.1}$$

Let's start the derivation by evaluating the following term:

$$\begin{aligned} P_e(Q|\mathbf{x}) &= 1 - \arg\max_{Q_j}\ p(Q_j|\mathbf{x}) \\ &= 1 - \left[\left\{ \arg\max_{Q_j}\ P(Q_j|\mathbf{x}) \right\} \sum_{j=1}^{N_Q} P(Q_j|\mathbf{x}) \right] \\ &\leq 1 - P(Q_j|\mathbf{x})^2 = \sum_{j=1}^{N_Q} P(Q_j|\ \mathbf{x}) \Big(1 - P(Q_j|\ \mathbf{x}) \Big) \qquad \cdots(*) \\ &\leq -\sum_{j=1}^{N_Q} P(Q_j|\ \mathbf{x})\ \ln\ P(Q_j|\ \mathbf{x}). \end{aligned} \tag{B.2}$$

Note that the inequality in $(*)$ is only valid for $p(Q_j|\mathbf{x}) \leq 1/N_Q$. Finally, an expectation is applied to the above formulation as

$$\begin{aligned} P_B &= \int_{\mathbb{X}} \left[1 - \arg\max_{Q_j} \; p(Q_j|\mathbf{x}) \right] \; p(\mathbf{x}) \; d\mathbf{x} \\ &\leq -\int_{\mathbb{X}} \sum_{j=1}^{N_Q} P(Q_j|\mathbf{x}) \; \ln \; P(Q_j|\mathbf{x}) p(\mathbf{x}) \; d(\mathbf{x}) \\ &\leq -\ln 2 \cdot \int_{\mathbb{X}} \sum_{j=1}^{N_Q} P(Q_j|\mathbf{x}) \; \mathrm{ld} \; P(Q_j|\mathbf{x}) \; d(\mathbf{x}) \\ &\leq \ln 2 \cdot H(Q|\mathbf{X}). \end{aligned} \tag{B.3}$$

Appendix C

Working with Entropy

C.1 Differential Entropy of a One-Dimensional Feature Vector

A one-dimensional feature vector X with the probability density function $p(x)$ is considered. The feature vector X is regarded as a continuous random variable. To derive the differential entropy (also referred to as continuous entropy) a histogram is first created from the density function $p(x)$ followed by letting the width of the histogram bins goes to zero. The histogram is created by partitioning the continuous random variable X into small intervals with equal width Δ

$$I_{\Delta_i} = \{X : i\Delta \leq X < (i+1)\Delta\}. \tag{C.1}$$

The probability P that X falls into an interval I_{Δ_i} is given by

$$P(I_{\Delta_i}) = \widetilde{p}_i\Delta\,; \qquad \widetilde{p}_i\Delta = \int_{i\Delta}^{(i+1)\Delta} p(x)\,dx\,; \qquad \sum_{i=-\infty}^{\infty} \widetilde{p}_i\Delta = 1,$$

where $\widetilde{p}_i$ is an approximation of the density function $p(x)$ in interval I_{Δ_i}. The values $P(I_{\Delta_i})$ as function of i yields the histogram and they represent a true probability function as they fulfill the properties of (5.14). In case if the density function $p(x)$ is not known, the histogram is obtained from the observed data. In this case $P(I_{\Delta_i})$ is given by

$$P(I_{\Delta_i}) = \frac{n_i}{N}; \qquad N = \sum_i n_i,$$

where n_i denotes the number of occurrences that x falls into the $i-$th interval.

The entropy of the histogram describes the information needed to determine into which interval an observed value x will fall. It is given by

$$\begin{aligned} H(I_\Delta) &= -\sum_{i=-\infty}^{\infty} \widetilde{p}_i\Delta \text{ ld } (\widetilde{p}_i\Delta) \\ &= -\sum_{i=-\infty}^{\infty} \widetilde{p}_i\Delta \text{ ld } \Delta - \sum_{i=-\infty}^{\infty} \widetilde{p}_i\Delta \text{ ld } \widetilde{p}_i \\ &= -\text{ld } \Delta + H_\Delta(X), \end{aligned} \tag{C.2}$$

where

$$H_\Delta(X) = -\sum_{i=-\infty}^{\infty} \widetilde{p}_i\Delta \text{ ld } (\widetilde{p}_i).$$

The value of $H(I_\Delta)$ depends on the value of Δ. The entropy increases with smaller Δ because the number of intervals increases the uncertainty.

To finally calculate the entropy of a continuous random variable X the condition of $\Delta \to 0$ is set. In this case $H(I_\Delta)$ goes to ∞ but the entropy $H_\Delta(X)$ converges to the differential entropy $H(X)$ defined as

$$H(X) = -\int_{-\infty}^{\infty} p(x) \text{ ld } p(x)\, dx.$$

It is obvious that the differential entropy is not the limit of the discrete entropy for $n \to \infty$ hence the properties of discrete entropy do not necessarily apply to it.

C.2 Differential Entropy of a Multidimensional Feature Vector

Given a feature vector $\mathbf{X} = [\, X_1, \cdots, X_d \,]^T$ where $\mathbf{X}$ is regarded as a continuous multivariate random variable with the distribution $p(x_1, \cdots, x_d)$, the differential entropy is given as

$$H(\mathbf{X}) = -\int_{\mathbb{X}_1} \cdots \int_{\mathbb{X}_d} p(x_1, \cdots, x_d) \text{ ld } p(x_1, \cdots, x_d)\, dx_1 \cdots dx_d.$$

In general this expression can be handled analytically only in special cases, e.g., for Gaussian distributions. In addition to that, approximation via a multidimensional histogram is difficult to evaluate especially in the case where the dimension d is high. $H(\mathbf{X})$ can be handled easier if the

variables $X_1, \cdots, X_d$ are statistically independent. In this case we have

$$p(x_1, \cdots, x_d) = \prod_{i=1}^{d} p(x_i),$$

and the entropy is calculated as

$$\begin{aligned} H(\mathbf{X}) &= -\int_{\mathbb{X}_1} \cdots \int_{\mathbb{X}_d} \prod_{r=1}^{d} p(x_r) \sum_{i=1}^{d} \text{ld } p(x_i) \; dx_1 \cdots dx_d \\ &= -\sum_{i=1}^{d} \int_{\mathbb{X}_i} p(x_i) \text{ ld } p(x_i) \, dx_i \\ &= \sum_{i=1}^{d} H(X_i). \end{aligned} \tag{C.3}$$

The result shows that in the case of statistical independency the differential entropy $H(\mathbf{X})$ can be determined via the summation of all one-dimensional differential entropies.

C.3 Entropy of a Mixed Distribution

If we regard the distribution $p(x, Q_j)$ which is the joint distribution of the feature vector and the state we have to deal with a distribution which is partly discrete and partly continuous. In order to deal with its entropy the partitioning of X as in (C.1) is done. This leads to the joint discrete distribution

$$P(I_{\Delta_i}, Q_j) = \widetilde{p}_i(Q_j),$$

with

$$\begin{aligned} \widetilde{p}_i(Q_j) &= \int_{i\Delta}^{(i+1)\Delta} p(x, Q_j) \, dx, \\ \sum_{i=-\infty}^{\infty} \widetilde{p}_i(Q_j)\Delta &= P(Q_j), \end{aligned}$$

where $P(I_{\Delta_i}, Q_j)$ describes the probability that X in the interval I_{Δ_i} and the state Q_j have both been observed. The related entropy $H(I_\Delta, Q)$ is given by

$$H(I_\Delta, Q) = -\text{ld } \Delta + H_\Delta(X, Q),$$

where

$$H_\Delta(X,Q) = -\sum_{j=1}^{N_Q}\sum_{i=-\infty}^{\infty} \widetilde{p}_i(Q_j)\,\Delta\ \mathrm{ld}\ (\widetilde{p}_i(Q_j)).$$

The entropy $H_\Delta(X,Q)$ converges for $\Delta \to 0$ to the entropy

$$H(X,Q) = -\sum_{j=1}^{N_Q}\int_{i=-\infty}^{\infty} p(x,Q_j)\ \mathrm{ld}\ \Big(p(x,Q_j)\Big)\,dx.$$

C.4 Entropy of a Monomodal Gaussian Distribution

Gaussian Distribution

The one-dimensional Gaussian distribution $\mathcal{N}(\mu,\sigma^2)$ of a random variable X is defined as

$$\mathcal{N}(\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}}\, e^{-\frac{(x-\mu)^2}{2\sigma^2}},$$

where μ and σ^2 denote the mean and variance of X, respectively. The differential entropy of this distribution is given by [Papoulis 1991]

$$H(X) = \frac{1}{2}\ \ln\ (2\pi e\sigma^2). \tag{C.4}$$

Multivariate Gaussian Distribution

Given a multivariate random variable $\mathbf{X} = [\ X_1,\ \cdots,\ X_d\ ]^T$ with mean $\mu_\mathbf{X} = E\{\mathbf{X}\}$, covariance matrix $\Sigma_\mathbf{X} = E\{(\mathbf{X}-\mu_\mathbf{X})(\mathbf{X}-\mu_\mathbf{X})^T\}$, and its matrix determinant $|\Sigma_\mathbf{X}|$, the multivariate Gaussian distribution $\mathcal{N}(\mu_\mathbf{X},\Sigma_\mathbf{X})$ is defined by

$$\mathcal{N}(\mu_\mathbf{X},\Sigma_\mathbf{X}) = \frac{1}{\sqrt{(2\pi)^d|\Sigma_\mathbf{X}|}}\, e^{-\frac{1}{2}(\mathbf{x}-\mu_\mathbf{X})\Sigma_\mathbf{X}^{-1}(\mathbf{x}-\mu_\mathbf{X})^T}.$$

The entropy $H(\mathbf{X})$ of this distribution is given by [Papoulis 1991]

$$H(\mathbf{X}) = \frac{1}{2}\ \ln\ \Big((2\pi e)^d\ |\Sigma_\mathbf{X}|\Big) = \frac{d}{2}\ \ln\ (2\pi e) + \frac{1}{2}\ \ln\ |\Sigma_\mathbf{X}|. \tag{C.5}$$

C.5 Marginal Distribution of the Feature Vector

Given a multivariate random variable $\mathbf{X} = [\, X_1,\ \cdots,\ X_d \,]^T$, the marginal distribution $p(x_i)$ of a distribution $p(x_1,\ \cdots,\ x_d)$ is defined as

$$p(x_i) = \int_{\mathbb{X}_1} \cdots \int_{\mathbb{X}_{i-1}} \int_{\mathbb{X}_{i+1}} \cdots \int_{\mathbb{X}_d} p(x_1,\ \cdots,\ x_d)\, dx_1 \cdots dx_{i-1}\, dx_{i+1} \cdots dx_d.$$

For the distribution of $p(x_1,\ \cdots,\ x_d)$ as given by

$$p(x_1,\ \cdots,\ x_d) = \sum_{j=1}^{N_Q} p(x_1,\ \cdots,\ x_d,\ Q_j) = \sum_{j=1}^{N_Q} P(Q_j)\ p(x_1,\ \cdots,\ x_d|Q_j),$$

the marginal distribution is calculated as

$$p(x_i) = \sum_{j=1}^{N_Q} P(Q_j) \int_{\mathbb{X}_1} \cdots \int_{\mathbb{X}_{i-1}} \int_{\mathbb{X}_{i+1}} \cdots \int_{\mathbb{X}_d} p(x_1,\ \cdots,\ x_d|Q_j)\, dx_1 \cdots dx_{i-1}\, dx_{i+1} \cdots dx_d.$$

For a statistically independent random variable $\mathbf{X}$ we have

$$p(x_1,\ \cdots,\ x_d|Q_j) = \prod_{i=1}^{d} p(x_i|Q_j),$$

and

$$p(x_i) = \sum_{j=1}^{N_Q} P(Q_j) p(x_i|Q_j), \tag{C.6}$$

which relates the marginal distribution $p(x_i)$ to the marginal distribution of $p(x_i|Q_j)$.

Means and Variances of the Marginal Distribution

The means and variances of the marginal distributions $p(x_i)$ and $p(x_i|Q_j)$ are shown as

$$\mu_{X_i} = \int_{\mathbb{X}_i} x_i\ p(x_i)\, dx_i \qquad \sigma^2_{X_i} = \int_{\mathbb{X}_i} (x_i - \mu_{X_i})^2\ p(x_i)\, dx_i,$$

$$\mu_{X_i|Q_j} = \int_{\mathbb{X}_i} x_i\ p(x_i|Q_j)\, dx_i \qquad \sigma^2_{X_i|Q_j} = \int_{\mathbb{X}_i} (x_i - \mu_{X_i|Q_j})^2\ p(x_i|Q_j)\, dx_i.$$

Using (C.6), which implies the statistical independency of the feature vectors in a state, yields

the following relationship of the means and variances between the marginal distributions $p(x_i)$ and $p(x_i|Q_j)$:

$$\begin{aligned}
\mu_{X_i} &= \int_{\mathbb{X}_i} x_i \sum_{j=1}^{N_Q} P(Q_j)\; p(x_i|Q_j)\; dx_i \\
&= \sum_{j=1}^{N_Q} P(Q_j)\; \mu_{X_i|Q_j}, \qquad \text{(C.7)} \\
\sigma^2_{X_i} &= \int_{\mathbb{X}_i} (x_i - \mu_{X_i})^2 \Big[\sum_{j=1}^{N_Q} P(Q_j)\; p(x_i|Q_j)\Big]\; dx_i \\
&= \sum_{j=1}^{N_Q} \int_{\mathbb{X}_i} \Big[(x_i - \mu_{X_i|Q_j})^2 + (\mu^2_{X_i} - \mu^2_{X_i|Q_j}) + 2\;(\mu_{X_i|Q_j} - \mu_{X_i})\; x_i\Big]\; P(Q_j)\; p(x_i|Q_j)\; dx_i \\
&= \sum_{j=1}^{N_Q} P(Q_j)\; \Big[\sigma^2_{X_i|Q_j} + (\mu^2_{X_i} - \mu^2_{X_i|Q_j}) + 2\;(\mu_{X_i|Q_j} - \mu_{X_i})\; \mu_{X_i|Q_j}\Big] \\
&= \sum_{j=1}^{N_Q} P(Q_j)\; \Big[\sigma^2_{X_i|Q_j} + (\mu_{X_i} - \mu_{X_i|Q_j})^2\Big]. \qquad \text{(C.8)}
\end{aligned}$$

Appendix D

Modeling the Temporal Statistical Dependency of the Feature Vector

Until now we only assume that we have to classify a state Q given an independent realization of the feature vector random variable $\mathbf{X} = \mathbf{x}$ with $\mathbf{x} = [\, x_1 \ \cdots \ x_d \,]^T$. In practice, the length V of the feature vectors associated to a state is a varying random variable which takes on the following set of possible values $V = \{1, \ \cdots, \ N_V\}$ with N_V denotes the maximum possible sequence length. This implies that we can not simply use the set $\{(\mathbf{x}, \ Q_j)\}$ since it neglects the feature vector temporal dependency given by the sequence. The set $\{(\mathbf{x}, \ Q_j)\}$ is therefore broken down into several subsets $\{(\mathbf{x}^v, \ Q_j)\}$, where $\mathbf{x}^v = \{\mathbf{x}(\ell), \ \cdots, \ \mathbf{x}(\ell + v - 1)\}$ denotes a possible sequence of $\mathbf{X}$ with length v.

To begin with the analysis of temporal dependency, let's define a random variable $\mathbf{X}_j^v$ to describe the feature vectors in the subset $\{(\mathbf{x}^v, \ Q_j)\}$ which takes on values defined in the set $\mathbb{X}_j^v$. The corresponding mutual information is given by

$$I(\mathbf{X}_j^v; Q_j) = H(\mathbf{X}_j^v) - H(\mathbf{X}_j^v | Q_j), \tag{D.1}$$

where

$$\begin{aligned} H(\mathbf{X}_j^v) &= -\int_{\mathbb{X}_j^v} p(\mathbf{x}_j^v) \ \mathrm{ld} \ p(\mathbf{x}_j^v) \, d\mathbf{x}_j^v, \\ H(\mathbf{X}_j^v | Q_j) &= -\int_{\mathbb{X}_j^v} p(\mathbf{x}_j^v | Q_j) \ \mathrm{ld} \ p(\mathbf{x}_j^v | Q_j) \, d\mathbf{x}_j^v. \end{aligned}$$

Further processing by taking into account all possible lengths in the state Q_j yields the following

mutual information:

$$\begin{aligned} I(\mathbf{X}_j^V;Q_j) &= H(\mathbf{X}_j^V) - H(\mathbf{X}_j^V|Q_j) \\ &= \sum_{v=1}^{N_V} P(v|Q_j) \left[H(\mathbf{X}_j^v) - H(\mathbf{X}_j^v|Q_j) \right] \\ &= \sum_{v=1}^{N_V} P(v|Q_j)\, I(\mathbf{X}_j^v;Q_j), \end{aligned} \tag{D.2}$$

where the following equations have been used:

$$H(\mathbf{X}_j^V) = \sum_{v=1}^{N_V} P(v|Q_j)\, H(\mathbf{X}_j^v), \tag{D.3}$$

$$H(\mathbf{X}_j^V|Q_j) = \sum_{v=1}^{N_V} P(v|Q_j)\, H(\mathbf{X}_j^v|Q_j). \tag{D.4}$$

Finally, the mutual information for all states $H(\mathbf{X}^V|Q)$ is calculated as

$$\begin{aligned} I(\mathbf{X}^V;Q) &= H(\mathbf{X}^V) - H(\mathbf{X}^V|Q), \\ &= \sum_{j=1}^{N_Q} P(Q_j) \left[H(\mathbf{X}_j^V) - H(\mathbf{X}_j^V|Q_j) \right] \\ &= \sum_{j=1}^{N_Q} P(Q_j) \sum_{v=1}^{N_V} P(v|Q_j) \left[H(\mathbf{X}_j^v) - H(\mathbf{X}_j^v|Q_j) \right] \\ &= \sum_{j=1}^{N_Q} P(Q_j) \sum_{v=1}^{N_V} P(v|Q_j)\, I(\mathbf{X}_j^v;Q_j), \end{aligned} \tag{D.5}$$

where we have used the following definitions:

$$H(\mathbf{X}^V) = \sum_{j=1}^{N_Q} P(Q_j)\, H(\mathbf{X}_j^V),$$

$$H(\mathbf{X}^V|Q) = \sum_{j=1}^{N_Q} P(Q_j)\, H(\mathbf{X}_j^V|Q_j),$$

and $H(\mathbf{X}_j^V)$ and $H(\mathbf{X}_j^V|Q_j)$ are calulated following (D.3) and (D.4), respectively.

To evaluate (D.5) the term $I(\mathbf{X}_j^v; Q_j)$ is needed. The approximation for the mutual information is given, for example, by the monogram approximation. Using the monogram assumption as in (5.29) and assuming a temporal statistically independent sequence of feature vectors $\mathbf{X}_j^v$ we obtain

$$\begin{aligned} H(\mathbf{X}_j^v) &= v_j \cdot H(\mathbf{X}), \\ H(\mathbf{X}_j^v|Q_j) &= v_j \cdot H(\mathbf{X}|Q_j), \end{aligned}$$

where v_j denotes the length v observed in the subset $\{(\mathbf{x}^v,\ Q_j)\}$. The mutual information is thus shown as

$$\begin{aligned} I(\mathbf{X}^V; Q) &= \sum_{j=1}^{N_Q} P(Q_j) \sum_{v=1}^{N_V} P(v|Q_j) \cdot v_j \cdot \Big[\, H(\mathbf{X}) - H(\mathbf{X}|Q_j) \,\Big] \\ &= \sum_{j=1}^{N_Q} P(Q_j) \cdot \overline{v}_j \cdot I(\mathbf{X}; Q_j), \end{aligned} \tag{D.6}$$

where $\overline{v}_j = \sum_{v=1}^{N_V} P(v|Q_j)\ v_j$. The result shows that $I(\mathbf{X}^V; Q)$ is obtained by first multiplying the mutual information $I(\mathbf{X}; Q_j)$ gained from an independent realization of feature vector random variable $\mathbf{X}$ in a particular state Q_j with the corresponding average length of the feature sequence $\overline{v}_j$. If $\overline{v}_j$ is sufficiently high, the mutual information $I(\mathbf{X}^V; Q)$ will increase and is no longer bounded by $H(Q)$.

Bibliography

[Aalburg et al. 2002] Aalburg, S. ; Beaugeant, C. ; Stan, S. ; Fingscheidt, T. ; Balan, R. ; Rosca, J.: Single- and Two-Channel Noise Reduction for Robust Speech Recognition in Car. In: *Proc. ISCA Tutorial and Research Workshop on Multi-Modal Dialogue in Mobile Environments (ITRW IDS)*, June 2002

[Agarwal and Cheng 1999] Agarwal, A. ; Cheng, Y.M.: Two-Stage Mel-Warped Wiener Filter for Robust Speech Recognition. In: *Proc. International Workshop on Automatic Speech Recognition and Understanding (ASRU)*, December 1999, pp. 67–70

[Alexandre and Lockwood 1993] Alexandre, P. ; Lockwood, P.: Root Cepstral Analysis: A Unified View. Application to Speech Processing in Car Noise Environments. In: *Speech Communication* 12 (1993), pp. 277–288

[Andrassy et al. 2001] Andrassy, B. ; Vlaj, D. ; Beaugeant, C.: Recognition Performance of the Siemens Front-End with and without Frame Dropping on the Aurora 2 Database. In: *Proc. INTERSPEECH - European Conference on Speech Communication and Technology (EUROSPEECH)*, September 2001, pp. 193–196

[Astrov et al. 2003] Astrov, S. ; Bauer, J.G. ; Stan, S.: High Performance Speaker and Vocabulary Independent ASR Technology for Mobile Phones. In: *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)* vol. 2, April 2003, pp. 281–284

[Bahl et al. 1988] Bahl, L.R. ; Brown, P.F. ; de Souza, P.V. ; Mercer, R.L.: A New Algorithm for the Estimation of Hidden Markov Model Parameters. In: *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, April 1988, pp. 493–396

[Bahl et al. 1974] Bahl, L.R. ; Cocke, J. ; Jelinek, F. ; Raviv, J.: Optimal Decoding of Linear Codes for Minimizing Symbol Error Rate. In: *IEEE Transactions on Information Theory* IT-20(2) (1974), March, pp. 284–287

[Bauer 2001] Bauer, J.G.: *Diskriminative Methoden zur automatischen Spracherkennung für Telefon-Anwendungen*, Technische Universität München, PhD Thesis, 2001

[Baum et al. 1970] Baum, L.E. ; Petrie, T. ; Soules, G. ; Weiss, N.: A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains. In: *The Annals of Mathematical Statistics* 41 (1970), Nr. 1, pp. 164–171

[Beaugeant et al. 2002] Beaugeant, C. ; Gilg, V. ; Schönle, M. ; Jax, P. ; Martin, R.: Computationally Efficient Speech Enhancement Using RLS and Psycho-acoustic Motivated Algorithm. In: *Proc. World Multi-Conference on Systemics, Cybernetics and Informatics (SCI)*, July 2002

[Beaugeant and Scalart 2001] Beaugeant, C. ; Scalart, P.: Speech Enhancement Using a Minimum Least Square Amplitude Estimator. In: *Proc. International Workshop on Acoustic Echo and Noise Control (IWAENC)*, September 2001, pp. 191–194

[Berouti et al. 1979] Berouti, M. ; Schwartz, R. ; Makhoul, J.: Enhancement of Speech Corrupted by Acoustic Noise. In: *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, April 1979, pp. 208–211

[Berstein and Shallom 1991] Berstein, A.D. ; Shallom, I.D.: An Hypothesized Wiener Filtering Approach to Noisy Speech Recognition. In: *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)* vol. 2, April 1991, pp. 913–916

[Boll 1979] Boll, S.F.: Suppression of Acoustic Noise in Speech using Spectral Subtraction. In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 27 (1979), April, pp. 113–120

[Cappé 1994] Cappé, O.: Elimination of the Musical Noise Phenomenon with the Ephraim and Malah Noise Suppressor. In: *IEEE Transactions on Speech and Audio Processing* 2 (1994), April, pp. 345–349

[Chen et al. 2005] Chen, C.-P. ; Bilmes, J. ; Ellis, D.P.W.: Speech Feature Smoothing for Robust ASR. In: *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)* vol. I, March 2005, pp. I–(525–528)

[Chen et al. 2002] Chen, C.-P. ; Filali, K. ; Bilmes, J.A.: Frontend Post-Processing and Backend Model Enhancement on the Aurora 2.0/3.0 Databases. In: *Proc. INTERSPEECH - International Conference on Spoken Language Processing (ICSLP)*, September 2002, pp. 241–244

[Chu and Chueh 1966] Chu, J. T. ; Chueh, J. C.: Inequalities Between Information Measures and Error Probability. In: *Journal of the Franklin Institute* 282 (1966), August, pp. 121–125

[Cohen 2004a] Cohen, I.: On the Decision-Directed Estimation Approach of Ephraim and Malah. In: *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, May 2004, pp. 293–296

[Cohen 2004b] Cohen, I.: Speech Enhancement Using a Noncausal A Priori SNR Estimator. In: *IEEE Signal Processing Letters* 11 (2004), September, pp. 725–728

[Cover and Hart 1967] Cover, T.M. ; Hart, P.E.: Nearest Neighbor Pattern Classification. In: *IEEE Transactions on Information Theory* IT-13(1) (1967), pp. 21–27

[Davis and Mermelstein 1980] Davis, S.B. ; Mermelstein, P.: Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences. In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 28 (1980), August, pp. 357–366

[de Veth et al. 2001] de Veth, J. ; Mauuary, L. ; Noé, B. ; de Wet, F. ; Sienel, J. ; Boves, L. ; Jouvet, D.: Feature Vector Selection to Improve ASR Robustness in Noisy Conditions. In: *Proc. INTERSPEECH - European Conference on Speech Communication and Technology (EUROSPEECH)*, September 2001, pp. 201–204

[Dempster et al. 1977] Dempster, A. P. ; Laird, N.M. ; Rubin, D.B.: Maximum Likelihood from Incomplete Data via the EM Algorithm. In: *Journal of the Royal Statistical Society, Series B,* 39 (1977), Nr. 1, pp. 1–38

[Droppo et al. 2001] Droppo, J. ; Deng, L. ; Acero, A.: Evaluation of the SPLICE Algorithm on the Aurora2 Database. In: *Proc. INTERSPEECH - European Conference on Speech Communication and Technology (EUROSPEECH)*, September 2001

[Ephraim and Malah 1983] Ephraim, Y. ; Malah, D.: Speech Enhancement Using Optimal Non-Linear Spectral Amplitude Estimation. In: *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, April 1983, pp. 1118–1121

[Ephraim and Malah 1984] Ephraim, Y. ; Malah, D.: Speech Enhancement using a Minimum Mean-Square Error Short-Time Spectral Amplitude Estimator. In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 32 (1984), December, pp. 1109–1121

[Ephraim and Malah 1985] Ephraim, Y. ; Malah, D.: Speech Enhancement using a Minimum Mean-Square Error Log-Spectral Amplitude Estimator. In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 33 (1985), April, pp. 443–445

[ETSI STQ-Aurora 2000] ETSI STQ-Aurora (Org.): *Speech Processing, Transmission and Quality Aspects (STQ); Distributed Speech Recognition; Front-End Feature Extraction Algorithm; Compression Algorithms*. ETSI ES 201 108 V1.1.2. April 2000

[ETSI STQ-Aurora 2001a] ETSI STQ-Aurora (Org.): *Description and Baseline Results for the Subset of the Speechdat-Car German Database used for ETSI STQ Aurora WI008 Advanced DSR Front-end Evaluation*. AU/273/00 V1.1. January 2001

[ETSI STQ-Aurora 2001b] ETSI STQ-Aurora (Org.): *Speech Recognition Performance Comparison between AMR Speech Coding and the DSR Front-End (ETSI ES 201 108)*. AU/411/02 V1.1. January 2001

[ETSI STQ-Aurora 2002] ETSI STQ-Aurora (Org.): *Speech Recognition Performance Comparison between AMR Speech Coding and the Advanced DSR Front-End (ETSI ES 202 050)*. AU/410/02 V2.0. March 2002

[ETSI STQ-Aurora 2003a] ETSI STQ-Aurora (Org.): *Speech Processing, Transmission and Quality Aspects (STQ); Distributed Speech Recognition; Advanced Front-End Feature Extraction Algorithm; Compression Algorithms*. ETSI ES 202 050 V1.1.3. November 2003

[ETSI STQ-Aurora 2003b] ETSI STQ-Aurora (Org.): *Speech Processing, Transmission and Quality Aspects (STQ); Distributed Speech Recognition; Extended Advanced Front-End Feature Extraction Algorithm; Compression Algorithms; Back-End Speech Reconstruction Algorithm*. ETSI ES 202 212 V1.1.1. November 2003

[ETSI STQ-Aurora 2003c] ETSI STQ-Aurora (Org.): *Speech Processing, Transmission and Quality Aspects (STQ); Distributed Speech Recognition; Extended Front-End Feature Extraction Algorithm; Compression Algorithms; Back-End Speech Reconstruction Algorithm*. ETSI ES 202 211 V1.1.1. November 2003

[Fano 1961] Fano, R. M.: *Transmission of Information: A Statistical Theory of Communications*. London : The MIT Press and John Wiley & Sons, Inc. New York, 1961

[Feder and Merhav 1994] Feder, M. ; Merhav, N.: Relations Between Entropy and Error Probability. In: *IEEE Transactions on Information Theory* 40 (1994), January, Nr. 1, pp. 259–266

[Fingscheidt et al. 2005a] Fingscheidt, T. ; Beaugeant, C. ; Suhadi, S.: Overcoming the Statistical Independence Assumption w.r.t. Frequency in Speech Enhancement. In: *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, March 2005, pp. 1081–1084

[Fingscheidt et al. 2004] Fingscheidt, T. ; Setiawan, P. ; Stan, S.: Revisiting Some Model-Based and Data-Driven Denoising Algorithms in Aurora 2 Context. In: *Proc. Electronic Speech Signal Processing Conference (ESSP)*, September 2004

[Fingscheidt et al. 2005b] Fingscheidt, T. ; Setiawan, P. ; Stan, S.: Approaches to Robust Speech Recognition in Mobile Devices. In: *Proc. German Acoustical Society Conference (DAGA)*, March 2005

[Freeman et al. 2001] Freeman, P. E. ; Doe, S. ; Siemiginowska, A.: Sherpa: A Mission-Independent Data Analysis Application. In: *Proc. SPIE - The International Society for Optical Engineering* vol. 4477, 2001, pp. 76–87

[Fujimoto and Nakamura 2005] Fujimoto, M. ; Nakamura, S.: Particle Filter Based Non-Stationary Noise Tracking for Robust Speech Recognition. In: *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, March 2005, pp. 257–260

[Fukunaga 1990] Fukunaga, K.: *Introduction to Statistical Pattern Recognition, Second Edition*. Academic Press, 1990

[Furui 1981] Furui, S.: Cepstral Analysis Technique for Automatic Speaker Verification. In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 29 (1981), April, pp. 254–272

[Furui 1986] Furui, S.: Speaker-Independent Isolated Word Recognition Using Dynamic Features of Speech Spectrum. In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 34 (1986), February, pp. 52–59

[Furui and Lee 1995] Furui, S. ; Lee, C.: Robust Speech Recognition - An Overview. In: *IEEE International Workshop on Automatic Speech Recognition and Understanding (ASRU)* (1995), December, pp. 93

[Gales 1997] Gales, M.J.F.: Maximum Likelihood Linear Transformations for HMM-based Speech Recognition / Cambridge University Engineering Department. May 1997 (CUED/F-INFENG/TR 291). – Tech. Report. rev. January 1998

[Gales and Young 1996] Gales, M.J.F. ; Young, S.J.: Robust Continuous Speech Recognition Using Parallel Model Combination. In: *IEEE Transactions on Speech and Audio Processing* 4 (1996), September, pp. 352–359

[Gemello et al. 2004] Gemello, R. ; Mana, F. ; De Mori, R.: A Modified Ephraim-Malah Noise Suppression Rule for Automatic Speech Recognition. In: *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, May 2004, pp. 957–960

[Ghitza 1994] Ghitza, O.: Auditory Models and Human Performance in Tasks Related to Speech Coding and Speech Recognition. In: *IEEE Transactions on Speech and Audio Processing* 2 (1994), January, pp. 115–132

[Golić 1987] Golić, J.: On the Relationship Between the Information Measures and the Bayes Probability of Error. In: *IEEE Transactions on Information Theory* IT-33 (1987), September, Nr. 5, pp. 681–693

[Gong 1995] Gong, Yifan: Speech Recognition in Noisy Environments. In: *Speech Communication* 16 (1995), pp. 261–291

[Gray 1984] Gray, R. M.: Vector Quantization. In: *IEEE ASSP Magazine* (1984), pp. 4–29

[Grumm 1999] Grumm, D.: Optimizing Functions Using ASCFIT. In: *Astronomical Society of the Pacific (ASP) Conference* 172, Astronomical Data Analysis Software and Systems VIII (1999), pp. 365–368

[Grundlehner et al. 2005] Grundlehner, B. ; Lecocq, J. ; Balan, R. ; J.Rosca: Performance Assessment Method for Speech Enhancement Systems. In: *Proc. IEEE BENELUX Signal Processing Symposium - DSP Valleyś Annual Research & Technology Symposium (SPS-DARTS)*, April 2005

[Gustafsson 1999] Gustafsson, S.: *Enhancement of Audio Signals by Combined Acoustic Echo Cancellation and Noise Reduction*. Wissenschaftsverlag Mainz, 1999

[Haeb-Umbach and Ney 1992] Haeb-Umbach, R. ; Ney, H.: Linear Discriminant Analysis for Improved Large Vocabulary Continuous Speech Recognition. In: *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, March 1992, pp. 13–16

[Hansen and Clements 1991] Hansen, J.H.L. ; Clements, M.A.: Constrained Iterative Speech Enhancement with Application to Speech Recognition. In: *IEEE Transactions on Signal Processing* 39 (1991), June, pp. 795–805

[Hauenstein and Marschall 1995] Hauenstein, A. ; Marschall, E.: Methods for Improved Speech Recognition over Telephone Lines. In: *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, May 1995, pp. 425–428

[Haykin 2002] Haykin, S.: *Adaptive Filter Theory - Fourth Edition*. Upper Saddle River, New Jersey : Prentice-Hall, 2002

[Hellman and Raviv 1970] Hellman, M. E. ; Raviv, J.: Probability of Error, Equivocation, and the Chernoff Bound. In: *IEEE Transactions on Information Theory* IT-16 (1970), July, Nr. 4, pp. 368–372

[Hermansky 1990] Hermansky, H.: Perceptual Linear Predictive (PLP) Analysis of Speech. In: *Journal of the Acoustical Society of America* 87 (1990), Nr. 4, pp. 1738–1752

[Hermansky et al. 1993] Hermansky, H. ; Morgan, N. ; Hirsch, H.: Recognition of Speech in Additive and Convolutional Noise Based on RASTA Spectral Processing. In: *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, April 1993, pp. 83–86

[Hilger and Ney 2001] Hilger, F. ; Ney, H.: Quantile Based Histogram Equalization for Noise Robust Speech Recognition. In: *Proc. INTERSPEECH - European Conference on Speech Communication and Technology (EUROSPEECH)*, September 2001, pp. 1135–1138

[Hirsch and Pearce 2000] Hirsch, H.-G. ; Pearce, D.: The Aurora Experimental Framework for the Performance Evaluation of Speech Recognition Systems under Noisy Conditions. In: *Proc. ISCA Tutorial and Research Workshop on Automatic Speech Recognition (ITRW ASR)*, September 2000

[Höge 1984] Höge, H.: A Parametric Representation of Short-Time Power Spectra Based on the Acoustic Properties of the Ear. In: *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)* vol. 9, March 1984, pp. 49–51

[Höge 1999] Höge, H.: Estimating an upper Bound for the Error Rate for Speech Recognition using Entropy. In: *International Journal of Electronics and Communications* 53 (1999), Nr. 4, pp. 205–214

[Höge et al. 2004] Höge, H. ; Geißler, C. ; Setiawan, P. ; Steinert, K.: Evaluation of Microphone Array Front-Ends for ASR - an Extension of the AURORA Framework. In: *Proc. International Conference on Language Resources and Evaluation (LREC)*, May 2004

[Höge et al. 2008] Höge, H. ; Hohenner, S. ; Kämmerer, B. ; Kunstmann, N. ; Schachtl, S. ; Schönle, M. ; Setiawan, P.: Automotive Speech Recognition. In: Tan, Zheng-Hua (Ed.) ; Lindberg, B. (Ed.): *Automatic Speech Recognition on Mobile Devices and over Communication Networks (Advances in Pattern Recognition)*. Springer-Verlag, London, 2008, pp. 347–374

[Hu and Loizou 2008] Hu, Y. ; Loizou, P. C.: Evaluation of Objective Quality Measures for Speech Enhancement. In: *IEEE Transactions on Audio, Speech, and Language Processing* 16 (2008), January, pp. 229–238

[Iskra et al. 2002] Iskra, D. ; Grosskopf, B. ; Marasek, K. ; Huevel, H. van den ; Diehl, F. ; Kiessling, A.: SPEECON - Speech Data for Consumer Devices: Database Specification and Validation. In: *Proc. International Conference on Language Resources and Evaluation (LREC)*, May 2002

[ITU-T P.800 1996] ITU-T P.800 (Org.): *Methods for subjective determination of transmission quality*. ITU-T Recommendation P.800. August 1996

[ITU-T P.835 2003] ITU-T P.835 (Org.): *Subjective Test Methodology for Evaluating Speech Communication Systems That Include Noise Suppression Algorithm*. ITU-T Recommendation P.835. November 2003

[ITU-T P.862 2001] ITU-T P.862 (Org.): *Perceptual Evaluation of Speech Quality (PESQ): An Objective Method for End-to-End Speech Quality Assessment of Narrow-Band Telephone Networks and Speech Codecs*. ITU-T Recommendation P.862. February 2001

[ITU-T P.862.2 2007] ITU-T P.862.2 (Org.): *Wideband Extension to Recommendation P.862 for the Assessment of Wideband Telephone Networks and Speech Codecs*. ITU-T Recommendation P.862.2. November 2007

[Juang 1991] Juang, B.-H.: Speech Recognition in Adverse Environments. In: *Computer Speech and Language* 5 (1991), pp. 275–294

[Juang and Katagiri 1992] Juang, B.-H. ; Katagiri, S.: Discriminative Learning for Minimum Error Classification. In: *IEEE Transactions on Signal Processing* 40 (1992), December, Nr. 12, pp. 3043–3054

[Juang and Rabiner 1990] Juang, B.-H. ; Rabiner, L. R.: The Segmental K-Means Algorithm for Estimating Parameters of Hidden Markov Models. In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 38 (1990), September, Nr. 9, pp. 1639–1641

[Kim et al. 1999] Kim, Doh-Suk ; Lee, Soo-Young ; Kil, Rhee M.: Auditory Processing of Speech Signals for Robust Speech Recognition in Real-World Noisy Environments. In: *IEEE Transactions on Speech and Audio Processing* 7 (1999), January, pp. 55–69

[Kim 1998] Kim, N.S.: Statistical Linear Approximation for Environment Compensation. In: *IEEE Signal Processing Letters* 5 (1998), January, pp. 8–10

[Kim 2002] Kim, N.S.: Feature Domain Compensation of Nonstationary Noise for Robust Speech Recognition. In: *Speech Communication* 37 (2002), pp. 231–248

[Kobayashi and Imai 1984] Kobayashi, T. ; Imai, S.: Spectral Analysis Using Generalized Cepstrum. In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 32 (1984), October, pp. 1087–1089

[Kolda et al. 2003] Kolda, T. G. ; Lewis, R. M. ; Torczon, V.: Optimization by Direct Search: New Perspectives on Some Classical and Modern Methods. In: *Society for Industrial and Applied Mathematics (SIAM) Review* 45 (2003), Nr. 3, pp. 385–482

[Kullback and Leibler 1951] Kullback, S. ; Leibler, R. A.: On Information and Sufficiency. In: *Annals of Mathematical Statistics* 22 (1951), pp. 79–86

[Lewis et al. 2000] Lewis, R. M. ; Torczon, V. ; Trosset, M. W.: Direct Search Methods: Then and Now. In: *Journal of Computational and Applied Mathematics* 124 (2000), December, pp. 191–207

[Li et al. 2004] Li, Jin-Yu ; Liu, Bo ; Wang, Ren-Hua ; Dai, Li-Rong: A Complexity Reduction of ETSI Advanced Front-End for DSR. In: *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)* vol. I, May 2004, pp. I–(61–64)

[Lim 1978] Lim, J.S.: Evaluation of a Correlation Subtraction Method for Enhancing Speech Degraded by Additive White Noise. In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 26 (1978), October, pp. 471–472

[Lim 1979] Lim, J.S.: Spectral Root Homomorphic Deconvolution System. In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 27 (1979), June, pp. 223–233

[Lim and Oppenheim 1978] Lim, J.S. ; Oppenheim, A.V.: All-Pole Modeling of Degraded Speech. In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 26 (1978), June, pp. 197–210

[Lim and Oppenheim 1979] Lim, J.S. ; Oppenheim, A.V.: Enhancement and Bandwidth Compression of Noisy Speech. In: *Proc. IEEE* vol. 67, December 1979, pp. 1586–1604

[Linde et al. 1980] Linde, Y. ; Buzo, A. ; Gray, R. M.: An Algorithm for Vector Quantizer Design. In: *IEEE Transactions on Communications* (1980), January, pp. 702–710

[Liporace 1982] Liporace, L. A.: Maximum Likelihood Estimation for Multivariate Observations of Markov Sources. In: *IEEE Transactions on Information Theory* IT-28 (1982), September, Nr. 5, pp. 729–734

[Lippmann 1987] Lippmann, R. P.: An Introduction to Computing with Neural Nets. In: *IEEE ASSP Magazine* 4 (1987), April, pp. 4–22

[Liu et al. 2006] Liu, W.M. ; Jellyman, K.A. ; Mason, J.S.D. ; Evans, N.W.D.: Assessment of Objective Quality Measures for Speech Intelligibility Estimation. In: *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)* vol. 1, May 2006, pp. I–(1225–1228)

[Lockwood and Boudy 1992] Lockwood, P. ; Boudy, J.: Experiments with a Nonlinear Spectral Subtractor (NSS), Hidden Markov Models and the Projection for Robust Speech Recognition in Cars. In: *Speech Communication* 11 (1992), June, Nr. 2-3, pp. 215–228

[Macho and Cheng 2001] Macho, D. ; Cheng, Y.M.: SNR-Dependent Waveform Processing for Improving the Robustness of ASR Front-End. In: *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, May 2001, pp. 305–308

[Macho et al. 2002] Macho, D. ; Mauuary, L. ; Noé, B. ; Cheng, Y.M. ; Ealey, D. ; Jouvet, D. ; Kelleher, H. ; Pearce, D. ; Saadoun, F.: Evaluation of a Noise-Robust DSR Front-End on Aurora Databases. In: *Proc. INTERSPEECH - International Conference on Spoken Language Processing (ICSLP)*, September 2002, pp. 17–20

[Makhoul 1975] Makhoul, J.: Linear Prediction: A Tutorial Review. In: *Proc. IEEE* vol. 63, April 1975, pp. 561–580

[Martin 1994] Martin, R.: Spectral Subtraction Based on Minimum Statistics. In: *Proc. European Signal Processing Conference (EUSIPCO)*, September 1994, pp. 1182–1185

[Martin 2001] Martin, R.: Noise Power Spectral Density Estimation Based on Optimal Smoothing and Minimum Statistics. In: *IEEE Transactions on Speech and Audio Processing* 9 (2001), July, pp. 504–512

[Martin 2002] Martin, R.: Speech Enhancement Using MMSE Short Time Spectral Estimation with Gamma Distributed Speech Priors. In: *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)* vol. I, May 2002, pp. I–(253–256)

[Martin and Breithaupt 2003] Martin, R. ; Breithaupt, C.: Speech Enhancement in the DFT Domain Using Laplacian Speech Priors. In: *Proc. International Workshop on Acoustic Echo and Noise Control (IWAENC)*, September 2003, pp. 87–90

[Mauuary 1998] Mauuary, L.: Blind Equalization in the Cepstral Domain for Robust Telephone Based Speech Recognition. In: *Proc. European Signal Processing Conference (EUSIPCO)* vol. 1, September 1998, pp. 359–362

[McAulay and Malpass 1980] McAulay, R.J. ; Malpass, M.L.: Speech Enhancement using a Soft-Decision Noise Suppression Filter. In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 28 (1980), April, pp. 137–145

[Moddemeijer 1989] Moddemeijer, R.: On Estimation of Entropy and Mutual Information of Continuous Distributions. In: *Signal Processing* 16 (1989), Nr. 3, pp. 233–246

[Moddemeijer 1999] Moddemeijer, R.: A Statistic to Estimate the Variance of the Histogram Based Mutual Information Estimator Based on Dependent Pairs of Observations. In: *Signal Processing* 75 (1999), Nr. 1, pp. 51–63

[Moreno et al. 2000] Moreno, A. ; Lindberg, B. ; Draxler, C. ; Richard, G. ; Choukri, K. ; Allen, J. ; Euler, S.: SpeechDat-Car: A Large Speech Database for Automotive Environments. In: *Proc. International Conference on Language Resources and Evaluation (LREC)*, June 2000

[Moreno et al. 1996] Moreno, P.J. ; Raj, B. ; Stern, R.M.: A Vector Taylor Series Approach for Environment-Independent Speech Recognition. In: *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, May 1996, pp. 733–736

[Neal 1993] Neal, R.M.: Probabilistic Inference Using Markov Chain Monte Carlo Methods / Department of Computer Science, University of Toronto. September 1993 (CRG-TR-93-1). – Tech. Report

[Nilsson and Kleijn 2007] Nilsson, M. ; Kleijn, B.: Mutual Information and the Speech Signal. In: *Proc. INTERSPEECH - European Conference on Speech Communication and Technology (EUROSPEECH)*, August 2007, pp. 502–505

[Noé et al. 2001] Noé, B. ; Sienel, J. ; Jouvet, D. ; Mauuary, L. ; Boves, L. ; de Veth, J. ; de Wet, F.: Noise Reduction for Noise Robust Feature Extraction for Distributed Speech Recognition. In: *Proc. INTERSPEECH - European Conference on Speech Communication and Technology (EUROSPEECH)*, September 2001, pp. 433–436

[Papoulis 1991] Papoulis, A.: *Probability, Random Variables, and Stochastic Processes, Third Edition*. McGraw-Hill, Inc., 1991

[Pearce 2000] Pearce, D.: Enabling New Speech Driven Services for Mobile Devices: An Overview of the ETSI Standards Activities for Distributed Speech Recognition Front-Ends. In: *Proc. Applied Voice Input/Output Society Conference (AVIOS)*, May 2000

[Pearce and Hirsch 2000] Pearce, D. ; Hirsch, H.-G.: The Aurora Experimental Framework for the Performance Evaluation of Speech Recognition Systems under Noisy Conditions. In: *Proc. INTERSPEECH - International Conference on Spoken Language Processing (ICSLP)* vol. 4, October 2000, pp. 29–32

[Portnoff 1980] Portnoff, M.R.: Time-Frequency Representation of Digital Signals and Systems Based on Short-Time Fourier Analysis. In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 28 (1980), February, pp. 55–69

[Press et al. 1992] Press, W. H. ; Teukolsky, S. A. ; Vetterling, W. T. ; Flannery, B. P.: *Numerical Recipes in C: The Art of Scientific Computing, Second Edition*. Cambridge University Press, 1992

[Rabiner et al. 1978] Rabiner, L. R. ; Rosenberg, A. E. ; Levinson, S. E.: Considerations in Dynamic Time Warping Algorithms for Discrete Word Recognition. In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 26 (1978), December, pp. 575–582

[Rabiner 1989] Rabiner, L.R.: A Tutorial on Hidden Markov Model and Selected Applications in Speech Recognition. In: *Proc. IEEE* vol. 77, February 1989, pp. 257–286

[Rabiner and Juang 1993] Rabiner, L.R. ; Juang, B.-H.: *Fundamentals of Speech Recognition*. Englewood Cliffs, New Jersey : Prentice-Hall, 1993

[Ramabadran et al. 2004] Ramabadran, T. ; Sorin, A. ; McLaughlin, M. ; Chazan, D. ; Pearce, D. ; Hoory, R.: The ETSI Extended Distributed Speech Recognition (DSR) Standards: Server-Side Speech Reconstruction. In: *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)* vol. I, May 2004, pp. I–(53–56)

[Sarikaya and Hansen 2001] Sarikaya, R. ; Hansen, J.H.L.: Analysis of the Root-Cepstrum for Acoustic Modeling and Fast Decoding in Speech Recognition. In: *Proc. INTERSPEECH - European Conference on Speech Communication and Technology (EUROSPEECH)*, September 2001, pp. 687–690

[Scalart and Vieira Filho 1996] Scalart, P. ; Vieira Filho, J.: Speech Enhancement Based on A Priori Signal to Noise Estimation. In: *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, May 1996, pp. 629–632

[Setiawan 2004] Setiawan, P.: Microphone Array Front-Ends for ASR - Proposing for a Standardization. In: *Proc. International Workshop on Advances in Speech Technology (AST)*, July 2004

[Setiawan et al. 2003a] Setiawan, P. ; Aalburg, S. ; Fingscheidt, T. ; Stan, S. ; Ruske, G.: A Text-Independent Speaker Verification Approach for Mobile Devices. In: *Proc. Electronic Speech Signal Processing Conference (ESSP)*, September 2003

[Setiawan et al. 2005a] Setiawan, P. ; Beaugeant, C. ; Fingscheidt, T. ; Stan, S.: Least-Squares Weighting Rules Formulations in the Frequency Domain. In: *Proc. Electronic Speech Signal Processing Conference (ESSP)*, September 2005

[Setiawan et al. 2005b] Setiawan, P. ; Fingscheidt, T. ; Höge, H.: Noise Reduction Approaches in Mobile Devices for Robust Speech Recognition in Car Noise. In: *Proc. International Workshop on Advances in Speech Technology (AST)*, June 2005

[Setiawan et al. 2003b] Setiawan, P. ; Fingscheidt, T. ; Stan, S. ; Höge, H.: On Robustness to Speech-Level Variations for Speech Enabled Services. In: *Proc. German Pattern Recognition Society (DAGM) Speech Processing Workshop*, September 2003

[Setiawan et al. 2008] Setiawan, P. ; Schandl, S. ; Taddei, H. ; Wan, H. ; Dai, J. ; Zhang, L. ; Zhang, D. ; Zhang, J. ; Shlomot, E.: On the ITU-T G.729.1 Silence Compression Scheme. In: *Proc. European Signal Processing Conference (EUSIPCO)*, August 2008

[Setiawan et al. 2004] Setiawan, P. ; Stan, S. ; Fingscheidt, T.: Revisiting Some Model-Based and Data-Driven Denoising Algorithms in Aurora 2 Context. In: *Proc. INTERSPEECH - International Conference on Spoken Language Processing (ICSLP)*, October 2004

[Setiawan et al. 2005c] Setiawan, P. ; Suhadi, S. ; Fingscheidt, T. ; Stan, S.: Robust Speech Recognition for Mobile Devices in Car Noise. In: *Proc. INTERSPEECH - European Conference on Speech Communication and Technology (EUROSPEECH)*, September 2005

[Shannon 1948] Shannon, C. E.: A Mathematical Theory of Communication. In: *Bell System Technical Journal* 27 (1948), July and October, pp. 379–423 and 623–656

[Siemens 2000] Siemens (Org.): *The Siemens Feature Extraction Module SFEM for Speech Recognition*. V1.2. October 2000

[Siemund et al. 2000] Siemund, R. ; Höge, H. ; Kunzmann, S. ; Marasek, K.: SPEECON - Speech Data for Consumer Devices. In: *Proc. International Conference on Language Resources and Evaluation (LREC)*, June 2000

[Sim et al. 1998] Sim, B.L. ; Tong, Y.C. ; Chang, J.S. ; Tan, C.T.: A Parametric Formulation of the Generalized Spectral Subtraction Method. In: *IEEE Transactions on Speech and Audio Processing* 6 (1998), July, pp. 328–337

[Sorin et al. 2004] Sorin, A. ; Ramabadran, T. ; Chazan, D. ; Hoory, R. ; McLaughlin, M. ; Pearce, D. ; Wang, F. ; Zhang, Y.: The ETSI Extended Distributed Speech Recognition (DSR) Standards: Client Side Processing and Tonal Languange Recognition Evaluation. In: *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)* vol. I, May 2004, pp. I–(129–132)

[Stern et al. 1996] Stern, R.M. ; Acero, A. ; Liu, Fu-Hua ; Ohshima, Y.: Signal Processing for Robust Speech Recognition. In: Lee, Chin-Hui (Ed.) ; Soong, Frank K. (Ed.) ; Paliwal, K.K. (Ed.): *Automatic Speech and Speaker Recognition: Advanced Topics*. Kluwer Academic Publ., Boston, 1996, pp. 351–378

[Stockham, Jr. 1966] Stockham, Jr., T.G.: High-Speed Convolution and Correlation. In: *Proc. American Federation of Information Processing Societies (AFIPS) Spring Joint Computer Conference*, 1966, pp. 229–233

[van Trees 1968] van Trees, H.L.: *Detection, Estimation, and Modulation Theory, Part I*. John Wiley & Sons, Inc., 1968

[Varga et al. 2002] Varga, I. ; Aalburg, S. ; Andrassy, B. ; Bauer, J.G. ; Beaugeant, C. ; Geißler, C. ; Höge, H.: ASR in Mobile Phones - An Industrial Approach. In: *IEEE Transactions on Speech and Audio Processing* 10 (2002), November, pp. 562–569

[Viterbi 1967] Viterbi, A.J.: Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm. In: *IEEE Transactions on Information Theory* IT-13 (1967), April, Nr. 2, pp. 260–269

[Wang and Lim 1982] Wang, D.L. ; Lim, J.S.: The Unimportance of Phase in Speech Enhancement. In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 30 (1982), August, pp. 679–681

[Wolfe and Godsill 2001] Wolfe, P.J. ; Godsill, S.J.: Simple Alternatives to the Ephraim and Malah Suppression Rule for Speech Enhancement. In: *Proc. IEEE Workshop on Statistical Signal Processing*, 2001, pp. 496–499

[Wu et al. 2005] Wu, J. ; Huo, Q. ; Zhu, D.: An Environment Compensated Maximum Likelihood Training Approach Based on Stochastic Vector Mapping. In: *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, March 2005, pp. 429–432

[Yang 1993] Yang, Jin: Frequency Domain Noise Suppression Approaches in Mobile Telephone Systems. In: *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, April 1993, pp. 363–366

[Yapanel et al. 2001] Yapanel, U. ; Hansen, J.H.L. ; Sarikaya, R. ; Pellom, B.: Robust Digit Recognition in Noise: An Evaluation Using the AURORA Corpus. In: *Proc. INTERSPEECH - European Conference on Speech Communication and Technology (EUROSPEECH)*, September 2001, pp. 905–908

[Young et al. 2005] Young, S. ; Evermann, G. ; Gales, M. ; Hain, T. ; Kershaw, D. ; Moore, G. ; Odell, J. ; Ollason, D. ; Povey, D. ; Valtchev, V. ; Woodland, P.: *The HTK Book (for HTK Version 3.3)*. `http://htk.eng.cam.ac.uk` : Cambridge University Engineering Department (CUED), April 2005

[Young and Chase 1998] Young, S.J. ; Chase, L.L.: Speech Recognition Evaluation: A Review of the US CSR and LVCSR Programmes. In: *Computer Speech and Language* 12 (1998), Nr. 4, pp. 263–279