Classification of Sound Environments for Hearing Aid Applications

Christine Oldenborg Voetmann

Kongens Lyngby 2012
IMM-MScEng-2012-43

Technical University of Denmark
Informatics and Mathematical Modelling
Building 321, DK-2800 Kongens Lyngby, Denmark
Phone +45 45253351, Fax +45 45882673
reception@imm.dtu.dk
www.imm.dtu.dk IMM-MScEng-2012-43

Abstract

The goal of this project is to create a Matlab based framework for sound environment classification including an investigation of the most robust features for classification of various sound environments.

Hearing aids use different amplification strategies targeted at different situations/sound environments. The different amplification strategies are normally chosen by the user with a remote control or via a program switch mounted on the hearing aid. Modern hearing aids contain various detectors which are used to automatically change a number of parameters of the hearing aid. The detectors are typically not fully descriptive for the sound environment. This project is seeking to improve the classification of the various sound environments relevant for the hearing aid user and focus is on two classes; car environment against miscellaneous environments.

The final framework is build up by a number of sound files covering the different sound environments, a list of features is configured from the openSMILE toolkit [12] and a classification tree is used as the classifying algorithm. By using the build robust framework a sensitivity of $91.6\% \pm 4.69\%$ and a specificity of $96.44\% \pm 3.13\%$ is achieved, but an expansion of the framework is recommended before an implementation in a hearing aid.

Resumé

Formålet med dette speciale er at skabe et Matlab baseret framework til lydmiljø klassifikation og herunder undersøge de mest robuste features til klassifikation af forskellige lydmiljøer.

Høreapparater bruger forskellige strategier til forstærkning af forskellige situationer/lydmiljøer. Normalt vælges denne forstærkning af brugeren med en fjernbetjening eller ved at skifte program på en knap på høreapparatet. Moderne høreapparater indeholder forskellige detektorer der bruges til automatisk at skifte mellem en række parametre i høreapparatet. Disse detektorer beskriver typisk ikke lydmiljøer fyldestgørende. Dette projekt søger at forbedre klassifikationen af de forskellige lydmiljøer der er relevante for en høreapparatsbruger, og har fokus på to klasser; bil miljø mod diverse andre lydmiljøer.

Det endelige framework er opbygget af et antal lydfiler der dækker de pågældende lydmiljøer, en liste af features konfigureret fra openSMILE værktøjet [12] og et klassifikationstræ benyttes som klassifikations algoritme. Ved at benytte det opbyggede robuste framework, opnås en sensitivitet på $91.6\% \pm 4.69\%$ og en specificitet på $96.44\% \pm 3.13\%$, men en udvidelse af frameworket anbefales inden en implementering i et høreapparat.

Preface

This thesis was prepared at the department of Informatics and Mathematical Modelling at the Technical University of Denmark in partial fulfilment of the requirements for acquiring the M.Sc. in Biomedical Engineering. The project was conducted in cooperation with Oticon A/S in the period from September 5th, 2011 to June 5th, 2012. The development of the framework was all done at the facilities of Oticon A/S in Smørum. The workload corresponded to 30 ECTS points.

The work has been supervised by:

- Associate Professor Jan Larsen, Department of Informatics and Mathematical Modelling
- Project Manager, Thomas Kaulberg, Embedded Systems department at Oticon A/S
- DSP Development Engineer, Sigurdur Sigurdsson, Embedded Systems department at Oticon A/S

Kgs. Lyngby, June 5th 2012

Christine Oldenborg Voetmann

Acknowledgements

I would like to thank Jan Larsen and Thomas Kaulberg for their supervision and many great ideas, for the support and great discussions. My deepest appreciation goes to Sigurdur Sigurdsson who has provided help and extensive support through the entire project, without this, the project would not have been taken to the same level. In addition, I would like to thank Dorthe Hofman-Bang at Oticon A/S for great discussions of what a hearing aid user asks for. A thanks goes to Oticon A/S for giving me the opportunity to do my Master thesis co-operating with them.

Finally, I would like to thank my family for their great support through the project period.

Nomenclature

P_{CC}	probability of correct classification
ACF	Auto correlation function
BM	Basilar membrane
BTE	Behind-the-ear
CF	Characteristic frequency
CGAV	spectral center of gravity
CGFS	fluctuations of the spectral center of gravity
CIC	Completely-in-the-canal
CS	compressed sensing
dB	decibel
FA	false alarm rate
FN	False negative
FP	False positive
GA	genetic algorithm
GMM	Gaussian mixture model
HATS	head and torso simulator
HMM	hidden Markov model
HR	hit rate

ICA	independent component analysis
IFT	inverse Fourier transform
ITC	In-the-canal
ITE	In-the-ear
k-NN	k-nearest neighbour
kHZ	kilohertz
LPC	linear prediction coefficients
MFCC	Mel-frequency cepstral coefficient
misc	miscellaneous
OH	overall hit rate
RITE	Receiver in the ear
rms	root mean square
SBS	sequential backward search
SFS	sequential forward search
SNR	signal-to-noise ratio
SPL	sound pressure level
TN	True negative
TP	True positive
ZCR	zero-crossing rate

Contents

Chapter 1

Introduction

1.1 Motivation

Hearing loss is a big problem in today's society. Many with a hearing impairment still have issues when it comes to the hearing aids on the market today, a great number of all hearing aids end in a drawer without being used [15]. It is believed that bad overall benefit is partly associated with poor selection of program modes for different situations. User satisfaction with hearing aids is investigated in this work and it is seen that an automatic program selection is found to be a valuable and desirable function appreciated by the user even if its performance is not perfect. This has led to many studies trying to find a way to satisfy the hearing aid users and has also motivated this work.

Difficult sound environments are of as much importance than all other sound environments and the more sound environments a hearing aid can automatically detect, the more satisfied a user will hopefully be. This has led to the focus in this study where classification of car environment versus miscellaneous environments is explored.

1.2 Project aim

The goal of this project is to create a Matlab based framework for sound environment classification and to investigate the most robust features for classification of various sound environments.

Scope:

- A list of sound environments must be chosen
- A number of sound recordings covering the different sound environments must be generated
- A list of features to investigate must be chosen
- A classification method must be chosen
- A Matlab based framework for the analysis must be created,

Specifications:

- The Framework must be easy to extend, both when it comes to sound environments and features
- The Framework must provide means to optimize performance of the classification
- The Framework must provide analysis of the classification to indicate the robustness the classification

1.3 Structure

In Chapter 2 background information is given on the ear, hearing loss, hearing aids and user satisfaction. Chapter 3 includes the state of the art and in Chapter 4 a data description is provided. Chapter 5 provides a technical description of the relevant features and the classifier used in this work followed by Chapter 6 which describes the classification system and used performance measures. Results from all the tests the framework has been put through can be seen in Chapter 7 and finally a conclusion is provided in Chapter 8.

Chapter 2

Background

Basic knowledge about the human ear is important in order to understand hearing loss. The anatomy of the ear, concepts related to hearing and the two main types of hearing loss are presented in this chapter. The most common types of hearing aids are introduced along with a description of the user's opinion of the need of hearing aids.

2.1 The Ear and the Auditory System

The organs of hearing, the ears, are made up of three main parts; the outer ears, the middle ears and the inner ears. The anatomy of the ear can be seen in Figure 2.1. The inner ear functions in both hearing and balance, whereas the outer and middle ear only is involved in hearing. The outer ear consists of the pinna and the external ear canal. The pinna modifies the incoming sound and is important in the ability of localizing sounds. Acoustic signals reach the outer ear as sound waves and are conducted through the external ear canal towards the tympanic membrane. The tympanic membrane, or eardrum, is a thin membrane that forms an airtight barrier between the outer and the middle ear. Sound waves reaching the tympanic membrane, through the external ear canal, cause it to vibrate about its equilibrium point in time with the sound pressure waves.

The middle ear is an air filled cavity containing three tiny bones, the auditory ossicles; the malleus (hammer), the incus (anvil) and the stapes (stirrup). They

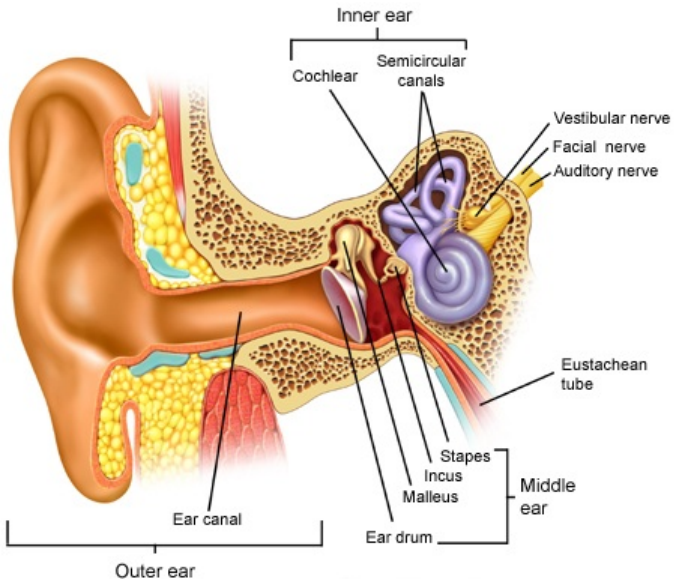


Figure 2.1: Anatomy of the ear [1].

transmit and amplify the vibrations from the tympanic membrane to the cochlea in the inner ear through the oval window, one of two covered openings of the middle ear separating it from the inner ear. The vibration of the tympanic membrane causes vibration of all three ossicles and this transfers the vibration to the oval window. Size difference between the tympanic membrane and the oval window results in about a 20-fold amplification of the vibration when crossing the middle ear. Amplification is required to cause adequate vibration in the liquid of the inner ear. The middle ear improves sound transmission and reduces the amount of reflected sound.

The inner ear consists on one side of the cochlea, and on the other side of the balance organ, which is not important for hearing, see Figure 2.1. The cochlea is the part of the inner ear that is stimulated by sound. In short terms, the cochlea transforms the mechanical vibrations into electrical nerve impulses that travel via the auditory nerve to the brain, where they form the actual impression of sound. The cochlea, which is shaped like a spiral shell of of snail, has liquid filled canals and cavities with bony rigid walls. Along its way, two membranes divide it, the vestibular membrane and the basilar membrane (BM).

The cochlea starts at the point where the oval window is situated, this is known as the base while the other end, the inner tip, is known as the apex. At the apex

there is a small opening called the helicotrema between the BM and the walls of the cochlea. Vibrations of the fluid in the cochlea are transmitted through the vestibular membrane which cause distortion of the basilar membrane. These distortions, together with weaker waves coming through the helicotrema, cause waves in the scala tympani fluid and result in the vibration of the membrane of the round window. When the oval window is set in motion, the BM is moving because of a pressure difference that is applied across the membrane.

Sounds of different frequencies strongly affect the displacement of the BM by its mechanical properties, which vary from base to apex. At the base it is narrow and stiff while it is wider and much less stiff at the apex. This cause sounds with high-frequencies to produce maximum displacement of the BM near the base with little movement of the remainder of the membrane. Low-frequency sounds, on the other hand, produce displacement all along the BM but reaches its maximum closer to the apex. The BM movement results in a frequency to place mapping where each place on the BM gives a maximum displacement to a different frequency called the characteristic frequency (CF). BM displacement translates mechanical movement to neural activity through movement of the outer hair cells. The cochlea contains approximateliy 12000 outer hair cells and approximately 3500 inner hair cells placed along the cochlea from the base to the apex [22]. Outer hair cells are related to the BM mechanical properties. Each inner hair cell is connected to several neurons in the main auditory nerve, and the inner hair cell microvilli are bent as they move against the tectorial membrane. Higher amplitude of the BM movement generates a higher firing rate in the neurons. This section is based on inspiration from [19] and [30].

The process of sound transduction is summarised in Figure 2.2. Here the pathway of conversion of sound energy into a neural signal that is interpreted by the brain as sound perception is shown. The sound waves travel through the various parts of the ear and the conversion of waves into mechanical signals lead to action potentials in the auditory nerve which finally result in an interpretation in the brain and hearing occurs.

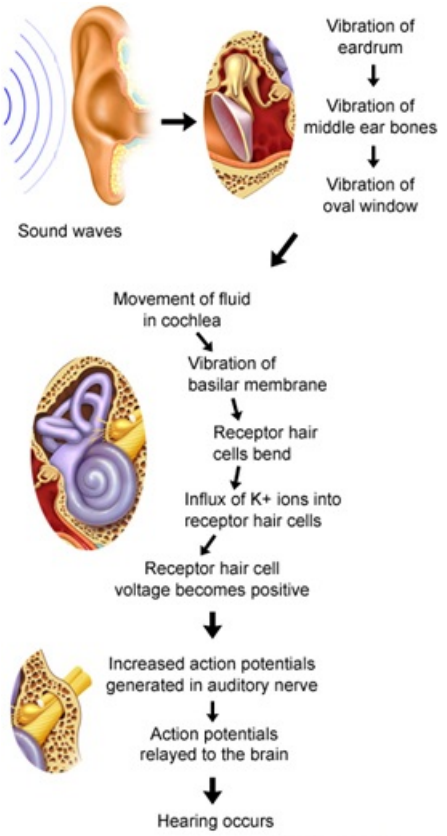


Figure 2.2: Sound transduction from the conversion of sound energy into a neural signal [1].

2.2 Hearing Loss

There are two main categories used for the type of hearing loss that can occur: conductive and sensorineural. They can appear isolatedly or simultaneously [19].

Conductive hearing loss occurs when there is a defect outside the cochlea, usually in the middle ear, and this reduces the transmission of sound to the cochlea. The cochlea itself and the neuronal pathways for hearing function normally. Causes for conductive loss can be infections of the middle ear (otitis media),

growth of bone over the oval window (ostosclerosis), injuries to the bones in the middle ear, abnormalities at the eardrum or wax in the ear canal. A conductive loss causes a non-normal attenuation of the incoming sound, soft sounds are no longer audible and intense sounds are reduced in loudness. This attenuation is thus frequency dependent and linear and can usually be compensated for with a simple hearing aid because the amplified sound waves it produces may provide normal stimulation to the cochlear once the blockage has been passed. Surgical treatment can be effective if the degree of hearing loss justifies this. This type of hearing loss can be accounted for up to 10% of all hearing losses [29].

The term sensorineural hearing loss is used when the hearing loss arises from a defect in the cochlea, in the auditory nerve or in higher centres in the auditory system. Sound waves are transmitted normally to the cochlea, but the ability to respond to the sound waves is impaired. Hearing loss arising from a defect in the cochlea is known as a cochlear loss and includes damage to the inner and outer hair cells whereas, if neural disturbances occurs at a higher point in the auditory pathway than the cochlear, it is known as retrocochlear loss. Acoustic trauma, drugs or infections can cause a cochlear sensorineural hearing loss [22]. It is usually permanent, complicated to compensate for with a hearing aid and cannot be treated by surgery. Even though a hearing aid has difficulties compensating for the hearing loss, they are often used to amplify sound waves by applying more gain to soft sounds and less gain to loud sounds, helping to overcome the altered loudness perception of reduced sound volume and sound clarity. Sensorineural hearing loss can be accounted for up to 90 % of all hearing losses [29].

Hearing loss due to ageing is the most common and is called presbyacusis. In the elderly, the extent of loss increases with frequency and a slowly growing permanent damage to the hair cells is the cause. The National Institute of Deafness and Other Communication Disorders claims that about 30-35 percent of adults between the ages of 65 and 75 years have a hearing loss. It is estimated that 40-50 percent of people at age 75 and older have a hearing loss [22].

If conductive and sensorineural components appear simultaneously, the hearing loss is called a mixed hearing loss. This includes damages to the outer or middle ear and the cochlea or sensory nerve or all at the same time. A central hearing loss may also occur, but there is currently no treatment available for this type of hearing loss, why this will only briefly be mentioned here. This type of hearing loss is caused by a disorder in the central auditory nervous system and usually manifest itself in poor word recognition and speech reception. This type of hearing loss is rare and is usually caused by a tumor or other changes in the neural structure [29].

2.3 Hearing Aids

There are four types of hearing aids that are the most common, these are listed below.

- Completely-in-the-canal (CIC)
- In-the-canal (ITC)
- In-the-ear (ITE)
- Behind-the ear (BTE)

The BTE has the largest physical size, is the oldest of the styles and comes is in different variants. These include one with standard tubing and custom earmold, one with a thin tube and a dome or one with a receiver in the ear (RITE) . Five of the different styles can be seen in Figure 2.3.

Figure 2.3: Different hearing aid styles. From left to right is the CIC, ITC, ITE, BTE, one with a thin tube and one with a receiver in the ear (RITE). Figure from [2].

Each style has its advantages and disadvantages, but since progress in technology has made it possible to reduce the size of the hearing aid components, especially the smaller styles have become popular since they can be hidden in the ear. But since they are blocking up the ear canal, they usually have a built in vent to prevent the occlusion effect where you hear both the sound waves carried through the air and sound transmitted from the bones of the scull, e.g. from chewing and breathing [29]. Here the BTE with receiver in the ear leave the ear canal open, called open fitting, which is an advantage because of the wearing comfort and no occlusion occurs.

2.4 User Satisfaction with Hearing Aids

Nearly all hearing aid users in the western world wear digital hearing aids. In many studies the user satisfaction with hearing aids have been tracked, e.g. the MarkeTrak study conducted in America since 1991 where hearing aid users have participated [15]. This study is an ongoing study that is repeated with a couple of years interval to track the trends of the hearing aid market and the users.

The hearing loss population is increasing along with the binaural rates while the average age of hearing aids has dropped. New technology improves the hearing aids all the time and this is one of the reasons why the average age of hearing aids has dropped. In the mentioned MarkeTrak study, the top ten factors related to overall customer satisfaction was registered [15]:

1. Overall Benefit (71 %)
2. Clarity of Sound (70 %)
3. Value (performance of the hearing aid relative to price) (68 %)
4. Natural Sounding (66 %)
5. Reliability of the Hearing Aid (65 %)
6. Richness or Fidelity of Sound (65 %)
7. Use in Noisy Situations (63 %)
8. Ability to Hear in Small Groups (63 %)
9. Comfort with Loud Sounds (60 %)
10. Sound of Voice (occlusion) (60 %)

The intensity of satisfaction is important to the user. The more passionate they are about their hearing aid experience, the more likely they are to wear them, recommend them to their friends and develop brand loyalty. All three are elements that, along with the perception of benefit, are very important when it comes to the utility of hearing aids. An important part of the experience is the ability to choose different settings in different listening situations.

A possible improvement of the utility of hearing aids is an automatic switching mode that can automatically sense the current acoustic situation and automatically switch to the best mode. Nowadays the user can select between several

modes for different situations, but this requires that the user recognises the acoustic environment and then switch to the best mode using a switch on the hearing aid or a remote control. In [9], a study was conducted where hearing impaired subjects tested the usefulness, acceptance and problems of an automatic program selection mode in a hearing aid from the users point of view. 63 subjects tested if the automatic program switch mode of the test instrument changed between modes in the desired way and if the switching was found to be helpful. It was found that adjusting for individual preferences in the frequency of switching mode could be useful, mostly the programs switched expectedly to a program that matched the situation quite well and 75% of the test subjects found the automatic system to be "quite useful" or "very useful" why an automatic program selection was found to be a valuable and desirable function appreciated by the user even if its performance is not perfect. This has highly motivated the work of others along with the work in this study. Focus has mainly been on recognising speech, speech in noise and music, which is seen clearly in the following state of the art chapter.

Chapter 3

State of the Art

Classification of sound environments is a topic of interest in many contexts, especially for the hearing aid companies. Some classification already occurs in the hearing aids on the market, but some sound environments have been found difficult to classify. Some of these difficult environments are interesting to the hearing aid user, since most users sees it as an advantage if the hearing aid can readjust to the desired settings for the certain environment. Therefore, developing an automatic classification algorithm expanded with more environments, even the difficult ones, is desired. The get an overview of the newest research in the field, this project was initiated with a literature study. It turned out that little has been published regarding classification of sound environments, but methods applicable for this field has been used in other contexts, why focus in the literature study has been on these methods. It has been explored what features seem to be of most use, what classifiers are most common and how this affects the classification rates in the fields of investigation.

Different articles and reports on the subject will be presented, while trying to follow a common structure in each of the presentations. The summaries will therefore cover the following points when possible:

- Data Description
- The Method
- The Results
- Other Remarks of Relevance

3.1 The Quest of Environmental Sound Classification

Many hearing aid users would find it helpful if they should not themselves change settings of their aids going from one listening environment to another. This consumer wish has led to a research field with many different and interesting approaches, all trying to get a robust classification of predefined environments (classes) at a low computational cost in order to permit an implementation in future hearing aids. Following is a number of studies focusing on this particular problem. All have a common approach, trying to find appropriate features along with a more or less simple classifier, but they still all differ in their choices of both features and classifiers. Without a common standard of features and classifiers, there is still room for improvement within the field, but the best points from each study are taken into account in the work of this project.

3.1.1 Sound Classification in Hearing Aids Inspired by Auditory Scene Analysis

Authors: M. Büchler, S. Allegro, S. Launer, and N. Dillier, ENT Department, University Hospital Zurich, Zurich, Switzerland and Phonak AG, Staefa, Switzerland, 2005 [9]

The purpose of this study is to find appropriate features for a sound classification system for the automatic recognition of the acoustic environment. The features are chosen as a combination of well-known auditory grouping features with features that are inspired by auditory scene analysis. These are evaluated with different types of pattern classifiers. The goal was to find a combination of features and classifier that gives a good hit rate for reasonable computational effort [9].

Data Description: A sound database was generated and used for evaluation. This contained four different sound classes: speech, speech in noise, noise and music. The database contains 300 real-world sounds of 30-second length each, sampled at 22 kHz/16 bit. The sounds were either recorded in the real world or in a sound proof room or taken from other media. The database has the following distribution of the four classes; 60 speech signals, 80 speech in noise, 80 noise and 80 music. Speech in noise signals contains speech signals mixed with noise signals at a signal-to-noise ratio (SNR) between +2 and -9 dB. Each of the signals were manually labelled with the one of the four classes they belong to .

Method: A combination of 11 auditory features (2 from amplitude modulation, 2 from spectral profile, 2 from harmonicity and 5 from amplitude onsets) and 6 classifiers (rule-based, minimum distance, Bayes, neural network, hidden Markov models (HMM) and a two-stage classifier (best HMM and rule-based)) were to be tested. Not all combinations of the features were chosen for the evaluation since this would have provided about 2^{14} different feature sets. Therefore an iterative strategy was developed heuristically to find the best feature set by trying to combine features that describe different attributes of the signal. Each of the about 30 sets of features then was processed for each classifier in order to find the optimal combination. Classification was calculated once per second for each of the sounds (resulting in 30 calculations per sound), and the output for a given sound was taken as the class that occurred most frequently. 80 % of the sounds were used for training of the classifier and tested on the remaining 20 %. The test/training split was chosen at random and repeated 100 times so tha actual score was the mean of these 100 cycles.

Results: In Figure 3.1 the results for the six classifiers with the best parameter and feature set can be seen. It is seen that the simpler approaches reach a hit rate of around 80 % which, with the more complicated systems can be approved to around 90 %. The features in the best feature sets that are not exactly self-explanatory are: m1, m2, m3 which are values for the modulation depth of three modulation frequency ranges; CGFS is fluctuations of the spectral center of gravity and describes dynamic properties of the spectral profile and CGAV is the spectral center of gravity which is a static characterization of the spectral profile [9].

It seems that the proper decision to make about what set of features gives the best result depends on what classifier is chosen. This is important to have in mind when a feature set is chosen in this project. An investigation of several features seems to be recommendable.

3.1.2 An Efficient Robust Sound Classification Algorithm for Hearing Aids

Authors: P. Nordqvist and A. Leijon, Department of Speech, Music and Hearing, Royal Institute of Technology, Stockholm, Sweden, 2004 [23]

The purpose of this study is, by an efficient robust sound classification algorithm, to enable a hearing aid to automatically change its behaviour for different listening environments according to the user's preferences [23]. The aim is to make the classification robust and insensitive to changes within one listening, since the user moves around and focus are mainly on the classes speech in quiet and

Classifier type, best parameters	Best feature set	Training set score (%)	Test set score (%)							
		Overall hit rate	Overall hit rate	Speech		Speech in noise		Noise		Music	
				Hit	False	Hit	False	Hit	False	Hit	False
Rule-based	*Tonality, pitch variance, m1, m2, m3*	—	78	79	1.3	67	9.9	88	11.8	78	7.0
Minimum distance, Euclidean	*Tonality, pitch variance, m1, m2, m3, CGFS, onsetv, beat*	84	83	86	3.5	83	11.2	80	7.0	85	1.0
Bayes, 15 intervals	*Tonality, pitch variance, m1, m2, m3, CGFS, onsetm, onsetc*	86	85	90	4.3	83	10.1	84	5.0	82	1.5
Two-layer perceptron, 8 hidden nodes	*Tonality, width, CGAV, CGFS, onsetc, beat*	89	87	86	1.7	86	7.0	89	5.9	87	2.8
Ergodic HMM, 2 states	*Tonality, width, CGAV, CGFS, onsetc, onsetm*	90	88	92	2.2	84	7.0	84	5.3	91	2.2
Two-stage (best HMM and rule-based)	*Tonality, pitch variance, width, CGAV, CGFS, onsetc, onsetm*	—	91	92	1.4	87	5.3	91	3.8	93	2.3

Figure 3.1: Classification results for all six classifiers tested in the study by Büchler et al. For each classifier, the score for best parameter and feature set is given [9].

speech in noise since the authors find these to be the most important listening environments,and not only this, but also classification between speech in various types of noise.

Data Description: The input stimuli are speech mixed with a variety of background noises. There is a total of 47795 test stimuli, each 14 s long, representing 185 h of sound. The presentation level of the speech lies between 64 and 74 dB SPL , the level is randomly chosen and so is the SNR with values between 0 and +5 dB. Training material consist of speech in traffic, speech in babble and clean speech. Test sounds include these along with a range of other background noises. In this implementation, a single sound source or a combination of two sound sources is defined as a listening environment. Music environments are not included in this study.

Method: The present work mainly uses features from the modulation characteristics of the signal, namely the cepstrum which is the result of taking the Inverse Fourier transform (IFT) of the logarithm of the spectrum of a signal. The absolute sound pressure level and the absolute spectrum shape contain information about the current listening environment, but since they are to easily affected by easily changeable factors, their values are not taken into account in this study. Focus lies on the classifier, and here HMMs are chosen. First a sound source classification occurs where the layer consist of one HMM for each included sound source, here the state probabilities are calculated. The output

data from this classification are further processed by a hierarchical HMM in order to determine the listening environment. The environment model consists of five states and a transition probability matrix. A state diagram of this model can be seen in Figure 3.2.

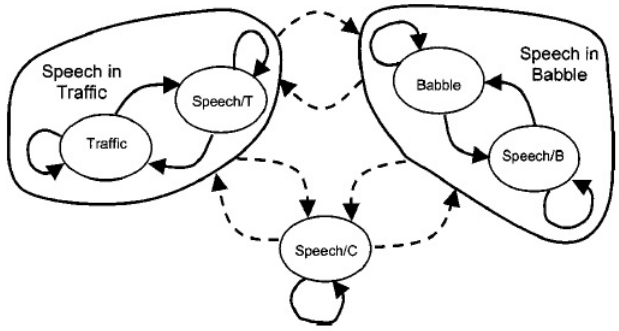


Figure 3.2: State diagram for the environmental hierarchical HMM containing five states. The dashed arrows indicate transitions between listening environments, these have low probability. The solid arrows represent transitions between states within one listening environment, these have relatively high probabilities. From [23].

Results: It is obvious that the sounds included in the training of the classifier were the easiest ones to correctly classify. For both clean speech and speech in traffic noise, the hit rate was 99.5 %, and for speech in babble noise it was 96.7 %. The false alarm rates were low, 0.2, 0.3 and 1.7 % respectively. The classifier was tested with the test sound shifted abruptly which made the classifier output shift from one environment to another within 5-10 s after the change of stimulus, except for clean speech to another listening environment which took about 2-3 s. Given environments with a varied number of speakers (1, 2, 4 or 8), the signals with one or two peakers were classified as clean speech and the others as speech in babble. Adjusting the SNR made a speech signal of 64 dB SPL be classified as speech in babble with a SNR interval between 0 and +5 dB and with a SNR of +10 dB or greater the signal was classified as clean speech. The impact of reverberation turned out to classify speech from a distant speaker (outside the reverberation radius) as speech in babble whereas speech from the listener itself was classified as clean speech.

The classification can be said to be robust with respect to level and spectrum variations, since these features are not used. This system is flexible and easy to update since the definitions of the listening environments can be changed, and sound sources can be added or removed. This is highly recommended for a

future system to make it easier to test all kinds of situations without to many alterations.

3.1.3 Computational Auditory Scene Recognition

Authors: V. Peltonen, J. Tuomi, A. Klapuri, J. Huopaniemi and T. Sorsa, Signal Processing Laboratory, Tampere University of Technology, Tampere, Finland and Speech and Audio Systems Laboratory, Nokia Research Center, Nokia Group, Finland, 2002 [27]

The purpose of this study is to classify auditory scenes into predefined classes. For this a newly developed concept, auditory scene recognition, is used. This is aimed at recognising a listening environment only using audio information, so recognition of the context is of interest here instead of analysing and interpreting discrete sound events. This work is conducted in 2002 and thus is some of the first within this field.

Data Description: A variety of different auditory scenes were used for real-world recordings. 226 measurements were made using two different configurations, 55 recordings using a binaural setup and 171 recordings using a stereo setup. Six classes were categorised according to common characteristics of the scenes, the six being outdoors, vehicles, public, offices, home and reverberant places. Some of the recordings can be associated with more than one class, but for one recording, multiple class labels was not allowed.

Method: Two different but almost equally effective systems are used. For each of the systems, different features were tested. Temporal, frequency and cepstral features are tested with the two classifiers; a k-nearest neighbour classifier (k-NN) and a Gaussian mixture model (GMM) . For the k-NN classifier it turned out that increase of the number of neighbours only had a minor effect on the performance, why a 1-NN classifier was chosen and for the GMM the optimal order was found to be five. The training set included all the recorded audio material and the test set included the material from 17 of 26 possible scenes. Test set duration was 30 s, training set was 160 s for all the cases. The classification performance was evaluated using the leave-one-out methods for cross-validation. This can be beneficial since the system never before has heard the particular sound while the training data is utilized maximally.

Results: Not all combinations of features were examined due to computation time. 11 combinations were chosen and tested with both of the classifiers. This resulted in a number of recognition rates for test sets of 30 s length. 26 trained scenes were used, which gives a random guess rate of 4 %. All the recognition

rates for the 11 feature set combinations can be seen in figure 3.3.

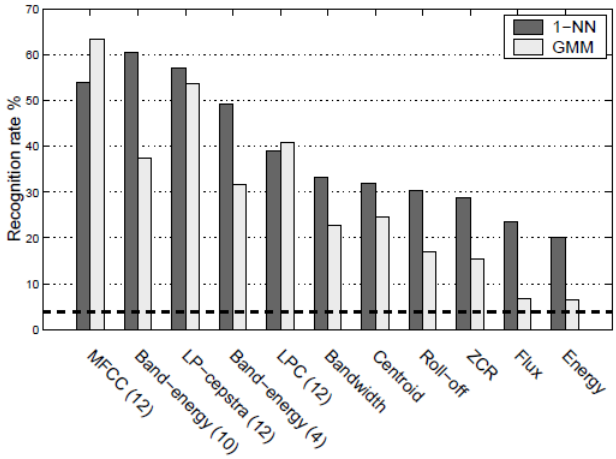


Figure 3.3: Recognitions rates obtained using the 1-NN and GMM for different features. The dashed line indicates the random guess rate. From [27].

This work suggest that focus in the future work within the environment recognition process should be put on modelling distinct sound events. This has been focus for several studies following in the years after this work but is still of interest since nothing yet is as good as the human solution to this problem.

3.1.4 Adaptive Environment Classification System for Hearing Aids

Authors: L. Lamarche, C. Giguère, W. Gueaieb, T. Aboulnasr, and H. Othman, School of Information Technology and Engineering (SITE), University of Ottawa, Ontario, Canada, 2010 [16]

The purpose of this study is, on the long-term, to develop fully trainable hearing aids in which both the acoustical environments encountered in everyday life and the settings preferred by the user in each environment can be learned. A framework is designed for adaptive classification which allow classes to be be added, deleted and tuned based on the environments the user encounters, without intervention or offline training [16].

Data Description: A sound database consisting of real-world sound files assembled from a wide range of sources was used. Each sound file with the specifications 30 s long, 20 kHz sampling and 16 bits, mono and with labels according to the class of the sound. In the study, a total of 960 sound files were used, belonging to the classes speech, noise and music. Speech and noise were divided into test and training files whereas music files were only used in the testing phase in order to evaluate adaptive classification.

Method: Only three features were considered in this work to maintain low complexity. The first two envelope-related features are depth of amplitude modulations in two modulation frequency ranges, 0-4 Hz (feature 1) and 4-16 Hz (feature 2). The third feature carries information about the fine structure of the signal and is the temporal variance of the instantaneous frequency (feature 3). These three features were chosen for their ability to distinguish between speech, noise and music environments [16]. The characteristic feature vectors are stored in a buffer which supplies this information every 15-60 s depending on the rate at which the classifier needs to be updated. Two adpative classification systems are developed and tested; the minimum distance classifier using an Euclidian metric and the Bayesian classifier. Both are static classifiers which, in this work, are extended to adaptive sound classification systems that can split and merge classes based on the feature patterns of the environments they encounter.

Results: Performance of the adaptive classifiers was compared to a best-case non-adaptive fully supervised three-class system trained on the entire data [16]. Classification accuracy was measured by the hit rate (HR), overall hit rate (OH) and the false alarm rate (FA). For both of the classifiers four post-splitting options were considered; globally re-estimating the prototypes of the splitting and new classes (PS1), locally estimating the prototype of the splitting class while globally re-estimate the prototype of the new class (PS2), keeping the original splitting class prototype unchanged while locally estimating the prototype of the new class (PS3) and keeping the original splitting class prototype unchanged while globally re-estimating the prototype of the new class (PS4) [16]. A splitting and a merging algorithm were tested for both of the classifiers. For the minimum distance classifier, the results can be seen in Figure 3.4 and for the Bayesian classifier, the results can be seen in Figure 3.5. In all the cases, the results are compared to the results from non-adaptive supervised learning.

Comparing the two splitting algorithms for the adaptive classifiers, it can be seen, that the Bayesian classifier achieves the highest OH with a maximum of 86.8 % (PS4 option) compared to the best minimum distance classifier option which with option PS3 gives an OH of 83.0 %. These adaptive classifiers are proposed only to be used with trainable hearing aids so a tracking of the behaviour of the user could create a fully learning classification system where both the class environments encountered by the user and the preferences for each class

	Testing accuracy (%)						
		Speech		Noise		Music	
	OH	HR	FA	HR	FA	HR	FA
Supervised	86.1	96.3	10.0	79.8	8.6	80.2	1.6
K-means (no PS)	81.1	97.8	11.6	79.4	14.5	66.9	0.6
K-means (PS1)	80.1	97.8	11.6	79.6	15.9	64.2	0.5
K-means (PS2)	80.2	97.8	11.5	79.7	15.8	64.5	0.5
K-means (PS3)	83.0	97.2	10.9	80.3	12.5	71.7	0.7
K-means (PS4)	82.6	97.1	10.8	80.6	13.2	70.5	0.6

(a) Splitting accuracy

	Testing accuracy (%)				
		Speech		Noise/music	
	OH	HR	FA	HR	FA
Supervised	91.4	97.3	12.0	88.0	2.7
Merging (no PS)	90.0	98.5	16.0	84.0	1.5
Merging (PS1)	89.6	98.5	17.6	82.4	1.5
Merging (PS2)	89.3	98.4	18.8	81.2	1.6
Merging (PS3)	88.3	96.5	22.4	77.6	3.5
Merging (PS4)	88.9	97.5	23.2	76.7	2.5

(b) Merging accuracy.

Figure 3.4: Accuracies of the adaptive minimum distance classifier, without and with four post-splitting options, compared to non-adaptive supervised learning. From [16].

	Testing accuracy (%)						
		Speech		Noise		Music	
	OH	HR	FA	HR	FA	HR	FA
Supervised	89.7	97.1	7.0	83.7	5.7	86.4	2.3
EM (no PS)	81.1	94.9	8.4	83.4	15.6	67.2	2.5
EM (PS1)	86.1	94.5	8.2	79.6	8.2	82.0	4.1
EM (PS2)	85.8	95.1	8.3	79.8	8.7	80.6	3.8
EM (PS3)	81.3	96.5	8.9	83.6	15.4	66.1	2.1
EM (PS4)	86.8	96.7	8.6	79.3	7.2	81.9	3.8

(a) Splitting accuracy.

	Testing accuracy (%)				
		Speech/noise		Music	
	OH	HR	FA	HR	FA
Supervised	91.8	94.4	11.8	88.2	5.6
Merging (no PS)	82.6	93.4	34.6	65.8	6.6
Merging (PS1)	89.4	92.6	15.0	85.0	7.4
Merging (PS2)	88.9	94.0	18.4	81.6	6.0
Merging (PS3)	83.2	95.5	35.4	64.6	4.5
Merging (PS4)	90.0	93.4	14.9	85.1	6.6

(b) Merging accuracy.

Figure 3.5: Accuracies of the adaptive Bayesian classifier, without and with four post-splitting options, compared to non-adaptive supervised learning. From [16].

could be learned.

3.1.5 Evaluation of Sound Classification Algorithms for Hearing Aid Applications

Authors: JJ. Xiang, M. F. McKinney, and K. Fitz andT. Zhang, Starkey Laboratories, Washington, USA, 2010 [34]

In this study, more sophisticated features and classifiers are tested in a number of experiments in order to asses their impact on automatic acoustic environment classification performance.

Data Description: A database composed of sounds from five classes is used. These classes are: speech, music, wind noise, machine noise and others with

a duration of 40, 14, 12, 73 and 22 minutes respectively. Music comes from a database which contains 80 15 s audio music samples, the remaining samples are recorded by the author for this study. The class speech contains both clean and noisy speech, where the noisy speech is generated by randomly mixing signals of clean speech with noise signals at three levels of SNR: -6 dB, 0 dB and 6 dB. The class others contain all sounds that are not described by the other classes.

Method: A low-level feature set and MFCCs are used in this study, the first one including both temporal and spectral features including the logarithms of these features. In the MFCC set, the first 12 coefficients are included. The feature set is specific to the choice of classifier where focus in this study is on a quadratic Gaussian classifier, a GMM and an ergodic HMM. The feature selection is performed for each of the classifiers.

Results: A combination a each of the classifiers with the two sets of features has been evaluated. The result of this can be seen in Figure 3.6. There is no significant difference in classification performance between the two feature sets given that more than five features are used in both cases.

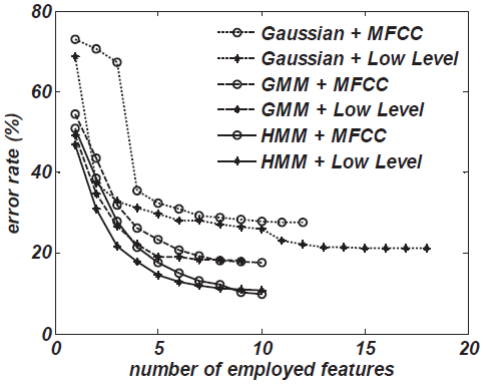


Figure 3.6: Error rate as a function of the number of employed features. Performance is evaluated for the possible combinations of each of the classifiers with the two sets of features. From [34].

The advantage of using advanced classification models with the low-level feature set becomes obvious in this study. When the computational cost is limited, the low-level feature set is definitely recommendable. 5-7 features should be used in order to balance the performance and the computational cost in the most suitable way. This should be taken into account in future work.

3.1.6 Feature Selection for Sound Classification in Hearing Aids Through Restricted Search Driven by Genetic Algorithms

Authors: E. Alexandre, L. Cuadra, M. Rosa, and F. López-Ferreras, Department of Signal Theory and Communications, University of Alcalá, Madrid, Spain, 2007 [5]

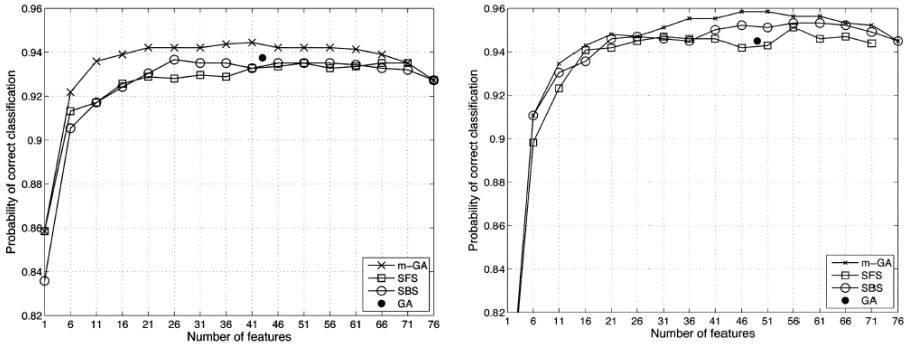
The purpose of this study is to develop an automatic sound classifier for digital hearing aids that aims to enhance listening comprehension when the user goes from one listening environment to another [5].

Data Description: The sound database in this study consists of 2936 files from three main classes; speech in quiet, speech in noise and noise. Each of the files have a length of 2.5 s with 22050 Hz sampling frequency with 16 bits per sample. The classes contain 509, 1455 and 972 files respectively. Music files have in this case been categorized as noise sources. The speech in noise signals exhibit different SNR ranging from 0 to 10 dB. All the files were randomly divided in to three groups, training, validation and testing with a division corresponding to 35 %, 15 % and 50 %.

Method: 38 features were considered in this study; mean and variance were calculated for 16 different spectral and temporal features, the high zer crossing ration and the low short-time energy ratio were calculated along with 20 Mel frequency cepstral coefficients. All these features were calculated from both the original time-domain sound signal and from the linear prediction coefficients (LPC) resulting in the creation of a feature vector containing the final 76 features. The classifier chosen is the two-layer Fisher linear discriminant which is a genetic algorithm (GA). Results are produced using 4 options; a conventional GA without the m-features operator, a GA with the m-features operator, a sequential forward search (SFS) and a sequential backward search (SBS).

Results: The two layers of the algorithm each represent a split problem, the first layer classifying speech/nonspeech and the second layer classifying clean/noisy speech. Each of these two problems give a probability of correct classification. This is calculated for each of the four options mentioned in the methods for different numbers of features, resulting in the functions seen in Figure 3.7.

Using more features does not necessarily improve the probability of correct classification, P_{CC}, but it definitely requires a larger computational cost. With the GA using m-features operator, only 11 features are needed for the speech/nonspeech classification to reach the same P_{CC} as the unconstrained GA, both of them performing better than the SFS and the SBS. This method allows a subset of



(a) Speech/nonspeech problem **(b)** Clean/noisy speech problem.

Figure 3.7: Probability of correct classification, P_{CC} as a function of the number of features reached for the unconstrained GA, the GA with the m-features operator, the SFS and the SBS. From [5].

signal-describing features to be selected in order to get a high P_{CC}. This is desirable for any classification system developed for hearing aids.

3.1.7 Pitch Based Sound Classification

Authors: A. B. Nielsen, L. K. Hansen and U. Kjems, Intelligent Signal Processing, Technical University of Denmark, Lyngby, Denmark and Oticon A/S, Smørum, Denmark, 2006 [21]

The purpose of this study is to create a classification system based solely on the pitch to classify three classes; music, speech and noise. In such a system a pitch estimator, pitch features and a classification model is necessary. To enhance efficiency of this system, effort has gone to finding features that separate the classes well instead of focusing on a complex classification model.

Data Description: Training data comes from a database consisting of three clean classes; speech, music and noise. The speech was taken from two different clean speech databases and was supplemented with other clean speech sources in different languages, totalling 42 minutes. The music, totalling 50 minutes, comes from various recordings from different genres. The noise contains various noise sources such as traffic, factory noise and many people talking and has a total duration of 40 minutes. The test set consist of public available sounds, 35 miutes of speech, 38 minutes of music and 23 minutes of noise. Applied settings gives approximately 40 pitch samples per second and overlap is used to obtain

a classification every second. These settings makes the training set size around 7000 samples and the test set is approximately 5500 samples [21].

Method: A total of 28 features are found calculating the pitch and a measure for the pitch error. Four features yielded the best performance, these four are: the standard deviation of the reliability signal, the pitch abs-difference based on histograms, the distance from the pitch to a 12'th octave musical note and the difference between the highest and the lowest pitch in a reliable window. For classification, a procilistic model is used based on the soft-max output function. The model is trained using maximum likelihood and three inputs are used; linear, quadratic including the squares of the features and a quadratic where all the covariance combinations are used.

Results: 1 s classification windows lead to the results seen in Figure 3.8 and Figure 3.9

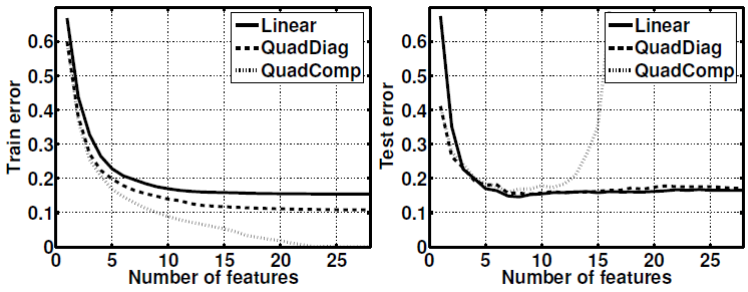


Figure 3.8: Negative log likelihoods of the training and test error. The test error shows no improvement when using more than 7 features. From [21].

In general, the more complex models show better training error, but when it comes to test error not much is gained from using the more complex systems, from a number of five features the linear model performs better. Some of the functionalities of this system would give good results if they were implemented in hearing aids, because this could possibly increase the classification functionality.

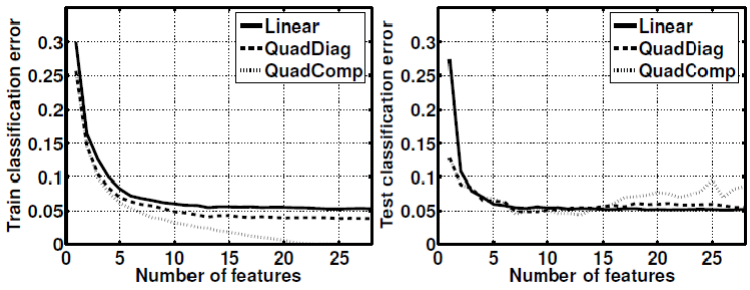


Figure 3.9: Classification error for both training and test data with 1 s windows. A test classification error of just below 0.05 is achieved. From [21].

3.1.8 An Efficient Code for Environmental Sound Classification

Authors: R. Arora and R. A. Lutfi, Department of Electrical and Computer Engineering and Auditory Behavioral Research Laboratory, University of Wisconsin, Wisconsin, USA and Department of Communicative Disorders and Auditory Behavioral Research Laboratory, University of Wisconson, Wisconsin, 2009 [6]

The purpose of this study is to develop an automated sound recognition system that effectively deals with efficient encoding of potential signals and the interference produced by sound sources considered as noise. A new approach is tested using compressed sensing (CS).

Data Description: 50 environmental sounds were used in the simulations, 25 targets and 25 interferers. These sounds come from high-quality sound effects CDs where the sounds have been shown to be easily identified by human listeners. All recordings were normalised in duration to 3.6 s by zero padding when necessary and equated in total rms. The sounds were down-sampled from 44.1 kHZ to 4kHz and then contained 14400 samples. Target-to-interference ratios were introduced ranging from -20 to 20 dB.

Method: Compressed sensing is used by projecting the signal onto a basis that has nothing in common with the structure of the signals and shares no features with the signal. The one basis that can live up to this property for all signals is the random basis, which has noise waveforms as basis functions. In this way, the basis is ensured to have some measurable correlation with any signal, positive or

negative. Only a small number of these correlations are required to recover the signal without error. Almost all signals (except continuous broadband noise) are sparse in either the frequency or the time domain. This sparsity can be used advantageous in classification of environmental sounds.

Results: Selected at random are M Gaussian noise waveforms each of length N to construct an $M \times N$ matrix as the random basis to be used in all conditions, M is ranging from 1 to 256 [6]. CS achieves a near perfect classification with only 128 projections of an arbitrary set of sounds, even with a target-to-interference ratio as low as -20 dB.

The listening situations in this work are not as realistic as what a human listener might encounter. If the features of CS is implemented in a computational model, it still remains to be seen if this model would eventually approach the performance of the human classifier. The results of this work encourage speculations as to if and how CS might be incorporated in order to obtain this result.

3.2 Approaches Developed for Improvement of Speech Perception

Many approaches have been developed to improve speech perception in hearing aids. The one included here has interesting sound signal recordings.

3.2.1 New Idea of Hearing Aid Algorithm to Enhance Speech Discrimination in a Noisy Environment and its Experimental Results

Authors: S. M. Lee, J. H. Won, S. Y. Kwon, Y.-C. Park, I. Y. Kim and S. I. Kim, Department of Biomedical Engineering, College of Medicine, Hanyang University, Seoul, Korea and Department of Computer Science, College of Engineering, Yonsei University, Wonju, Korea, 2004 [18]

The purpose of this study is to improve speech perception in a noisy environment. This s done using an algorithm that combines independent component analysis (ICA) with multi-band loudness compensation.

Data Description: The authors recorded mixed signals using a hearing aid in a real room. The speech source was located 1 min front of the hearing aid, and

the noise source was placed 1 m behind it [18]. The speech signals was either a one-syllable signal from a male or a two-syllable signal from a female. The noise signals used were a car, babble and factory noises.

Method: The mixture signals received in the front and rear microphones can be separated by using ICA. This is used for speech in a noisy environment. Afterwards the loudness perception of the hearing impaired person is restored using an eight-band loudness correction algorithm by using the procedure referred to as the frequency sampling method. The output can be selected to be either from the front or rear direction. This is implemented to make it possible to choose the front direction only, since the hearing impaired, in most cases, are interested in speech from the front.

Results: The proposed method was compared to a spectral subtraction method. Figure 3.10 shows a source signal of the one-syllable male talker and car noise. The recorded signal of the male in the car is separated using both the proposed method and the spectral subtraction method.

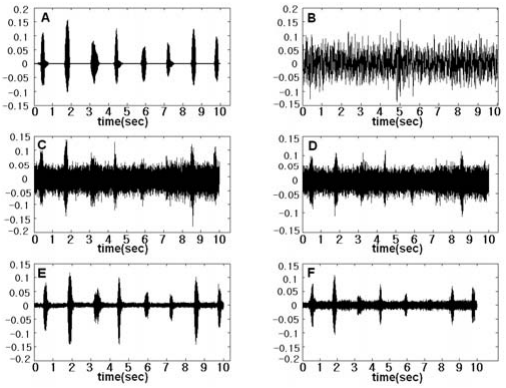


Figure 3.10: Speech and noise input and output signals. A: original one-syllable male talker, B: car noise, C: mixed signal from the front microphone, D: mixed signal from the rear microphone, E: output speech signal extracted by the proposed method, F: output speech signal extracted by the spectral subtraction method. From [18].

The SNR improves drastically when extracting the speech signal using the proposed method compared to the spectral subtraction method. This seems to be the tendency in various noise environments, all getting higher SNR values with the method separating signals using ICA and restoring the loudness perception by using an eight-band loudness correction algorithm.

Chapter 4

Data Description

For this project, a number of sound files were generated using the Greenhouse sound database at Oticon. This database consist of numerous sound recordings, including impulse responses from different rooms with a large variety of reverberation. These rooms can be used to describe more or less realistic sound environments, recordings from a car, a cantina (café), a staircase and a bathroom form basis for realistic sound environments whereas impulse responses from anechoic rooms and meeting rooms at Oticon form a basis of environments simulating realistic environments of rooms with the same size and reverberation time. All together, impulse responses from 9 environments where used, these being:

- Atlantic
- Canada
- Café
- Car
- Cellar
- Faroe Islands
- Germany
- Japan North
- Staircase

Only one loudspeaker was used in the measurement setup. This was done to assure that the input to the HATS was the same in every recording without colouring from different loudspeakers. The impulse responses all have a sample rate of 48 kHz with 24 bit recording. The impulse responses have been post-processed to remove the very long tails, so that only the part of the tail above the noise floor is kept. In some of the environments, different types of hearing aids were used for the recording. The setup on a HATS can be seen in Figure 4.1.

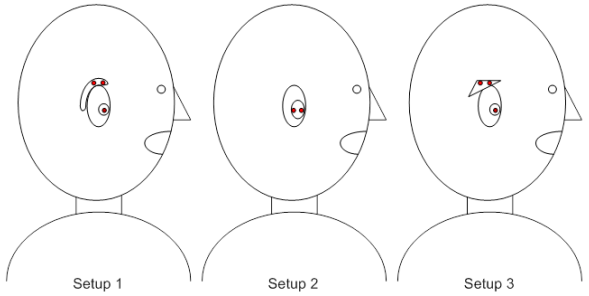


Figure 4.1: Different setups for impulse recordings on a HATS. Setup 1 shows the microphone placements for a Agil BTE shell. Also the sound at the eardrums should be recorded simultaneously (all together 6 simultaneous recordings). Setup 2 shows the microphone placements for ITE shell recordings (all together 4 simultaneous recordings). Setup 3 shows the microphone for Dual BTE shell recordings. Here too, a simultaneous recording at the ear drum should be made (all together 6 simultaneous recordings). [26].

The setup with six microphones placed on a HATS mannequin was used in all cases with a setup like the first one in Figure 4.1. Sound reflections from the rooms were recorded on the HATS with sound sources from different directions. Setup 1 was chosen for further sound file generation in all the environments.

4.1 Description of Sound Environments

The listed environments are all (except car) names of a room or a location at Oticon. Following is a description of each of them, mainly by a visualisation of the rooms describing their size, placement of the listener and placement of loudspeakers corresponding to sources in the environment. In each of the environments different placement of sources were considered. In all the rooms a

realistic environment has been set up including different noise sources (more about these in Section 4.2) as well as situations trying, in their best way, to imitate the placement of the car.

4.1.1 Atlantic

The setup in the bathroom Atlantic can be seen in Figure 4.2. This setup was chosen since it is the one of the Atlantic setups that can resemble position of a target source in a car the most. Further explanation of the content and placement of sources in this environment will be given in Chapter 7.

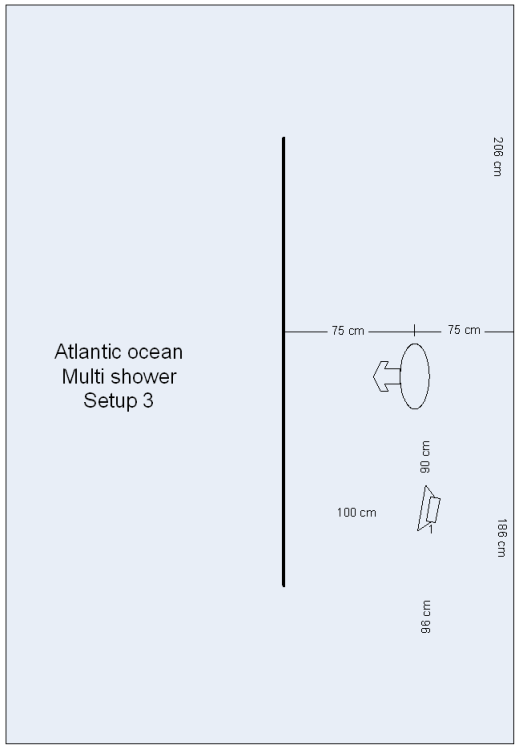


Figure 4.2: Setup of the measurements in the bathroom "Atlantic". [33].

4.1.2 Canada

The setup in the meeting room Canada can be seen in Figure 4.3. Recordings in Canada have loudspeaker placements equally distributed in a circle around the listener. This placement is useful in simulating realistic situations in such a room, but not so helpful when a car situation is imitated. Further explanation of the content and placement of sources, including the imitation of a car situation, in this environment will be given in Chapter 7.

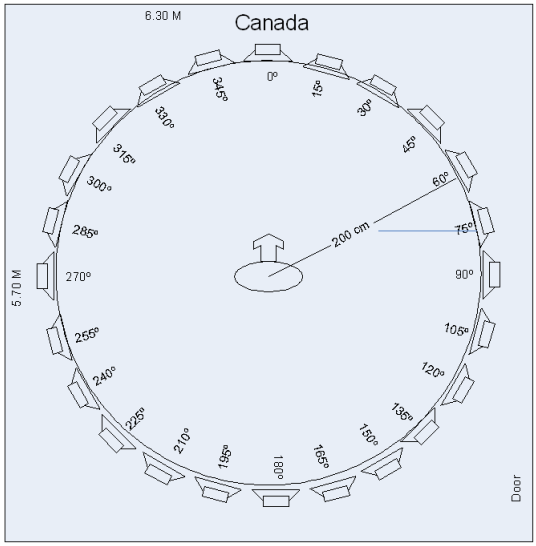


Figure 4.3: Setup of the measurements in the meeting room "Canada". [33].

4.1.3 Café

The setup in the canteen Café can be seen in Figure 4.4. Recordings in Café have loudspeaker placements in different positions around the listener, corresponding to placement of other people by the tables in the canteen. This placement can both be used for a realistic setup in the canteen and one that imitates a car situation. Further explanation of the content and placement of sources in this environment will be given in Chapter 7.

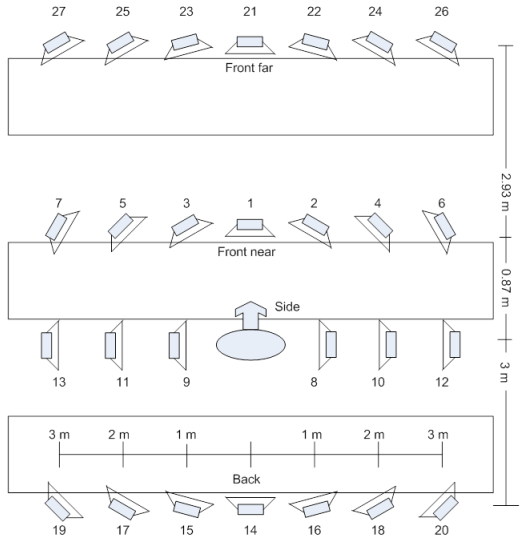


Figure 4.4: Setup of the measurements in the canteen "Café". [33].

4.1.4 Car (Ford Scorpio)

Recordings have been made in a Ford Scorpio with possible source placement as seen in Figure 4.5. Placement of the HATS in the car environment is based on a possible expansion of the recordings in the car, recording engine noise, wind noise and so on while the car is driving. Therefore the HATS has not been placed in the driving seat. The further recordings have not been made yet, and the placement of the HATS is a bit unrealistic in this environment, since a listener in the passenger seat in a realistic situation would turn its head towards the target source. This is not possible in this recording, so this would mainly be a realistic setup for a car in a country where the driver sits at the right front seat.

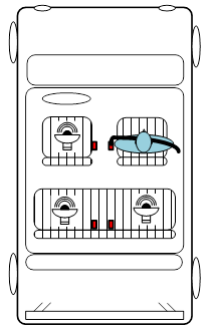


Figure 4.5: Setup of the measurements in a Ford Scorpio "Car". [26].

4.1.5 Cellar

Recording in the Cellar can be seen in Figure 4.6. The recordings were set up at the end of a long corridor in a corner space next to an elevator. This room is therefore not a closed room like the others, but are still considered this in the further work. Recordings in Cellar have loudspeaker placements equally distributed in a circle around the listener. There are not as many possible source placements as in some of the other rooms, but there a still a fair amount making it possible to simulate different situations in a cellar environment. Further explanation of the content and placement of sources, including the imitation of a car situation, in this environment will be given in Chapter 7.

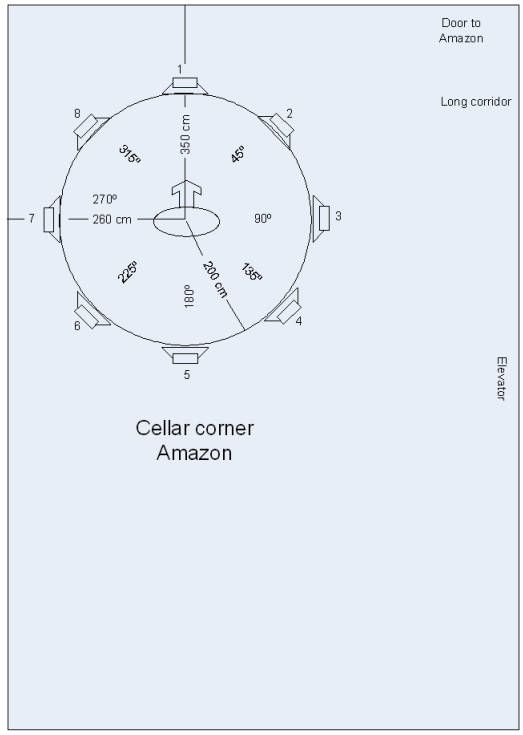


Figure 4.6: Setup of the measurements in the cellar "Cellar". [33].

4.1.6 Faroe Islands

Recording in the sound proof room Faroe Islands can be seen in Figure 4.7.Recordings in the soundproof room Faroe Islands have a loudspeaker placements equal to the one in Canada thus with reversed rotation direction, but the same remarks go for this room as well.

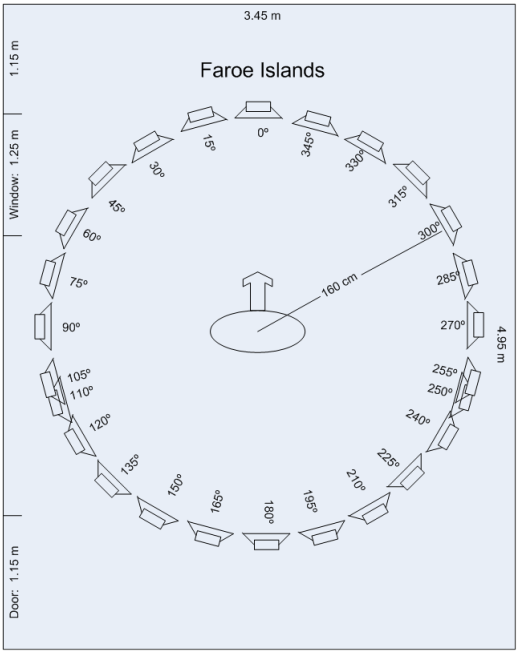


Figure 4.7: Setup of the measurements in the sound proof room "Faroe Islands". [33].

4.1.7 Germany

Recording in the meeting room Germany can be seen in Figure 4.8.In the meeting room Germany, four different setups have been used for recording. Here setup D has been chosen in order to be able to imitate a car situation in the best possible way. Further explanation of the content and placement of sources, including the imitation of a car situation, in this environment will be given in Chapter 7.

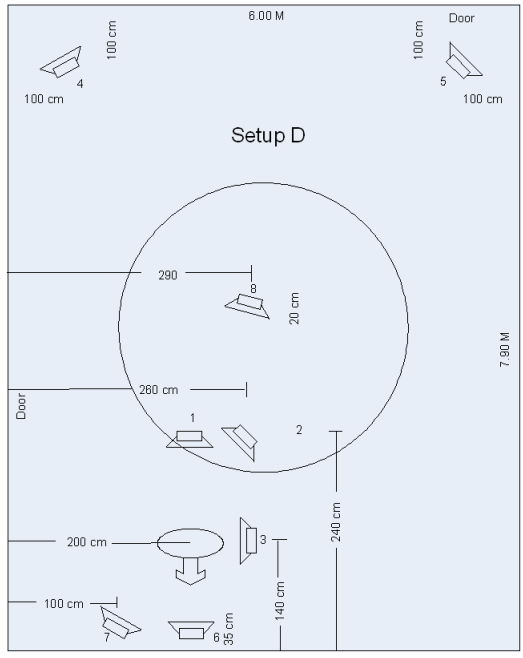


Figure 4.8: Setup of the measurements in the meeting room "Germany". [33].

4.1.8 Japan North

In the meeting room Japan North, the HATS was placed between two tables with the nose 90 °C away from the window. Measurements was done in the angles 0 °C, 45 °C, 90 °C, 135 °C, 180 °C, 225 °C, 270 °C, 315 °C, with the loudspeaker placement starting 50 cm from the center of the HATS, and increased with 50 cm for each measurement until a wall was reach. A photograph of the setup can be seen in Figure 4.9.The very flexible placement of the loudspeakers make it possible to simulate many situations in this room. This comes in handy both when imitating a car situation and other more realistic source placements. See Chapter 7 for further details.

Figure 4.9: Photograph of the setup of the measurements in the meeting room "Japan North". [32].

4.1.9 Staircase

Recording in the Staircase can be seen in Figure 4.10. The staircase goes froom the cellar to the second floor and the setup was placed at the north east ground floor. Recordings in the Staircase have a much different loudspeaker placement than the other environments. There are two possible source placements at the same level as the listener and one placement on the level below and one on the level above the listener. This can cause problems when an imitation of the car situation is considered, but the placement of the sound sources in this situation and in more realistic situations will be given in Chapter 7.

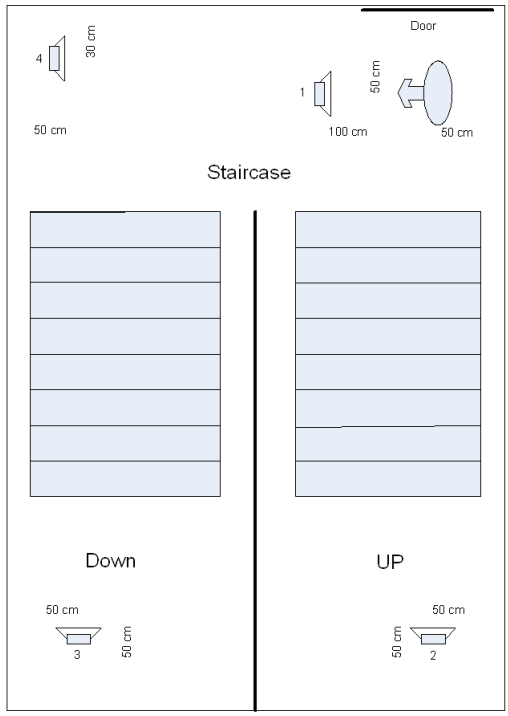


Figure 4.10: Setup of the measurements in the staircase "Staircase". [33].

4.2 Sound Source Signals

In all the generated sound files a number of speech signals are used either as target or background speakers. Along with these a number of noise signals are used as well to generate environments with both speakers and realistic noise. A description of the speech and noise signals used follow here.

4.2.1 Speech Signals

Three speech signals are chosen from an English speaker setup. One female speaker is used for target source in all generated sound signals while a dialogue between two male speakers are used as non-target speech in the setups where speaker noise is included. The target source is taken from the "English monologues, some with raised effort" and the noise speaker sources comes from "English dialogues - 2 male voices". More information about the three files can be found in Appendix B.

4.2.2 Noise Signals

Many sounds can be found in the Greenhouse database, and a lot of them can make sense as noise sounds in the generated environments. To be sure that a classifying algorithm does not base differences in the environments on differences from recording equipment, it is important to chose a set of noise sounds that all are recorded with the same equipment. ICRA2 [7] is an example of such a sound set. These recordings have been made as a part of a project at the Technical University of Denmark and contains a broad variety of sounds, some more realistic to occur in the given environments than others. From these sounds the most usable ones have been chosen, and an analysis of which sounds that could be realistic in which environments form the basis of the included noise sounds. Examples of noises that can occur in most of the environments are hair dryer, vacuum cleaner, ventilation, music of different genres and keyboard typing. A list of all the possible realistic noise files to be chosen from can be seen in Appendix B.

4.3 Generating Sounds

A simulation tool called Acoustic Simulation is used to generate all the sound files used in this work. The tool is developed at Oticon and uses sounds from the Greenhouse database to create new sounds which can be a single sound or combinations of different sounds. Most importantly it is possible to create different acoustic environments using recorded impulse responses from different rooms convoluted with any number of wanted sound sources in the signal. The level of the target source and the noise sources can each be specified along with an overall SNR which gives the final input level at the listeners eardrum. The placement of the sources is also to be specified according to the possible placements of the given room (depending on the loudspeaker placements in the original recordings).

For all the environments described earlier, a number of sound files have been generated. First of all, the number of speech sources in the signal is varied. The target source, the female speaker, is present in all generated sounds. The two male speakers are either not included (the filenames end with _1source), one is included (the filenames end with _2sources) or both are included (the filenames end with _3sources). Each of the created signals in this work contains a target source that is set to a level of 65 dB in all cases. Every other included source in any of the signals is set to a level of 55 dB. There are many sounds fit for noise sources, and for each environment, it is carefully considered which of the noise signals mentioned in Section 4 that are realistic for the environment. Sound signals are then created placing these noise signals in realistic positions for each of the environments by generating a file in Matlab for all the wanted environments, one for each combination of speakers with the potential noise sources for that environment. A list of all the possible scenarios for each of the environments can be seen in Appendix B.

The signals are created by specifying which source at what position should be a part of the sound signal. Impulse responses have been measured for different positions in the chosen environments, so by specifying the position of a source, the source signal is convolved with the impulse response from that position. All the convoluted signals are then added and result in the final sound signal from the specific environment with the specified speaker and noise sources at their respective positions.

When creating the sound signals, it is possible to define an overall level of both the target signal and the noise sources at the eardrum along with the signal to noise ratio. If nothing is specified, the target source and the noise sources will be added with the level they each where specified to have. When specifying both the overall input level at the eardrums and the signal to noise ratio, the sources

are scaled to fit these levels. If no scaling is specified, the levels of the target and noise sources will create a signal with a SNR depending on the included sources. If an overall input level at the eardrum is specified along with an SNR, the target and noise sources will scale accordingly to the chosen parameters. This is not a realistic situation, but can be used in cases where testing evolves around these parameters.

In this work, three cases are compared, one with no scaling, one with an overall level set to 65 dB with a SNR = 0 dB and one with an overall level of 65 dB with a SNR = 10 dB. The testing of these three cases can be seen in Section 7.2.

4.4 Sound Data

Different types of datasets have been created, first of all a small dataset has been generated to perform some of the preliminary tests. The small dataset consist of two different environments, Car and Canada. For each of the environments, three sound files are generated, one including only the target source, one including the target source and one noise source and one including the target source and two noise sources. All of the sources in this dataset are speakers, those described in Subsection 4.2.1 are used in all the generated sound files. In testing the target direction, a special dataset is generated, this is described in the relevant section. Using a small dataset for the preliminary tests seem most reasonable since they will show the tendency of the behaviour no matter how much data is included but it will save a lot of time on the calculations. In testing the target direction, all environments that fit the specifications for this test are included.

A big dataset has also been generated and used to test the final framework. This set include the nine environments mentioned earlier where for all eight environments that are not the Car, nine sets of sound signals have been generated containing different noise sources (these all differ from environment to environment representing possible setups including only realistic noise signals). A tenth situation is generated containing only speaker sources. For each of these ten setups three signals are generated containing one, two or three speakers, the first one only including the target source and the others containing also either one or both of the noise speaker sources. For the Car, nine situations are created using realistic noise sources, one situation is created containing only speaker sources and ten situations are created using semi-realistic noise sources. All together this results in 280 sound signals.

In the tests where the big dataset is used, a division is created in order to generate a training set and a test set. The optimal division would be a "leave-one-environment-out" method, but this is not possible because of the constraints of only having one Car setup. Since the optimal split is not possible, data is divided into a training and test set so half of the sounds are used for training and the other half is used for testing. The same setups but with different number of speakers are used in the same part of the split so none of the tested sounds are used for the training. For the Car, the ten semi-realistic situations are used for training and the ten realistic situations are used for testing. For the other eight environments, all the even numbered signals are used for training and all the odd numbered signals are used for testing. The assignment of an even or an odd number to the situation is done randomly, only the tenth situation where no other sounds than the speakers are included is not randomly chosen to have this number. This is done to assure the training always includes the situation with no background noise. In all the environments, only realistic setups are used for testing.

Chapter 5

Technical Description of the Classification System

A classification system typically consist of a number of steps, going from a sound to a classification of the sound using feature extraction and pattern classification. Many sound classification algorithms have been developed and described through the years, but only few are designed for hearing aid applications.

The general structure of a sound classification system can be seen in Figure 5.1. From sound data, a number of characteristic features can be extracted, this is done in the feature extraction step. These features are then used with some pattern classifier to give an output that is recognised as a sound class. In this work, the desired sound classes are the environments, that is, identification of the environment entered by the hearing aid user.

Figure 5.1: General block diagram of a sound classification system for identification of sound environments.

5.1 Audio Features

The features of a sound signal has to be extracted in order to classify the signal into a given class, the features will decide the class of the signal. Feature extraction involves the analysis of the input of the sound signal, the extraction techniques can be classified as temporal (time-domain) and spectral (frequency-domain) analysis. Temporal analysis uses the waveform of the sound signal itself whereas spectral analysis uses spectral representation of the sound signal for analysis. Two types of acoustic features exist, physical and perceptual features. The perceptual features describe the sensation of a sound described by how a human perceives it. Examples of these are loudness, brightness and timbre. Physical features refer to features that can be calculated mathematically from the sound wave such as spectrum, spectral centroid and fundamental frequency. It is only the physical features that are further grouped into spectral and temporal features.

All features are extracted by breaking the input signal into smaller windows or frames and compute the variation of each feature over time by computing one feature value for each of the windows or frames. Feature extraction is of utmost importance in classification of sound signals why the selection of the best feature set makes the classification problem more efficient.

Selecting the best feature set is a crucial step in building a classification system. The selection of features can either be done manually based on results from previous classification systems, or an algorithm can be used to find the most suitable features that can discriminate between the classes to be classified. In this work the last approach is implemented, more about this implementation and feature selection can be found in Subsection 5.1.8. The feature sets that turn out to be of importance differ from the other sets and depend on the input signals. It turns out that every training set gets a different set of important features. Even though the specific feature sets turn out to differ from training set to training set, there are still common features that are always included regardless of what training set, of sound signals, is looked upon. Many features could be mentioned in the following, but focus is in the ones that are included in the feature extraction of this work. These features are described in the following subsections.

5.1.1 Zero-Crossing Rate

The zero-crossing rate (ZCR) counts the number of times the sign of the signal amplitude changes, the number of time-domain zero crossings within one

window. The feature measures the frequency content of the signal and can be calculated as follows

$$ZCR = \frac{\sum_{n=1}^{W-1} |\text{sgn}(x(n)) - \text{sgn}(x(n-1))|}{2W} \tag{5.1}$$

where x is the time-domain signal, W is the size of the window and sgn is the sign function defined as

$$\text{sgn}(x) = \begin{cases} 1 & x > 0 \\ 0 & x = 0 \\ -1 & x < 0 \end{cases} \tag{5.2}$$

ZCR is often used in speech processing. Here the counts of zero-crossings can be used to help distinguish between voiced and un-voiced speech. Un-voiced sounds are very noise-like and have a high ZCR. ZCR can be used to make a rough estimation of the fundamental frequency for single-voiced signals while for complex signals it can be used as a simple measure of noisiness. The ZCR can also be used to determine if a signal has a DC offset. If there are few zero-crossings, it might mean that the signal is offset from the zero-line.

5.1.2 Mel-Frequency Scale Spectrum

The Mel-frequency scale is a linear frequency spacing below 1000 Hz and a logarithmic spacing above 1000 Hz. This scale is computed since we now that human ears amplify tones under 1000 Hz with a linear scale and for frequencies higher that 1000 Hz frequencies are amplified logarithmically. This gives rise to placing more filters in the low frequency regions and less number of filters in high frequency regions. The scale is based on pitch comparison and the reference point between this scale and frequency measurement in Hz is defined by assigning a perceptual pitch of 1000 Mels to a 1000 Hz tone, 40 dB above the threshold of the listener. To compute a Mel-frequency value from a frequency value in Hz, the following approximate formula can be used [25]

$$\text{Mel}(f) = 2595 \log_{10}(1 + \frac{f}{700}) \tag{5.3}$$

The Mel spectrum is computed by multiplying the power spectrum of a sound signal by each of the triangular Mel weighting filters spaced uniformly and inte-

grating the result. In this work a range from 0 to 8000 Hz is considered divided into $K = 26$ uniformly distributed Mel weighting filters.

5.1.3 MFCC

When the Mel spectrum is computed it is possible to calculate the cepstrum of this spectrum by taking the logarithm of the powers at each of the Mel frequencies, take the discrete cosine transform of the list of Mel log powers, as if it were a signal and then the Mel frequency cepstral coefficients MFCCs are the amplitudes of the resulting spectrum. A schematic illustration of these calculations can be seen in Figure 5.2

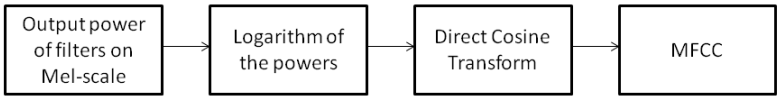


Figure 5.2: Schematic illustration of the steps in the calculation of MFCCs.

The MFCC is computed by [36]:

$$MFCC(d) = \sum_{k=1}^{K} X_k \cos\left[d(k-0.5)\frac{\pi}{K}\right], \quad d = 1, 2, \cdots, D \tag{5.4}$$

where $MFCC(d)$ is the d^{th} MFCC and K is the number of Mel weighting filters. In this work 13 coefficients are included (0-12), that is $D = 13$.

5.1.4 Spectral Features

Spectral features are in useful for distinguishing energy content in signals. Some of those that turn out to be of most importance are mentioned here.

5.1.4.1 Spectral Roll-Off

The $X \cdot 100$ percent spectral roll-off point, P, is determined as the frequency below which $X \cdot 100$ percent of the total signal energy fall. If only the spectral roll-off is mentioned, it refers to the 95 % roll-off point.

5.1.4.2 Spectral Flux

Spectral flux measures the change in the shape of the power spectrum. It is defined as the Euclidean distance between the power spectra of two successive/close frames. For N FFT bins is computed as

$$SF_k = \sum_{n=1}^{N-1} \Big[|X_k(n)| - |X_{k-1}(n)| \Big]^2, \tag{5.5}$$

where k is the index of the frame.

Spectral flux is efficient in discriminating speech/music, since speech in the frame-to-frame spectra fluctuate more than in music, particularly unvoiced speech.

5.1.4.3 Spectral Frequency Band Energy

The spectral frequency band compute energy in the given spectral band by rectangular summation of FFT bins in this band (FFT magnitudes).

$$\mathrm{E}_{\mathrm{bands[n]}} = \mathrm{LoFrq(Hz)} - \mathrm{HiFrq(Hz)} \tag{5.6}$$

5.1.4.4 Spectral Centroid

The spectral centroid represents the midpoint of the spectral power distribution. The spectral centroid, SC, at time t is computed by

$$SC = \frac{\sum_{\forall f} f \cdot X_t(n)}{\sum_{\forall f} \cdot X_t(n)}, \tag{5.7}$$

where $X_t(n)$ is the spectral magnitude at time t in bin n.

5.1.4.5 Spectral Maximum and Minimum Position

The position of the maximum and minimum magnitude spectral bin (in Hz)

5.1.5 Power Cepstrum

To calculate the power cepstrum, a squared magnitude of the Fourier transform of the logarithm of the magnitude of the Fourier transform is applied, that is the cepstrum, $c(n)$, is given by

$$c(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \log \left| X(\mathrm{e}^{j\omega}) \right| \mathrm{e}^{j\omega n} \mathrm{d}\omega, \quad -\infty < n < \infty \tag{5.8}$$

The power cepstrum can be used for identification of any periodic structure in a power spectrum and is ideal in detecting periodic effects such as harmonic patterns. Power cepstrum is generally used in conjunction with spectral analysis, since it identifies items which spectral analysis does not identify while it suppresses information about the spectral content [24].

5.1.6 Log Energy

This component computes logarithmic (log) signal energy from frames. The logarithmic energy (LOGenergy), E_t, can for the frame size, N, be computed as [12]

$$E_t = \log \left[\frac{\sum_{n=0}^{N} x_n^2}{N} \right] \tag{5.9}$$

5.1.7 Fundamental Frequency

The fundamental frequency and the probability of voicing is computed via an ACF/Cepstrum based method. The input must be an ACF field and a Cepstrum field, concatenated exactly in this order. The output is then the fundamental frequency (pitch), F_0, and the envelope of the fundamental frequency can be calculated from exponential decay smoothing.

5.1.8 Feature Extraction

One of the most important things of a classification system is the feature extraction. It is therefore of great importance that the right features are chosen in order to get the best possible classification. In a classification, the best feature set depends on the classifier it is used together with. In this work a classification tree is used in order to investigate which features that describe the sound environment signals in the best way without them being a part of a more complex classifying system. For the purpose of feature extraction, the openSMILE [12] feature extraction toolkit is used. It is a modular and flexible feature extractor for signal processing and machine learning applications. It is a purely C++ function under the GNU license. The toolkit combines features from music information retrieval and speech processing and makes it possible to extract large audio feature spaces both off-line and in realtime on-line processing. A binary version of the tool is available, which makes it possible to use the tool without compiling any source code. The feature extraction can thus be implemented as a part of a Matlab function (this is done in the `generate_features` function) with a specified configuration file in order to get an output in form of a .csv file containing all the values of the calculated features for the specified sound signal.

In this work, the configuration file "emo_large.conf" is used in order to extract 57 low-level descriptors along with the first and second derivatives of these descriptors in a combination with 39 possible functionals, all in order to extract a large set of 6669 features, 1st level functionals of low-level descriptors.

The following (audio specific) low-level descriptors are computed by the emo_large configuration file [12]:

- Frame Energy
- Critical Band spectra (Mel)
- Mel-Frequency-Cepstral Coefficients (MFCC)
- Fundamental Frequency (via ACF/Cepstrum method)
- Probability of Voicing
- Power Cepstrum
- Zero-Crossing rate
- Spectral features (Magnitude of: arbitrary band energies, roll-off points, centroid, maxpos, minpos, flux)

This configuration extracts the features from 25 ms audio frames (sampled at a rate of 1 s). A Hamming function is used to window the frames and a pre-emphasis with k = 0.97 is applied using a 1-st order difference equation: $y[n] = x[n] - k \cdot x[n-1]$. It provides feature sets containing 6669 features given by the logarithmic energy, Mel spectra from 26 bands with a range from 0 to 8 kHz by applying overlapping triangular filters equidistant on the Mel scale to an FFT magnitude spectrum, 13 MFCC (0-12) from the 26 Mel-frequency bands, and applies a cepstral liftering filter with a weight parameter of 22, Pitch (F0), Probability of voicing, F0 envelope, zero-crossing rate, spectral features (5 arbitrary band energies; bands[0]=0-250 Hz, bands[1]=0-650 Hz, bands[2]=250-650 Hz, bands[3]=1000-4000 Hz, bands[4]=3010-9123 Hz, 4 roll-off points; rollOff[0] = 0.25, rollOff[1] = 0.50, rollOff[2] = 0.75, rollOff[3] = 0.90, centroid, maximum position, minimum position, flux) and delta and delta delta.

The suffix _sma appended to the names of the low-level descriptors indicates that they were smoothed by a moving average filter with window length 3. The suffix _de appended to _sma suffix indicates that the current feature is a 1st order delta coefficient (differential) of the smoothed low-level descriptor.

In all the features that refers directly to the input data, the wave file, an abbreviation of pcm_ is put in front of the feature name [3].

In order to map contours of low-level descriptors onto a vector of fixed dimensionality, the following functionals are applied:

- Extreme values and positions
- Regression (linear and quadratic approximation, regression error)
- Moments (standard deviation, variance, kurtosis, skewness)
- Percentiles and percentile ranges
- Peaks
- Means (arithmetic, quadratic, geometric)

A specified list of all the functionals can be found in Appendix C.

5.2 Classifying Algorithm

In this work a classification based decision tree is used as a the classifying algorithm. This section will focus on a description of the tested classification

algorithm, first the general theory behind classification trees and then specifically the Matlab function used.

5.2.1 Classification Tree

Using a binary classification tree is a way to analyse data in a flexible, non-parametric way and gives an interesting way of looking at data. The use of decision trees can be dated back to the 1960s since the use of trees was unthinkable before computer-based calculations became possible [8]. The general classification problem is rather simple, measurements are made in some case and from these measurements it is desirable to predict which class the case belongs to. The goal is then to find a systematic way of predicting this class.

The entire construction of a tree revolves, according to [8], around three elements:

1. The selection of the splits
2. The decisions when to declare a node terminal or continue splitting it
3. The assignment of each terminal node to a class

A tree classifier is constructed by repeated splits of a measurement space into two subgroups until a terminal subset called node is reached. Each terminal node is denoted by a class label. More than one terminal node might be from the same class label and putting all of those with the same class label together, gives the partition corresponding to the classifier. The tree classifier predicts a class for a measurement by a number of steps. The first split decides which of the two child nodes the measurement belongs to. The splitting continues until a terminal node is reached where the class is predicted by the class label attached to that terminal node. The first issue of the construction is how to select the splits to reduce the measurement space into smaller and smaller pieces, making sure that the data in the child nodes are purer than the data in the parent node. In the root node, a search is made through all possible splits to find the one that gives the largest decrease in impurity. This procedure is repeated for each of the following nodes separately. For each branch node, the left child node corresponds to the points that satisfy the condition, and the right child node corresponds to the points that do not satisfy the condition. To decide when a terminate node is reached, a heuristic rule is designed such that when no significant decrease in impurity is possible, the node is not split further and becomes a terminal node. The tree is a convenient classifier since, if a measurement from an unknown

class is dropped into the tree and ends up in a terminal node, it is classified as the class from that terminal node [8]. One way to see how accurate a classifier works, is to test the classifier on cases whose correct classification has been observed. In this way it is possible to estimate the true misclassification rate.

An example of the principle of mapping from a two-dimensional feature space to a decision space can be seen in Figure 5.3. For every point in the feature space, a class is defined by mapping it to the decision space. The borders between the classes are found by training the classifier, here they are made up for the simple example. Once the borders are fixed, a test dataset, independent of the training set, is used to find the performance of the classifier. Going from the feature space to the decision space in this small example would provide a tree as the one also seen in the figure. The first split purifies the child nodes and the left child node satisfy the condition given for the split. At the terminal nodes a classification is made assigning the given class to each of the points from the feature space ending in that node. As can be seen this will lead to some misclassification, but this is unavoidable in most cases, especially when a simple classifier as a classification tree is used.

Setting a stopping rule for the tree can be very complicated, the splitting can be stopped either to soon at some terminal nodes or be continued too far in other parts of the tree. So instead of an attempt to stop the splitting at the right set of terminal nodes, the splitting should be continued until all terminal nodes are very small, resulting in a large tree. This tree can then be pruned upwards resulting in in a decreasing sequence of subtrees and then cross-validation or test sample estimates can be used to pick out the subtree having the lowest estimated misclassification rate. This method is implemented in the Matlab function described in 5.2.2.

5.2.2 Matlab Function `classregtree`

A Matlab function, `classregtree`, is a part of the Statistics toolbox in Matlab, and with this function, a classification tree is build. The entire function with all its posibilities is inspired by the theory presented in [8]. In the function, input is given as X, an n-by-m matrix of predictor values, and y, a vector of n response values as a function of the predictors. When the function is run, output is given as t, a binary tree where each branching node is split based on the values of a column of X, see equation 5.10 for notation.

$$t = classregtree(X, y) \tag{5.10}$$

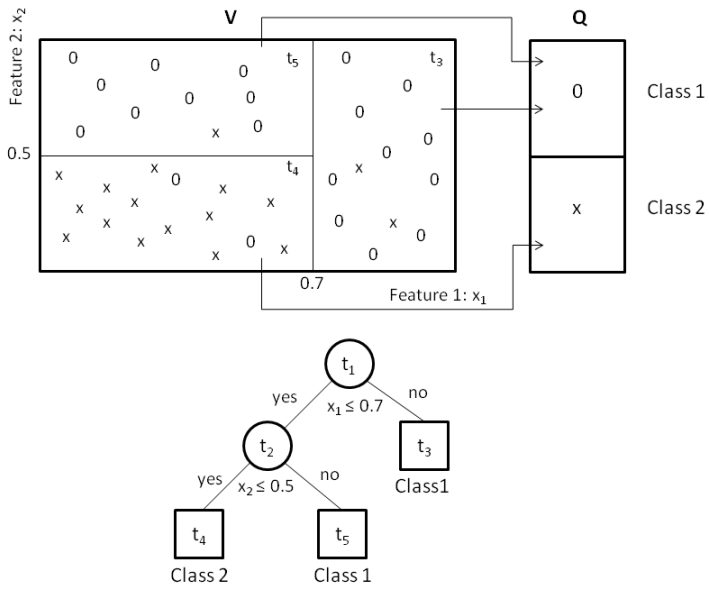


Figure 5.3: Pattern classification viewed as mapping a feature space, V, into a decision space, Q. The division of the square can be represented by the given classification tree. Here for a two-dimensional feature space. With inspiration from [8].

At the expense of more computation, the training set can be divided into a number of cross-validation samples, v, and then one sample can be used to test the performance of the classifier on the remaining $(v-1)$ samples and finally average the v such estimates. When choosing the number of cross-validations, it is important to notice that in [8] they have tested the adequate number of partitions of the sample, v. Adequate accuracy was gained with a 10-fold cross-validation, $v = 10$. In some cases, smaller values of v has given adequate accuracy, but no situations have ever implied that taking a value of v larger than 10 would significantly improve the accuracy for the selected tree. Therefore $v = 10$ in this framework to ensure an adequate accuracy in the cross-validation without it being to computationally time consuming with a large dataset.

In Matlab the `test` function can be specified to either test or cross-validation. For the training set, the test version is calculated. This results in a cost vector, the standard error of each cost value, a vector containing the number of terminal nodes for each subtree, and a scalar containing the estimated best

level of pruning. When cross-validation is chosen, the function uses the 10-fold cross-validation to compute the cost vector by partitioning the sample into 10 subsamples, chosen randomly with roughly equal size and class proportions. For each subsample, the function fits a tree to the remaining data and uses it to predict the subsample. It pools the information from all subsamples to compute the cost for the whole sample. The same values are generated for the test set as for the training set, but using the best level from the cross-validation of the training set to decide the point of the smallest tree within 1 standard error of the minimum cost tree.

The best level of pruning, derived from the cross-validation, can then be used to specify the pruning level in the `prune` function. This function takes the classification tree of input and prunes it to a specified level. In this work, the best level of pruning is used. If this level is 0, no pruning occurs. Classification trees are pruned by first pruning the branches giving less improvement in error cost.

Chapter 6

Description of the Classification System

The sequence of the classification framework, used in this work, is described in this chapter along with the performance measures that are used to evaluate the classifications. The description is supported by a flowchart very shortly presenting the steps of the framework in order to clarify the steps along with their input and output.

6.1 Classification Framework

The basic outline is run from the function `run_test`. This file starts by running the function `matlabsetup` which is used to add and remove Matlab and Java paths for the current workspace and to check for specific built in Matlab toolbox versions. The `run_test` file is specified for each new test bench in order to perform the correct steps for that certain test bench. All of the following functions and files will automatically run when the `run_test` function is executed. Then a framework of generating sounds, extracting features, generating classifier data and running classifier is initiated with the function `generate_sounds`. This function calls the m-files in the sound_specs folder. These files are created as functions to call the acoustic simulator tool and create output structures that can be used further on to generate sound files in .wav format. These sounds are then used to extract features using the openSMILE toolbox in `generate_features`. The feature extraction is based on the configure file chosen from the setup in the toolbox. First of all it is tested using the

emo_large.conf file which extracts 6669 features. The features extracted are saved in .csv format to further use in the generation of the classification data. This data is used in `generate_classifier_data` to make a .mat file containing all samples, the features calculated for each sample and a label marking if the file is from a car environment or not. If the feature file begins with "car_" a vector is generated containing ones in the same number of rows as the feature file contains otherwise a row of zeros is created. This is used further on to load all data and create a test and training split in `generate_classifier_input`. This split is used in `run_classifier` to create a set of training data and test data and from this using the `classregtree` function define which of the features that are relevant for a correct classification of the given test data set. Here a cross-validation is also performed in order to find the best combination of features for a potential pruning of the classification tree. From this a confusion matrix is calculated in `analyze_data` to find the correct and false classification rate for the two possible classes. Finally all the features used for classification are extracted and plotted using `run_feature_extraction`. A visualisation of the sequence of the framework can be seen in Figure 6.1.

For some of the tests, the framework is altered a bit. This goes for example for the tests where one single dataset is being created (the last ones of the conducted tests, see Section 7.2). In these cases, the framework works the same way until the function `generate_classifier_data` where a .mat file is created, but all the functions up till this point is run both for a predefined training and test set. From here on the function `generate_classifier_input_single_dataset` is used to create a single input file for the `run_classifier` function where the split into training and test data was decided already before any of the functions are run. The predetermined split is still used in the `classregtree` function to define which features that are relevant for a correct classification of the given test dataset. A possible pruning is still of interest in the single dataset tests along with the confusion matrix and extraction and plots of the relevant features. In a final test the features from the pruning are the only ones in focus, so here feature extraction is only based on these features. There is no point in pruning the resulting classification tree from this test further why pruning is disabled in this test and no cross-validation is performed.

Note the acoustic simulator tool can only be used when the computer is connected to the network at Oticon. If the user is not connected to this network, only sounds that have already been generated can be used. In this situation, the entire `run_test` should not be executed, but a sound_files folder should already be available containing a number of .wav files and then the rest of the functions can be run one by one from `generate_features` until `run_feature_extraction`.

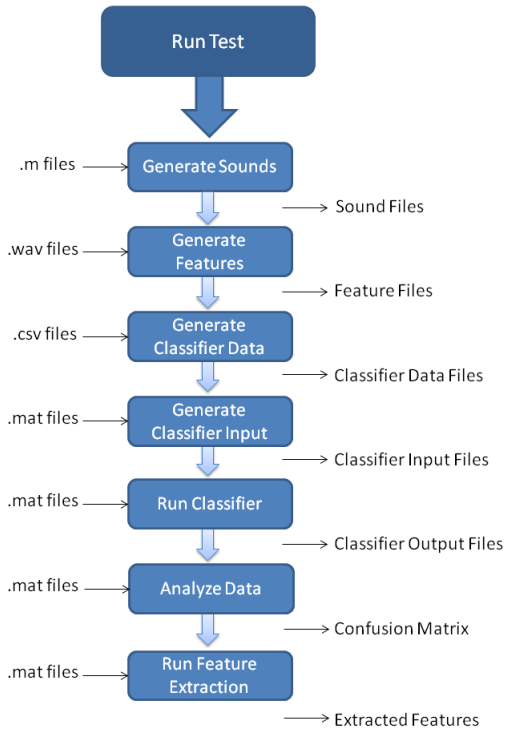


Figure 6.1: The sequence of the classification framework going from generation of sounds over feature extraction to the classification of the sounds and further analysis of the classified signals.

6.2 Performance Measures

To be able to compare the outcome of the classifying system within different tests and with other classifying systems, it is important with some standardised measures of how the system performs and for this, parameters that define the performance of the system are needed. In sound environment studies, there is no golden standard of how to present the outcome of the classifying system. Displaying scores of a classifier can be done in many ways, and some of the most used ways, as seen in Chapter 3, are classification error/recognition rate or by hit rate, overall hit rate and false alarm rate. A confusion matrix, and scores like sensitivity and specificity can be also be used.

6.2.1 Classifiaction Rate

The correct classification rate and classification error rate are measures of how well the classification system fits the data. A classification system strives to obtain good classification wherefore the correct classification rate and classification error rate are often used as measures of how well the system performs. The correct classification rate, p, for a test set with N sound signals can be calculated as [31]:

$$p = \frac{1}{N} \sum_{n=1}^{N} l^{(n)}, \tag{6.1}$$

where $l = 1$ if the sound signal is correctly classified and 0 otherwise. The classification error rate can then be calculated as $1 - p$, so the two measures are said to be complementary, and therefore it does not matter which of these two that are used.

A test set needs to be large in order for a classification rate to be calculated correctly [28]. The estimate for a classification count has a binomial(N, p) distribution. Therefore it is possible to calculate the standard error bars as

$$SE = 2 \cdot \sqrt{\frac{p \cdot (1 - p)}{N}} \tag{6.2}$$

6.2.2 Confusion Matrix

In a confusion matrix it is easy to visually represent the outcome of a classification system. All correct classifications are located in the diagonal of the matrix, and a misclassification will then be represented by any non-zero value outside the diagonal, in this way it is easy to see if the system is confusing two classes. It is desirable to know which classes are confused, which can be found by the calculation [28]:

$$e_{ij} = \mathbf{P}\{\text{decision } j \mid \text{class } i\} \tag{6.3}$$

A visual representation of a confusion matrix for a two class classification system can be seen in Figure 6.2. Here a correct classification is marked with a darker blue color in the diagonal of the matrix.

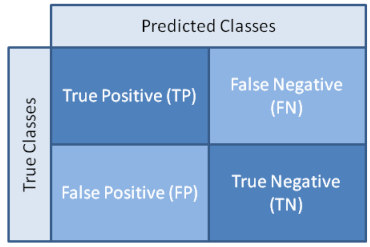


Figure 6.2: Confusion matrix for a two class system. In the confusion matrix, the rows represent the true classes and the columns the predicted classes.

6.2.3 Sensitivity and Specificity

From the rates in the confusion matrix it is possible to calculate two often used statistical measures of the performance, namely the sensitivity and the specificity. These measures are often used when presenting medical data, since sensitivity describes how good a test is at detecting what is being tested for, so how well a test detects a medical condition and specificity is the opposite of sensitivity, that is it describes how many of the true negatives in a test that are detected. For any test, there is usually a trade-off between the measures. The equations for the measures of the two are as follows

$$Sensitivity = \frac{TP}{TP + FN} \qquad Specificity = \frac{TN}{FP + TN} \tag{6.4}$$

If the values in the confusion matrix is given in percentages, the two diagonal entries correspond to the sensitivity and specificity respectively.

The true positive (TP) and the true negative (TN) values can also be expressed as the number of correctly classified sounds in the classes whereas the false positive (FP) and the false negative (FN) value also can be expressed as the number of sounds of other classes wrongly classified in this class. For each class a hit rate can be calculated, this indicates how many sounds are correctly classified out of the total number of sounds in that class. A false alarm rate can also be calculated for each class, here the number of sounds wrongly classified as a class out of the total number of sounds not coming from this class is calculated. The overall hit rate can then be found as the mean of the hit rates of all classes. Using the TP, TN, FP, FN values or the hit rate, false alarm rate and overall hit rate are thus two different ways of expressing the performance.

Chapter 7

Evaluation of the Classification System

In this section, a description of the tests that have been made during the development of the framework, will follow. These tests include a couple of preliminary tests that are conducted to clarify how the generation of the sounds handle the output and to optimize the settings for the sound files. The following will be tested

- Preliminary tests
  - a clarification of the number of channels that are used
  - the elimination of the possibility to use cues in the classification based on the target sources in the signals
  - the impact of the direction of the target source
- the importance of the split of data into training set and test set
- the influence of noise sources in all environments, both realistic and semi-realistic noise sources
- the influence of a scaling of the signals to obtain a predetermined SNR compared to signals with fixed target and noise levels for each source

7.1 Preliminary Tests

7.1.1 Number of Channels

Here it is tested whether the number of channels included has a saying in the results. When the sounds are generated, it is possible to specify the number of channels to include, four channels is the maximum possibility. As described earlier, the four channels to be included is front and rear in both left and right side. These correspond to the positions that have been measured on the HATS during the recordings of the impulse responses.

This test is important to clarify if only one channel or all the possible channels have to be included when generating the sounds in order to represent the environment as realistically as possible. It is not obvious from the documentation of the feature extractor if it extracts information from a single channel or from all the includes channels. This is important to clarify, since source direction will have more influence on the classification if all channels are included. If it turns out that only one of the channels is included and repeated in the three other channels, there is no point in including all four channels, and it is then possible to include omnidirectional noise, since there is no difference in how the sounds enter the hearing aid in the different channels. There is also a possibility that the classifier chooses the channel that gives the smallest classification training error. If there is a difference in the channels, and all four channels actually are represented with their difference, more caution has to be put into the generation of the signals, especially when it comes to the noise sources since it then will not be possible to include omnidirectional noise as easily as in the case where only one channel is included and repeated.

Small datasets were created from three situations, one including only one channel, the second has the first channel repeated in all four channels and the third have the channels set to get their input from each of their respective positions. These datasets were created to discover if a difference occurred between them. A difference between the last described situation and the others would mean that every channel is included, and the setup would resemble a realistic hearing aid situation which would be good for the reliability of the test but would make the addition of noise signals more difficult. A visual inspection of the time signals in all three situations were used along with the results in a confusion matrix from each of the situations. The signals can be seen in Figure 7.1 and the matrices in Table 7.1.

The number of channels included is tested using the small dataset described earlier. The test is conducted using the leave-one-out method, training on five

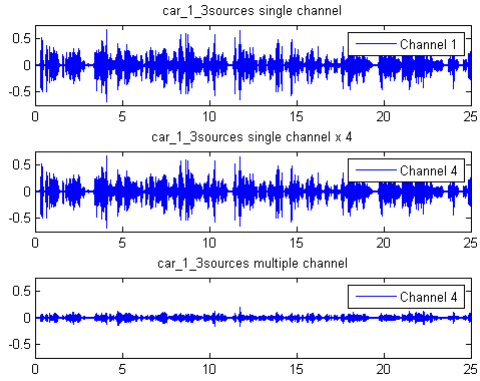


Figure 7.1: Time signals from three different situations, one including only one channel (top), one where the first channel was repeated in all four channels (middle) and one where the channels were set to get the input from each of their respective positions (bottom). One channel from each of the signals is shown.

sound signals and testing the sixth sound signal with this setup. From the sound signals and the confusion matrices it can be seen that there is a difference from the single channel to the multiple channel setup. The total scale of the signal was in this test set to be 65 dB at the eardrum, so when four channels are added instead of just one, the amplitude of each of the signals is decreased. Also the correct classification is decreased, apparently it becomes more difficult to distinguish a car environment from a non-car environment when all possible directions of the sound is included. Even though it results in lower TP value for the car, it is necessary to include all four channels from here on out to create the most realistic hearing aid sound setup since the feature extraction actually is based on all channels. This also means that adding noise becomes more difficult, and care must be taken when especially omni-directional noise is being simulated.

7.1.2 Elimination of Number of Speakers

The importance of elimination of number of speakers is tested using the small dataset described in Section 4.4. In the previous test, the number of speakers was not taken into account. This means that the number of speakers could have been a potential cue in the classification of the sound environments. To make sure this is not the case, the framework is altered such that when a sound

Table 7.1: Classification rates for the test of number of channels from three different situations. **Top:** including only one channel, **Middle:** the first channel was repeated in all four channels and **Bottom:** the channels were set to get the input from each of their respective positions.

	car	misc
car	0.8933 ± 0.2521	0.1067
misc	0.1333	0.8667 ± 0.2775

	car	misc
car	0.8933 ± 0.2521	0.1067
misc	0.1333	0.8667 ± 0.2775

	car	misc
car	0.8000 ± 0.3266	0.2000
misc	0.1333	0.8667 ± 0.2775

signal with one source is used for testing, no other sound signals with only one source is used for the training of the classifier etc. So instead of training with the leave-one-out method, the classifier is trained by a "leave all other sound signals with the same number of speaker sources out" principle. The confusion matrix for the multiple channel setup tested in the previous subsection should be compared to the one trained with the new method proposed here. The two matrices can be seen in Table 7.2.

Table 7.2: Classification rate for the test where number of speakers is eliminated as a cue option. **Top:** Multiple channel obtained with the leave-one-out method. **Bottom:** Multiple channel with the "leave all other sound signals with the same number of speaker sources out" principle.

	car	misc
car	0.8000 ± 0.3266	0.2000
misc	0.1333	0.8667 ± 0.2775

	car	misc
car	0.8667 ± 0.2775	0.1333
misc	0.0933	0.9067 ± 0.2375

From the confusion matrices it can be seen that eliminating the number of speakers as a possible classification cue results in a higher correct classification rate for both the car and the miscellaneous (misc) situation. This result turns out very beneficial since the input to the classifier from here on has to be separated in this way, to make sure that it is not the different number of speakers that gives rise to the correct classification rate.

7.1.3 The Impact of Target Direction

The impact of target direction is tested by using the environments where a car situation is possible to simulate, that is placing one target source to the left of the listener and the possibility to include two noise sources, one placed behind the listener and one placed behind the target source. This setup is possible in four of the environments; Café, Car, Japan North and Staircase. In these four environments two sets of sound signals are generated, one where the speakers are placed as in the car, and one where, in the three environments that are not a car, the speakers are rotated relative to the listener so the target source is in front of the listener, and the two possible speaker noise sources are placed one behind the target source and one next to the first noise source respectively. Calculations for these situations have been made in order to compare if it becomes easier to classify the car environment if all the speaker sources are not placed as in the car, but a bit more realistically considering the environment they are placed in. The confusion matrices for both the situations can be seen in Table 7.3.

Table 7.3: Classification rate for testing the target direction. **Top:** Placement of the speakers is as similar as possible to the possible placements in a car. **Bottom:** Direction of the target is in front of the listener.

	car	misc
car	0.7867 ± 0.2365	0.2133
misc	0.0489	0.9511 ± 0.1245

	car	misc
car	0.8800 ± 0.1876	0.1200
misc	0.0978	0.9022 ± 0.1715

It can be seen from the results of the two setups, that placing the target source in front of the listener and the speaker noise sources according to the target source, improves the correct classification rate of the car environment compared to when all speakers are placed exactly as in the car. This means that the placement of the sources in the car environment is an important factor in classifying this environment. This makes good sense and was also expected, since the location of the speakers and noise sources is well defined in the car environment compared to other environments, there are not many alternative ways for the speakers to be placed, so of course this is an important factor to exploit when looking for car environment classification.

7.2 Single dataset

Until the last of the mentioned preliminary tests was conducted, the placement and the number of talkers was the exact same as in the car, or was resembling as much as possible, given the certain environment structure. From here on, the placement of both the target, the speaker noise sources and all other noise sources will be considered from situation to situation in order to create as realistic environments as possible. Furthermore, the framework is altered again such that it is possible to create one single dataset as input for the classifier. The split into training and test set is conducted before the creation of the single dataset such that none of the sound signals used for the training set is repeated in the test set. The datasets and the split into training and test set are as described in Section 4.4.

Splitting in this way is necessary since sound files only exist from one car situation. It would be preferable if more then one car situation existed, since then it would be possible to run the framework with a "leave-one-environment out" method. But since impulse response recordings have only been made for one car situation, training of the car environment must come from the same car situation as is used for testing.

7.2.1 Test of the Scaling of the Sound Signals at the Eardrum

When creating the sound signals, it is possible to specify both the overall input level at the eardrums along with a signal to noise ratio. This is described in detail in Chapter 4. In the following, it will be tested what this scaling does for the classification of the signals. First of all, a scenario with fixed target and noise levels for each source is presented, here there is neither specified an overall input level nor a SNR. In this case the overall level depends on the number and placements of the target and noise sources. Two other cases is studied as well, both where an overall input level is set to 65 dB, one with an SNR of 0 dB and one with an SNR of 10 dB.

7.2.1.1 Fixed Target and Noise Levels for Each Source

Making sure that no overall input level at the eardrum is defined and that no SNR is specified, makes the signals a combination of the sources, and any level of noise depends only on how many noise sources is included, their specified

level and their placement. Creating signals in this way and testing them with the framework result in the following confusion matrix.

Table 7.4: Testing the situation with fixed target and noise levels for each source

	car	misc
car	0.9160 ± 0.0469	0.0840
misc	0.0356	0.9644 ± 0.0313

7.2.1.2 Final SNR Set to 0 dB

Setting the overall input level at the eardrum to 65 dB and the SNR to 0 dB makes the signals a great confusion between the speech target and the noise signals. Focus in this case will be unclear since the SNR forces the levels of both target and noise to be equal. This results in the following confusion matrix.

Table 7.5: Testing the situation where the overall input level is set to 65 dB with a SNR of 0 dB

	car	misc
car	0.8027 ± 0.0673	0.1973
misc	0.0629	0.9371 ± 0.0410

7.2.1.3 Final SNR Set to 10 dB

Setting the overall input level at the eardrum to 65 dB and the SNR to 10 dB makes the signals focus much more on the speech target than on any of the noise signals. This results in the following confusion matrix.

Table 7.6: Testing the situation where the overall input level is set to 65 dB with a SNR of 10 dB

	car	misc
car	0.8720 ± 0.0565	0.1280
misc	0.0516	0.9484 ± 0.0374

A scaling of the overall input level at the eardrum is not the most realistic way to represent how a hearing aid processes the sounds, but in some cases it can be useful when trying to simulate certain situations. From the results above it can be seen that the situation with fixed target and noise levels for

each source results in the highest TP value for the car of 91.6% compared to only 80.27% and 87.2% in the other tests. The combination of all the possible noise signals stands more out when there is no limit set for the overall level of these. This gives a more realistic representation of the environments, since all the environments are created to make it possible for the classifier to distinguish between realistic environments. Testing the possibility of specifying the overall input level and a SNR results in the conclusion that this is not beneficial for the correct classification rate. Creating signals with fixed target and noise levels for each source gives a much higher correct classification and is therefore considered as the best possible way to represent the different environments. This is also the most realistic way to represent these. From this test it is decided that only situations with fixed target and noise levels for each source are considered in the final framework and in any of the following tests.

7.2.2 Further Analysis of the Situation with Fixed Target and Noise Levels for Each Source

From the previous test, it is obvious that the situation with fixed target and noise levels for each source results in the highest correct classification rate. This situation will therefore be used for the further investigations of the possibilities for the classification. First of all the training set will be analysed with a cross-validation to find the best possible feature set for a pruning of the classification tree and to reduce the number of features considered.

For the single dataset with fixed target and noise levels for each source, a test is conducted to find if it is possible to prune the feature set and by this reduce the number of terminal nodes in the classification tree. For this, a cross-validation is used, see Section 5.2.2. Calculating a cross-validation for the tree can be used to decide if the tree should be pruned. When a cross-validation is used, three error rates can be plotted, for train data, validation data and test data, these error rates can be seen in Figure 7.2. In the figure, the solid line shows the estimated cost for each tree size, the dashed line marks one standard error above the minimum, and the red square marks the smallest tree under the dashed line. In the plot of the test set, the red square marks the smallest tree from the cross validation of the training set. Both of these squares are set to mark where a possible pruning of the tree could result in a tree with less calculations (minimum cost) and therefore also fewer terminal nodes.

The error rate for the train data decreases with a higher number of terminal nodes. The error rate is a measure of how well the classifier performs with the specified number of terminal nodes. A low error rate is preferable since this indicates an improvement in the classifier, but the more terminal nodes that are

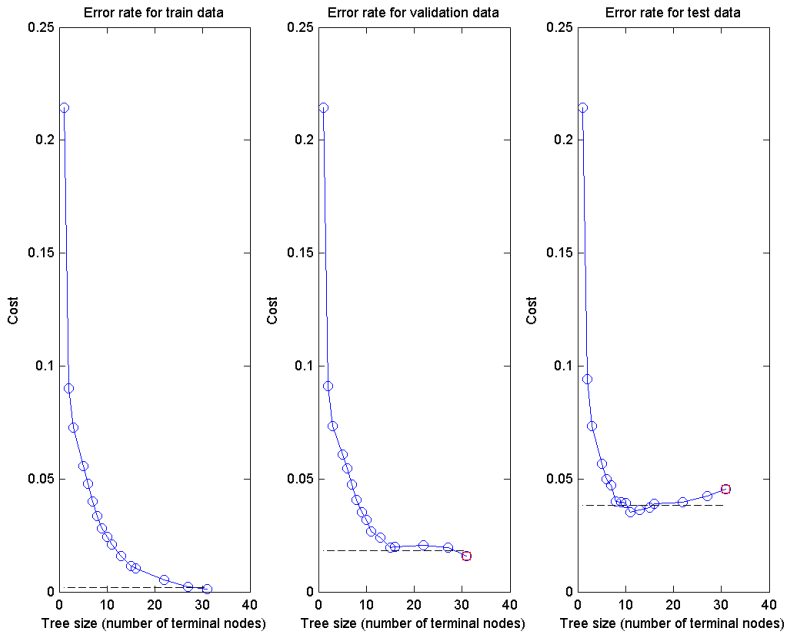


Figure 7.2: **Left:**Plot of the error rate for the training set, **Middle:** the validation set, **Right:** for the test set for the situation with fixed target and noise levels for each source. The solid line shows the estimated cost for each tree size, the dashed line marks one standard error above the minimum, and the red square marks the smallest tree under the dashed line for the validation set.

included in the tree, the higher is the computational cost. The cross-validation is therefore used to see if a pruning of the tree can reduce the number of terminal nodes while still getting an acceptable low error rate. It can be seen in this case presented here, that it is not possible to prune the tree to get a lower acceptable error rate, the largest tree considered is the only tree where the error rate falls beneath the one standard error above the minimum for the validation set.

The curve for the training set behaves as expected, with a level in the beginning corresponding to chance, since the car environment forms 21.43% of the training data. Increasing the tree size then decreases the error rate for each step as expected. The validation set behaves almost in the same way, but an increasing error rate can occur with an increase in tree size, this all depends on which part of data is used for the subsamples that are validated. Looking at the test set

rate could give rise to a suspicion that a pruning of this tree could be beneficial. This is difficult to be sure of, and the number of terminal nodes is in this case already fairly small, so a pruning is not conducted in this case.

The test of possible pruning was conducted for two other situations as well, the situations also considered in the test of scaling. The error rate plots for these tests can be seen in Appendix C. For the examples shown in the appendix, both of the cases would result in a pruned tree in order to reduce the cost by reducing the number of terminal nodes.

Deciding that no pruning is necessary in this case, makes it possible to move on to look at the features selected for the classification. 30 features are selected for the feature set. A full list of these features can be seen in Appendix C along with visual representations of all of them. Here the three most important features are shown in order to give an idea of how the values are for the features in the sound files from the different environments. These can bes seen in Figure 7.3 and Figure 7.4.

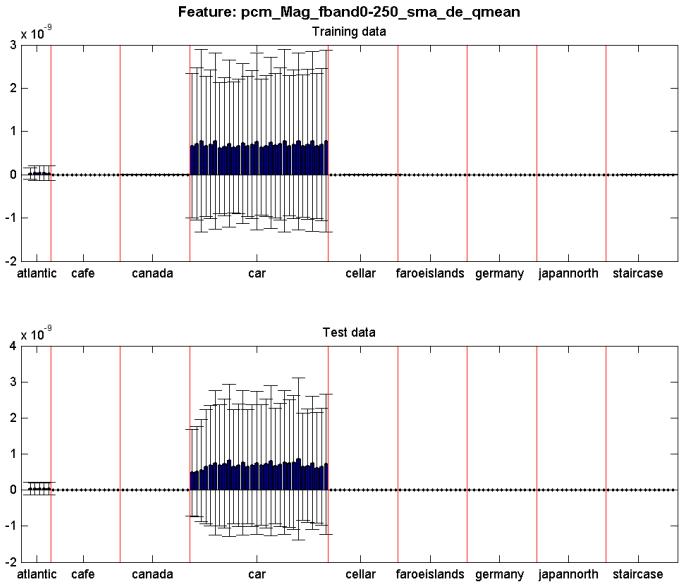
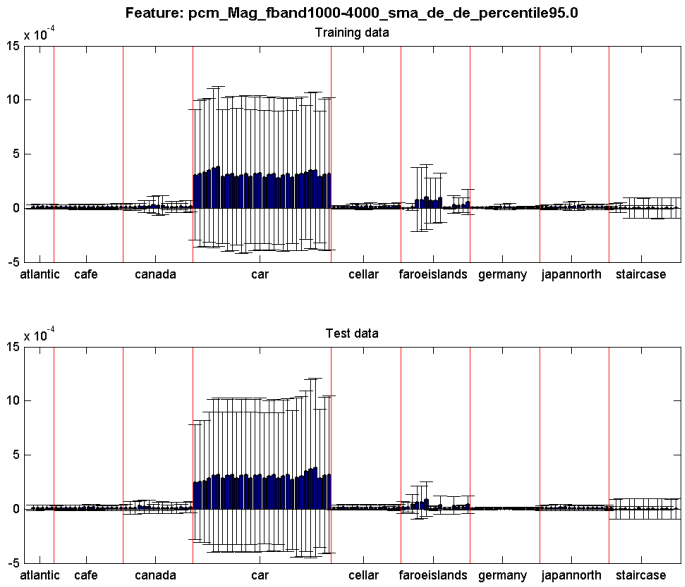
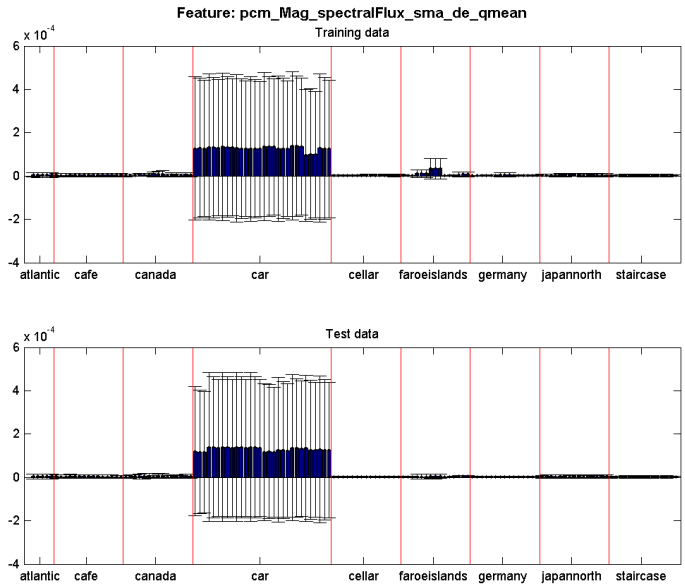


Figure 7.3: Feature used for the first split



(a) Feature used for splitting the left child node from the first split



(b) Feature used for splitting the right child node from the first split

Figure 7.4: Feature number 2 and 3

From the three most important features a clear difference can be seen between the car environment and the other environments. When looking at the feature set it becomes clear that the first three features all are spectral features. These are used for the first splits and using these thus reduces the impurity of the first nodes the most. The dataset is designed such that the speaker sources are the same in all files, the noises sources are somewhat the same depending on the environment and the biggest difference in the sound files are the environments themselves (in form of the impulse responses used). One of the big differences between especially the car environment compared with the other environments is the reverberation time of this environment. For a car, the reverberation time is very short compared to any other of the included environments. A car is a small environment with many different interior materials. But since most of the noises in a car does not have directional cues for the human ear (mostly low frequency noise occurs in a car) they tend to not cause many reflections that a human ear would catch. The reverberation time is therefore an important factor in differentiating between a car environment and the other environments, and since an environment with short reverberation time has a lower signal energy than an environment with a long reverberation time (because of the tail in the signal), spectral features can be used to distinguish between these situations. From the first two features, it seems that a difference especially can be seen in the frequency bands 0-250 Hz and 1000-4000 Hz.

The Mel-frequency spectra and the MFCCs seem to provide some important information as well (feature number 4-6). These measures, along with the zero-crossing rate (feature number 7), are often used in speech/music/noise classification. Since the noise sources in most of the environments contain speech, music and pure noise signals, it makes good sense that the given features are important for the classification. The car environment includes noise sources that are not included in any of the other environments. This is in particular the low-frequency noise from inside a moving car. Even though many different kinds of noise sources are placed in all the environments, those in the car stand out enough for the features to catch the differences and use this in the classifying process.

In general, the features used for classifying the car environment from the other environments revolve around certain measures. These are summarised in Table 7.7.

7.2.3 Test of Specified Features

The best feature set only contains 30 features out of 6669 possible features. There is no point in calculating the values for all the other features when these

Table 7.7: The important features for use in car environment classification

Spectral Features
Spectral frequency band energy (0-250 Hz)
Spectral frequency band energy (1000-4000 Hz)
Spectral flux
Spectral centroid
Spectral maximum position
Mel spectrum features
MFCCs
Zero-crossing rate
Loarithmic energy

30 features are sufficient for the classification. Thus, in order to minimize the calculation time and computational cost, the framework is once again expanded so it is possible to specify a feature set that should be used when the sound signals are classified. This expansion will be useful in a situation where a new test set is to be classified, since there is no point in extracting multiple features for a new test set when a feature set already has been defined. This implementation will especially come in handy when the framework is expanded to the ability of classifying more specific environments and not just car vs. misc (as it is now it is only beneficial if it is this specific classification that is of interest).

In this following test, the features chosen are the 30 features from the feature set found in the previous test. It is specified that only these features should be taken into account, and this time there is no interest in a possible pruning level, since all the features selected are the relevant ones for this specific training set. To see how the classifier performs with this specified feature set, the error rate for the training and test data is plotted and can be seen in Figure 7.5. The confusion matrix for this test is also calculated and can be seen in Table 7.8.

Table 7.8: Classification rate for the no scale test where the features are reduced to only include the specific feature set specified by the previous test.

	car	misc
car	0.9107 ± 0.0482	0.0813
misc	0.0313	0.9687 ± 0.0294

The classification tree is now only build from the specified features and in this situation the classifier finds it is necessary to include some of the features more than once and leave some of the features out. The full list of features can be seen in Appendix C and it can be seen that up to and including feature number 9, the

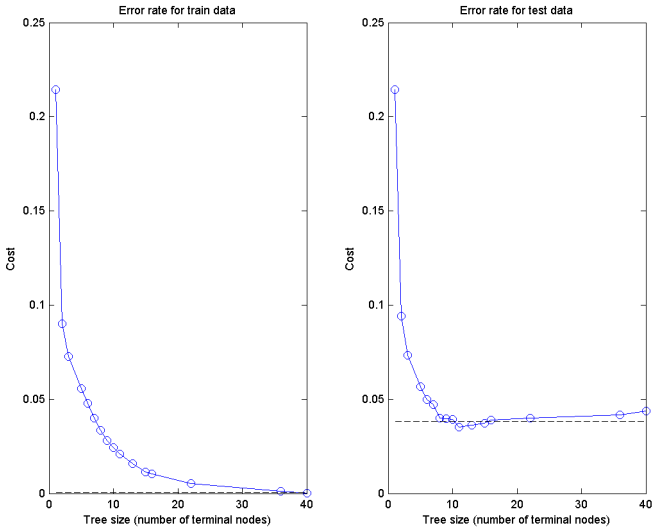


Figure 7.5: Left: Plot of the error rate for the training set **Right:** for the test set for the test where the features are reduced to only include the relevant feature set. The solid line shows the estimated cost for each tree size, the dashed line marks one standard error above the minimum.

two feature sets are identical. Including the features more than once result in a lower TP value for the car and a higher TN value. This was not expected since the test set is the exact same as was used in the further analysis of the situation with fixed target and noise levels for each source. The classification rates were expected to have the same values for the two tests. The small deviation could arise from the dataset from which the features are calculated. For the previous test, the calculations were based on the cross-validated data whereas here they are based on the training data. Only very small differences arise from the different treatment of the training data, but it seems that these small differences are enough for the classifier to find it necessary to include some of the features more than once which then make the classification rates deviate a little from each other. Since the deviation is so small, this way of reducing the feature input is still interesting and definitely something worth keeping in the final framework because of the time saving and reduction in computational cost. Especially the cross-validation is very time-consuming but not necessary when the feature set already is found.

Chapter 8

Conclusion

In this thesis a framework has been build to identify the different steps in a sound environment classification system. The framework is build as a standard classification system and goes through the steps of generating sounds, extracting features, classifying the sounds and analyse the classifications. The framework is build for classification between two classes as it is, namely the car environment and miscellaneous environments, but it is build with the intention of expanding the framework in the future to make it possible to distinguish between more classes.

For feature extraction, a configuration using the openSMILE [12] toolkit was implemented. This made it possible to extract 6669 low-level features and in this way made it possible to investigate, from a broad variety, which features were important in this sound environment classification. It turned out that spectral features, Mel-frequency scale spectrum features, MFCCs, zero-crossing rate and logarithmic energy were the most important ones. This arises from the nature of the tested sounds, since the differences in the sound files from the car environment to the other environments are found to be biggest in the environments themselves and in the possible sound sources in the environments.

A classification tree was used as the classifying algorithm, and this turned out to be a good match for the large feature set, since this made it easy for the system to decide which features were important.

Preliminary tests were made in order to learn how the features were extracted and to prepare the framework for the final build. It turned out that the best classification rates were obtained with a framework were all used sounds were

as true to their environment as possible and where the target and noise sources had fixed levels when creating the sounds. Using the framework with the best settings resulted in a sensitivity of $91.6\% \pm 4.69\%$ and a specificity of $96.44\% \pm 3.13\%$.

Despite of the results, an expansion of the framework is still recommended before an implementation in a hearing aid.

Future Work

The work in this thesis give rise to a number of ideas for further improvements. First of all an expanded sound database including realistically recorded noise sources for a car situation would be of great use along with more than one car situation. A flexibility for moving the HATS according to target direction would also give more realistic setups.

When it comes to the framework, it is build so it is possible to easily extend, both when it comes to sound environments and features. Features are investigated closely in this work, but an extension of classifier investigation could also be interesting to maybe identify a classifier that could result in even better classifications. The framework is robust in classifying car from miscellaneous sounds, but this should be expanded to make it possible to classify even more sound environments. This framework is not ready to implement in a hearing aid before it can classify several sound environments.

The next step in this framework would be to implement an identification of the sources present in the environment (people, noise etc.), what these sources express (speech, music, noise) and where are they placed relative to the listener.

Bibliography

[1] Bigpond health. Homepage: http://www.virtualmedicalcentre.com/anatomy/ear/29#C34, Last accessed 1st of May 2012.

[2] Digital hearing care. Homepage: http://www.digitalhearingcare.org.uk/blog/index.php/tag/oticon/, Last accessed 1st of May 2012.

[3] Private communication. with F. Eyben (author of openSMILE), April 2012.

[4] E. Alexandre, L. Cuadra, and R. Gil-Pita. Sound classification in hearing aids by the harmony search algorithm. *Music-Inspired Harmony Search Algorithm, Studies in Computational Intelligence*, 191:173–188, 2009.

[5] E. Alexandre, L. Cuadra, M. Rosa, and F. López-Ferreras. Feature selection for sound classification in hearing aids through restricted search driven by genetic algorithms. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(8):2249–2256, 2007.

[6] R. Arora and R. A. Lutfi. An efficient code for environmental sound classification. *J. Acoust. Soc. Am.*, 126(1):7–10, 2009.

[7] A. P. Bjerg and J. N. Larsen. Recording of natural sounds for hearing aid measurements and fitting. Master's thesis, Technical University of Denmark, 2006.

[8] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Chapman & Hall/CRC, 1st edition, 1984.

[9] M. Büchler, S. Allegro, S. Launer, and N. Dillier. Sound classification in hearing aids inspired by auditory scene analysis. *EURASIP Journal on Applied Signal Processing*, 18:2991–3002, 2005.

[10] S. Chu, S. Narayanan, and C.-C. J. Kuo. Environmental sound recognition with time-frequency audio features. *IEEE Transactions on Audio, Speech, and Language Processing*, 17(6):1142–1158, 2009.

[11] L. Cuadra, R. Gil-Pita, E. Alexandre, and M. Rosa-Zurera. Joint design of gaussianized spectrum-based features and least-square linear classifier for automatic acoustic environment classification in hearing aids. *Signal Processing*, 90:2628–2638, 2010.

[12] F. Eyben, M. Woellmer, and B. Schuller. openear - introducing the munich open-source emotion and affect recognition toolkit. In *Proc. 4th International HUMAINE Association Conference on Affective Computing and Intelligent Interaction 2009 (ACII 2009), IEEE*, Amsterdam, The Netherlands, 2009.

[13] D. Fabry and J. Tchorz. Results from a new hearing aid using "acoustic scene analysis". *The Hearing Journal*, 58(4):30–36, 2005.

[14] G. Keidser. Many factors are involved in optimizing environmentally adaptive hearing aids. *The Hearing Journal*, 62(1):26–31, 2009.

[15] S. Kochkin. Marketrak viii: Consumer satisfaction with hearing aids is slowly increasing. *The Hearing Journal*, 63(1):19–32, 2010.

[16] L. Lamarche, C. Giguère, W. Gueaieb, T. Aboulnasr, and H. Othman. Adaptive environment classification system for hearing aids. *J. Acoust. Soc. Am.*, 127(5):3124–3135, 2010.

[17] Y. Lavner and D. Ruinskiy. A decision-tree-based algorithm for speech/music classification and segmentation. *EURASIP Journal on Audio, Speech, and Music Processing*, 2009, 2009.

[18] S. M. Lee, J. H. Won, S. Y. Kwon, Y.-C. Park, I. Y. Kim, and S. I. Kim. New idea of hearing aid algorithm to enhance speech discrimination in a noisy environment and its experimental results. *Proceedings of the 26th Annual International Conference of the IEEE EMBS*, pages 976–978, 2004.

[19] B. C. J. Moore. *An Introduction to the Psychology of Hearing.* Academic Press, 5th edition, 2003.

[20] J. Moragues, A. Serrano, L. Vergara, and J. Gosálbez. Acoustic detection and classification using temporal and frequency multiple energy detector features. *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP) 2011*, pages 1940–1943, 2011.

[21] A. B. Nielsen, L. K. Hansen, and U. Kjems. Pitch based sound classification. *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP) 2006*, 3:788–791, 2006.

[22] P. Nordqvist. *Sound Classification in Hearing Instruments*. PhD thesis, Royal Institute of Technology, Sweden, 2004.

[23] P. Nordqvist and A. Leijon. An efiicient robust sound classification algorithm for hearing aids. *J. Acoust. Soc. Am*, 115(6):3033–3041, 2004.

[24] M. P. Norton and D. G. Karczub. *Fundamentals of Noise and Vibration Analysis for Engineers*. Cambridge University Press, 2nd edition, 2003.

[25] D. OŚhaughnessy. *Speech communication: Human and Machine*. Addison-Wesley, 1987.

[26] M. S. Pedersen and T. Kaulberg. *Car Recordings*, 2010. http://p4db/specialFileView.cgi?TYPE=WINWORD&FSPC=//depot/projects/greenhouse/doc/tracks/reverberant%5froom%5frecordings/HX%5f20101116%5fcar%5frecordings.doc&REV=3.

[27] V. Peltonen, J. Tuomi, A. Klapuri, J. Huopaniemi, and T. Sorsa. Computational auditory scene recognition. In *Proc. of ICASSP*, Florida, USA, May 2002.

[28] B. D. Ripley. *Pattern Recognition and Neural Networks*. Cambridge University Press, 1st edition, 1996.

[29] A. Schaub. *Digital Hearing Aids*. Thieme, 1st edition, 2008.

[30] R. R. Seeley, T. D. Stephens, and P. Tate. *Anatomy & Physiology*. McGraw-Hill, 7th edition, 2006.

[31] S. Sigurdsson. *A Probabilistic Framework for Detection of Skin Cancer by Raman Spectra*. PhD thesis, Technical University of Denmark, 2003.

[32] J. Skovgaard. *Measure Setup Japan North*, 2010. http://p4db/specialFileView.cgi?TYPE=WINWORD&FSPC=//depot/projects/greenhouse/doc/tracks/reverberant%5froom%5frecordings/Measure%20setup%20Japan%20North.doc&REV=1.

[33] J. Skovgaard and M. S. Pedersen. *Reverberant Room Recordings*, 2006. http://p4db/specialFileView.cgi?TYPE=WINWORD&FSPC=//depot/projects/greenhouse/doc/tracks/reverberant%5froom%5frecordings/GR%5f080102%5fImpulse%5fresponses%5fsetup.doc&REV=8.

[34] J. Xiang, M. F. McKinney, and K. F. andT. Zhang. Evaluation of sound classification algorithms for hearing aid applications. *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP) 2010*, pages 185–188, 2010.

[35] H. Zhang, N. Nasrabadi, T. S. Huang, and Y. Zhang. Transient acoustic signal classification using joint sparse representation. *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP) 2011*, pages 2220–2223, 2011.

[36] F. Zheng, G. Zhang, and Z. Song. Comparison of different implementations of mfcc. *J. Computer Science & Technology.*

Appendix A

Matlab Scripts

All commented Matlab scripts, and sound files that are needed to run the scripts if the reader is not placed at Oticon while running the framework, can be found in the zip-file uploaded along with the thesis. The files have been saved in the folders from which they have to be run. All scripts (except for the run_test functions and the files used to generate the sound signals (sound_specs folder)) must be placed in the main folder. An addpath has to be run to add the path to the main folder, this can be done from `initialize`. In each of the test bench folders the specific `run_test` function has to be placed (along with a folder called sound_files if the reader is not placed at Oticon while running the framework).

For the tests with a single dataset, only the sound files from the no scale test is included in the zip-file since the sound files take up a lot of space. With these sound files it is thus possible to run both the "single_data_set_no_scale" test and the "select_features_no_scale" test.

Appendix B

Speaker Signals, Noise Sources and Positions

Information is provided for the speaker signals, for all realistic noise signals and finally the setup of all the combinations of sound signals in all the environments are provided.

B.1 Speaker Signals

The target source is the same in every sound file. It has the following specifications:

Length:	151.7626 s
Sampling rate:	44100 Hz
Resolution:	16 bits
Number of Channels:	1
Filename:	VWA-HP_-6dB.wav
Title:	EnglishSpeakers - VWA-HP_-6dB
Description:	English monologues, some with raised effort
Sound ID:	GR_02699
Sound folder:	english_speech

Table B.1: Information about the target speaker signal

In some of the sound files, one or both of the following speaker noise signals are included. They are a part of the same dialogue and have the following specifications:

Length:	162.6326 s
Sampling rate:	44100 Hz
Resolution:	16 bits
Number of Channels:	1
Filename:	VWA_0dB.wav
Title:	EnglishSpeakers - VWA_0dB
Description:	English monologues, some with raised effort
Sound ID:	GR_02701
Sound folder:	english_speech

Table B.2: Information about the first of the possible speaker noise signals

Length:	162.6326 s
Sampling rate:	44100 Hz
Resolution:	16 bits
Number of Channels:	1
Filename:	VWA_0dB_Comp.wav
Title:	EnglishSpeakers - VWA_0dB_Comp
Description:	English monologues, some with raised effort
Sound ID:	GR_02702
Sound folder:	english_speech

Table B.3: Information about the second of the possible speaker noise signals

B.2 Possible Noise Signals - ICRA2 files

Those of the ICRA2 sound files that make sense in anyway to include in any of the environments are listed below. [7]

GR_00804: Catina HATS
GR_00810: Hair dryer HATS
GR_00814: Vacuum Cleaner HATS
GR_00820: Industrial Dishwasher HATS
GR_00826: Traffic Noise (High Intensity) HATS
GR_00832: Ventilation HATS
GR_00835: Bathwater HATS
GR_00838: Coffee Machine HATS
GR_00844: Keyboard Typing HATS
GR_00850: Children Playing Inside HATS
GR_00871: Forest Birds (Very Soft) HATS
GR_00880: Classic Music HATS
GR_00883: Football Match (Stadium 6000-8000 People) HATS
GR_00886: Jazz Music HATS
GR_00889: Rock Music HATS

GR_00892: Soft Music HATS
GR_00895: Badminton Match (Inside) HATS
GR_00898: Choir In Church HATS
GR_00901: Choir In Church With Organ HATS
GR_00904: Flute (Fast) HATS
GR_00907: Flute (Soft) HATS
GR_00910: Car Slow In City HATS
GR_00919: Party (60 People) HATS
GR_00925: Car 60 KMT HATS
GR_00928: Car Accelerating HATS
GR_00931: Car Motorway HATS
GR_00943: Party Close With Music HATS

B.3 Noise Signals and Placement in the Environments

For each of the environments, a list of the nine combinations of sound signals are provided (the tenth always only include the speaker noise sources and none of the ICRA2 noises). The list is given as the combination presented with the name of the noise source followed by the position/positions of this source in a bracket (in some of the environments the positions are numbered, in others they are identified by their angles). The number of the positions corresponds to the numbers used in the .m files from which the sound signals are created, and these numbers corresponds to possible positions (where impulse responses have been recorded) in the given environment.

B.3.1 Atlantic

1. Includes noise sources in 1 position: Hair dryer (1)
2. Includes noise sources in 1 position: Vacuum Cleaner (1)
3. Includes noise sources in 1 position: Ventilation (1)
4. Includes noise sources in 1 position: Bathwater (1)
5. Includes noise sources in 1 position: Classic Music (1)
6. Includes noise sources in 1 position: Jazz Music (1)
7. Includes noise sources in 1 position: Rock Music (1)
8. Includes noise sources in 1 position: Soft Music (1)
9. Includes noise sources in 1 position: Flute (1)

B.3.2 Café

1. Includes noise sources in 10 positions: Cantina (1 and 23), Traffic Noise (6), Jazz Music(7), Industrial Dishwasher (19), Children Playing Inside (20), Forest Birds (20), Vacuum Cleaner (26), Coffee Machine (27), Ventilation (27)
2. Includes noise sources in 5 positions: Coffee Machine (20), Flute (22, 24 and 26), Ventilation (27)
3. Includes noise sources in 7 positions: Children Playing Inside (4, 6, 10, 12, 18 and 20), Soft Music (13)
4. Includes noise sources in 4 positions: Coffee Machine (7), Vacuum Cleaner (13), Ventilation (20), Industrial Dishwasher (26)
5. Includes noise sources in 10 positions: Cantina (1, 2, 3, 8, 9, 15, 16, 21, 22 and 23)
6. Includes noise sources in 7 positions: Forest Birds (6, 12, 20 and 26), Rock Music (7), Vacuum Cleaner (24), Ventilation (27)
7. Includes noise sources in 10 positions: Traffic Noise(6 and 26), Football Match (14, 15, 16, 17, 18, 19 and 20), Ventilation (27)
8. Includes noise sources in 10 positions: Party Close with Music (2, 3, 4, 5, 8, 9, 15, 16, 24 and 25)
9. Includes noise sources in 10 positions: Party (3, 4, 8, 9, 16, 19, 20, 22, 26 and 27)

B.3.3 Canada

1. Includes noise sources in 3 positions: Badminton Match (75, 90 and 105)
2. Includes noise sources in 10 positions: Soft Music (195, 210, 225, 240, 255, 270, 285, 300, 315 and 330)
3. Includes noise sources in 10 positions: Children Playing Inside (45, 90, 135, 195, 225, 270, 285, 300, 315 and 345)
4. Includes noise sources in 10 positions: Keyboard Typing (15, 45, 90, 105, 120, 135, 150, 165, 180 and 195)
5. Includes noise sources in 1 position: Coffee Machine (135)
6. Includes noise sources in 2 positions: Vacuum Cleaner (90 and 270)
7. Includes noise sources in 1 position: Hair dryer (15)
8. Includes noise sources in 5 positions: Classic Music (90), Vacuum Cleaner(135), Keyboard Typing (195), Coffee Machine (270), Hair dryer (345)
9. Includes noise sources in 10 positions: Party Close with Music (30, 60, 90, 120, 165, 195, 225, 255, 300 and 330)

B.3.4 Car

Including Realistic Noises:

1. Includes noise sources in 3 positions: Car Slow In City (1, 2 and 3)
2. Includes noise sources in 3 positions: Car 60 KMT (1, 2 and 3)
3. Includes noise sources in 3 positions: Car Accelerating (1, 2 and 3)
4. Includes noise sources in 3 positions: Car Motorway (1, 2 and 3)
5. Includes noise sources in 3 positions: Traffic Noise (1, 2 and 3), Forest Birds (1, 2 and 3)
6. Includes noise sources in 3 positions: Car Slow In City (1, 2 and 3), Traffic Noise (1), Rock Music (2)
7. Includes noise sources in 3 positions: Car 60 KMT (1, 2 and 3), Jazz Music (1, 2 and 3)
8. Includes noise sources in 3 positions: Car Accelerating (1, 2 and 3), Keyboard Typing (1, 2 and 3), Soft Music (1)
9. Includes noise sources in 3 positions: Car Motorway (1, 2 and 3), Classic Music (1), Keyboard Typing (2)

Including Unrealistic Noises:

1. Includes noise sources in 3 positions: Children Playing Inside (1, 2 and 3)
2. Includes noise sources in 3 positions: Party Close with Music (1, 2 and 3)
3. Includes noise sources in 3 positions: Party (1, 2 and 3)
4. Includes noise sources in 1 position: Hair dryer (2)
5. Includes noise sources in 2 positions: Ventilation (1), Flute (2)
6. Includes noise sources in 1 position: Vacuum Cleaner (1)
7. Includes noise sources in 1 position: Coffee Machine (3)
8. Includes noise sources in 3 positions: Bathwater (1, 2 and 3)
9. Includes noise sources in 3 positions: Football Match (1, 2 and 3)
10. Includes noise sources in 3 positions: Badminton Match (1, 2 and 3)

B.3.5 Cellar

1. Includes noise sources in 1 position: Hair dryer (8)
2. Includes noise sources in 1 position: Vacuum Cleaner (6)
3. Includes noise sources in 3 positions: Coffee Machine (3), Industrial Dishwasher (7), Ventilation (8)

4. Includes noise sources in 5 positions: Keyboard Typing (2, 3, 6, 7 and 8)
5. Includes noise sources in 1 position: Classic Music (2)
6. Includes noise sources in 1 position: Ventilation (7)
7. Includes noise sources in 2 positions: Hair dryer (2), Soft Music (6)
8. Includes noise sources in 5 positions: Children Playing (2, 3, 6, 7 and 8)
9. Includes noise sources in 8 positions: Party Close with Music (1, 2, 3, 4, 5, 6, 7 and 8)

B.3.6 Faroe Islands

1. Includes noise sources in 3 positions: Badminton Match (255, 270 and 285)
2. Includes noise sources in 10 positions: Soft Music (30, 45, 60, 75, 90, 105, 120, 135, 150 and 165)
3. Includes noise sources in 10 positions: Children Playing Inside (15, 45, 60, 75, 90, 135, 165, 225, 270 and 315)
4. Includes noise sources in 10 positions: Keyboard Typing (165, 180, 195, 210, 225, 240, 255, 270, 315 and 345)
5. Includes noise sources in 1 position: Coffee Machine (225)
6. Includes noise sources in 2 positions: Vacuum Cleaner (90 and 270)
7. Includes noise sources in 1 position: Hair dryer (345)
8. Includes noise sources in 5 positions: Classic Music (270), Vacuum Cleaner(225), Keyboard Typing (165), Coffee Machine (90), Hair dryer (15)
9. Includes noise sources in 10 positions: Party Close with Music (30, 60, 105, 135, 165, 195, 240, 270, 300 and 330)

B.3.7 Germany

1. Includes noise sources in 1 position: Hair dryer (8)
2. Includes noise sources in 1 position: Vacuum Cleaner (5)
3. Includes noise sources in 3 positions: Coffee Machine (4), Ventilation (5), Football Match (7)
4. Includes noise sources in 5 positions: Keyboard Typing (1, 2, 3, 4 and 5)
5. Includes noise sources in 1 position: Classic Music (3)
6. Includes noise sources in 1 position: Ventilation (5)
7. Includes noise sources in 2 positions: Hair dryer (7), Soft Music (8)
8. Includes noise sources in 5 positions: Children Playing Inside (3, 4, 5, 7 and 8)
9. Includes noise sources in 8 positions: Party Close with Music (1, 2, 3, 4, 5, 6, 7 and 8)

B.3.8 Japan North

In Japan North it is both possible to specify angle and distance. In each bracket distance will be noted with a d in front.

1. Includes noise sources in 1 position: Football Match (270 d:300)
2. Includes noise sources in 1 position: Soft Music (225 d:300)
3. Includes noise sources in 10 positions: Children Playing (270 d:50, 100, 150, 200 and 250, 315 d:50, 100, 150, 200 and 250)
4. Includes noise sources in 10 positions: Keyboard Typing (90 d:100, 180 d:100, 0 d:200, 45 d:200, 90 d:200, 135 d:200, 180 d:200, 225 d:200, 270 d:200 and 315 d:200)
5. Includes noise sources in 1 position: Coffee Machine (45 d:250)
6. Includes noise sources in 1 position: Vacuum Cleaner (45 d:200)
7. Includes noise sources in 5 positions: Traffic Noise (135 d:450 and 500, 180 d:350, 225 d:450 and 500)
8. Includes noise sources in 6 positions: Hair dryer (45 d:150), Coffee Machine (90 d:400), Keyboard Typing (135 d:200) Forest Birds (180 d:350), Classic Music (270 d:300), Vacuum cleaner (315 d:200)
9. Includes noise sources in 10 positions: Party Close with Music (90 d:100, 180 d:100, 0 d:200, 45 d:200, 90 d:200, 135 d:200, 180 d:200, 225 d:200, 270 d:200 and 315 d:200)

B.3.9 Staircase

1. Includes noise sources in 3 positions: Jazz Music (2), Ventilation (3), Vacuum Cleaner (4)
2. Includes noise sources in 2 positions: Ventilation (2), Vacuum Cleaner (3)
3. Includes noise sources in 1 position: Ventilation (2)
4. Includes noise sources in 1 position: Vacuum Cleaner (2)
5. Includes noise sources in 1 position: Jazz Music (2)
6. Includes noise sources in 2 positions: Soft Music (2), Ventilation (4)
7. Includes noise sources in 2 positions: Vacuum Cleaner (2), Rock Music (3)
8. Includes noise sources in 1 position: Choir In Church With Organ (2)
9. Includes noise sources in 1 position: Choir In Church (2)

Appendix C

Feature Investigation

C.1 Features

C.1.1 Functionals

The 5 **Extremes** values include: max - The maximum value of the contour, min - The minimum value of the contour, range = max-min, maxPos - The absolute position of the maximum value (in frames), minPos - The absolute position of the minimum value (in frames)

The 10 **Regression** values include: linregc1 - The slope (m) of a linear approximation of the contour, linregc2 - The offset (t) of a linear approximation of the contour, linregerrA - The linear error computed as the difference of the linear approximation and the actual contour, linregerrQ - The quadratic error computed as the difference of the linear approximation and the actual contour, qregc1 - quadratic regression coefficient 1, qregc2 - quadratic regression coefficient 2, qregc3 - quadratic regression coefficient 3, qregerrA - linear error between contour and quadratic regression line (parabola), qregerrQ - quadratic error between contour and quadratic regression line (parabola), centroid - centroid of contour (this is computed as a by-product of the regression coefficients)

The 4 **Moments** values include: variance, standard deviation, skewness, kurtosis

The 8 **Percentiles** values include: quartiles - quartile 1-3 (0.25, 0.5, 0.75), iqr - inter-quartile ranges (2-1, 3-1, 3-1), percentile - array of $p \cdot 100$ percent percentiles to compute (here set to 0.95 and 0.98), interp - set to 1, percentile values are linearly interpolated, instead of being rounded to the nearest index in the sorted input array.

The 1 **Crossings** values include: zero-crossing rate

The 4 **Peaks** values include: numPeaks - the number of peaks, meanPeakDist - mean distance between peaks (relative to the input segment length, in seconds, or in frames), peakMean - arithmetic mean of peaks, peakMeanMeanDist - (arithmetic mean of peaks - arithmetic mean of all values)

The 7 **Means** values include: amean - arithmetic mean, absmean - arithmetic mean of absolute values, qmean - quadratic mean, nzabsmean - arithmetic mean of absolute values (of non-zero values only), nzqmean - quadratic mean (of non-zero values only), nzgmean - geometric mean (of absolute values of non-zero values only), nnz - number of non-zero values (relative to the input segment length, in seconds, or in frames)

C.1.2 Error Figures

Error rate plots for the two situations not included in Section 7.2 can be seen in Figure C.1 and C.2. For both of these situations it can be seen that a pruning would result in an error rate that is acceptable low with fewer terminal nodes than what the train data suggests is necessary.

C.1.3 List of features

For the analysis of the best features in Section 7.2.2, the full list of features in their prioritised numeration is provided in Table C.1 (notice that one of the features is included in the set twice, namely 'pcm_LOGenergy_sma_range').

When the specified feature set is used in the classification system, it seems that some of the features are included more than once. The full feature list for the specified feature set can be seen in Table C.2.

The following three features are not included in the new feature set: 'pcm_LOGenergy_sma_linregerrA', 'mfcc_sma[1]_range', 'pcm_LOGenergy_sma_maxPos'.

C.1.4 Plot of features

The full list of features are plotted in Figure C.3 to Figure C.16 (feature 1, 2 and 3 are not shown here since they are plotted in section 7.2.2).

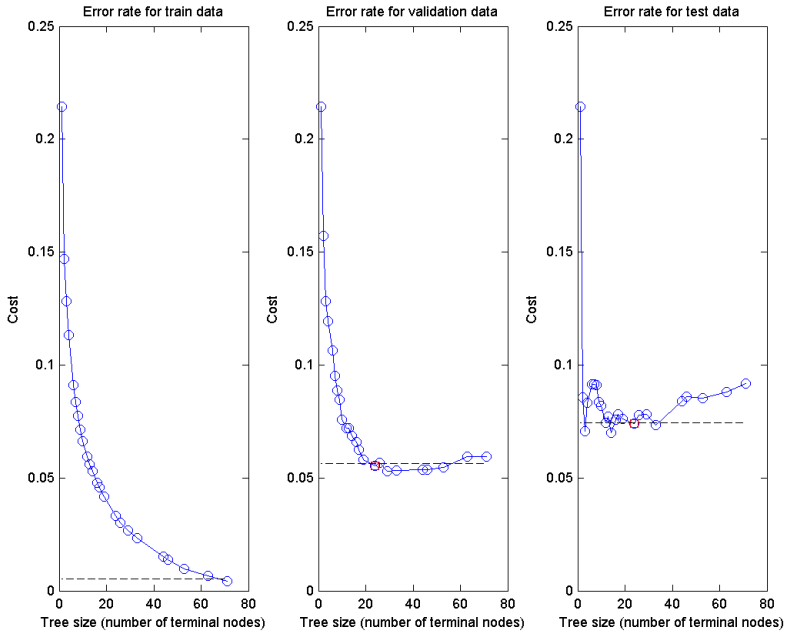


Figure C.1: **Left:**Plot of the error rate for the training set, **Middle:** the validation set, **Right:** for the test set for the situation where the final SNR is set to 0dB. The solid line shows the estimated cost for each tree size, the dashed line marks one standard error above the minimum, and the red square marks the smallest tree under the dashed line for the validation set.

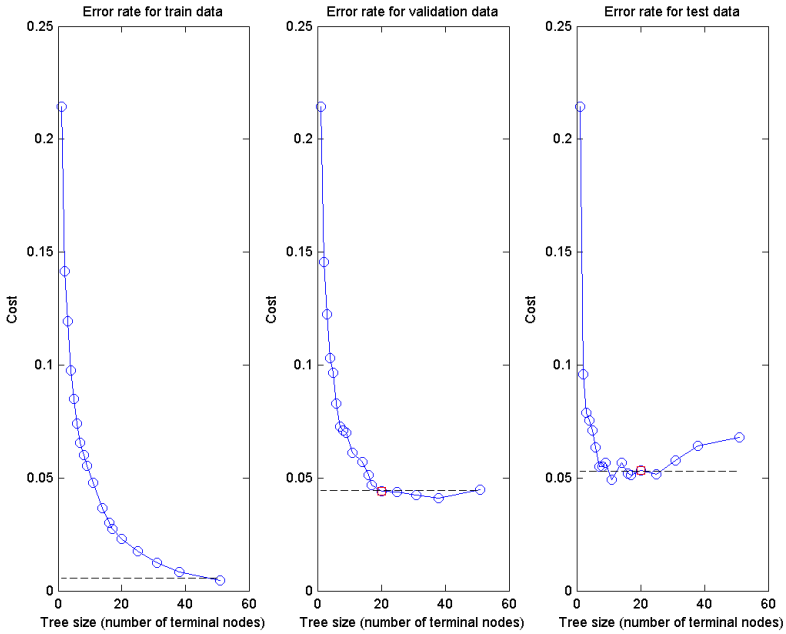


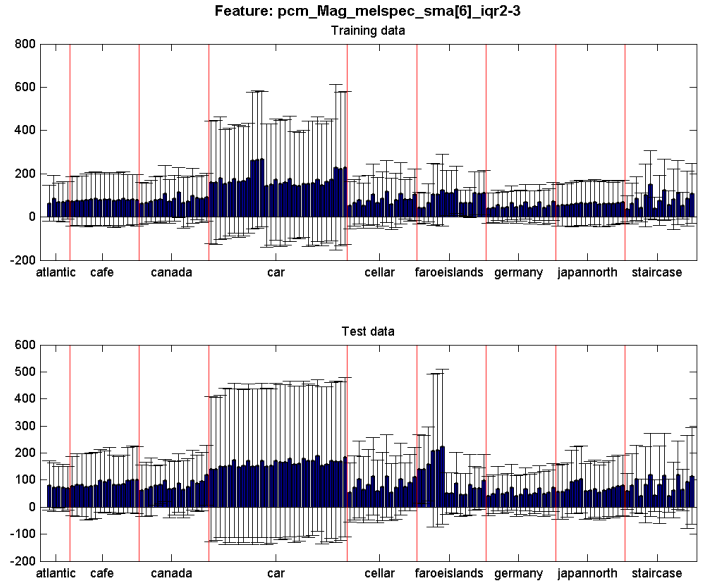
Figure C.2: **Left:**Plot of the error rate for the training set, **Middle:** the validation set, **Right:** for the test set for the situation where the final SNR is set to 10dB. The solid line shows the estimated cost for each tree size, the dashed line marks one standard error above the minimum, and the red square marks the smallest tree under the dashed line for the validation set.

1:	'pcm_Mag_fband0-250_sma_de_qmean'
2:	'pcm_Mag_fband1000-4000_sma_de_de_percentile95.0'
3:	'pcm_Mag_spectralFlux_sma_de_qmean'
4:	'pcm_Mag_melspec_sma[6]_iqr2-3'
5:	'pcm_Mag_melspec_sma_de[21]_linregc1'
6:	'mfcc_sma[2]_linregc2'
7:	'pcm_Mag_fband0-250_sma_de_de_range'
8:	'pcm_zcr_sma_de_quartile1'
9:	'mfcc_sma[5]_linregc2'
10:	'pcm_LOGenergy_sma_linregerrA'
11:	'pcm_Mag_melspec_sma_de[24]_peakMeanMeanDist'
12:	'mfcc_sma[1]_range'
13:	'pcm_Mag_melspec_sma[1]_qregerrA'
14:	'pcm_Mag_spectralCentroid_sma_de_de_linregc1'
15:	'mfcc_sma[11]_nzgmean'
16:	'pcm_Mag_melspec_sma_de_de[23]_peakMean'
17:	'pcm_Mag_melspec_sma[14]_iqr1-2'
18:	'pcm_Mag_spectralCentroid_sma_de_skewness'
19:	'pcm_LOGenergy_sma_de_de_qregc3'
20:	'pcm_LOGenergy_sma_range'
21:	'pcm_Mag_melspec_sma[14]_qregc1'
22:	'pcm_LOGenergy_sma_range'
23:	'pcm_Mag_melspec_sma[15]_peakMean'
24:	'pcm_LOGenergy_sma_maxPos'
25:	'pcm_Mag_melspec_sma_de[0]_quartile1'
26:	'pcm_Mag_melspec_sma_de[7]_linregc1'
27:	'pcm_Mag_fband0-250_sma_de_de_centroid'
28:	'pcm_Mag_spectralMaxPos_sma_de_de_centroid'
29:	'pcm_LOGenergy_sma_de_skewness'
30:	'pcm_Mag_melspec_sma_de_de[10]_skewness'

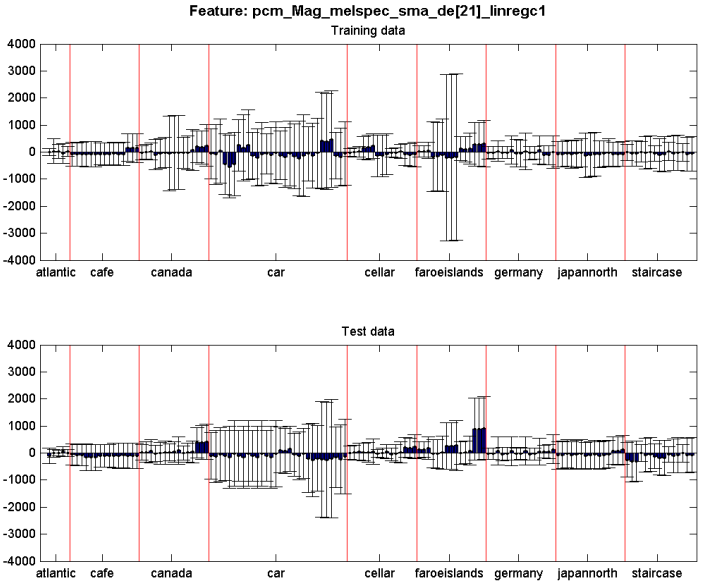
Table C.1: Full list of features for the situation with the situation with fixed target and noise levels for each source single dataset.

1:	'pcm_Mag_fband0-250_sma_de_qmean'
2:	'pcm_Mag_fband1000-4000_sma_de_de_percentile95.0'
3:	'pcm_Mag_spectralFlux_sma_de_qmean'
4:	'pcm_Mag_melspec_sma[6]_iqr2-3'
5:	'pcm_Mag_melspec_sma_de[21]_linregc1'
6:	'mfcc_sma[2]_linregc2'
7:	'pcm_Mag_fband0-250_sma_de_de_range'
8:	'pcm_zcr_sma_de_quartile1'
9:	'mfcc_sma[5]_linregc2'
10:	'pcm_Mag_fband0-250_sma_de_qmean'
11:	'pcm_Mag_melspec_sma_de[24]_peakMeanMeanDist'
12:	'pcm_Mag_spectralFlux_sma_de_qmean'
13:	'pcm_Mag_melspec_sma[1]_qregerrA'
14:	'pcm_Mag_spectralCentroid_sma_de_de_linregc1'
15:	'mfcc_sma[11]_nzgmean'
16:	'pcm_Mag_melspec_sma_de_de[23]_peakMean'
17:	'pcm_Mag_melspec_sma[14]_iqr1-2'
18:	'pcm_Mag_spectralCentroid_sma_de_skewness'
19:	'pcm_LOGenergy_sma_de_de_qregc3'
20:	'pcm_Mag_fband0-250_sma_de_de_range'
21:	'pcm_Mag_melspec_sma[14]_qregc1'
22:	'pcm_zcr_sma_de_quartile1'
23:	'pcm_Mag_melspec_sma[15]_peakMean'
24:	'pcm_Mag_fband0-250_sma_de_qmean'
25:	'pcm_Mag_melspec_sma_de[0]_quartile1'
26:	'pcm_Mag_melspec_sma_de[7]_linregc1'
27:	'pcm_Mag_fband0-250_sma_de_de_centroid'
28:	'pcm_Mag_spectralMaxPos_sma_de_de_centroid'
29:	'pcm_LOGenergy_sma_de_skewness'
30:	'pcm_Mag_melspec_sma_de_de[10]_skewness'
31:	'pcm_Mag_spectralMaxPos_sma_de_de_centroid'
32:	'mfcc_sma[11]_nzgmean'
33:	'pcm_LOGenergy_sma_range'
34:	'mfcc_sma[11]_nzgmean'

Table C.2: Full list of features for the situation with specified input feature set.

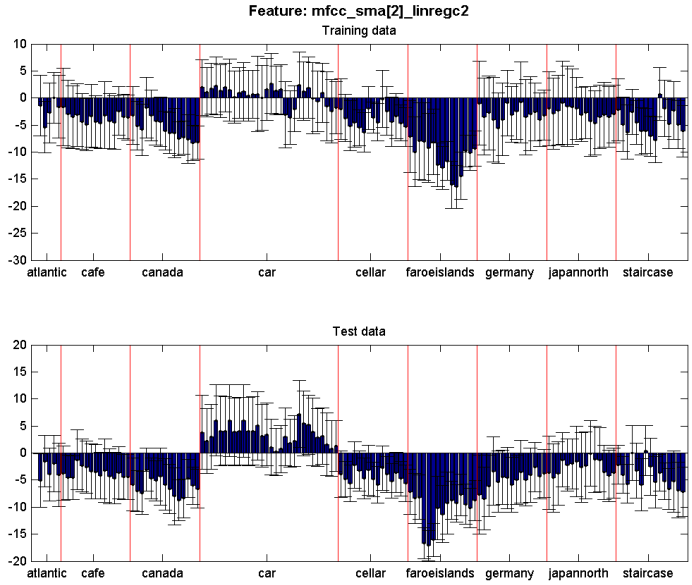


(a) Feature used for splitting the left child node from the left node of the second split

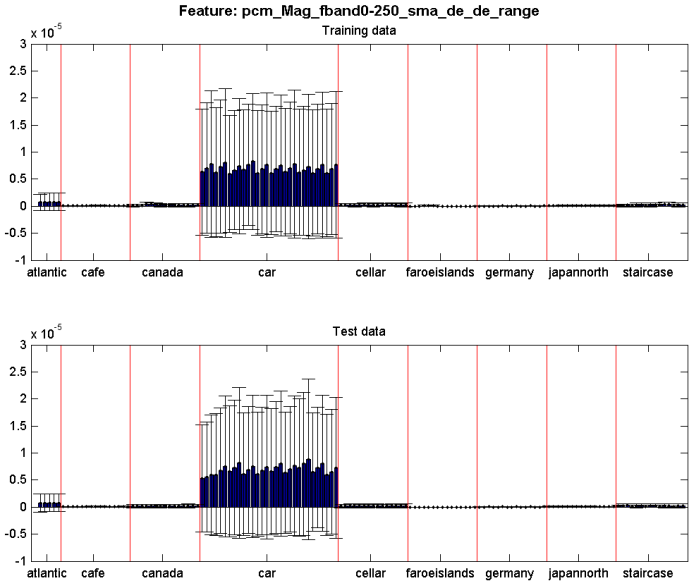


(b) Feature used for splitting the right child node from the left node of the second split

Figure C.3: Feature number 4 and 5



(a) Feature used for splitting the left child node from the right node of the second split



(b) Feature used for splitting the right child node from the right node of the second split

Figure C.4: Feature number 6 and 7

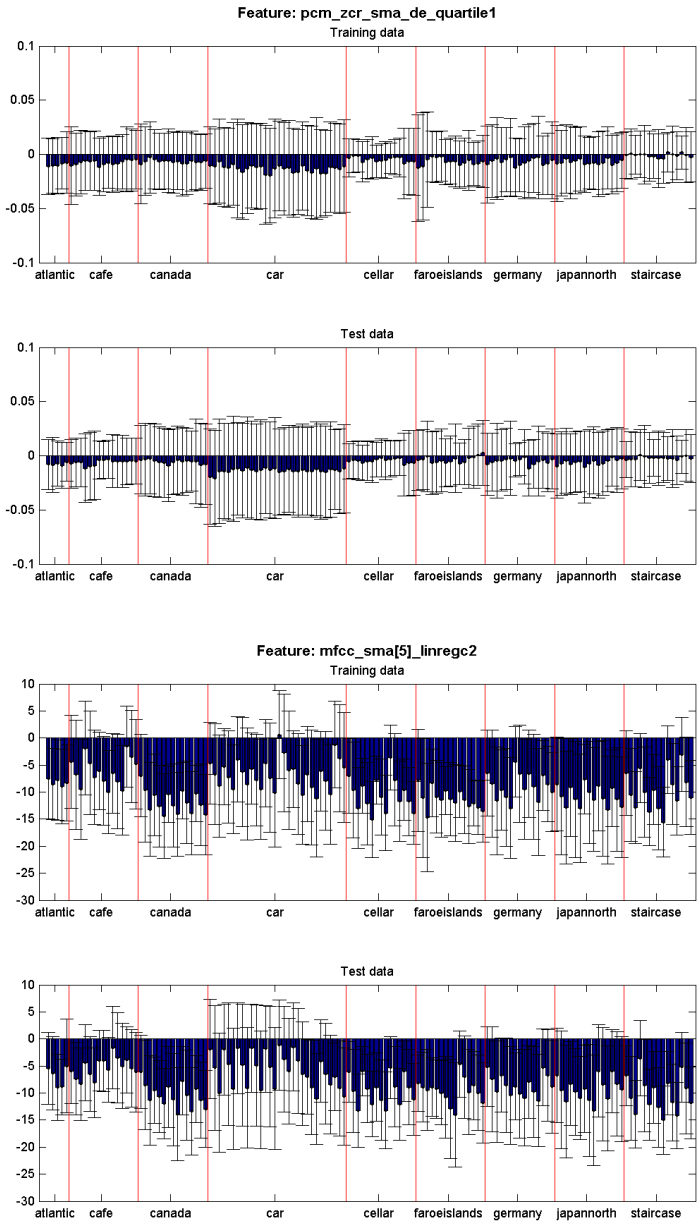


Figure C.5: Feature number 8 and 9

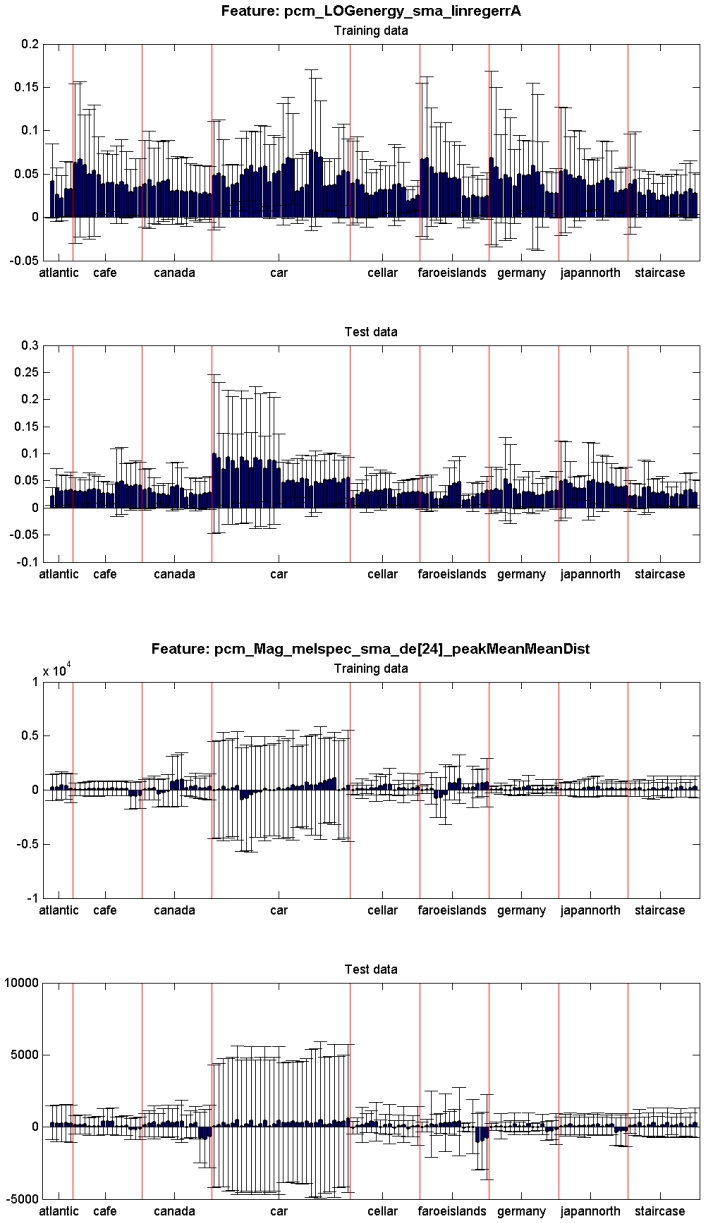


Figure C.6: Feature number 10 and 11

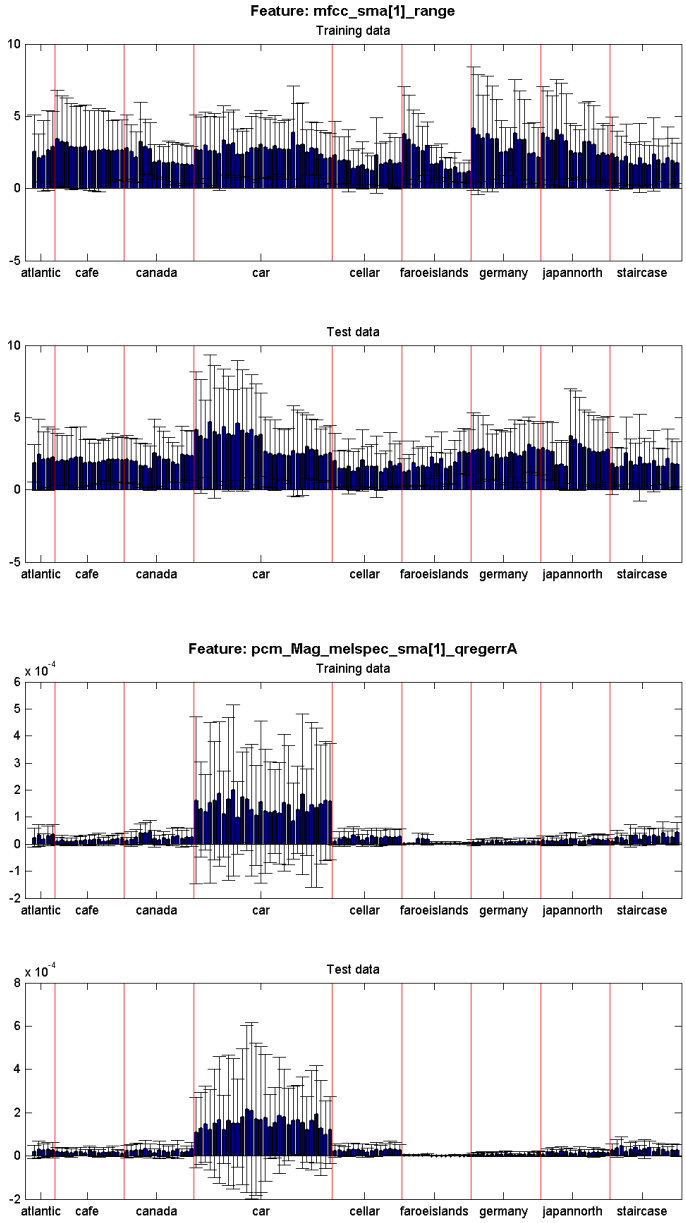


Figure C.7: Feature number 12 and 13

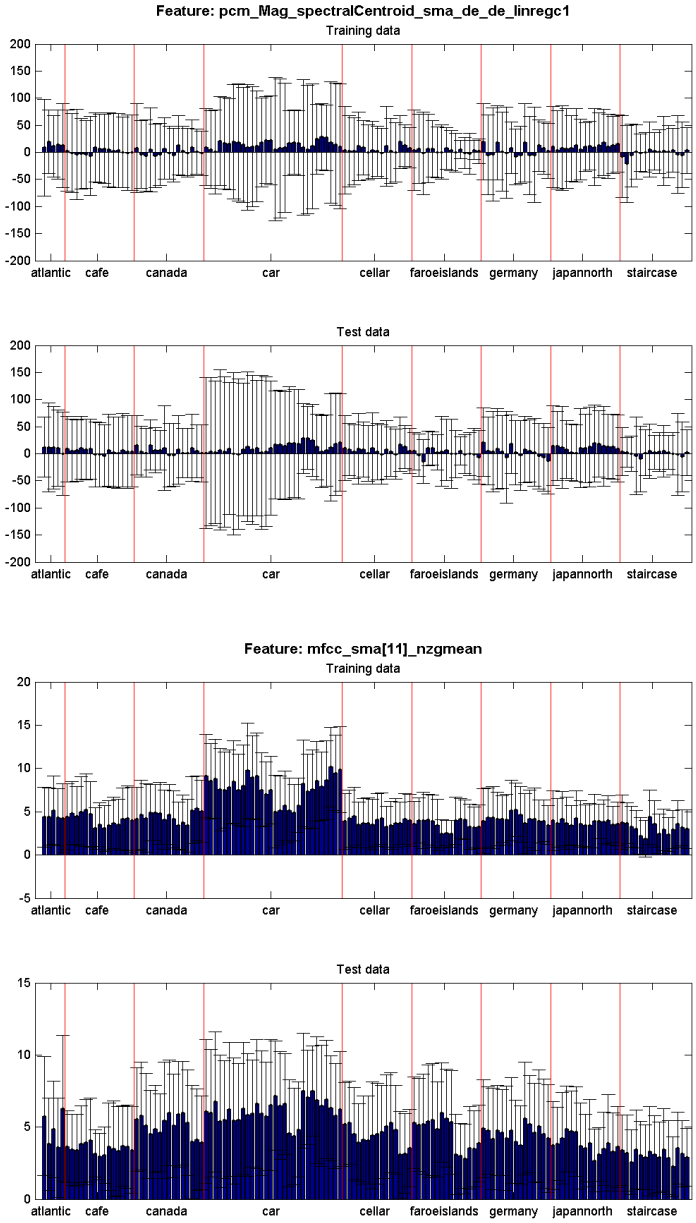


Figure C.8: Feature number 14 and 15

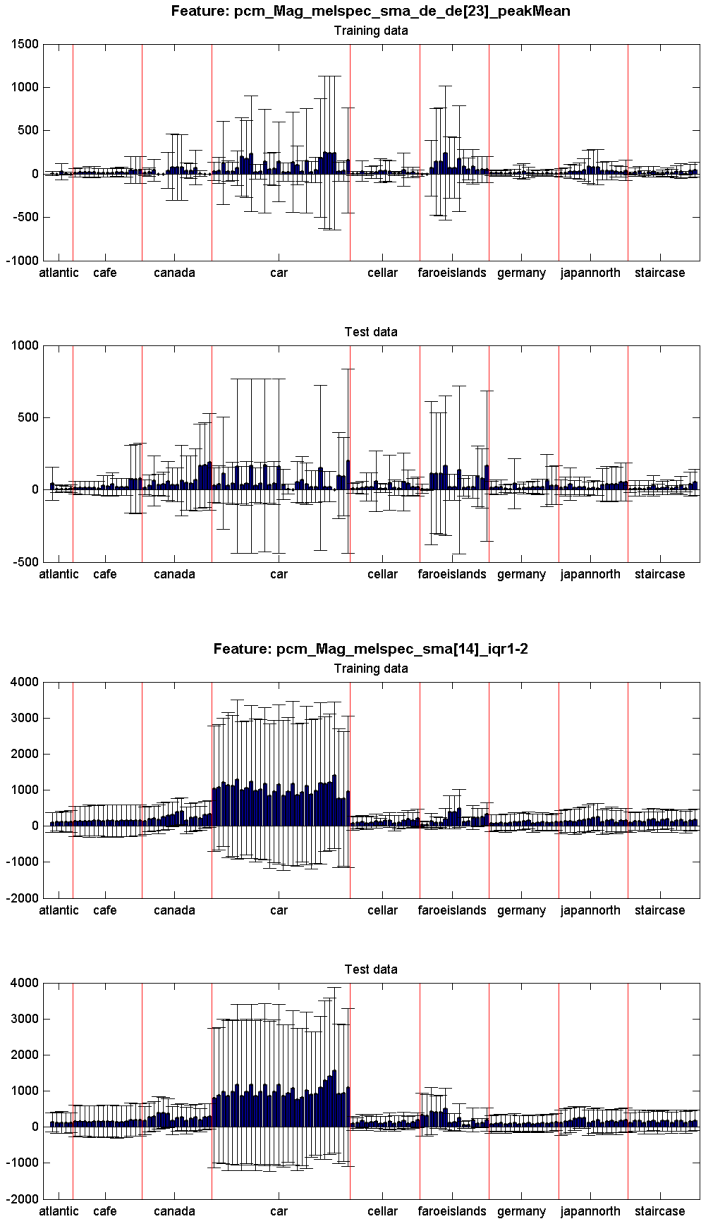


Figure C.9: Feature number 16 and 17

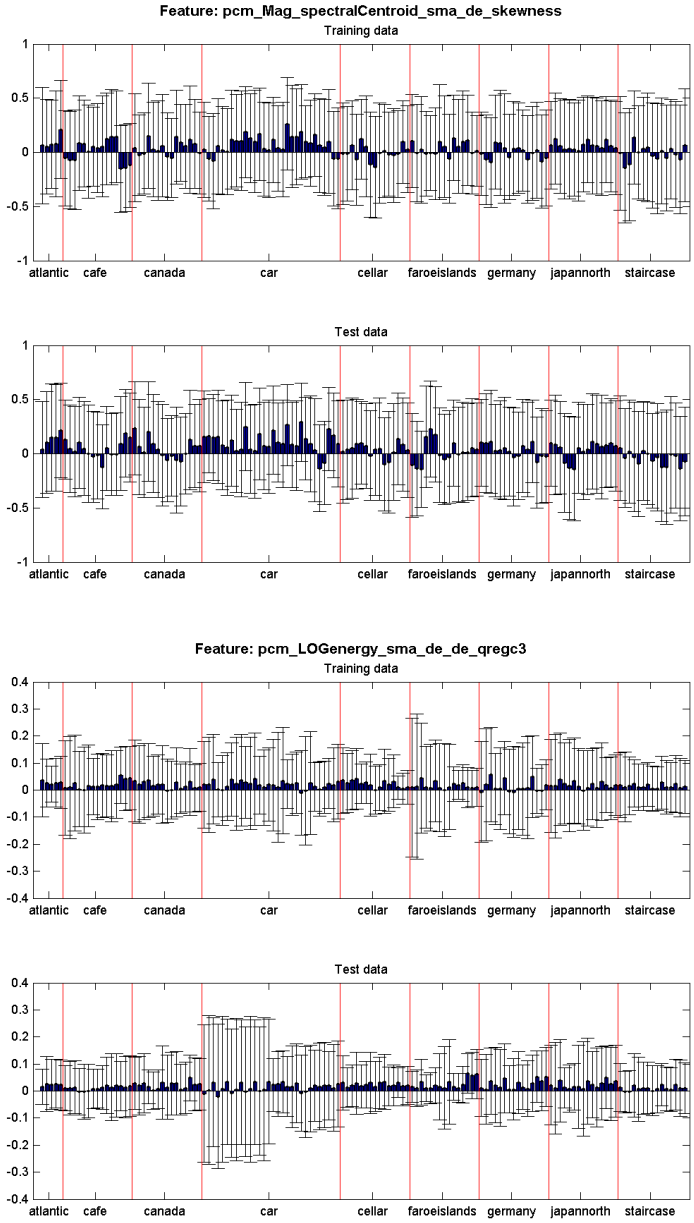


Figure C.10: Feature number 18 and 19

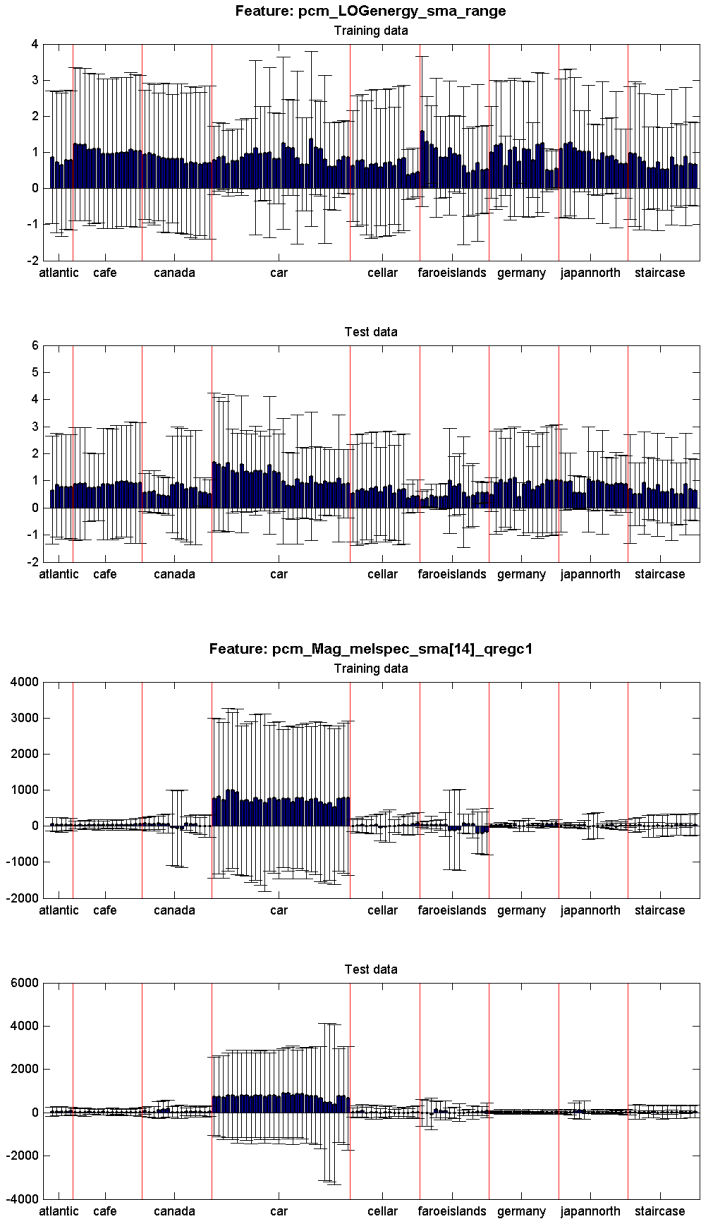


Figure C.11: Feature number 20 and 21

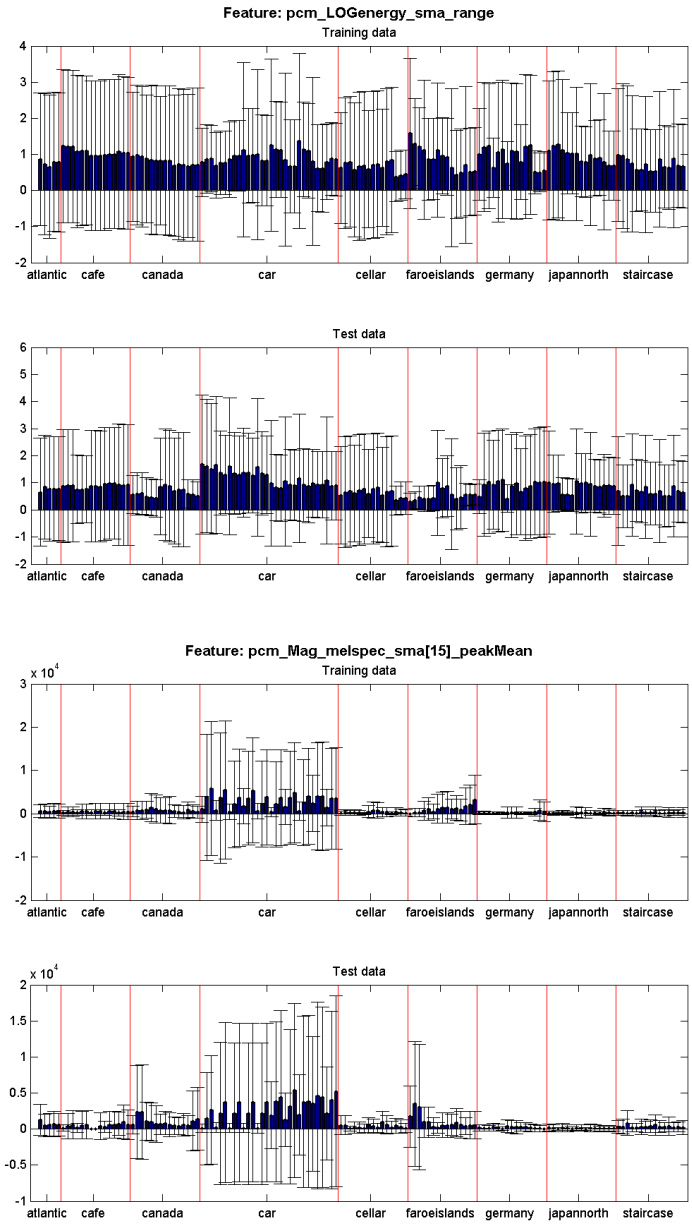


Figure C.12: Feature number 22 and 23

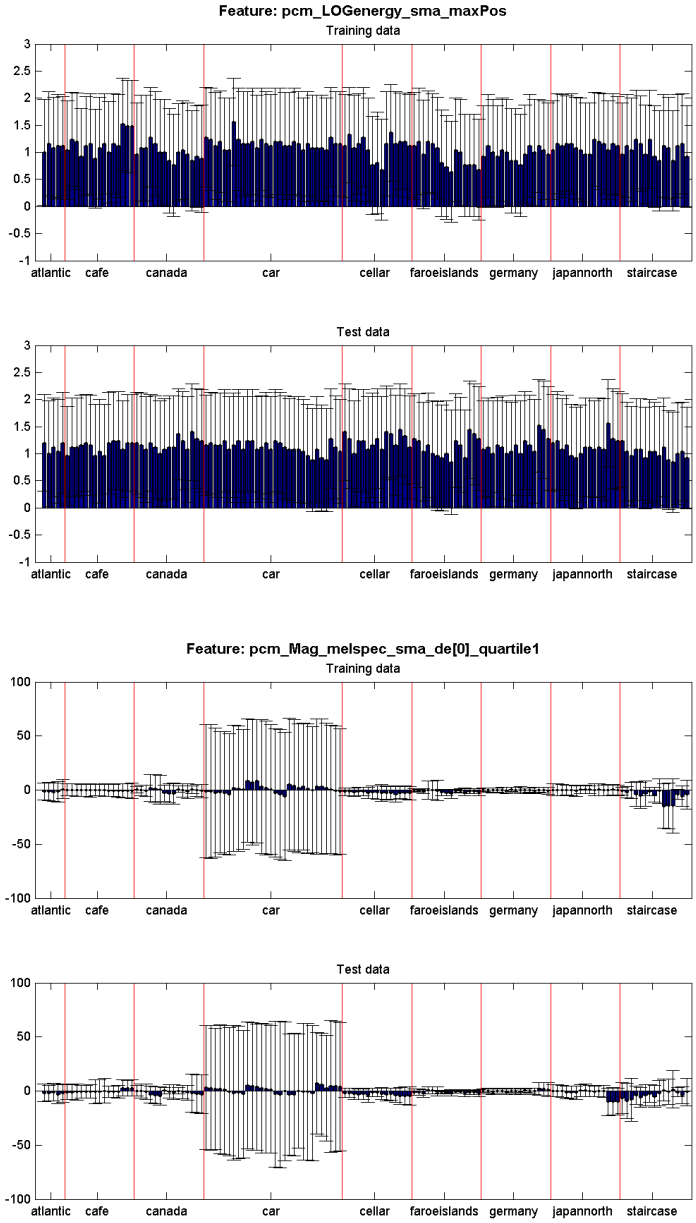


Figure C.13: Feature number 24 and 25

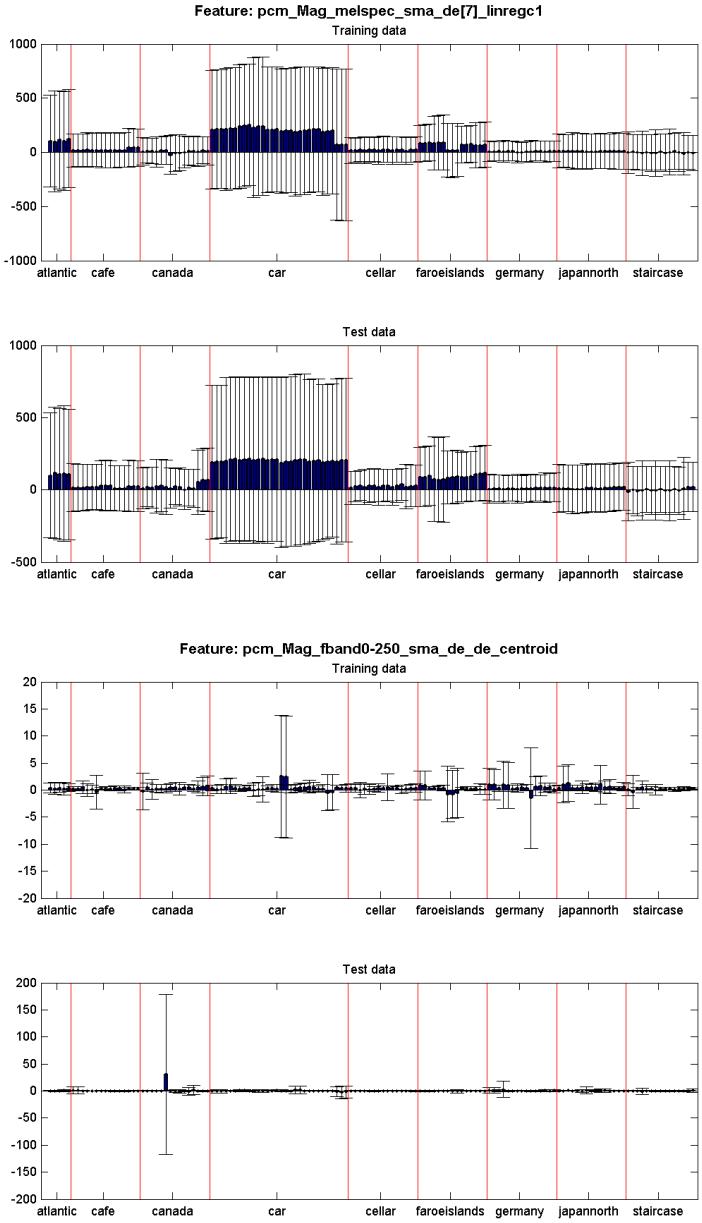


Figure C.14: Feature number 26 and 27

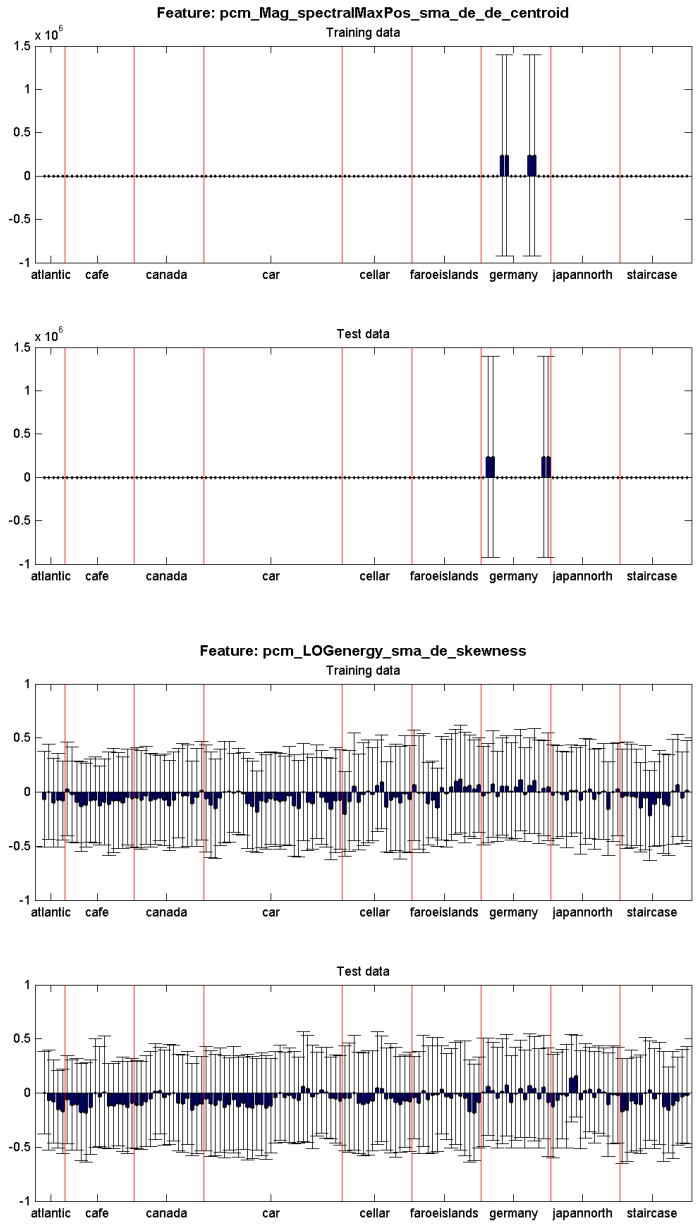


Figure C.15: Feature number 28 and 29

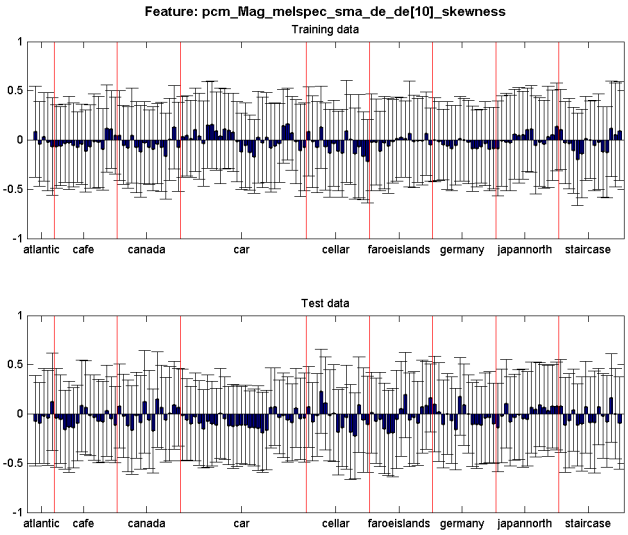


Figure C.16: Feature number 30