Analysis of Human Behaviour by Machine Learning

Kit Melissa Larsen s062261
Louise Mejdal Jeppesen s062254

Kongens Lyngby 2012
IMM-MSc-2012-33

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-MSc-2012-33

Abstract

This thesis deals with automation of manual annotations for use in the analysis of the interaction pattern between mother and child. The data applied in the thesis are provided by Babylab at the Institute of Psychology, University of Copenhagen, and consists of the three recording modalities; sound, motion capture and video. The focus of this thesis, with respect to the available data, is the recordings of 21 four-months old children and their mothers.
The aim of the thesis is to automatically, by the use of machine learning, regenerate labels that have been extracted manually at Babylab. With this, a much time consuming task would be relieved from their shoulders. Furthermore, the human subjectivity of the labels would be removed with the objective replacement of a machine.

The re-annotation of labels introduces the area of supervised classification which is used for the task of speaker identification as well for emotion recognition in this thesis. A thorough investigation of different classification approaches forms the basis of the results of the two aforementioned tasks, for the sound data provided by Babylab. These results have a reliability in the same order as that of the manual codings, and are therefore considered very promising for the future work at Babylab.
It is also investigated whether the uniqueness of this particular data set, i.e. that three recording modalities are available, is beneficial to the two tasks of speaker identification and emotion recognition. This is tested by including information from the motion capture data to the sound data. The results show no effect as well as an actual high rate of deterioration of the classifier performance for the two tasks, respectively.
Besides being included in the two classification tasks, the motion capture data

provides stable annotations on several aspects of the mother-child interaction. These have therefore been extracted in an automated way in this thesis.
The video modality has also been superficially investigated, with respect to the child's facial expressions. This has been considered as a possible support to the two classification tasks as well as for the direct application in the analyses performed at Babylab of mother-child interaction. This showed interesting prospects that should definitely be pursued by Babylab in the future.

Resumé

Dette speciale omhandler automatisering af manuelle annotationer til brug i analyse af interaktionsmønstre mellem mor og barn. Data behandlet i dette studie er udlånt af Babylab, Institut for Psykologi, Københavns universitet, og består af de tre modaliteter: lyd, video og motion capture. Fokus i dette speciale, baseret på det foreliggende datasæt, er optagelserne af 21 4 mdr. gamle børn og deres mødre.
Formålet med specialet er, ved brug af machine learning metoder, at opnå automatiske annotationer af de aspekter af mor-barn interaktionen som er blevet manuelt annoteret af Babylab. Herved vil en utrolig tidskrævende opgave blive fjernet fra skuldrene af Babylab. Derudover vil den menneskelige subjetivitet blive udskiftet med computerens objektivitet.
Med den automatiske annotering af labels introduceres supervised klassifikation der anvendes til speaker identification og emotion recognition i dette speciale. En grundig undersøgelse af forskellige klassifikationsmetoder lægger til grund for resultaterne af de to førnævnte problemer baseret på lyddata. Reliabiliteten af disse resultater er i samme størrelsesorden som reliabiliteten af de manuelle kodninger og er derfor af yderst lovende karakter i forhold til Babylabs fremtidige arbejde.

Ydermere undersøges det hvorvidt Babylabs unikke data, der baserer sig på tre datamodaliteter, kan udnyttes i de to klassifikationsproblemer speaker identification og emotion recognition. Dette testes gennem kombination af lyddata og motion capture data. Disse tests viser at henholdsvis ingen ændring og en decideret forværring af resultaterne opnås ved at inkludere motion capture information.
Udover at kunne bruges i de to klassifikationsproblemer, bidrager motion capture

dataen med stabile annotationer af forskellige aspekter af mor-barn interaktionen. Derfor er flere af de manuelle kodninger blevet genskabt automatisk i dette speciale.
Mulige annotationer fra videomodaliteten er også blevet berørt i et lille sideløbende studie. Dette med tanken at automatisere annotationer af barnet ansigtsudtryk, både som support i de føromtalte klassifikationsproblemer, såvel som til direkte brug af Babylab i analysen af interaktionen mellem mor og barn. Resultaterne for dette var lovende og bør bestemt blive undersøgt nærmere af Babylab i fremtiden.

Preface

This thesis was prepared at the department of Informatics and Mathematical Modelling at the Technical University of Denmark in fulfilment of the requirements for acquiring a M.Sc. in Medicine and Technology. The thesis corresponds to a workload of 35 ECTS-credits.

Lyngby, 02-April-2012

Kit Melissa Larsen s062261
Louise Mejdal Jeppesen s062254

Acknowledgements

Throughout the study that underlies this thesis, many different people have been involved and the final outcome would not have been the same without the support and knowledge from all of these.

First of all we would like to thank the staff at Babylab at the Institute of Psychology, University of Copenhagen. Lektor Simo Køppe, lektor Susanne Harder and lektor Mette Skovgaard Væver have been indispensable, with their supervising in the research area of psychology. Their enthusiastic ideas at our meetings have been very contributing to the outcome of this study.

Furthermore we would like to thank Ph.d.-student Jens Fagertun from IMM for all his help in the study on Active Appearance Models as well as his help in training the model.

A great thanks should be given to our two supervisors from IMM, Professor Lars Kai Hansen and Assistant Professor Morten Mørup. Without the many discussions on ideas and approaches for this thesis, the last six months would not have been as motivating and exciting.
Finally, Morten Mørup should be granted a special thanks for his always happy and enthusiastic being as well as for his willingness to help whenever stopping by his office. Thank you.

Kit Melissa Larsen & Louise Mejdal Jeppesen

Abbreviations

Abbreviation	Description
IMM	Informatics and Mathematical Modelling
GMM	Gaussian Mixture Model
ANN	Artificial Neural Network
TREE	Decision Tree Classifier
HMM	Hidden Markov Model
MNR	Multinomial Regression
KNN	K-Nearest Neighbour
LDC	Linear Discriminant Classification
SVM	Support Vector Machine
f-b algorithm	Forward-Backward Algorithm
EM algorithm	Expectation Maximization Algorithm
MFCC	Mel Frequency Cepstrum Coefficients
LPCC	Linear Prediction Cepstral Coefficients
zcr	Zero-Crossing Rate
mocap	Motion Capture
AAM	Active Appearance Model
EAM	Elastic Appearance Model

Contents

Chapter 1

Introduction

Analysis of the interaction pattern between mother and child (also referred to as a dyad) has been an important topic in the research area of psychology in the last many decades [9], [28], [29], [45], [64]. This stretches from the interactions in vocal rhythms between mother and child, to the facial expressions of the child and to the distinct mother-child movement patterns. In [64] vocalizations, facial expressions and gazes at the mother's face were investigated during a face-to-face interaction between a mother and a child. The study provides strong evidence that the emotional facial expressions of the infant are correlated with vocalizations and with gazes at their parents faces. In [9] the vocalizations and turn-taking in vocalizations of the mother and child were investigated. The results showed that vocalization of one of the dyad members was more likely to occur when the other member was vocalising.
The types of research mentioned so far are of great interest to psychologists because the physical relationship between mother and child is of uttermost importance for the child's future well being, [53], [19].

The data processed in this thesis is provided by Babylab, Institute for Psychology, University of Copenhagen. Their goal of this research is to investigate the many aspects of early child development through interaction patterns. The data provided include the three recording modalities: sound, video and motion capture. The recording set-up are 10 minutes of talk and play between the mother

and her child. The details of the recordings will be described in chapter 3.
To be able to analyse the interactions, extraction of relevant information from the data is necessary. This is obtained at Babylab by manually annotating several different physical aspects from all three modalities, individually. The general issue with manual codings is that there can be large differences in the inter-coder agreement of labels. Also the time aspect of the manual codings should be considered.
The intention of this thesis is to automate this annotation process through the use of machine learning methods. Likewise, it is of interest to combine the information extracted from the three modalities for a possible improvement of the annotation precision. For Babylab this annotation automation would ease the future workload and reduce the processing time significantly. Of more importance is the complete removal of human errors if the optimized automatic annotations are implemented. A note here, is that automatic annotation errors will be the consequence, with the amount depending on the performance of the automatic method.

The annotations carried out in this thesis are described in details in chapter 2 where also a specification of the problems investigated is outlined. In chapter 3, the data dealt with during this study is described. Due to the fact that three different data modalities are provided, the aspect of time synchronization is of great importance before any further data processing can take place. The synchronization of the modalities is described in details in chapter 4.
Chapter 5 covers the topic of speaker identification. In this chapter the speech signal is described in general, section 5.1, as well as the preprocessing techniques that is a necessity in dealing with speech signals, section 5.2. Before classification in the speaker identification problem can be executed, feature extraction must be carried out. This process is described in section 5.3. Section 5.4 deals with the different classification methods investigated in the speaker identification problem. Finally, chapter 5 is rounded off by section 5.5 that discuss the methods with which the model can be evaluated.
The classification of the child's emotional state is approached in chapter 6. This chapter has the same structure as chapter 5, where preprocessing, feature extraction, classification and model evaluation constitute the topics of section 6.1 to 6.4.
Chapter 7 describes the automatic annotations obtained from motion capture, whereas chapter 8 addresses the possibilities of combining the three modalities. This is carried out by including the motion capture annotations as features in the problems of speaker identification and emotion recognition. It is furthermore discussed in this chapter how the third data modality, video, can be included as well.
The results obtained during the thesis are presented in chapter 9. This chapter is divided into four sections, where section 9.1 presents the results from

the speaker identification problem, section 9.2 the results from the emotional recognition, section 9.3 the results for the annotations in motion capture and finally section 9.4 the results when combining the sound and the motion capture modalities. For the sake of overview of the many obtained results, each result section is provided with a brief summary of that particular topic.
Chapter 10 rounds of the report with a conclusion as well as a discussion of the perspectives regarding the future work.

Chapter 2

Problem Statement

As mentioned in the introduction, the data provided by Babylab include sound, video and motion capture. The purpose of this thesis is to obtain automatic annotations of the states or actions occurring between the mother and child during the recordings. These include annotations obtained by analysing the modalities separately, but also annotations derived by combining the information extracted from two or all three modalities. The approach in this thesis is to include and apply relevant machine learning methods to achieve applicable results. The annotations in focus are therefore chosen based on the interests of the psychologists at Babylab and on the possibility of angling these towards the intelligent data processing branch of pattern recognition. Especially, it is of interest to work with those problems that have already been approached by Babylab, because this provides the advantage of having the ground truth. The problems then become supervised learning.
An important note regarding the choice of annotations to be included in this project, is that the manual labels made at Babylab are numerous, meaning that a selection has been made among these, because of the limited time prospect of the thesis. Working with an "untouched" data set and trying to comply with all the expectations from the psychologists at Babylab confines the possibility of developing new methods for the annotation automation. This thesis will therefore integrate state-of-the-art methods regarding the two major topics of the report, speaker identification, chapter 5, and emotion recognition, chapter 6, as the starting point for the analyses carried out during the study.

2.1 Problem Specification

The annotations to be automated, and thereby to be the focus of this thesis, are explained in the following, under the appertaining modality. Furthermore, the interaction patterns of interest, across and in between the three data modalities, are described in the last section, 2.1.4. In this section the synchronization issue when analysing data across modalities is also discussed.

2.1.1 Sound

The identification of the speaker throughout the 10 minute recordings has been manually executed by Babylab for 21 dyads from sound file listening.
Speaker identification is also a well-known machine learning problem where improvements are continually attained, [24], [27], [33], [42], [55], [57]. This is therefore chosen as one of the focus areas of this project.
During the recording session, four possible states are observed: the child is speaking, the mother is speaking, both are speaking or no one is speaking. This makes the speaker identification a four-class problem which is thoroughly investigated in chapter 5.
Besides the speaker identification task, Babylab's annotations from the sound signals cover the emotional state of the child (protest/not protest) and the mother's vocalization (speech/song). Solving these problems are therefore additional machine learning tasks. The emotion recognition problem is examined in section 6, whereas the vocalization of the mother is left for future work, as described in chapter 10.

2.1.2 Motion Capture

Of interest to the psychologists is the physical relationship between the mother and her child. The motion capture data supplies the analysts with information that can give a comprehension of this. One of the advantages of this recording modality is that the position of the mother and child in relation to each other is known.
From the marker coordinates, the changes in distance between the mother and child can be calculated. Likewise, the physical energy of the child can be obtained by calculating the covered distance of the child. This is interpreted by the psychologists at Babylab as the movement of the right arm, which is calculated in this thesis through the coordinates of the right wrist marker. This

information could be a relevant feature in the speaker identification task as well, because of a possible connection between speech and movement.
Another annotation of interest to the psychologists at Babylab is the child's head orientation, because of the correlation between sudden movements of the mother and head aversion of the child as well as distance between the mother and child and the head orientation of the child. The annotations from the motion capture modality are studied in chapter 7.

2.1.3 Video

The advantage of video as a signal is that it provides the visual understanding of the interaction between the mother and her child. This can be used to extract information of the child's emotional state by identifying the facial characteristics on a frame-by-frame basis. These features alone can be used in a classifier, but they could also support the classifier mentioned in section 2.1.1 above, where sound qualities of the child's emotional state are used as features. Furthermore, identifying facial expressions of the child could possibly support the speaker identification, also mentioned in section 2.1.1 above.
A great issue with the video data is the poor image resolution. The child is positioned rather distant from the cameras, making the actual number of pixels visualizing the child's face limited to around 70×70 pixels. The possibility of detecting face characteristics could therefore be very difficult. Although not an actual part of this thesis, the aspect of video annotations is discussed further in section 8.2.

2.1.4 Interaction Patterns across Data Modalities

Before combining the information extracted from the three modalities, a structuring of the data must be performed. This involves time synchronization of the data to achieve exact comparability of the modalities. This will be done between the sound and video, and between sound and motion capture. By solving these two synchronization problems the third problem, video and motion capture, is given.
When the goal of automating the annotations already executed at Babylab has been reached, the actual analysis of the interaction pattern between the mother and her child can take place. Due to the different research areas of the psychologists at Babylab, many different aspects of the interaction pattern are important for them to establish. One of these is the correlation between the vocalizations of mother and child. Also the correlation between the mother's vocalization and the child's energy level is of interest. In an overall perspective, the psychologists

are interested in a clarification of which actions of the mother causes actions of the child and vice versa. This will hopefully show a generalizable pattern across the dyads.
The challenge in the interaction pattern between mother and child across the data modalities arises in the temporal aspect. With temporal aspect is meant that a displacement or delay can occur when comparing the modalities and the causes of for example a movement. If, for example, the mother begins to speak and the child responds with a movement of the hands, this movement will probably be delayed with respect to the vocalization of the mother. When analysing the interaction pattern, this action/cause-delay is therefore importing to keep in mind.

2.1.5 Summary

The annotations to be automated in this thesis are summarized in the following table, where the respective chapters/sections are indicated for the sake of overview. The task of synchronization as well as the only superficially touched subject on extraction of facial expressions from video are included as well.

Annotation	Modality	Description	Chapter
Synchronization	Sound, motion capture, video	Time synchronizing the three recording modalities	4
Speaker identification	Sound (motion capture)	Classifying the four states: mother speaks, child speaks, both speak and no one speeks	5
Emotion recognition	Sound (motion capture)	Classifying the two states of the child: protest and no protest	6
Head orientation of child	Motion capture	Determining the angular head orientation from vector calculus	7
Distance between mother and child	Motion capture	Calculate the distance between two motion capture markers representing the heads of the mother and child	7
Physical energy level of child	Motion capture	Calculate the covered distance of the right arm from the wrist marker	7
Facial expressions	Video	Extraction of the child's facial expressions	8.2, B

Table 2.1: The annotations to be automated in this thesis. The task of synchronization as well as the extraction of facial expressions from the video modality is mentioned as well.

Chapter 3

Data

The data used in this thesis are recording sessions of the interaction between a mother and her child and includes sound, video and motion capture. The dyad interaction have been recorded at Babylab when the child was at the ages 4 months, 7 months, 10 months and 13 months, respectively. In this study only the data for the 4 months old children, dyads 001 - 021, will be analysed. Each session has a duration of 10 minutes. This chapter explains briefly each modality and how these have been recorded. Furthermore a section is included that briefly introduces the manual annotations provided by Babylab.

3.1 Sound

The sound is recorded externally through microphones. Depending on the specific recording session set-up, either two or three microphones are used. In all recordings one microphone is placed on the mother's head, reaching her mouth, and the same is the case for the child. In some recording sessions an extra microphone is hung from the ceiling. For the purpose of this master thesis, only the two microphones positioned on the child and mother have been considered, meaning that two channels are used in the data processing. Channel 1 is the child's microphone and channel 2 the mother's. It is to be noted that the

mother's utterances are registered in the child's microphone and the other way around.
The sampling frequency of the audio signals is 48000 Hz which corresponds to about 28.8 millions samples per channel during the 10 minute session. The audio signals are in the format *.wav*.

3.2 Motion Capture

Markers are attached to both the mother and child for the purpose of motion capture recordings. The position of the markers can be seen in figure 3.1.

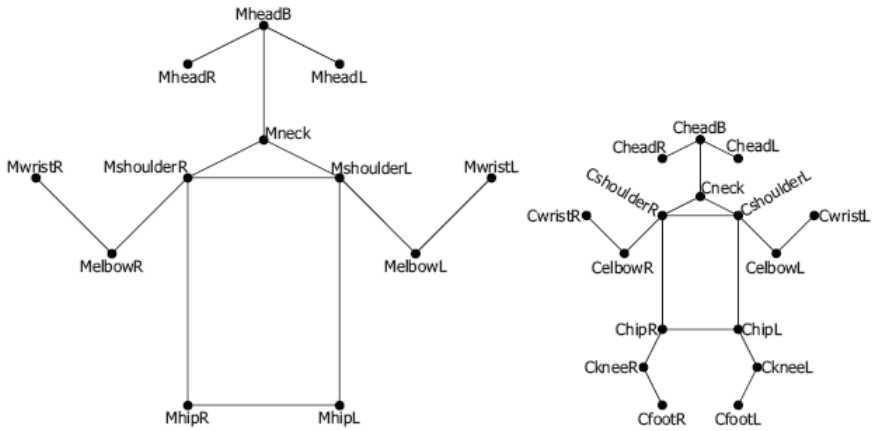


Figure 3.1: The position of the markers, to the left the mother, to the right the child. Figure from [34].

The motion capture data are recorded by the system Qualisys where 8 infra red cameras collect the 3-D positions of the markers placed on the mother and child. The sampling frequency is 60 Hz, corresponding to approximately 36000 frames per dyad per session. Despite the 8 cameras collecting the marker positions, some markers remain unidentified by Qualisys, because they, in one or more frames, are completely shadowed by either the mother or the child. For this reason, student assistants from Babylab identify these manually, if possible, after the recording session. The data recorded in Qualisys can be directly saved as a *.mat*-file and thereafter loaded into Matlab. Figure 3.2 shows the experimental set-up of the room as viewed from Qualisys.

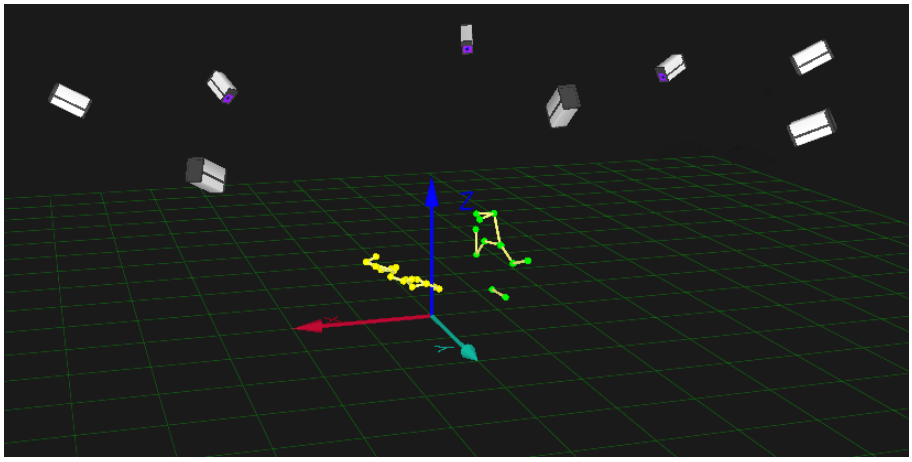


Figure 3.2: The experimental set-up of the room with the 8 infra red cameras as visualised in Qualisys. The markers illustrating the mother are shown in green and the markers illustrating the child are shown in yellow. The coordinate system of the room is likewise illustrated, with the red arrow indicating the x-axis, the turquoise arrow indicating the y-axis and the blue arrow indicating the z-axis. The point of origin of the coordinate system is located at the exact same position for all sessions.

3.3 Video

In all of the recording sessions two video cameras are included. These cameras record the interaction between the mother and child with a sampling frequency of 25 Hz, corresponding to around 15000 frames per camera per recording session. Each video file is in the format *.avi* and consists of one video track and two audio tracks. The position of the video cameras has not been the same for all sessions, but for all the latest recordings, the two cameras are located with the focus as shown in figure 3.3.

(a) (b)

Figure 3.3: The experimental set-up with the focus of each of the two cameras. (a) The focus of video camera 1. (b) The focus of video camera 2.

3.4 Annotations

As mentioned in the problem statement, chapter 2, Babylab has different coding groups that are in charge of making specific annotations manually. The number of dyads for which annotations have been made, differs depending on the coding group. None of the annotations have been made for all dyads. The annotations already made by Babylab are mentioned in the following under the modality that is used by Babylab for the specific annotation.

Sound

- Speaker identification with the classes
  - child speaking
  - mother speaking
  - both speaking
  - silence
- Child's emotional state with the classes
  - protest
  - no protest (satisfied)
- Mother vocalising with the classes

- singing
- speaking

Motion Capture

- Distance between faces
- Child's physical energy level

Video

- Child's head position
- Joint attention
- Child's facial expressions
- Gaze

The sound signal annotations, i.e. speaker identification and emotion recognition, are executed in the free-ware program Praat, where a basic script indicates the intervals of mother speaking and child speaking, respectively, from an intensity measure. From this, the coder's job is to listen to the sound file and manually move or remove the suggested intervals of speech. For the manual emotion recognition task, the intervals indicating that the child is speaking, are divided, by the coder, into protest and no protest. The same is the case for the mother's vocalizations, i.e. the coder is to determine whether the mother is speaking or singing.
The distance between the mother's and child's faces is calculated in Excel by coders at Babylab. For this, the marker coordinates of the heads from Qualisys are used. Excel is also used to annotate the child's physical energy level where the right wrist marker is used as indicator.
The video coding group at Babylab annotates the above mentioned physical interaction patterns. Regarding the child's head orientation, the coders are to determine how much the child's head position deviates with respect to the mother from the starting position, that is the child facing the mother. This is elaborated in chapter 7, where this annotation is automated through the use of motion capture marker coordinates.
The joint attention, that provides information on the joint focus of both mother and child on an object in the room, is extracted by Babylab from the video files. To automate these it would probably be more correct to apply the head direction from the motion capture head marker coordinates through vector calculus.

This is not approached in this thesis, but instead left for future work.
The child's facial expressions are extracted from the video files, where an important factor to the psychologists at Babylab is that the sound is off. The sound of the child could possibly affect the coder in deciding on a different label than if only the visual information is available. The facial expressions include the positions of the mouth, cheeks, eyes and forehead. The group at Babylab that are conducting these annotations follow a particular scheme that can be seen in appendix A. The facial expression annotations will not be automated in this thesis, but a small test will be conducted in order to obtain an idea of the possibilities within this area. This can be seen in section 8.2 and in appendix B.
The last annotation that have been extracted by Babylab is the gaze of the child. For this, the video recordings have been applied, which is the only recording modality that enables detection of eye direction. This annotation is not attempted automated in this thesis due to the poor pixel resolution of the child, as earlier mentioned.

Chapter 4

Synchronization

To be able to combine the three recording modalities and make use of the information extracted from one modality in the analysis of another, time synchronization across the modalities is a necessity.
The external sound recording is started manually before each session and this action is then directly connected to a trigger, that starts the video and the motion capture recordings. This, naturally, creates a synchronization problem. After loading all three measurement modalities into Matlab, but before further data processing, synchronization is performed. The delay estimations are carried out between the sound and video and between sound and motion capture. By solving these two separate synchronization problems the third problem, video and motion capture, is given.
The psychologists at Babylab are aware of the synchronization issues but have only been capable of solving the sound to video synchronization problem. Their approach is to, manually for each recording, mark out three clear sounds during the 10 minute sessions and find the time delay between these sounds in the video recordings and in the external sound recordings. The average of these three time delays has been assumed to explain the issue of synchronization between sound and video respectively. For this, and for much of Babylab's other analyses, the free-ware program Praat is used.

4.1 Sound versus Video

As explained in chapter 3, the external sound file contains two channels, i.e. the sound recorded from the child's microphone and the sound recorded from the mother's microphone. The video files consist of two audio tracks and a video track. It is, with good reason, assumed that the three tracks constituting the video file are fully synchronized. This assumption makes it possible to identify the sound-to-video time delay through analysis using the cross-correlation between of one of the audio tracks in the video file and one of the channels in the external sound file. The set-up of this approach is shown in figure 4.1.
The applied cross-correlation method is given by equation (4.1).

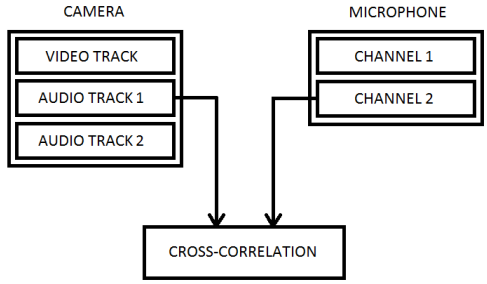


Figure 4.1: The set-up for the cross-correlation approach. The shown combination of video and sound signals is the one used in this thesis.

$$\theta_{fg}(n) = \sum_{m} f(m)g(n+m) \tag{4.1}$$

The cross-correlation function between two signals is calculated by retaining the first signal at the same position, whilst the second signal is moved on top of the first, one sample n at a time. For each position n of the second signal, the sum of the multiplication of the two signals at each sample is calculated. The position of the moving signal that gives the largest correlation value, will correspond to the time lag where the two signals are most alike. It should be noticed that the cross-correlation formula given by (4.1) is not normalized. The segments of the signals being cross-correlated with each other in this study have the same length and the normalization would therefore not have a high impact.

The audio signal from the video file and the external sound signal will be very similar because all recordings take place in a closed room. This causes the correlation value to have a large peak at the time lag corresponding to the synchronization difference. It should be mentioned here, that the audio signal from

the video file is delayed in itself with respect to the external sound signals, because of the position of the cameras compared to the head microphones, recall figure 3.3. This delay would in the signal correspond to the sound delay with the given distance, but because of the small distance and speed of sound measure being 340.29 m/s, this delay is assumed negligible.

Figure 4.2 shows the cross-correlation result for dyad 011. Here the external sound signal is held at the same position and the audio signal from the video file is moved one sample at a time. This is done for three smaller intervals of the two signals, i.e. in the beginning, the middle and the end, respectively.
It is possible to calculate the time delay using the entire signal, but some issues are associated with this approach. The first problem is that a computer with much processing power is needed because of the full signal size (10 minutes with a sampling frequency of 48000 Hz). Another issue that is possibly present, is that a further delay or reduction in delay between the two signals during the 10-minute sessions could occur, due to the time settings in the two recording devices. If the time delay between the two signals is found at several signal intervals, this uncertainty is taken into account. That three intervals are used in the calculation of the time-delay also reflects the approach of the psychologists at Babylab.
A necessity for the cross-correlation method to work is to represent the two signals with the same sampling frequency. With the sound signal having a sampling frequency of 48000 Hz and the audio track from the video signal having one of 32000 Hz, the sampling frequency of 16000 Hz is the largest common sampling frequency obtainable when down-sampling the signals. Both signals are therefore down-sampled accordingly.
In figure 4.2 it is observed that the three peaks (although the middle one being very small) are positioned around the same time lag. The exact time lag between the two signals and the corresponding delay in seconds for the three intervals are shown in table 4.1.
The synchronization differences in seconds are calculated as in the following ex-

Interval	Time lag in samples	Delay in seconds
1	38,678	2.4174
2	38,763	2.4227
3	38,846	2.4279
Average	**38,762 ± 84**	**2.4227 ± 0.0053**

Table 4.1: The time lag and delay in seconds for dyad 011, for the three intervals. The average of the three are likewise shown.

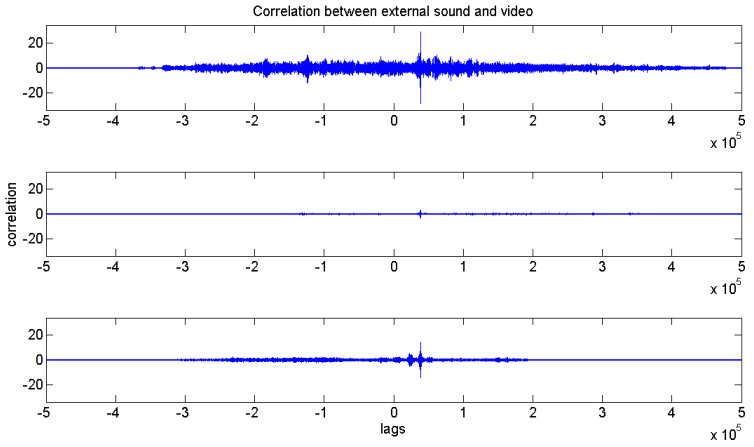


Figure 4.2: The three cross-correlations between the external sound signal and the audio signal from the video file, dyad 011.

ample: (38,678 samples)/(16000 samples/s) = 2.4174 seconds. Since the time lag is positive, the external microphone signal is delayed 2.4174 seconds compared to the audio signal in the video file. The mean of the three time intervals is 2.4227 ± 0.0053. In the manual annotations from Babylab a result of 2.4355 ± 0.0008 seconds was obtained. Thus, the delay obtained through the automatic method is extremely close to the manually obtained delay.
To adjust the delay and remove the synchronization difference, the first 38,762 samples, as being the average of the three intervals, should be removed from the external audio signal. An action that makes the two files (video and sound) start at the same time.

In practice, there are a few issues that have been discussed prior to the actual calculations. As mentioned in the beginning of this section, each video file contains two audio tracks and the external sound file contains two sound channels. This means that there are four possible combinations when applying the cross-correlation method for each video camera. Since the two external sound channels are synchronized and so are the two audio tracks from the video files, only one signal from each recording modality is required to make the above explained calculations.
It has been chosen to use channel 2 from the external sound file, representing the mother. In general, the mother speaks much more often and much louder than the child, making the speech signal from the mother presumably more identifiable in the video microphones as they are positioned further away (see

figure 3.3, section 3.3). Furthermore the channel 1, representing the child is quite noisy which would make it hard to identify this channel with the video microphones.
Regarding the two video files, the automatic approach used in this thesis takes its starting point in the work already done by Babylab. Therefore, for the sake of comparison, the video file used by Babylab for the synchronization will be used here as well.

The results for the average time delay of the sound-to-video synchronization are shown in table 4.2. What is seen in the table is that when taking the standard deviation into consideration, the results obtained through the cross-correlation method are extremely close to the results obtained with the manual method. What is furthermore observed from the table, is that the results for some dyads are missing. For dyad 005 and dyad 007, no data is provided from Babylab. For dyad 013, 014, 016, 018, 019, 020 and 021 there is no sound on the video files, making it impossible to extract the time-delay through this approach.

4.2 Sound versus Motion Capture

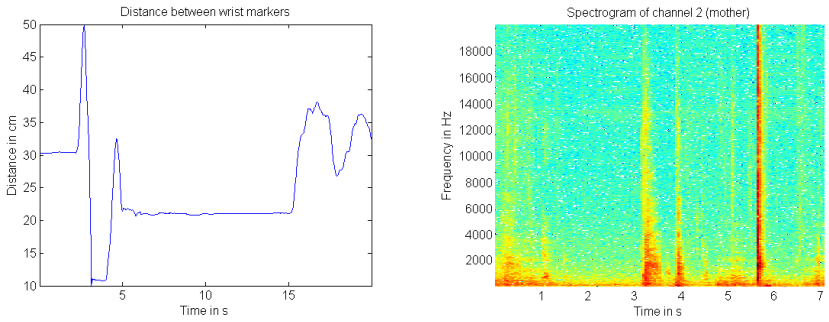
Two approaches to the issue of synchronization between sound and motion capture have been executed. The first is the correlation of distance profiles. These represent the mutual movement between mother and child throughout the 10-minute session and are calculated from the mocap file and from the external sound files, respectively. Several uncertainties regarding this method caused the results to be incorrect. The details on the calculations and the results are discussed in appendix C. The reason that this method was implemented in the first case implemented is due to its general applicability, in that the distance profiles can be calculated for all dyads.

The second method uses the starting information given by the mother, in the form of a clap. The time of the clap can be extracted from the mocap files, as the frame where the distance between the mother's wrist markers is minimized. Figure 4.3(a) illustrates the distance profile of the mother's wrist markers for the first 20 seconds for dyad 011.
From the external sound signal, the time of the clap can be extracted through the use of the spectrogram. This is done by locating the time where the sum of the power at each frequency reaches it maximum. Figure 4.3(b) shows the first 7 seconds of the spectrogram for dyad 011. Several issues have been considered

Dyads	Video (1 or 2)	Cross-corr method	Manual method
001	video 2	2.4788 s ± 0.0015 s	2.4811 s ± 0.0008 s
002	video 2	2.4593 s ± 0.0122 s	2.4629 s ± 0.0012 s
003	video 2	2.2569 s ± 0.0079 s	2.2698 s ± 0.0058 s
004	video 2	2.7417 s ± 0.0151 s	2.7514 s ± 0.0002 s
006	video 1	2.7399 s ± 0.0017 s	2.7410 s ± 0.0062 s
008	video 1	2.0270 s ± 0.0025 s	2.0535 s ± 0.0014 s
009	video 2	-0.8516 s ± 0.0120 s	-0.8509 s ± 0.0029 s
010	video 1	2.3623 s ± 0.0040 s	2.3629 s ± 0.0016 s
011	video 2	2.4227s ± 0.0053s s	2.4355 s ± 0.0008 s
012	video 1	2.7211 s ± 0.0052 s	2.7354 s ± 0.0007 s
015	video 2	2.1416 s ± 0.0064 s	2.1389 s ± 0.0010 s
017	video 2	2.1615 s ± 0.0101 s	2.1685 s ± 0.0010 s

Table 4.2: The time delay between external sound and video for each dyad. Both the results from the automatic approach developed in this thesis and those obtained from the manual method are shown. The delay shown is the mean of the three time delay for the three time intervals together with the corresponding standard deviation.

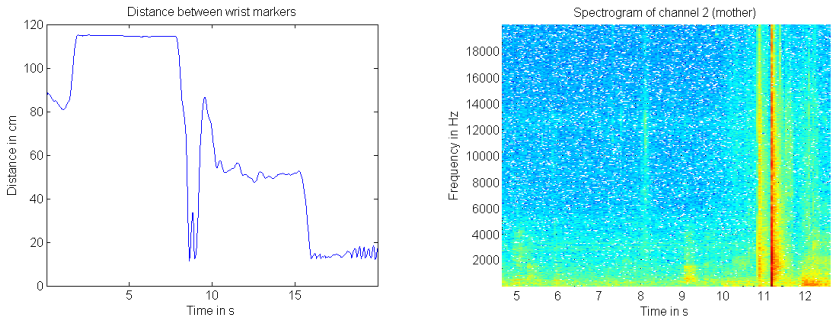
during the practical development of the method. First, sometimes the mother holds her hands as close or closer to each other than during the clap. This, of course, will result in a wrong time-of-clap estimation. To avoid this scenario, only the first 20 seconds of the mocap files will be used in the wrist distance profile, since it is assumed that the mothers perform the clap during this interval. Regarding the clap-identification using the spectrogram, several sounds holds the same amount of power (or more) as the clap. This makes it uncertain if the time instant with the highest power actually corresponds to the time of the clap. The approach has therefore been to first identify the time of the clap from the wrist distance profile, denoted here as T_{clap}. The interval of $T_{clap} \pm 4$ seconds is subsequently analysed, spectrogram-wise. The choice of this particular interval is chosen based on the delays found between the external sound files and the video files, which are assumed to correspond, more or less exactly, to the delays between the external sound files and the mocap files.
Figure 4.3 illustrates a case where the mother only claps once. In several of the sessions the mother claps twice, inducing another problem. The distance



(a) (b)

Figure 4.3: Example for dyad 011. (a) Distance profile of the mother's wrist markers of the first 20 seconds. The clap can be identified as the minimum of the curve at around 3.5 seconds. (b) Spectrogram of the first 7 seconds. The clap can be identified at close to 6 seconds as the darker red column.

between the wrists during the two claps are not necessarily exactly the same, which makes it uncertain which of the two claps are extracted by the algorithm. Likewise for the clap in the spectrogram, it is of uncertainty whether the first or the second clap holds most power to it. This is actually a problem for dyad 001 illustrated in figure 4.4. Here, the first minimum of the distance profile in figure 4.4(a) corresponds to the global minimum of the first 20 seconds, whereas the second clap holds most power to it, which is clear from figure 4.4(b). The calculated time delay between the external sound file and the mocap file for dyad 001 is 2.55 seconds, but the true time delay (from second clap in distance profile to second clap in spectrogram) is 2.23 seconds. It should be stated at this point, that the case of uncertainty about the time instant of the clap only causes a problem if the process is to be executed totally automatically. If the figures of the distance profiles as well as the spectrograms of the clap are visually inspected, no doubt is in evidence which time instant of the clap belongs to the first or second clap.
Other problems that have been discovered during the development and use of this method include the fact that not all mothers perform the clap, and that the strength of the clap is crucial to the detection of the clap from the spectrogram. The dyads for which the true delay has been found by the algorithm are listed in table 4.3. It should be stated that the precision of this method is limited of the sampling rate of the mocap files of 60 Hz. This causes a limit of the precision of the clap of $1/60 = 16.7$ ms. Furthermore, if the mother claps very slowly, the clap would occur over more frames, and the uncertainty about the exact frame of the clap arises.



(a) (b)

Figure 4.4: Example with two claps, dyad 011.(a) Distance profile of the mother's wrist markers of the first 20 seconds. The clap with the minimum distance is identified at around 8.6 seconds. (b) Spectrogram of the interval [4.6 : 12.6] seconds. The clap with the maximum power can be identified at around 11.2 seconds as the darker red column.

To briefly sum up the issues of applying this method, it should be recalled that

Dyads	True time delay
011	2.5167 s
015	2.2500 s
021	2.1667 s

Table 4.3: The dyads for which the true time delay between external sound and mocap has been extracted, through the use of the clap method and the corresponding true time delay.

the method is dependent on visual inspection of the profiles considered, as well as is limited to the rate at which the mocap files have been recorded.

4.3 Video versus Motion Capture

Since the recording session is started at the starting time of the external sound recording and because this triggers the video and Qualisys recordings, the expectation is that there is no difference in synchronization between the video and

the motion capture modalities. With this in mind, it is still of importance to make the investigation, because a synchronization difference, in the worst case, could deteriorate the results of the multi-modal studies of this thesis.
To extract synchronization information between the video files and the Qualisys files, the synchronization differences found in the above mentioned methods for sound-to-video and sound-to-mocap can be compared. Table 4.4 shows the time delays found for both problems, where this has been possible.
For dyad 001 the time difference between video and mocap has the same mag-

Dyads	sound-to-video	sound-to-mocap	Difference
001	2.4788 s	2.2300 s	-0.2488
011	2.4227 s	2.5167 s	0.0940
015	2.1416 s	2.2500 s	0.1084

Table 4.4: The dyads for which both the sound-to-video and sound-to-mocap synchronization difference have been extracted and the corresponding time delays. The column difference shows the difference between the sound-to-video and the sound-to-mocap.

nitude but opposite sign compared to the other two dyads. This indicates that the order in which the video and infrared cameras are started is random. When looking at the column *difference* in table 4.4 it can be seen that the difference between the sound-to-video and sound-to-mocap is very small. When taking the uncertainty about the individual measurements into account, it could seem like no delay is in evidence between sound-to-video and sound-to-mocap. The foundation necessary to make a conclusion on the video-to-mocap time difference is very vague, but from the three results in the table the tendency is that the time delay is so small that it could be thought of as not existing.

Chapter 5

Speaker Identification

In speaker identification the task is to identify a given voice from a group of known voices. To be able to do this, it is necessary to extract information from the speech signal that can reveal the identity of the speaker. Information on the words spoken are, on the other hand, of lesser importance. In contrast, in the task of speech recognition, the speaker-carrying qualities of the speech signal are irrelevant and instead information on the utterances (word or sentences) are to be extracted. Speaker identification can either be text-dependent or text-independent. If the task is text-independent, the system only relies on vocal tract characteristics of the speaker, whereas in text-dependent speaker identification information on the spoken utterances are included as well, [57], [25]. Text-independence is therefore most often assumed in speaker identification, since this does not make any assumption about the speech, and therefore can be more widely used, [12].
Regarding the mother/child interaction, it is of great interest for BabyLab to obtain automatic annotations of whether the child or the mother is speaking, if they both are speaking at the same time or if there is silence, see table 5.1. In this case the speaker identification is a text-independent, 4-class problem. The amount of data available for the speaker identification task is 15 dyads each providing 10 minutes of spoken interaction.
In the following section, 5.1, details on speech and speech perception is given. Speech as a signal and the general preprocessing performed before speaker identification is possible, is explained in section 5.2. Sections on the extracted fea-

Class	Class definition
1	Child speaking
2	Mother speaking
3	Both speaking
4	No speech

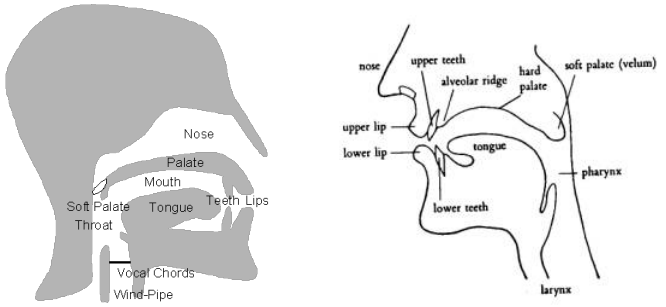
Table 5.1: The class definitions.

tures, 5.3, and on the performed classifications, 5.4, follows subsequently, where the section on classification includes a detailed explanation on the applied classifiers. In the last section, 5.5, different approaches for generalizing the model as well as boosting the performance of the classifiers are discussed.

5.1 Speech and Speech Perception

This section is not provided to give an exhaustive explanation on the anatomy of speech production, but instead to outline the nature of speech and of speech perception, to obtain an understanding of the feature extraction from the sound signals. The perception of speech takes place in the human auditory system. A total comprehension of speech perception, would provide the solution to how the speech signal should be modelled to identify speakers from each other, due to the fact that speaker identification for the human brain is a rather simple task. The following description takes basis in [12], [50].
In figure 5.1(a) the anatomy of the vocal tract system is shown. The production of speech starts in the lungs, forcing air up through the vocal cords. These, as seen in figure 5.1(a), has the ability of vibrating, where the frequency of this vibration is controlled by the muscles in the larynx. The frequency at which the vocal cords vibrate are typically higher for female speakers than for male speakers and the sound is hereby given its so called fundamental frequency. The mouth, throat and nose all contribute to modifying the sound from the vocal cords, giving the sound its tone. The ability of pronouncing vowels and consonants, and thereby pronouncing utterances, stems from the movement of the articulators of speech which is the pharynx, soft plate, lips, jaw and tongue, seen in figure 5.1(b). From this, it is clear that the voice of one person is individual from another.

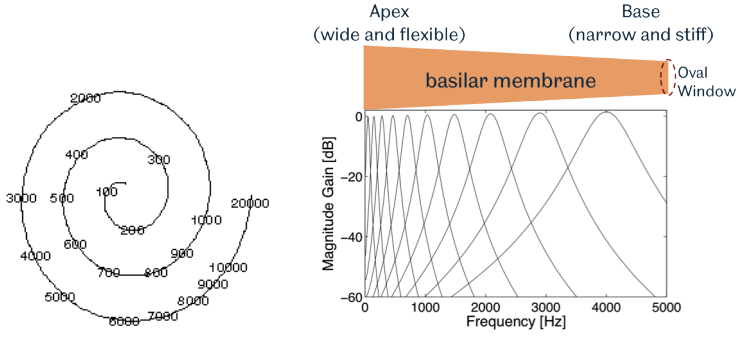
Concerning speech perception, the human ear has the ability of separating a



(a) (b)

Figure 5.1: Figure showing (a) the anatomy of the vocal tract, figure from [1]. (b) the articulators of speech, figure from [2].

sound into its frequency components, an ability called frequency selectivity. Frequency selectivity takes place on the basilar membrane of the ear. Each position along the membrane is more sensitive to one particular frequency than to all other frequencies, see figure 5.2(a). Thus, the spectral composition of a sound can be extracted by the human auditory system. Mathematically, the basilar membrane can be represented by a bank of overlapping band-pass filters, which can be visualized as figure 5.2(b). It can be seen in the figure that the spacing of the filters is not linear but logarithmic which explains why the filters are more closely spaced at the lower frequencies than the higher ones. The



(a) (b)

Figure 5.2: The concept of frequency selectivity where (a) shows the frequency selectivity of the basilar membrane, figure from [3]. (b) The basilar membrane represented as a filter bank of band-pass filters, figure from [4].

impressive function of the ear regarding frequency selectivity encourages the use of mathematical models to extract the same information from a speech signal as the ear is capable of. This is approached in section 5.3 on feature extraction.

5.2 Preprocessing

Before analysing a signal, stationarity must be established since this is an assumption in most signal processing methods. A stationary signal is defined as a signal whose statistical parameters, such as mean and variance as well as frequency content, do not change over time, [12]. Figure 5.3 presents the signal from the mothers microphone extracted from 35 seconds to 40 seconds. When inspecting the figure, it is clearly seen that the frequency varies over time which means that the signal should be interpreted as a non-stationary signal.

A way to obtain stationarity is to divide the non-stationary signal into quasi-

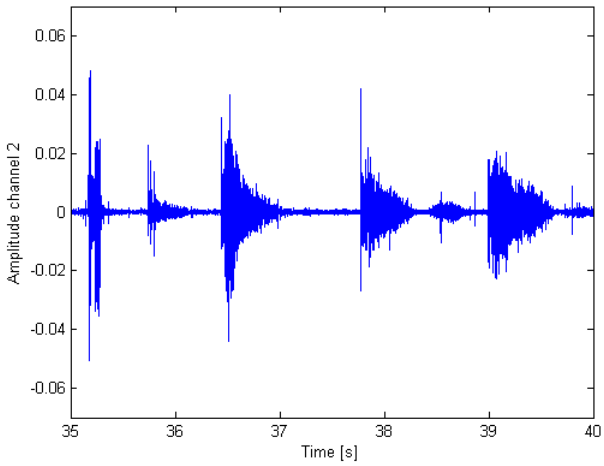


Figure 5.3: The signal from channel 2 shown from 35-40 seconds. It is noticed that the speech signal is a non-stationary signal.

stationary segments, where each of these segments are analysed separately. This means that the signal is divided into windows of a given sample size in which it is assumed that the characteristics of the signal do not change significantly, [54]. The window sizes dealt with in this thesis are 10 ms, 50 ms, 100 ms, 150 ms, 200 ms and 250 ms. With a signal sampling frequency of 48000 Hz this corresponds to window sizes of (480, 2400, 4800, 7200, 9600, 12000) samples.

The choice of the window sizes is first of all due to the stationary concept already mentioned. Second, the research project at Babylab was started with inspiration from [29] where the windows are chosen to be 250 ms. Third, the lower boundary at 10 ms stems from the accuracy of the manual annotations made in Praat by Babylab. Forth and last, windows of size from 5 ms to 100 ms are used in the literature regarding speaker identification, [57], [24], where in [24] it is also pointed out that the concept of stationarity holds for segment up to about 200 ms in size.
Depending on the window size, the amount of observations available varies. In table 5.2 the number of observations in each class is shown for the 14 dyads constituting the training set for the 6 different window sizes.
The manual annotations made at Babylab are, as mentioned, carried out in the

Class	10 ms	50 ms	100 ms	150 ms	200 ms	250 ms
Child	88172	17644	8913	5849	4434	3504
Mother	281861	56389	28468	18788	14275	11258
Both	115872	23175	11473	7735	5753	4617
No one	404881	80938	40212	27004	20064	16239

Table 5.2: The number of observations belonging to each class for the data set consisting of 14 dyads at 4 months, for each of the six different window sizes.

programme Praat with an accuracy of 10 ms, i.e. one class label exists for every 10 ms. These annotations are used as the ground truth to the speaker identification classifiers in this study. This implies that when increasing the signal's window size as input to the classifier, the true class vector must be changed accordingly. Majority voting is used to obtain the new class label vectors. For instance, if the window size is 50 ms, the class label of this window is determined by the majority of the 5 annotations from the 10 ms class vector. Hereby the number of class labels from Babylab matches the number of segments in the windowed sound signal.

As mentioned in section 5.1 the frequency with which the vocal cords vibrate varies, depending on the word or utterance being pronounced as well as the person pronouncing it. The spectral content of the four respective classes is therefore expected to vary. In figure 5.4 four spectrograms are shown, each representing one of the four respective classes. By using the true class labels, 450 ms of each class in channel 1 has been pointed out. The spectrograms show the frequencies up to 7000 Hz since the main part of the frequency content lies

in this area. Looking at the spectrograms in figure 5.4, it is observed that the

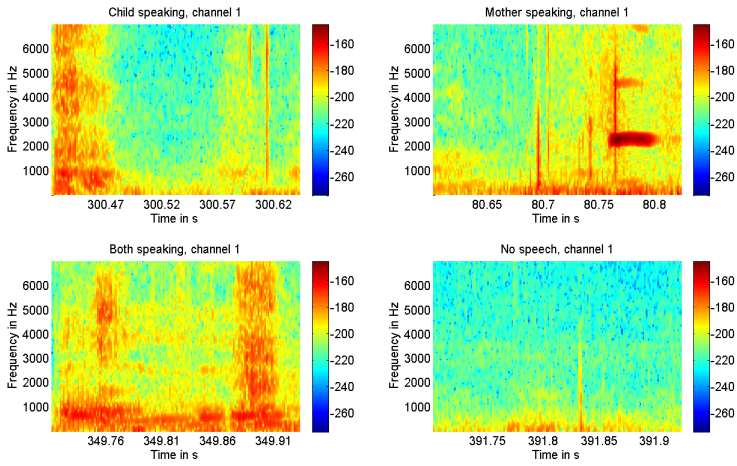


Figure 5.4: Spectrograms of the four respective classes with a duration of 450 ms showing the frequencies up to 7000 Hz.

spectral content of the four respective classes deviates from each other visually. Comparing the spectrograms representing the mother speaking and the child speaking it is seen that they have very different spectral content. The spectrogram of the child's speech seems to have no dominant frequency, but instead a frequency content that covers all the illustrated frequencies in the first part of the shown interval. The opposite is valid for the mother's spectrogram. Here, by far most of the frequency content is centred around 2000 Hz in the last part of the time interval, implying speech in this part of the signal. The spectrogram of both speaking is observed to have smaller time intervals of frequency content similar to both the mother's and the child's spectrograms. The last spectrogram represents the class where no one is speaking. No signal should be detected due to the labelled silence, which means that the frequency components represented is because of noise in the recordings.
To sum up, based on the difference in the spectral content of each of the classes, it appears that spectral features are useful in distinguishing between the four classes.

5.3 Feature Extraction

Feature extraction is performed to obtain a finite representation of each signal segment. To obtain the best possible classifier, the features extracted should represent those qualities of the sound signal that maximize the differences between the four classes and at the same time minimize or eliminate those of irrelevance for the classification. Features of unimportance could deteriorate the performance of the classifier which of course is undesirable. The curse of dimensionality is most often an issue with practical data sets, which is why only the features with the greatest impact on the classification should be included in the final model constellation. As indicated by it's name, curse of dimensionality occurs when the number of features is to large compared to the number of observations, in which case modelling of the data becomes more or less impossible. For a more thorough explanation of the curse of dimensionality see section 5.3.3. The features to be used as input to the classifier are divided into two types: the time-domain features and the frequency-domain features. This section holds a detailed explanation of each of the involved features, where the time-domain features are approached first, section 5.3.1, after which the frequency-domain features are explained, section 5.3.2.

5.3.1 Time-domain Features

In the time-domain, a feature that carry speaker-dependent information, and therefore could assist in the classification of the mother and child speech sequences, is the cross-correlation between the two channels. For a detailed explanation of the cross-correlation see chapter 4. The approach for calculating the cross-correlation in discrete time is shown in (5.1). This equation corresponds to equation (4.1).

$$\theta_{fg}(n) = \sum_{m} f(m)g(n+m) \tag{5.1}$$

In equation (5.1), f and g represent the two audio channels, with f being the mother's signal and g being the child's signal. In this case, if the peak of the cross-correlation is at a positive lag, the mother's signal is delayed compared to the child's, which therefore clearly indicates that the child is making an utterance. The opposite is for the same reason assumed valid for a peak at a negative lag. Furthermore, through testing, it was observed that the cross-correlation in many windows did not have a clear peak, suggesting that either no one or both are speaking. Based on these factors, the cross-correlation could be a relevant feature in the classification. It should be noticed that the cross-

correlation formula given by (5.1) is not normalized. The segments of the signals being cross-correlated with each other in this study have the same length and the normalization would therefore not have a high impact.
Another feature that has been used frequently in the literature is the zero-crossing rate (zcr) of the speech signal, [18]. For each time window, the number of times that the speech signal crosses the time axis, corresponding to a change of sign of the signal, is a simple representation of the frequency content at that specific part of the speech signal, [52]. Equation (5.2) displays the mathematical approach for calculating the zcr.

$$zcr = \frac{1}{2N}\sum_{n=1}^{N}|sgn(x(n)) - sgn(x(n-1))| \tag{5.2}$$

In equation (5.2), N is the total number of samples in the specific time window and x represents the windowed sound signal. All changes in the sign of x will be summed (if no change in sign occurs, the expression $|sgn(x(n)) - sgn(x(n-1))|$ is equal to zero), but because of the nature of the sgn function ($sgn(x) > 0 = 1$, $sgn(x) < 0 = -1$), the aforementioned expression will give the value 2 if a change in sign is observed. This is taken into account by dividing by two outside the sum. To obtain the rate of the zero-crossings, the output from the sum is divided by the number of samples in the time window.
A high zcr corresponds to a frequency content consisting primarily of high frequencies and vice versa for a low zcr. In general, most of the energy of voiced speech (movement of the vocal cords) is found below 3 kHz, whereas for unvoiced speech (speech produced only by air and the mouth movement) the energy majority falls in the higher frequencies, [52]. A difference in zcr could therefore possibly be found in the speech of the mother and of the child. Furthermore it is imaginable that the zcr for no speech (corresponding to noise) would differ from that of speech.
A third feature that is commonly used in speaker identification tasks is the energy of the windowed signal. This is given as the sum of squares of the amplitudes within a segment [18]. The equation for calculating the energy is shown in (5.3).

$$energy = \sum_{-\infty}^{\infty}|x(n)|^2 \tag{5.3}$$

The x in equation (5.3), represents the windowed audio signal. The amount of energy directly relates to whether or not speech is present in each frame, with a high energy level indicating a speech-filled window and vice versa for a low energy level. The energy is for that reason assumed to be a valuable feature in the separation of the windows of no speech from the remaining windows.

5.3.2 Frequency-domain Features

Regarding the frequency-domain features, especially the mel-frequency cepstral coefficients (MFCC's) have been applied in more recent studies on speaker identification, [24], [55], [46]. These coefficients are based on the Mel scale which explains the subjective relationship between the pitch of a sound and its acoustic frequency. Since the Mel scale represents a mathematical interpretation of the human ability to perceive tones, it is one of the most realistic approaches to sound perception in the area of speaker and speech identification. See section 5.1 for a more thorough description of the human perception.
The Mel scale has been interpreted in several different ways throughout the last decades, but the implementation used in this study is the Isound toolbox, [30], as represented by M. Slaney in the Auditory toolbox [61]. The survey conducted in this thesis on MFCC as can be read in the following, takes its basis in the two books [21], [12].
The MFCC interpretation by [61] consists of a filter bank of 40 overlapping, equal-area, triangular filters. Of the 40 filters, the first 13 have linearly-spaced center frequencies (f_c) with a distance of 66,7 Hz between each, whereas the last 27 have log-spaced f_c's separated by a factor of 1.0711703 in frequency. The center frequencies for the 40 filters are expressed in equation (5.4).

$$f_{c_i} = \begin{cases} 133.33333 + 66.66667 \cdot i & , i = 1, 2, ..., N_{lin} \\ f_{N_{lin}} F_{log}^{i-N_{lin}} & , i = N_{lin}+1, N_{lin}+2, ..., N_{lin}+N_{log} \end{cases} \tag{5.4}$$

To avoid confusion, i here indicates the filter index and is therefore unrelated to the complex i. In equation (5.4), f_{c_i} is the i'th center frequency of the filter bank, N_{lin} is the number of linear filters and N_{log} the number of log-spaced filters. $f_{N_{lin}}$ is therefore the center frequency of the last linear filter ($f_{c_{13}}$). $F_{log} = \exp(\ln(f_{c40}/1000)/N_{log})$, where f_{c40} is the center frequency of the last filter in the filter bank. Therefore $F_{log} = 1.0711703$ as mentioned above.
The entire filter bank cover the frequency range [133.3:6855] Hz where each filter is defined as in equation (5.5).

$$H_i(k) = \begin{cases} 0 & \text{for } k < f_{b_{i-1}} \\ \dfrac{2(k - f_{b_{i-1}})}{(f_{b_i} - f_{b_{i-1}})(f_{b_{i+1}} - f_{b_{i-1}})} & \text{for } f_{b_{i-1}} \le k \le f_{b_i} \\ \dfrac{2(f_{b_{i+1}} - k)}{(f_{b_{i+1}} - f_{b_i})(f_{b_{i+1}} - f_{b_{i-1}})} & \text{for } f_{b_i} \le k \le f_{b_{i+1}} \\ 0 & \text{for } f_{b_{i+1}} > k \end{cases} \quad \text{, i = 1,2,...,M} \tag{5.5}$$

In equation (5.5), $i = 1, 2, ..., M$ is the i'th filter of the M-sized filter bank, $k = 1, 2, ..., N$ is the k'th coefficient of the N-point DFT and $f_{b_{i-1}}$ and $f_{b_{i+1}}$ are

the lower and the higher boundary point, respectively. f_{b_i}, which is equal to the center frequency of the i'th filter (f_{c_i}), corresponds to the point of the filter where most of the original frequency content is passed through.
Figure 5.5 illustrates the equal-area filter bank. In theory, the first 13 filters

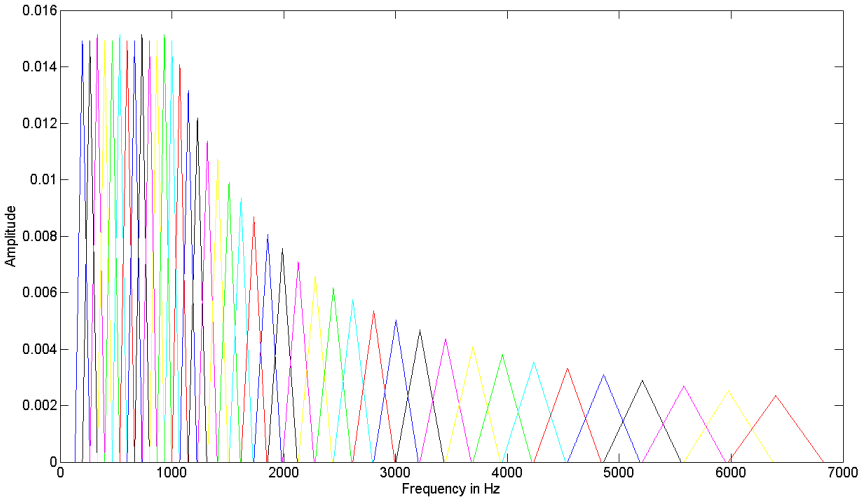


Figure 5.5: The 40 equal-area filter bank as introduced by [61]. In theory, the first 13 filters should have equal height due to the linear spacing between them, but due to round-off's in the spacing in Matlab, small variations can be observed. Every filter has a shape of a triangle and is represented by different colours.

should have equal height due to the linear spacing between them, but due to round-off errors in the spacing, small variations can be observed in the figure.
The approach to express the sound signal on the Mel scale is to take the Fourier transform of the windowed signal, to obtain the frequency spectrum of each segment. The window function used in this thesis for the MFCC extraction is a Hamming window. The frequency spectrum of each segment is then converted to the Mel scale by multiplying the magnitude of the spectrum with the aforementioned filter bank. The logarithm of the converted spectrum is taken, expressing the output of each filter in dB to obtain a more precise representation of the manner in which humans perceive sound. This step can be seen in equation (5.6).

$$S_i = log_{10}\left(\sum_{k=0}^{N-1}|S(k)|H_i(k)\right),\ i = 1, 2, ..., M \tag{5.6}$$

In equation (5.6), the $|S(k)|$ is magnitude of the DFT-obtained frequency spectrum and H_i is the Mel frequency filter for the ith filter.

By using the Discrete Cosine Transformation (DCT), the Mel Frequency Cepstral Coefficients can be extracted, as expressed in equation (5.7). It is to be noted that since the DCT is a fourier-related transform, see [5], using the DCT on the Mel frequency spectrum converts it to the Mel frequency cepstrum, with cepstrum being the spectrum of a spectrum.

$$MFCC(r) = \sqrt{\frac{2}{M}} \sum_{i=0}^{M-1} S_{i+1} \cos\left(\frac{(i+0.5)\pi r}{M}\right), \; r = 0, 1, ..., R-1 \qquad (5.7)$$

In equation (5.7), the S_{i+1} is the filter bank output from equation (5.6) where $i = 1, 2, ..., M$ with M being the number of filter banks. Since the sum index starts at $i = 0$, the filter bank output has the index $i + 1$. Equation (5.7) gives R unique MFCC's, where $R \leq M$. If R is chosen larger than M, these MFCC's mirrors those of the first M coefficients, [21].
Figure 5.6 illustrates the MFCC-extraction from the raw speech signal to the final Mel frequency cepstal coefficients are extracted.
As used in [27], the delta-MFCC's and delta-delta-MFCC's are likewise applied

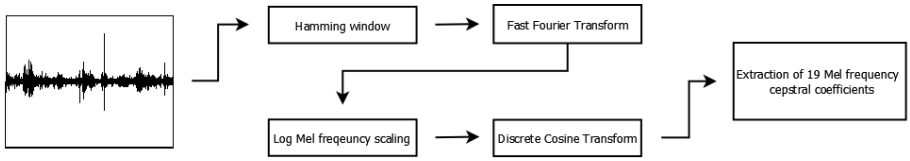


Figure 5.6: The approach to extract Mel frequency cepstral coefficients.

as features in this thesis. These features could give a more accurate representation of the speech signal because they represent the temporal changes of the MFCC's. The delta-MFCC's are the first-order derivatives of the MFCC's corresponding to the changes in MFCC value between two consecutive time windows. The delta-delta-MFCC's are the second-order derivatives of the MFCC's and they represent the changes between two consecutive time windows of the delta-MFCC, i.e. the acceleration between two consecutive windows of the MFCC's.

5.3.3 Feature Composition

In total, for each time window, 20 MFCC's are extracted. The 20 MFCC's are chosen based on the use of MFCC in the literature, [24], [47], [41]. The first MFCC (c_0) is removed since it only carries information about the mean value of the input signal and therefore have little speaker-dependent importance, [24].
19 delta-MFCC's and 19 delta-delta-MFCC's are also extracted. Furthermore the zcr and the energy for each time window are extracted and so is the cross-correlation between the two channels of the mother and the child. With respect

to the cross-correlation, the maximum value is pointed out together with the corresponding lag. The cross-correlation as a feature therefore consists of two values.
A total of 61 features are consequently constituting the feature vector. Each feature is listed in table 5.3 and supported by a short explanation. Whether

Features	Representation of each time window
MFCC's	Representation of specific qualities of the sound signal extracted from the Mel frequency spectrum
delta-MFCC's	Difference in MFFC between two consecutive windows
delta-delta-MFCC's	Difference of the difference of MFFC between two consecutive windows
Zero-Crossing Rate	The rate of the times the sound signal crosses the x-axis
Energy	The total energy
Cross-correlation	Correlation between the two signals

Table 5.3: Selected features for the speaker identification followed by a short description.

the features should be normalized or not, depend on the data set and on the classifier. Typically, in the area of speaker identification, feature normalization has been performed, [56], to even out the feature differences of several channels, which is often used with multiple speakers. In this thesis the recordings of the 15 dyads represent 15 channels and normalization is therefore likewise performed here.

As mentioned in the introduction to this section, the curse of dimensionality plays an important role in the decision of the number of features, and thereby dimensions, representing the data set. Bishop, [13], describes the concept from figure 5.7. As seen in the figure, the volume of the feature space increases more rapidly than the number of dimensions increase. In fact, the volume increases exponentially with the dimensionality of the space. The number of observations in a high-dimensional space is therefore often sparse due to the much larger volume in which the same amount of observations is represented in.
The number of observations in class 1 for each of the six respective window sizes in table 5.2 is observed to be lower than 10,000 for window sizes larger than 100 ms. If all features in table 5.3 are used in the classification task, the dimension

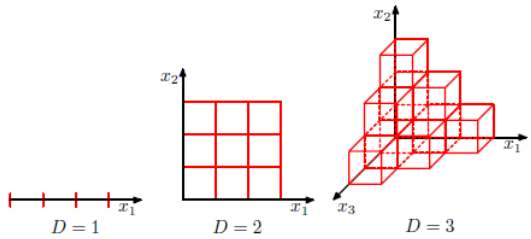


Figure 5.7: The concept of curse of dimensionality here shown for the first three dimensions. The volume of the space grows exponentially with the number of dimensions D of the space. Figure taken from [13].

of the feature space is 61. The data will therefore be sparsely represented in the 61-dimensional feature space and it will become more difficult to detect groups of similarities among the observations.

5.4 Classification

To obtain an expression of which of the four states are occurring at each time segment in the sound signal, a classifier is to be used. Different approaches have been proposed in the literature where the Gaussian Mixture Model (GMM) is the one appearing most often [57], [31], [33]. With this classifier, each state in the classification problem is to be modelled with a GMM. The belief is that the acoustic features representing vocal tract configuration, and thereby reflecting a speakers voice, can be modelled by the components in the GMM. Furthermore one of the advantages of the GMM is that it is capable of finding a complex non-linear structure of a given class. In section 5.4.1 the Gaussian Mixture model is described more thoroughly.
The speaker identification problems considered in the literature often includes analysis of a speech database, where a given number of speakers is to be identified. The sound signal considered in this thesis consists of a lot of noise as well as the fact that one of the speakers that is to be identified is a child of 4 months. Besides this, three other classes are to be identified, including the mother speaking, no one is speaking and both are speaking. Thus the four states that need to be classified in this problem are not directly comparable with the classes found in the literature.
With this in mind, it is in this thesis also investigated how four other classification methods behave in the problem of speaker identification, since the GMM might not be the best choice in this particular case.

Among the classifiers tested is the K-nearest neighbour algorithm (KNN), which simply assigns a new observation to the class determined through a majority voting of the K-nearest neighbours. Depending on the number of K, the algorithm is capable of finding complex structure in the feature space. The K-nearest neighbour algorithm is described in section 5.4.2.
The second algorithm investigated, besides the GMM, is the decision tree algorithm, also referred to as TREE in this thesis. A decision tree includes a number of conditions and maps these into equivalent classes, where the decisions are made based on the conditions, and where the classes are the consequences of the decisions. By constructing a TREE of the speaker identification problem, the identification of the speaker becomes hierarchical. In section 5.4.3 the theoretical part of decision trees is covered.
Furthermore the classification method multinomial logistic regression (MNR) is investigated. This has it's basis in the generalized linear model and is used only for classification of multiple groups. It applies the one-against-the-rest strategy through a softmax transformation of the linear functions from fixed basis functions. The advantage of MNR is that, compared to non-linear models, it is a relatively simple model to apply, [13]. The details on MNR can be read in section 5.4.4.
The last classifier applied in this study is the artificial neural network (ANN). ANN uses an adaptive approach by adjusting the parameters of the basis functions, in comparison to MNR that have fixed basis functions. Additionally, the ANN consists of multiple layers, where only the input and the output layer are actually known. The middle layers, called hidden layers, can be thought of as a black box, making this method much less transparent than the other methods used in this thesis. Section 5.4.5 explains the mathematical aspect of the model.

5.4.1 Gaussian Mixture Models

The following description of Gaussian Mixture Models is inspired by [13]. A Gaussian mixture model is a model consisting of a linear superposition of a specified number of components. Each component is Gaussian distributed. For K components the Gaussian mixture model is formulated as in equation (5.8).

$$p(\mathbf{x}|\mathbf{w}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\mu_k, \Sigma_k) \tag{5.8}$$

In equation (5.8), x is the data vector consisting of D features, $\mathcal{N}(x|\mu_k, \Sigma_k)$ is the D-variate Gaussian distribution with μ_k and Σ_k being the mean vector and the covariance matrix of the K component respectively. Finally π_k is the weight coefficients of each Gaussian mixture. For the sake of simplicity these parameters are collectively called $\mathbf{w}$, where $\mathbf{w} = \{\pi_k, \mu_k, \Sigma_k\}$ for $k = 1, ..., K$.

Each D-variate Gaussian density is given by (5.9).

$$\mathcal{N}(\mathbf{x}|\mu_k, \Sigma_k) = \frac{1}{(2\pi)^{D/2}|\Sigma_k|^{1/2}} \exp(-\frac{1}{2}(\mathbf{x} - \mu_k)' \Sigma_k^{-1}(\mathbf{x} - \mu_k)) \tag{5.9}$$

To obtain the total Gaussian mixture, the parameters π_k, μ_k and Σ_k for each component are to be estimated. As mentioned these are collectively called $\mathbf{w}$. The parameters are estimated by applying the maximum likelihood with the aim of finding the parameters that maximize the likelihood of the Gaussian mixture, given a training set. If a training set consisting of N observations are given, this data set can be represented by a $N \times D$ matrix $\mathbf{X}$ where each row in $\mathbf{X}$ corresponds to one observation with a number of D features. Using (5.8), the likelihood function is given by (5.10).

$$\ln p(\mathbf{X}|\pi, \mu, \Sigma) = \sum_{n=1}^{N} \ln \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_n|\mu_k, \Sigma_k) \tag{5.10}$$

As can be seen from equation (5.10), the summation over k takes place inside the logarithmic function. A consequence of this is that the derivative of the likelihood put to zero, will have no closed form solution and a numerical approach to the solution of the parameters is not obtainable. An iterative approach is therefore necessary and the most common method is the expectation-maximization-algorithm or just EM-algorithm.
The first step in the EM-algorithm is to choose some initial values for the parameters π_k, μ_k and Σ_k followed by the E and the M step. In the E step the initial values (in the first iteration) or current values (in the following iterations) of the parameters are used to evaluate the posterior probabilities for each of the observed components, once the observation $\mathbf{x}$ is observed. This is given by equation (5.11).

$$p(k|\mathbf{x}_n, \mathbf{w}) = \frac{\pi_k \mathcal{N}(\mathbf{x}_n|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}_n|\mu_j, \Sigma_j)} \tag{5.11}$$

The posterior probabilities are then applied to the M step where the parameters π_k, μ_k and Σ_k are updated with the formulas in equation (5.12), (5.13), (5.14) ensuring increase in the log likelihood function.

$$\mu_k = \frac{1}{N_k} \sum_{n=1}^{N} p(k|\mathbf{x}_n, \mathbf{w}) \mathbf{x}_n \tag{5.12}$$

$$\Sigma_k = \frac{1}{N_k} \sum_{n=1}^{N} p(k|\mathbf{x}_n, \mathbf{w})(x_n - \mu_k)(x_n - \mu_k)^T \tag{5.13}$$

$$\pi_k = \frac{N_k}{N} \tag{5.14}$$

Where N_k is defined as in (5.15).

$$N_k = \sum_{n=1}^{N} p(k|\mathbf{x}_n, \mathbf{w}) \tag{5.15}$$

This iterative approach continues until the change in the log likelihood function is below a given threshold or until the maximum number of iterations has been reached. As mentioned, initial values of the parameters are needed. If no prior information is given, these values are usually drawn randomly. Another opportunity is to initialize the values by selecting random seed points among the training points.

The number of components K in the Gaussian mixture is not known in advance and depends on the data to be modelled. One approach to determine K is to use cross-validation, where the negative log likelihood function is summed for each fold for a specified range of K, for example $K = [1 : 20]$. The K that gives the minimum of the negative log likelihood function, is the optimal K for that specific case.
Another approach for determining the number of components is to use the information criteria Akaike information criterion (AIC) or Bayesian information criterion (BIC). In these approaches a penalty term of the model complexity is added so that the model complexity is taken into account when analysing the log likelihood function with respect to the best K. In this way the number of components only increases if the increase in model complexity do not overcome the increase in the likelihood of the model. The method hereby prevents overfitting. The AIC and BIC are given by (5.16) and (5.17) where k is the total number of estimated parameters, N is the number of observations and L is the likelihood for the model. The model with the lowest AIC or BIC is to be chosen since this is a representation of the best trade-off between how well the model fits the data and how complex the given model is.

$$AIC = -2\ln L + 2k \tag{5.16}$$

$$BIC = -2\ln L + k\ln(N) \tag{5.17}$$

The term $-2\ln L$ in equations (5.16) and (5.17) represents how well the data is modelled because of the inclusion of the log likelihood function, whereas the terms $k\ln(N)$ and $2k$, respectively, represent the penalty term of the model complexity for the two criteria. It is clearly seen that for a given size of the data N, the BIC penalizes the model complexity the most and the tendency for this criterion is to choose a lower number of components than AIC.
At this point, the density estimation of a given data set has described using Gaussian mixtures models. For a classification problem, as speaker identification, each class is modelled with a Gaussian mixture, which supplies a density estimate for each class. It is therefore a necessity that the number of classes is known in advance. The Gaussian mixture for each class, C_i where $i = 1, 2, ..., C$ with C being the total number of classes, is then from (5.8), given by equation (5.18).

$$p(\mathbf{x}|\mathbf{w}, C_i) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\mu_k, \Sigma_k, C_i) \tag{5.18}$$

As indicated in equation (5.18) the parameters $\mathbf{w}$ for the Gaussian mixtures are given. When the distributions for all classes have been estimated, as expressed in (5.18), Bayes' theorem can be applied to provide the posterior probability of a given observation from the test set $\mathbf{x}$ belonging to class i, see equation (5.19).

$$p(C_i|\mathbf{x}) = \frac{p(\mathbf{x}|\mathbf{w}, C_i)p(C_i)}{p(\mathbf{x})} \tag{5.19}$$

In equation (5.19) $p(\mathbf{x}|C_i)$ is the probability density function as given in equation (5.18) and $p(C_i)$ is the prior probability of obtaining C_i. $p(\mathbf{x})$ in the denominator normalizes the posterior probability, since it represents the sum of the likelihood function times the prior for all the involved classes. For the four class problem in the speaker identification task in this thesis, the posterior probability of assigning an observation $\mathbf{x}$ to class C_1 is given by equation (5.20).

$$p(C_1|\mathbf{x}) = \frac{p(\mathbf{x}|C_1)p(C_1)}{p(\mathbf{x}|C_1)p(C_1) + p(\mathbf{x}|C_2)p(C_2) + p(\mathbf{x}|C_3)p(C_3) + p(\mathbf{x}|C_4)p(C_4)} \tag{5.20}$$

For each observation, the posterior probability of the observation belonging to each class is calculated and the observation is assigned to the class with the highest posterior probability.

5.4.2 K-Nearest Neighbour

The K-nearest neighbour algorithm can be used in classification problems as well as regression problems and is among the simplest of the machine learning algorithms for classification. As indicated by its name, the algorithm simply assigns an observation to the class that the K-nearest neighbours belong to, through a majority voting.
This can be expressed more formally by introducing a dataset with a total of N observations. If the number of observations in each class is N_k this means that $\sum_k N_k = N$. If a new observation x is to be classified, the distance between x and each of the N points in the data set is calculated. These distances are sorted in ascending order and the K-nearest neighbours are analysed through a majority voting of their class membership. This also means that the bigger the ratio $\frac{N_k}{N}$ in equation (5.21), the higher the probability of assigning the observation x to class k. Thus, the fraction $\frac{N_k}{N}$ is the so-called prior probability of the classes.

$$p(C_k) = \frac{N_k}{N} \tag{5.21}$$

The posterior probability of the classification is $p(C_k|x)$, which represents the probability of assigning the observation x to the class C_k. To find this, Bayes' theorem is applied as given by (5.22).

$$p(C_k|x) = \frac{p(x|C_k)p(C_k)}{p(x)} \tag{5.22}$$

The conditional probability $p(x|C_k)$ is expressed in (5.23), where K_k is the number of the K-nearest neighbour belonging to class k. The unconditional density of x is expressed in equation (5.24).

$$p(x|C_k) = \frac{K_k}{N_k} \tag{5.23}$$

$$p(x) = \frac{K}{N} \tag{5.24}$$

To obtain the posterior probability in (5.22), equation (5.21), (5.23) and (5.24) can be combined to equation (5.25).

$$p(C_k|x) = \frac{p(x|C_k)p(C_k)}{p(x)} = \frac{K_k}{K} \tag{5.25}$$

The result indicates that if the observation x is to be classified as the class having the highest posterior probability, it should be assigned to the class having the largest number of representatives among the K-nearest neighbours.
It should be noted here that no particular training phase is needed for the K-nearest neighbour algorithm, since the K-nearest neighbour represent observations in the training data where the true classes are known. But as a consequence of this, the algorithm is computationally expensive for large data sets because it needs to store the entire data set to calculate the distance from one observation to all other observations, for every observation x that is to be classified. To overcome this problem, different nearest neighbour search algorithms have been proposed which seek to reduce the calculations needed to find the nearest neighbours.
The number K is, as mentioned, the number of neighbours that are to decide which class an observation should be assigned to. If K is set to one, only the closest neighbour decides the class assigning of the observation. The logically interpretation of K would be that small values of K would result in relatively small regions of classes, and a large value of K would give fewer larger areas of classes, since K could be thought of as some kind of smoothing parameter. The optimal number of K is not known in advance and varies with the type of data. It can therefore be tested by trial and error where different values of K are chosen. The test and training error rates are found for each of these. The K with the lowest error rate should be chosen as the fixed K. The description of the K-nearest neighbour algorithm above has been inspired by [13].

5.4.3 Decision Tree

A decision tree can, as the K-nearest neighbour method, be used for both classification and regression. The purpose of the model is to predict a target value from a given number of inputs. In a classification problem, the target value to be predicted corresponds to the class label that each observation belongs to. The approach of a decision tree to obtain its class labels, is to ask a series of question until a conclusion is reached. The following detailed explanation on decision trees is inspired by [63].
The decision tree can, as indicated by its name, be visualized as a tree where the root contains all the observations of the training set. Climbing up the tree, nodes will be represented where questions are asked and branches from the nodes will identify the possible answers to the question, i.e. a split is made. If a question results in a split where all observations in one branch belongs to the same class, a leaf node is created and the node is said to be pure. This procedure continues until all observations has been assigned to a class. The principle of splitting until each node is pure, is referred to as Hunts algorithm.
Figure 5.8 shows an example of a decision tree. The Q's in the figure represents the questions that is to be asked at the nodes and the D's represents the decisions.
The challenge of decision trees is to choose the best split, which is where the

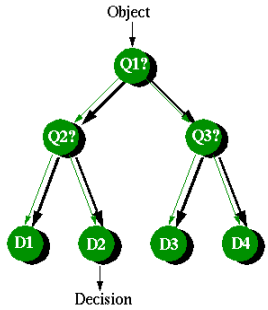


Figure 5.8: An example of a TREE where Q1 refers to question 1, Q2 to question 2 ans so on. Figure from [6].

result of splitting has the consequence of a leaf node. A measure of how good a split is, is the measure of impurity. Among the impurity measures are the Entropy and the Gini impurity, given by equation (5.26) and (5.27).

$$Entropy(t) = -\sum_{i=1}^{c} p(i|t) \log_2 p(i|t) \tag{5.26}$$

$$Gini(t) = 1 - \sum_{i=1}^{c} (p(i|t))^2 \tag{5.27}$$

In equation (5.26) and (5.27) $i = \{1, 2, ..., c\}$ where c is the number of classes and $p(i|t)$ is the fraction of objects belonging to class i. For a possible number of splits for a given node, the measure of impurity should be calculated for each of the splits. Considering a node and a proposed split the impurity should be calculated for the node, and each of the two branches. After this, the weighted average impurity should be calculated. The split with the lowest weighted average impurity should be chosen. The weighted impurity gain can be seen in (5.28).

$$\Delta_{impurity} = I(node) - \sum_{j=1}^{k} \frac{N(v_j)}{N} I(v_j) \tag{5.28}$$

In (5.28), $I(node)$ is the impurity measure of the node, $I(v_j)$ the impurity measure for the branch v_j, $N(v_j)$ the number of observation falling in v_j and N the total number of observations in the node. The created decision tree, based on the training set, is a description of the data and this tree can therefore be used as an input for decision making.
As mentioned, the Hunts algorithm continues splitting until each node is pure. This has the disadvantage of creating very complex models and sometimes this further results in an over-fitted model. To avoid over-fitting, pruning can be applied. Pruning is a technique where smaller parts of the tree can be removed if these parts only contribute minimally in the final outcome of the classifier. The complexity of the tree is thereby reduced and the predictability of the model should have increased, due to the removal of the over-fitted part. Another method to avoid over-fitting and thereby very complex trees is by controlling the number of observations in each node as a stop criteria for splitting. By doing this, no split is proposed if the number of observations in a node is lower than the given number.

5.4.4 Multinomial Regression

The multinomial logistic regression is an expansion of the binomial logistic regression model and is used for classification problems with more than two possible outcomes (classes). It has its basis in the generalized linear model and is thereby the only linear classifier out of the five tested in the problem of speaker identification. The following description of MNR is inspired by [13].
In general, for a two-class classification problem the posterior probabilities are given by Bayes theorem, as seen in the following equation, (5.29).

$$p(C_1|\mathbf{x}) = \frac{p(\mathbf{x}|C_1)p(C_1)}{p(\mathbf{x}|C_1)p(C_1) + p(\mathbf{x}|C_2)p(C_2)} = \frac{1}{1 + \exp(-a)} \tag{5.29}$$

C refers here to the class (1 or 2) and $\mathbf{x}$ is the data set. The last expression in (5.29) corresponds to the logistic sigmoid function, where a is defined in (5.30).

$$a = \ln \frac{p(\mathbf{x}|C_1)p(C_1)}{p(\mathbf{x}|C_2)p(C_2)} \tag{5.30}$$

Expansion to a multi-class problem is a generalization of Bayes theorem, as seen in (5.31).

$$p(C_k|\mathbf{x}) = \frac{p(\mathbf{x}|C_k)p(C_k)}{\sum_j p(\mathbf{x}|C_j)p(C_j)} = \frac{\exp(a_k)}{\sum_j \exp(a_j)} \tag{5.31}$$

Here C_k corresponds to class k. The quantities a_k are defined as in (5.32).

$$a_k = \ln(p(\mathbf{x}|C_k)p(C_k)) \tag{5.32}$$

The last expression in (5.31) is called the normalized exponential or the softmax function.

For logistic regression, the logistic sigmoid shown in (5.29) is applied in the two-class case to obtain the posterior probabilities, whereas the softmax function from (5.31) provides the posterior probabilities in the multi-class case. The logistic sigmoid and the softmax function are also referred to as the activation functions of the models. The activations a_k are then the input to the activation function and is in the two-class case of logistic regression given by (5.33) and in the multi-class case as (5.34).

$$a = \mathbf{w}^T \mathbf{\Phi} \tag{5.33}$$

$$a_k = \mathbf{w}_k^T \mathbf{\Phi} \tag{5.34}$$

Here $\mathbf{w}_k$ is the parameter vector, that is to be determined and $\mathbf{\Phi} = \mathbf{\Phi}(\mathbf{x})$ is the feature vector, where $\mathbf{\Phi}(\cdot)$ is a vector of fixed basis functions making a non-linear transformation of the data set $\mathbf{x}$. In both cases, (5.33) and (5.34), the activations are therefore linear functions of the feature vector, but non-linear in the original data space.

By applying the activation functions on the above-mentioned activations, the posterior probabilities of each observation belonging to each of the K classes is obtained. The multinomial logistic regression approach is known as the one-against-the-rest strategy because the methods evaluates K classifiers that each estimates the probability of a specific data point belonging to class k against the other $K - 1$ classes.

Before the posterior probabilities of the test set can be determined, the parameter vector $\mathbf{w}_k$ of the model is to be estimated, which is done by minimizing the negative log likelihood or also referred to as the error function of the classification problem, given by (5.35).

$$E(\mathbf{w}_1, ..., \mathbf{w}_K) = -\sum_{n=1}^{N} \sum_{k=1}^{K} t_{nk} \ln y_{nk} \tag{5.35}$$

In (5.35) $\mathbf{w}_k$ refers to the model parameters for model k where $k = 1, 2, ..., K$ as already mentioned is the classes in the problem. t_{nk} is the target vector expressing if the observation x_n belongs to class k by addressing it the value 1 if it belongs to the particular class and 0 if not. The term $y_{nk} = y_k(\boldsymbol{\phi}_n)$ represents the softmax transformation of the activations for observation x_n, because the

posterior probability from (5.31) is expressed as (5.36) when it is a function of the fixed basis functions $\boldsymbol{\phi}$.

$$p(C_k|\boldsymbol{\phi}) = y_k(\boldsymbol{\phi}) = \frac{\exp(a_k)}{\sum_j \exp(a_j)} \tag{5.36}$$

Minimizing the error function and thereby estimating the parameters $\mathbf{w}$ of the K models, corresponds to taking the gradient of the error function with respect to the parameters. Due to the non-linearity of the softmax function, an analytical solution to this is not obtainable, [13]. Thus, iterative optimization must be applied. Many different algorithms have been proposed to solve the problem where most of them applies a scheme as in (5.37).

$$\mathbf{w}^{\tau+1} = \mathbf{w}^{\tau} + \Delta\mathbf{w}^{\tau} \tag{5.37}$$

The new parameter $\mathbf{w}^{\tau+1}$ is calculated by a sum of the old parameter $\mathbf{w}^{\tau}$ and a weight step term $\Delta\mathbf{w}^{\tau}$, where τ represents the iteration step. To start the iterative approach in order to find the parameter $\mathbf{w}$ that minimizes the error function in (5.35), an initial value $\mathbf{w}^0$ should be chosen. This initial value is then moved in a direction determined by the term $\Delta\mathbf{w}^{\tau}$. The result of this becomes the new value of the parameter $\mathbf{w}^{\tau+1}$ and the process continues until some given number of iterations is reached or until no further reduction in the error function is obtained.
Many iterative algorithms integrates the gradient information of the error function into the method, including the one used in the MNR problem applied in this thesis. This method is referred to as the IRLS method.

5.4.5 Artificial Neural Network

An artificial neural network consists of different layers of artificial neurons, where the communication in the network takes place through these neurons. An example of a neural network with one hidden layer is given in figure 5.9. From the figure it can be seen that the neural network consists of inputs, represented by x's, hidden units represented by z's and outputs represented by y's. The input units send information to the hidden units and the information is then send to the output units. The network shown is said to have one hidden layer and a total of two layers which is due to the weights between the input and hidden layer and between the hidden layer and the output. The weights are represented by w's in the figure.

Mathematically, a neural network is build up of a fixed number of basis functions where the basis functions are adaptive in the case of their parameters. To build a neural network corresponding to the one in figure 5.9, M linear combinations of the inputs, $x = (x_1, ...x_D)^T$ are to be constructed, see equation (5.38)
.

$$a_j = \sum_{i=0}^{D} w_{ji}^{(1)} x_i, \quad x_0 \equiv 1 \tag{5.38}$$

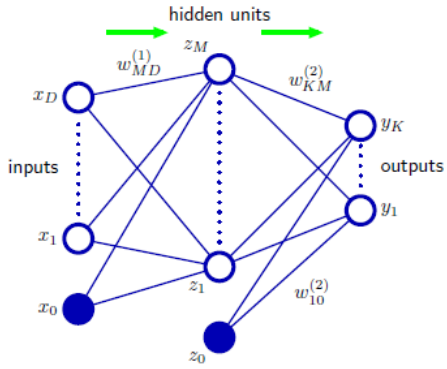


Figure 5.9: Neural network with one hidden layer, figure from [13].

In (5.38) $j = 1, ...M$ and represents the M linear combinations of the inputs whereas the i represents the i'th dimension of the input data x. The w_{j0} acts as a bias of the weights, and the w_{ji} are the weights of each of the inputs and the subscript (1) refers to the layer where the weight are present, here layer 1. To obtain the j'th output of the hidden units, a non-linear activation function is applied which is represented as $g(\cdot)$ in (5.40).

$$z_j = g(a_j) \tag{5.39}$$

$$= g\left(\sum_{i=0}^{D} w^{(1)}_{ji} x_i\right), \quad x_0 \equiv 1 \tag{5.40}$$

The non-linear activation function $g(\cdot)$ should be chosen with respect to the given data set and most often is a logistic sigmoid or a hyperbolic tangent ($tanh$). To obtain the k'th output of the neural network, another linear combination is constructed, now for the z_j's and a new activation function is applied, as in equation (5.41).

$$y_k = h\left(\sum_{j=0}^{M} w^{(2)}_{kj} z_j\right), \quad z_0 \equiv 1 \tag{5.41}$$

Equation (5.42) gathers (5.40) and (5.41) to obtain the total output of the neural network.

$$y_k = h\left(\sum_{j=0}^{M} w^{(2)}_{kj} g\left(\sum_{i=0}^{D} w^{(1)}_{ji} x_i\right)\right), \quad x_0 \equiv 1 \quad z_0 \equiv 1 \tag{5.42}$$

Here it should be noted that the output of the neural network in (5.42) can also be expressed as the non-linear activation function $h(\cdot)$ working on the activation a_k, as $y_k = h(a_k)$. Hereby a_k is given by (5.43).

$$a_k = \sum_{j=0}^{M} w^{(2)}_{kj} g\left(\sum_{i=0}^{D} w^{(1)}_{ji} x_i\right), \quad x_0 \equiv 1 \quad z_0 \equiv 1 \tag{5.43}$$

As mentioned, the basis functions in the neural network are adaptive during the training phase of the neural network. This means that the parameters, w, are optimized during the training phase. The way to estimate and update the parameters are by minimizing the error function given in (5.44) assuming a training set of input vectors $\mathbf{x}_n, n = 1, ...N$ with corresponding target vectors $\mathbf{t}_{kn}$ given. $\mathbf{t}_{kn}$ is the target vector expressing if the observation $\mathbf{x}_n$ belongs to class k by addressing it the value 1 if it belongs to the particular class and 0 otherwise. The training set is collectively called $\mathcal{D} = \{\mathbf{x}_n, \mathbf{t}_{kn}\}$.

$$E_D(\mathbf{w}, \beta) = -\sum_{n=1}^{N}\sum_{k=1}^{c} t_{kn} \ln y_k(\mathbf{x}_n, \mathbf{w}) \tag{5.44}$$

In (5.44), the coefficient β will be explained later. The task is to find a set of parameters, $\mathbf{w}$, that minimizes the error function. The point at which the error function reaches its minimum is where the gradient of the error function is equal to zero. Since no analytical solution to the problem can be found, the way to solve the problem is through an iterative approach. The general iterative optimization scheme was explained in section 5.4.4 and as mentioned there, many different algorithms have been proposed to solve the parameter optimization problem. The algorithm applied in this thesis for the ANN uses the BFGS algorithm for the optimization of the parameters (weights).
The speaker identification is a four-class problem, where the softmax function is used for classification. This is given by the last term in equation (5.31) under multinomial regression. This means that the activation function $h(\cdot)$ in (5.41) and (5.42) represents the softmax function. The softmax function as described in section 5.4.4 provides the posterior probabilities in the multi-class case.
In the ANN algorithm considered in this thesis, a modified version of the softmax function has been used. The modified version of the softmax includes the output from $c-1$ classes with c being the number of classes, and is given by (5.45).

$$p_0(C_k|\mathbf{x}) = \frac{\exp(a_k(\mathbf{x}))}{1 + \sum_{k=1}^{c-1} a_{k'}(\mathbf{x})} \tag{5.45}$$

The posterior probability for the remaining class c, is then given by (5.46).

$$p_0(C_c|\mathbf{x}) = 1 - \sum_{k=1}^{c-1} p_0(C_k|\mathbf{x}) \tag{5.46}$$

The annotation $p_0(C_c|\mathbf{x})$ with subscript 0 will be described later. The choice of this modified version of the softmax function is due to problems of evaluating the inverse Hessian matrix if the original softmax as shown in (5.31) was used. The evaluation of the inverse Hessian matrix takes place in the BFGS algorithm when updating the parameters as described later.
The ANN algorithm used in this thesis incorporates an outlier algorithm, [60]. The outlier algorithm aims at controlling random label noise by introducing an estimate of the outlier probability. What is meant by random label noise is that a target class label could erroneously have been assigned to another class than it actually belongs to. If this was the case, this data point would deteriorate

the modelling of the true class. [60] introduced the parameter *epsilon* = [0, 1] which represents the probability of assigning with random target label. A formulation of the posterior probability $p(C_l|\mathbf{x})$ therefore includes the probability of assigning with random target labels as given by (5.47).

$$p(C_l|\mathbf{x}) = p_0(C_l|\mathbf{x})(1-\epsilon) + \frac{\epsilon}{c-1}\sum_{k=1,k\neq l}^{c} p_0(C_k|\mathbf{x}) \tag{5.47}$$

In (5.47) $p_0(C_l|\mathbf{x})$ is the posterior probability of the data $\mathbf{x}$ belonging to the class C_l with zero outlier probability. $(1-\epsilon)$ represent the prior probability that $\mathbf{x}$ is not an outlier. The first term in (5.47) therefore represents the probability that $\mathbf{x}$ is not an outlier. The second term is the outlier contribution coming from classes other than C_l. (5.47) can be reduced to (5.49) by introducing the scaling of the outlier probability given by (5.48), where β is defined in the interval $\beta = [0; \frac{1}{c-1}]$.

$$\beta = \frac{\epsilon}{c-1} \tag{5.48}$$

$$p(C_l|\mathbf{x}) = p_0(C_l|\mathbf{x})(1-\beta c) + \beta \tag{5.49}$$

In order to control the weight parameters, a regularization term is added to the error function with one regularization parameter for each weight, with the approach from [38], [39] and [40]. When the regularization term is added to the error function in (5.44), the error function to be minimized is on the form (5.50).

$$\tilde{E}_D(\mathbf{w}) = E_D(\mathbf{w},\beta) + \frac{\alpha}{2}\sum_i w_i^2 \tag{5.50}$$

In (5.50) the weight decay term $\frac{\alpha}{2}\sum_i w_i^2$ acts as a regularization parameter for the weights. In this way the weight parameters prevent the model in over-fitting to the data and thereby to noise if present in data. The reason for this is to obtain a generalized model that provides the most optimal error rate.
The value of α should be chosen in a way such that it does not restrict the weights too much, but not too little either. If α for example is chosen too small, the weights may get too large which results in an over-fit to the training data and vice versa if chosen too big. MacKay [40] uses a Bayesian framework for the updating of the weights $\mathbf{w}$ where the decay parameter α and scaled outlier probability β, collectively called hyper parameters, are assumed given. The posterior probability of the weights $\mathbf{w}$ can be seen in (5.51).

$$p(\mathbf{w}|\mathcal{D},\alpha,\beta) = \frac{p(\mathcal{D}|\mathbf{w},\beta)p(\mathbf{w}|\alpha)}{p(\mathcal{D}|\alpha,\beta)} \tag{5.51}$$

In (5.51) $p(\mathcal{D}|\mathbf{w},\beta)$ represents the likelihood and is given by (5.52) where $E_D(\mathbf{w},\beta)$ can be seen in (5.50).

$$p(\mathcal{D}|\mathbf{w},\beta) = \exp[-E_D(\mathbf{w},\beta)] \tag{5.52}$$

Further in (5.51) $p(\mathbf{w}|\alpha)$ represents the prior and $p(\mathcal{D}|\alpha, \beta)$ is called the evidence. The prior is a zero mean Gaussian prior and is given by (5.53).

$$p(\mathbf{w}|\alpha) = \frac{\exp(\frac{-\alpha}{2}\sum_i w_i^2)}{\int \exp(\frac{-\alpha}{2}\sum_i w_i^2) d\mathbf{w}} \tag{5.53}$$

The optimization of the parameters is then carried out by minimizing the error function given in (5.50). Here, as mentioned, the BFGS algorithm is used, that applies a Gauss-Newton scheme for the approximation of the Hessian matrix. The optimization of the parameters in order to minimize the error function can be seen in (5.54).

$$\mathbf{w}^{\tau+1} = \mathbf{w}^{\tau} + \eta \mathbf{A}^{-1}(\mathbf{w}^{\tau})\mathbf{g}(\mathbf{w}^{\tau}) \tag{5.54}$$

In (5.54) η is the step size, determined by a line search algorithm. $\mathbf{A}$ is the Gauss-Newton approximation to the Hessian matrix and $\mathbf{g}(\mathbf{w})$ is the gradient of the error function (5.50) taken with respect to the weights. When the weights have converged, the parameters α and β should be updated. This process should be continued until α and β have converged as well. For more details on the updating of the hyper parameters see [60].

5.5 Model Evaluation

When applying a classifier, an essential thing is the evaluation of the performance of that given classifier. The question that arises is; what is a good performance? The error rate of the classifier reveals how many wrong decisions the classifier makes, when comparing the estimated class vector to the true class vector. If more classifiers are applied, the error rate of these can be compared if the training and test set remains the same for the respective models. Naturally, the performance of the classifier should be better than the outcome of assigning the observations in the test set randomly, called the by-chance error rate. The by-chance error rate depends on number of classes as well as the number of observations in each class. It is described more thoroughly in section 5.5.1.
To ensure a generalizable model different approaches can be used, depending, amongst other, on the data size. This is described in section 5.5.2.
As explained in section 5.3, different features have been extracted that are to be used as input to the classifiers presented in section 5.4. The numerous features selected for a given model may not be the most optimal and in the evaluation of the model performance, testing different feature combinations are therefore an essential part. This is described in section 5.5.3.3 together with the concept of combining the outcome from different models.
Two microphones are used in the recording session, meaning that two signals are available in the speaker recognition task. Combining the information from these could possibly boost the performance of the classifier. The aspect of combining

the channels is also described in section 5.5.3.3 and rounds off this section on model evaluation.

5.5.1 Data Imbalance

The by-chance error rate is, as mentioned in the introduction to this section, the error rate obtained by assigning each observation randomly if the class proportions are equal (balanced data set). If, for instance, the problem considered involved two classes and the class proportions were equal, the by-chance error rate would be 50 %.
When the class proportions are unequal, corresponding to an imbalanced data set, the by-chance error rate is more tricky to estimate. It can be estimated by assigning the observations according to the prior probabilities in two ways. One is to assign all observations to the class with the highest prior probability, as for example with two classes with a prior probability for class A of 40 % and of 60 % for class B. This gives a by-chance error rate of 40 % if all observations are assigned to class B.
The other method is more practical. For two classes with prior probability as above, 40 % of all the observations is ascribed to class A and 60 % of all observations to class B. This would result in an error rate of $1-0.60^2-0.40^2 = 48\%$. From this example it is clearly seen that the class proportions as well as the number of classes in the problem has an influence on the error rate. The way in which the classifier ascribes the observations depend on the specific classifier and data set. The by-chance comparison error rate for imbalanced data sets should therefore be accompanied by an analysis of the confusion matrix, which is a matrix where the rows represents the actual classes and the columns the predicted classes. In the case of the speaker identification problem this would result in a 4×4 confusion matrix. The matrix could, for example, reveal if the model is capable of fitting all the classes to some degree or if the model fits two classes perfectly but not the remaining two. This is discussed in detail in section 9.1.2.
In the case investigated during this thesis, the number of classes is four. If it is assumed that each class has an equal proportion of observations, meaning that the four prior probabilities are equal, the by-chance error rate is be 75 %. If the class proportions on the other hand are unequal, the by-chance error rate would be as seen in equation (5.55), shown for four classes where N_c is the number of observations in class c and N is the total number of observations.

$$Errorrate = 1 - \left(\left(\frac{N_1}{N}\right)^2 + \left(\frac{N_2}{N}\right)^2 + \left(\frac{N_3}{N}\right)^2 + \left(\frac{N_4}{N}\right)^2 \right) \tag{5.55}$$

Equation (5.55) is general in that more classes easily can be included.
With the number of observations given in table 5.2 for the window size of 150 ms, this results in an error rate of 67%, see (5.56). This error rate is obtained if the classifier simply assigns each new observation to one of the classes according to their prior probability only. The error rate obtained from the applied classifiers

should therefore be lower than 67% to be better than random.

$$Errorrate = 1 - \left(\left(\frac{5849}{59376} \right)^2 + \left(\frac{18788}{59376} \right)^2 + \left(\frac{7735}{59376} \right)^2 + \left(\frac{27004}{59376} \right)^2 \right) = 67\% \quad (5.56)$$

An important note to the above discussion on imbalanced data sets, is that the class imbalance proportion should be generalizable to the entire data set. That is, the amount of segments where, for instance, the mother is speaking, should be more or less the same across dyads. If this is not the case, the class priors calculated based on the training set only, will differ from the class priors of the test set and the classification is in risk of being degraded.
In table 5.4, the prior probabilities for each class in the training as well as the test set are shown. From the table it is seen that the class priors of the training and the test set are not completely identical. But what is also seen is that the classes *mother* and *no one* in both cases has the largest prior as well as the classes *child* and *both* has the smallest priors. Due to the fact that this is in evidence, it is assumed that the priors of the training set represents the general prior distribution over classes.

Class	Prior-train	Prior-test
Child	10 %	3 %
Mother	32 %	44 %
Both	13 %	2 %
No one	45 %	51 %

Table 5.4: The prior probabilities for the training set consisting of 14 dyads and the test set consisting of 1 dyad.

5.5.2 Generalizing the Model

To obtain a generalizable error rate, the k fold cross-validation method could be applied. In k-fold cross-validation the data set is split into k pieces where the $k-1$ pieces are used as a training set and the last piece is used as a test set. An illustration of this can be seen in figure 5.10 for $k = 4$.

The partitioning is made for k different combinations where each observation is only used in the test set once, leaving the observation as training point $k-1$ times. The advantage of this method is that the error rate is evaluated over the entire data set and the error rate thereby gives a generalized illustration of the model performance. The drawback of the cross-validation method is that the computations necessary increase since k number of models are to be fitted, [13].

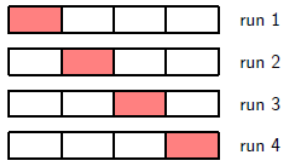


Figure 5.10: The concepts of k-fold cross-validation shown here for $k = 4$. Figure from [13].

In this thesis it would mean that a 15-fold cross validation should be applied, since the data set consists of 15 dyads. Each dyad should be used as test set once and as training set 14 times. Due to the size of the entire data set, see table 5.2, it is, based on [13], chosen not to perform this 15-fold cross validation. Instead a generalized model is generated based on 14 dyads which is tested using one dyad as a test set only, thus the hold-out method is applied. With this approach, the purpose of the model would be fulfilled since the goal is to make a model that Babylab can use on future data sets.

5.5.3 Boosting Performance

When constructing a classifier, different methods can be used in order to boost the performance of that given classifier. Three different approaches have been applied that are described in the following.

5.5.3.1 Combining Models

One way to boost the performance of a classifier is to combine the outcome of a given number of classifiers, to take advantage of the information from different sources. There are two ways in which this can be carried out. The first is the combinations of the investigated classification methods, here GMM, KNN, decision tree, MNR and ANN. The idea is that if the errors of the classifiers are independent of each other, then one classifier would make one type of errors whereas another classifier makes another type of errors. If the outcome of these classifiers are combined the performance would be boosted. The second approach to combine models, is by fixing the classification method and differentiating the classifiers through their features. This kind of model combination has actually been investigated in some studies on the speaker identification problem

and these results are promising, [42].
The results of this test can be seen in section 9.1.3.

5.5.3.2 Window Predictability

One of the advantages of testing several window sizes, as explained in section 5.2, is that sub-window predictability can be investigated. The thought is that an improvement in error rate is possible by using the output of smaller windows in the prediction of larger windows. This is done by performing a majority voting of the outcome of the smaller windows. For example, the majority of 3 consecutive 50 ms windows can be used to decide the outcome of one 150 ms window. Figure 5.11 illustrates this.
The figure shows a situation where the outcome of the majority voting results

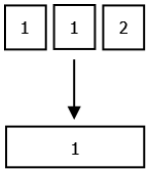


Figure 5.11: The outcome of the classifier at 50 ms shown at the top of the figure can be used in the prediction of the 150 ms outcome. In this example the outcome of the majority vote is class 1, as shown in the bottom of the figure.

in the class label *1* of the 150 ms window. By applying this method for all the observations for the sub-window size, the error rate calculated from these can be compared to the error rate obtained from the classifier with the larger window size.
This procedure is tested in section 9.1.4.

5.5.3.3 Combining Channels

The speaker identification problem investigated in this thesis, includes two microphones. The classifier performance has been investigated for each channel separately. To take advantage of the information from both recordings, a combination of the two channels, with the purpose of boosting classifier performance, has been carried out on the basis of the results in [37].
As mentioned in section 5.1, the speech signal is divided into quasi-stationary smaller segments. The outcome of the classifiers therefore represents the state

or class of each segment. In [37] the outcome of the classifiers for each microphone (there are three) are gathered into one single classification label through a majority voting. Since only two microphones are available in the case of mother/child speaker identification, it is difficult to make this majority voting. To overcome this, the voting is performed by exploiting the appertaining smaller window sizes, see section 5.2.
For instance, when the two channels with a window size of 150 ms are classified, the estimated class labels for each channel are compared. If the class labels for the 150 ms time window are equal, this label represents the label of the combined channels for this time window. That is, no majority voting takes place. If, on the other hand, the outcome of the two channels are unequal, as in figure 5.12, a majority voting of the appertaining six 50 ms windows is performed. In figure 5.12 the large rectangles represents the 150 ms windows whereas the smaller rectangles represents the 50 ms windows. As seen in the figure the class label of the 150 ms window is ascribed class 1 since the majority of the six 50 ms windows are ascribed by the classifiers to class 1.

In the case considered in this thesis it is expected that the signal from the

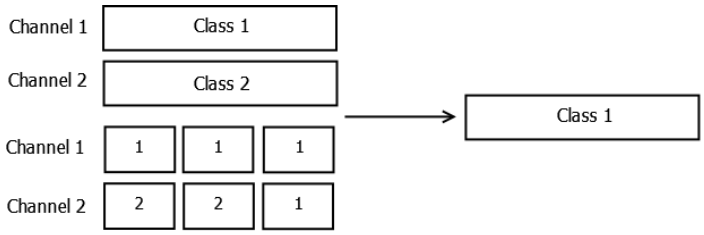


Figure 5.12: The outcome of the classifiers for channel 1 and 2 respectively represented both for the windows of 150 ms (large rectangles) and 50 ms (small rectangles). If the two outcomes of the 150 ms segments are unequal a majority voting of the smaller segments takes place.

mothers channel would result in the best performance due to the fact that the signal from the child's microphone in general is more noisy. The child makes a lot of sudden movements and this results in scratching and thereby noise in the microphone. On the other hand, the child's voice is of course lower in the microphone of the mother due to the distance between them. The outcome of the classifier from the mother's microphone would, as a consequence of this, possibly have difficulty in classifying when the child is speaking. The combination of the outcome from the classifiers from each channel therefore might contribute in a boosting of the performance in this case as well as in [37].
The test results obtained from this method are given in section 9.1.5.

Chapter 6

Emotion Recognition

As well as identifying the speaker, which was the focus of chapter 5, it is of great interest to the psychologists at Babylab to determine the child's emotional state. This is done at Babylab by manually annotating the child's spoken utterances as being either protests or not, see table 6.1.
Due to the numerous possibilities in many human-machine interactions, such as

Class	Class definition
1	Protest
2	No protest

Table 6.1: The class definitions for emotion recognition.

applications where the speaker's emotional state determines the response given by the system, [49], as well as for diagnostic purposes, [20], emotion recognition is a popular subject within the area of pattern recognition and machine learning. Many studies have been carried out with the aim of discovering the composition of classifier and features that provides the lowest error rate - and thereby the best emotion recognizer - for the given emotion database. These databases include both acted and natural emotional utterances as well as utterances spoken in different languages (see [17] for a thorough description of several emotion databases).

The emotions to be classified are usually the six archetypal emotions of joy, anger, sadness, fear, disgust and surprise, [48], [58], [16]. For these emotions, especially pitch, energy and speaking rate are used as features in the classifier. Furthermore, spectral features such as MFCC and LPCC are included in many studies as well, [32], [59].
Among the articles that focus their work on real-life emotions are [36], [59] and [62]. None of these studies base their emotion recognition on the same classifier method, and this is also the general image in the emotion recognition area: the best classifier is not a specific one, but is dependent on the data set to be analysed. The classifiers applied in the aforementioned studies are linear discriminant classification, K-nearest neighbours, artificial neural networks and hidden markov models (HMM), but also the classifiers support vector machine and decision tree have been applied in emotion recognition tasks.
In this thesis it has been chosen to work with the HMM classifier. HMM is used in many speech applications and likewise in many emotion classifications, [48], [14], [32], [58], [17].
Details on the preprocessing of the sound signal before classification is described in section 6.1. The choice of features will be explained in the subsequent section 6.2, while details on the chosen classifier will be given in section 6.3. In the last section 6.4 the model optimization will be discussed.

6.1 Preprocessing

At Babylab the emotional states of the child have been annotated manually as either protest or not protest based on the sound signal. As was explained in chapter 5, the entire speech signal has been annotated into the four classes, child speaking, mother speaking, both speaking or no one speaking. By extracting the intervals where the child is speaking, the new signal only consists of the two classes *protest* and *no protest*. This information can then be used as the ground truth.
When applying HMM, the temporal changes in the signal is accounted for by the model, which is why time segments of a certain length must be extracted. Since not one specific approach is used for all HMM emotion classification problems, it is in this thesis assumed that one emotional utterance is given by the child within 100 ms. This is assumed to be valid because the speaker is a 4 months old infant and is therefore not able to say any words or sentences. Only short sounds constitute the emotions that both the mother and the manual coder are able to assess. Likewise it is assumed that the HMM can capture the variations in the utterances.
Since the precision of the ground truth annotations is 10 ms, this window size is

chosen to be the smaller segment from which the feature vectors are extracted. I.e. one emotional utterance consists of 10 smaller segments from which the temporal changes are modelled by the HMM. The details on the HMM is given in section 6.3. Figures 6.1(a) and 6.1(b) illustrate spectrograms of 100 ms duration of the sound signal, where the child is in protest and not in protest, respectively. It should be noted, as for the spectrograms of the speaker identification task, figure 5.4, that only the frequencies up to 7000 Hz are shown because it is assumed that most of the frequency content lies in this area.

From the figures it is clear that there is a difference in spectral content when the

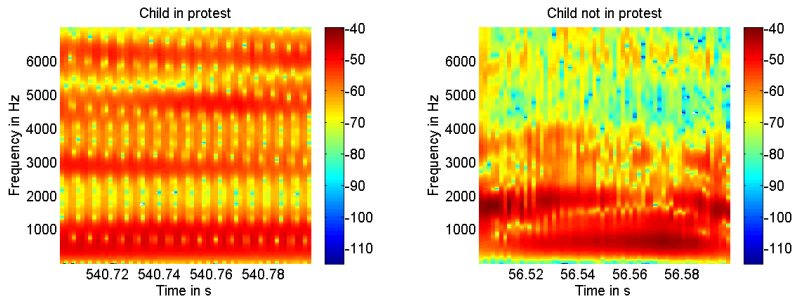


Figure 6.1: Spectrogram of 100 ms of the sound signal during an utterance of the child annotated by Babylab as (a) being protest and (b) not protest.

child is in protest and is not in protest, respectively. Based on visual inspection on several spectrograms of the child's emotional state, this seems to be the general picture. From the spectral features alone, the possibility of separating the two emotional states therefore appear achievable.

The emotion classification is based on 11 dyads, for all of which the ground truth is available. 10 dyads are used as training set and one as test set. The amount of sequences of 100 ms and of feature vectors of 10 ms are shown in table 6.2 for the training set.

Class	Number of sequences	Number of feature vectors
Protest	5212	52120
No protest	2355	23550

Table 6.2: The amount of data available in the training set.

6.2 Feature Extraction

As mentioned in the introduction to this chapter, the emotional states usually implemented as classes in an emotion recognition task are joy, anger, sadness, fear, disgust and surprise. Energy, pitch, zcr and MFCC have all been applied as features in these classification problems. For both speech and speaker recognition purposes, the delta and delta-delta cepstral coefficients are used as well. One article was found that investigated the classification of negative versus not-negative utterances in adults. Here energy and pitch was used as features, [36]. Since no articles have been found that focus their emotion recognition on the utterances of infants, the feature choices here are based on the available literature. Energy, zcr, MFCC and delta-MFCC are therefore all included as features in the emotion recognition performed in this thesis, but since they were all explained in detail in the chapter on speaker recognition, 5, section 5.3, these features will not be discussed here.

The pitch feature represents a quality of the signal that is related to fundamental frequency, see 5.1 for an explanation on fundamental frequency. Pitch is not a physical quantity of a sound signal, but is instead related to the human sensation of perceiving sounds, [50]. Often the pitch of a sound is confused with the fundamental frequency of a sound, and therefore the pitch estimation of a sound is actually estimation of the fundamental frequency.
As was discussed briefly in section 5.3 regarding zcr, a difference exists in the speech produced by voiced and unvoiced speech. Pitch and fundamental frequency is related only to voiced speech, since the vibration of the vocal cords defines the frequency. In many of the articles already mentioned regarding emotion recognition, the unvoiced regions of the speech are removed before classification is performed. Since the purpose of the emotion recognition is to be able to identify all of the child's utterances as either being protests or not, the pitch feature will therefore not be included in the emotion classification in this thesis. The combination of features that are included in this classification problem are therefore the above mentioned which are gathered in table 6.3.
In the speaker identification chapter, the features were normalized based on the multiple dyad set-up. In emotion recognition in general, not one approach is used regarding features, in spite of the often used multiple talker-scenario. Some studies extract statistical measures, such as mean and standard deviation of each feature, [36], whereas others apply not-normalized features that represent an instantaneous short time segment, [48].
In the emotion recognition set-up used in this thesis it is chosen to apply the raw instantaneous features and thereby not normalize.

Features	Representation of each time window
MFCC's	Representation of specific qualities of the sound signal extracted from the Mel frequency spectrum
delta-MFCC's	Difference in MFFC between two consecutive windows
Zero-Crossing Rate	The rate of the times the sound signal crosses the x-axis
Energy	The total energy

Table 6.3: Selected features for the emotion classification followed by a short description.

6.3 Classification

The HMM classifier takes, as mentioned in section 6.1, the temporal changes of a speech segment into consideration. This is useful in dealing with speech signals where the acoustics properties vary over time and are not always independent of each other.
The idea of the HMM is based on Markov chains, in which an observation is dependent on the previous x observations. The usual assumption of independent and identically distributed observations, that is fulfilled if all observations are drawn from the same probability distribution and are independent from each other, is therefore not given with these models.
The HMM introduces discrete latent variables, also referred to as states, where transition probabilities describe the moving between states. Furthermore, emissions are introduced, which describes the probability of causing each possible observation from each state. An example of a HMM is shown in figure 6.2, from [7]. As explained in the caption of figure 6.2, three states and four observations are included in this model. The a's represent the transition probabilities, whereas the b's represent the emission probabilities. For instance, the probability of moving from state 1 to 2 is equal to a_{12}, moving from state 2 to 1 is equal to a_{21}, etc. If a transition probability is not shown in the figure, it is equal to zero. This is the case for a_{11}, a_{22}, a_{33}, a_{13}, a_{31} and a_{32} which means that these transitions do not occur.
The transition matrix will in this case appear as (6.1).

$$\mathbf{A} = \begin{bmatrix} 0 & a_{12} & 0 \\ a_{21} & 0 & a_{23} \\ 0 & 0 & 0 \end{bmatrix} \tag{6.1}$$

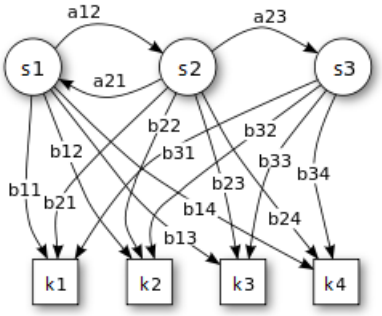


Figure 6.2: Example of an HMM. Here three states and four observations are included in the model. The a's represent the transition probabilities, whereas the b's represent the emission probabilities. For instance is the probability of moving from state 1 to 2 equal to a_{12}, etc. If a transition probability is not shown in the figure, it is equal to zero. This is the case for a_{13}, a_{31} and a_{32} which means that the transitions from state 1 to state 3, state 3 to state 1 and state 3 to state 2 never occurs. Also the transitions from a state to the same state also never occurs in this example. These transition probabilities, a_{11}, a_{22} and a_{33}, are therefore equal to zero. Figure modified from [7].

The corresponding emission matrix will for the same example be written as (6.2).

$$\mathbf{B} = \begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{14} \\ b_{21} & b_{22} & b_{23} & b_{24} \\ b_{31} & b_{32} & b_{33} & b_{34} \end{bmatrix} \tag{6.2}$$

From the above example it is clear that the transition and emission probabilities will vary according to the data set to be modelled.
In addition to the transition and emission probabilities, the HMM introduces the initial state probabilities that determines in which state the HMM is initiated. If S is the total number of states and K is the size of the codebook, the transition probability matrix, described by $\mathbf{A}$, has a size of $S \times S$. The emission probability matrix, $\mathbf{B}$, has a size of $S \times K$ and the initial state probability vector, π, a length of S. The parameters of the HMM model are represented by λ as in (6.3).

$$\lambda = (\mathbf{A}, \mathbf{B}, \pi) \tag{6.3}$$

A restriction of the transition and emission matrices is that each row must sum to 1. This is due to the finite and fixed number of states and observations indicating that the possible moves are confined between states and likewise from

state to observation.

The approach used in this thesis is isolated emotion recognition through discrete HMM. Here two models are estimated; one for each of the two emotions, based on their separate training set observations. For every observation in the test set the likelihood of the sequence given each of the two estimated models is calculated. The sequence is ascribed to the model, and thereby emotion, with the highest likelihood.
As explained in section 6.1, ten feature vectors, i.e. 10 observations of 10 ms duration each, constitute one emotional sequence. Each feature vector in the total training set, corresponding to 10 times the total number of sequences representing both emotions, are used in the estimation of one common codebook. This is obtained through the use of the K-means algorithm and has the purpose of representing the observations through a finite number of clusters in order to reduce the complexity of the model.
The K-means algorithm performs a partitioning of the feature vectors into K clusters. For this, the distortion measure J is a useful guideline of how well the clusters represent the observations. This is shown in equation 6.4, from [13].

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} j_{nk} ||\mathbf{o}_n - \boldsymbol{\mu}_k||^2 \tag{6.4}$$

Here, N is the total number of observations, K the total number of clusters, $\mathbf{o}_n$ the n'th observation and $\boldsymbol{\mu}_k$ the cluster center of the k'th cluster. j_{nk} is 1 for the value of k with which the squared distance measure $||\mathbf{o}_n - \boldsymbol{\mu}_k||^2$ is minimized. For all other k's j_{nk} is 0. This can also be expressed mathematically, as in 6.5. From [13].

$$j_{nk} = \begin{cases} 1 & \text{if k} = \arg\min_j ||\mathbf{o}_n - \boldsymbol{\mu}_j||^2 \\ 0 & \text{otherwise} \end{cases} \tag{6.5}$$

The minimized J corresponds to the cluster composition that represents the data set in the best possible way. To obtain this expression, the EM-algorithm is applied. As was explained in section 5.4.1 related to the GMM, the approach of this algorithm is to iteratively obtain the values of j_{nk} and $\boldsymbol{\mu}_k$ that minimizes J. The cluster centres, $\boldsymbol{\mu}_k$, are initiated randomly and fixed, while the j_{nk} is estimated through minimization of J. Next, the estimated j_{nk} is fixed while J is now minimized with respect to $\boldsymbol{\mu}_k$. This is repeated until J has converged or until the maximum number of iterations is reached.

With this procedure, the global HMM codebook is obtained, consisting of K cluster centres. The next step is to quantize all sequences in the training set for the estimation of the two emotional models. For each sequence, each observation is quantized by assigning it to the cluster to which the distance between the observation itself and the cluster center is the smallest. The output is therefore a series of sequences, in which each observation takes on a value from 1 to K

depending on the cluster assignment.
From the quantized training sequences, two HMM models are estimated; one based on the training set of the emotion *protest* and one on the emotion *no protest*. The forward-backward algorithm (f-b algorithm) is used for this. Here, the goal is, through a special case of the iterative EM-algorithm, to maximize the likelihood of the quantized training sequences O given the model λ, that is $P(O|\lambda)$, in order to obtain a reliable model estimate that can predict the emotions of the test set. The optimal models are found when the likelihood has converged or the maximum number of iterations has been reached.

The f-b algorithm consists of two passes; the forward pass and the backward pass. In general, the forward pass, denoted $\alpha(o_{1:t}, i)$, is used to calculate the joint probability of the model having generated the observation sequence, $O = o_1, o_2, ..., o_t$ and having arrived at state s_i at time t, where $i = 1, 2, ..., S$ indicates the state and where $t = 1, 2, ..., T$ with T being the number of observations in each sequence.. In mathematical terms this is written as (6.6).

$$\alpha(o_{1:t}, i) = P(o_1, o_2, ..., o_t, q_t = s_i|\lambda) \tag{6.6}$$

Here, the q_t refers to the state at time t of the state sequence $Q = q_1, q_2, ..., q_t$. The backward pass, referred to as $\beta(o_{t+1:T}, i)$, is applied for calculating the probability of having the observation sequence $O = o_{t+1}, o_{t+2}, ..., o_T$ given the state s_i at time t and the model λ. This is shown in (6.7).

$$\beta(o_{t+1:T}|i) = P(o_{t+1}, o_{t+2}, ..., o_T|q_t = s_i, \lambda) \tag{6.7}$$

The approach for obtaining α_T and β_T, i.e. the forward and backward passes for the entire observation sequence O, is based on recursion, meaning that the result from α_1 is used to estimate α_2 and so on, and likewise for β.
To estimate the α and β parameters for each time instant t of the entire sequence, (6.8) and (6.9), respectively, are used. From [51].

$$\alpha(o_t, j) = \sum_{i=1}^{S} a_{ij}\alpha(o_{t-1}, i)b(o_t|j), \qquad \alpha(o_1, i) = \pi_i b(o_1|i) \tag{6.8}$$

$$\beta(o_{t+1}|j) = \sum_{i=1}^{S} a_{ji}\beta(o_{t+2}|i)b(o_{t+2}|i), \qquad \beta(o_{T+1}|j) = 1 \tag{6.9}$$

The α and β parameters are scaled to avoid numerical problems. This is done through the use of (6.10) by multiplying it with (6.8) and (6.9), respectively, for each time instant of the sequence O. Hereby $\hat{\alpha}$ and $\hat{\beta}$ are obtained. From [23].

$$c_t = \left(\sum_{i=1}^{S} \alpha(o_t, i) \right)^{-1} \tag{6.10}$$

As observed in (6.8) and (6.9), the parameters of the model are part of the f-b pass calculations. To maximize the likelihood $P(O|\lambda)$, these parameters should be adjusted. This is done iteratively, through the use of the following updating

rules for the elements of $\mathbf{A}$, (6.11), $\mathbf{B}$, (6.12), and π, (6.13). Modified from [23].

$$\bar{a}_{ij} = \frac{\sum_{l=1}^{L}\sum_{t=1}^{T_l-1} a_{ij}\hat{\alpha}^{(l)}(o_t,j)b(o_{t+1}|i)\hat{\beta}^{(l)}(o_{t+2}|i)}{\sum_{l=1}^{L}\sum_{t=1}^{T_l-1}\hat{\alpha}^{(l)}(o_t,j)\hat{\beta}^{(l)}(o_{t+1}|j)/c_t^{(l)}} \tag{6.11}$$

$$\bar{b}_{jk} = \frac{\sum_{l=1}^{L}\sum_{o_t=k,t=1}^{T_l}\hat{\alpha}^{(l)}(o_t,j)\hat{\beta}^{(l)}(o_{t+1}^{T_l}|j)/c_t^{(l)}}{\sum_{l=1}^{L}\frac{1}{P(o^{(l)}|\lambda)}\sum_{t=1}^{T_l}\hat{\alpha}^{(l)}(o_t,j)\hat{\beta}^{(l)}(o_{t+1}^{T_l}|j)/c_t^{(l)}} \tag{6.12}$$

$$\bar{\pi}_i(1) = P(s_1 = i) = \frac{1}{L}\sum_{l=1}^{L}\frac{\hat{\alpha}^{(l)}(o_1,i)\hat{\beta}^{(l)}(o_2|i)}{c_1^{(l)}} \tag{6.13}$$

The updates in (6.11), (6.12) and (6.13) are given for the case of isolated HMM for multiple training sequences, which is denoted by $l = 1, 2, ..., L$.

For each iteration, the likelihood is calculated as the sum of (6.6) with respect to i. This is shown in (6.14).

$$P(O|\lambda) = \sum_{i=1}^{S}\alpha(o_{1:T},i) = \left(\prod_{\tau=1}^{T}c_\tau\right)^{-1} \tag{6.14}$$

The likelihood in (6.14) is converted into the log-likelihood in (6.15), to avoid numerical problems.

$$\log P(O|\lambda) = -\sum_{\tau=1}^{T}\log c_\tau \tag{6.15}$$

The log-likelihood is maximized through the iterative procedure. The optimal solution, i.e. the optimal model parameters λ, is obtained when no further increase in likelihood is found or when the maximum number of iterations is reached.

With the above described approach, the two HMMs are obtained. The task is now to use the estimated models to classify each emotional sequence in the test set. Each sequence is to be quantized through the codebook and then classified by estimating the likelihood of the test sequence given the two models. The test sequence is assigned to the emotional model that provides the largest of the two likelihoods.

6.4 Model Evaluation

For the emotion recognition task, different aspects should be considered with respect to the classification.
Section 6.4.1 deals with the aspect of an imbalanced data set, as was also the focus of section 5.5.1. As with the speaker identification problem, a generalized model of the child's emotional states is desirable, and a discussion of this therefore constitute the topic of section 6.4.2.

6.4.1 Data Imbalance

As was the case with the speaker identification problem in chapter 5, the two classes in the emotion recognition task, which is the focus of this chapter, are imbalanced. That is, there are more than twice the amount of the emotion *protest* compared to the emotion *no protest* in the training set. In the speaker identification problem, recall section 5.5, the ratios between the four respective classes were comparable across data sets, which is why it made sense to perform classification on the entire data set. In this problem, the ratio between the amount of the two emotions in the training set does not correspond to the same ratio of the test set. Here, the amount of the emotion *protest* outnumbers the emotion *no protest* by a little more than a factor 4. The prior probabilities of the training and test set for the two emotions, respectively, are shown in table 6.4.
Another issue of the emotion recognition task is that it is not clear whether the

Class	Priors of training set	Priors of test set
Protest	69 %	80 %
No protest	31 %	20 %

Table 6.4: The prior probabilities of the two emotions for the training and test set, respectively.

child on the day of the recording session is in a good or a bad mood. Therefore it can never be known if the number of *protest*s outnumbers the *no protest*-class, if it is the other way around or if there are an equal amount of both types of utterances. Hence the priors of the new observation (or dyad) is not known in advance and the use of the prior of the training can therefore possibly deteriorate the results.
Because of this insecurity of the imbalance between training and test set, it is chosen to balance the data set in the emotion recognition task. The balancing of the data set is done after the two HMM models are estimated. This means that all available sequences was used to estimate the HMM and the data balancing therefore was performed only on the test set.
Balancing the data set theorizes the emotion recognition problem, in that the possibility of applying the model on an new dyad data set is much limited. The thought here is therefore to illustrate the potential of applying an emotion classifier which then, in the future, has the potential of being improved through different methods for data imbalance, [15], [26].

6.4.2 Generalizing the Model

The approach applied here for model generalization corresponds to that of speaker identification, section 5.5.2, i.e. applying all but one dyad recordings as training set and the last as test set. For the emotion recognition task here, the ground truth, that is the manual labelling executed by Babylab, is available for 11 dyads. Therefore 10 are used as training set and 1 as test set. This approach is assumed valid, confer the discussion in section 5.5.2, considering the large data set as shown in table 6.2.
Furthermore, since the modelling of the HMM from the training set is initiated randomly, i.e. the transition matrix, emission matrix and the initial state probability vector are randomly chosen, every test regarding the HMM is run 15 times. The error rates shown in the result section, 9.2, are thereby the mean of these 15 runs.

Chapter 7

Motion Capture Annotations

The psychologists at Babylab focus much of their work on the physical relationship between the mother and child. For this, motion capture data is highly relevant. The manual annotations from Babylab regarding the motion capture modality that are to be automated in this thesis are listed below, summed up from chapter 3.

- Child's head position
- Distance between faces
- Child's physical energy level

The infant head position is currently being extracted by a group at Babylab by manually annotating according to four categories from the video recordings. The four categories are listed in table 7.1 below and refer to the angular interval between the child's and the mother's head positions where the angles chosen have been inspired by [35] and [10].
The angle of the child's head position is by Babylab determined with respect to a reference point in the room, that is not the mother. The angle annotations

Category	Angular Interval
En face	[0° : 30°]
Minor avert	]30° : 60°]
Major avert	]60° : 90°]
Arch	]90° : 180°]

Table 7.1: The category definition based on the four angular intervals: *en face*, *minor avert*, *major avert* and *arch*.

is applied in the analysis of the relationship between the mother and child by Babylab.
Likewise, the distance between the heads of the mother and child is annotated by Babylab. This is calculated in Excel from the marker coordinates MheadB and CheadB, see figure 3.1. By combining the child's head orientation and the distance between the mother and child, the concept of *chase and dodge*, as formulated in, amongst others, [11], is investigated by Babylab. The idea is that if the child feels intruded by the mother if she leans forward, too much or too fast, the reaction is that the baby moves its head back and away.
The last annotation that is to be automated in this thesis is the child's physical energy level. This is at Babylab interpreted as the covered distance of the right wrist marker. Currently this is calculated by Babylab in excel from the marker coordinates of motion capture.
The child's physical energy level can be used in the analyses of the existence of specific patterns between the mother's vocalizations and the child's movements as well as in the child's coordination of movement and vocal actions.
It should be noted that, as was also mentioned in section 3.2, in almost every mocap file, a number of not-identified mocap markers exists. This is a source of error, in that these must be estimated to make use of the entire 10 minutes. In some sessions only a few markers have not been identified in a few frames, whereas in others a countless number is missing. Examples of this is shown in the results section on mocap features, 9.3.
The approach to extract the three mocap annotations is explained in the following sections.

7.1 Child's Head Position

The infant head position with respect to the mother's head was investigated in [34]. This included both the child moving its head up and down, corresponding

to a movement in the XZ-direction of the mocap coordinate system, as well as it moving its head to the sides, corresponding to a movement in the XY-direction. The coordinate system is illustrated in figure 7.1.

[34] introduced two new points, namely CheadM and MheadM, which repre-

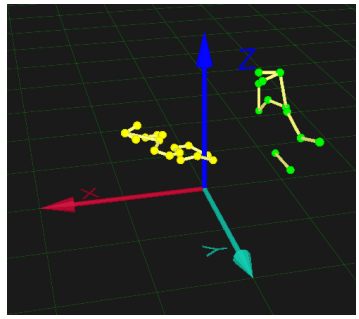


Figure 7.1: The 3-D coordinate system of the recording room from Qualisys. The markers of the mother and child are illustrated with green and yellow, respectively

sented the mid point of the child's and the mother's head, respectively. These were calculated as the mean point between the two markers CheadR and CheadL for the child, and as MheadR and MheadL for the mother. See figure 3.1 for a recollection of the position of these markers.

To calculate the orientation of the child's head at all frames regarding the XZ-direction, [34] first estimated the reference plane between the two heads. This is thought of as the plane between the mother and child, where they point their faces towards each other and is interpreted by Babylab as the plane where the child faces the mother, because it is assumed that the mother is always orientated towards her child.

This plane was estimated in [34] by drawing a vector from the CheadB marker to the CheadM marker and likewise from the MheadB marker to the MheadM marker. Where the two vectors were in parallel, it was assumed that the mother and child were looking at each other. Due to the fact that the CheadB marker is positioned on the top of the child's head, the vector between the CheadB and CheadM markers was pointing downwards instead of directly ahead as expected. The calculation of the child's head position therefore involved a manual estimation of the angle between the vector representing the child's direction and the corresponding vector for the mother. This was done once for every recording from the video data at a frame where the child visually directs its head towards the mother and vice versa, see [34] for more details.

Due to these corrections that must be included before calculating the child's head orientation in the XZ-plane, it is here decided not to use this as a feature, because of the manual aspect of it which is in conflict with the objective of this

thesis.

[34] also extracted the head position of the child with respect to the XY-plane, i.e. the child's side-to-side head moving. The child's head orientation was also calculated with respect to the mother in this task and was again carried out by representing the midpoints of the child's and the mother's head through the two created markers MheadM and CheadM, respectively. In this task, [34] applied the manually corrected back point on the head of the child.
In this thesis, the child's head position with respect to the XY-plane is calculated automatically. The two extra markers MheadM and CheadM has been found as in [34] and used for the calculations. After this a vector is drawn between the back marker and the new generated point, see figure 7.2. The two vectors represents the direction of the mother's and child's head and the angle between these two vectors can be perceived as the head position of the child with respect to the mother. When the angle between the two vectors is zero, they are facing each other. The approach is shown in figure 7.2.

The angle between the two vectors are given by the formula in (7.1) where

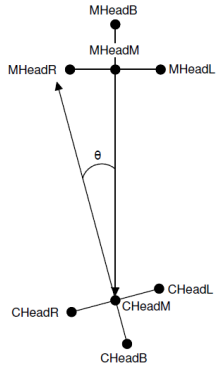


Figure 7.2: Illustration of how the angle between the mother and her child is calculated.

M and C are the vectors representing the mother's orientation and the child's orientation, respectively.

$$cos(\theta)_n = \frac{M \cdot C}{|M||C|} \tag{7.1}$$

In (7.1) the vectors M and C only include the x and y coordinates, since the z coordinate, as mentioned, only carry information about the position in the vertical direction and thereby not about the position of the child's head moving from side to side with respect to the mother's.

Equation (7.1) is therefore applied to extract the angular information for each frame of the mocap file.

7.2 Distance Between Faces

The distance between the mother and the child's faces is simply calculated for all frames of the mocap file with use of the formula given in equation (7.2). Here $n = 1, 2, ..., N$ where N is the total number of frames.

$$dist_n = \sqrt{(x_{Mn} - x_{Cn})^2 + (y_{Mn} - y_{Cn})^2 + (z_{Mn} - z_{Cn})^2} \tag{7.2}$$

In (7.2) x_M, y_M, z_M, x_C, y_C and z_C refer to the mothers x, y and z coordinates and the child's x, y and z coordinates, respectively. The coordinates that have been used in calculating the distances are the mean coordinate of the markers MheadR and MheadL in figure 3.1 for the mother and the mean of the corresponding markers for the child, CheadR and CheadL, in figure 3.1. The choice of these estimated markers bases on the belief that they represent the positions of the mother's and child's head.

7.3 Child's Physical Energy Level

The child's physical energy level has, as mentioned, been calculated as well. To calculate this, the child's right wrist marker has been used, named CwristR in figure 3.1, due to the fact that this marker is also being used at Babylab for this particular calculation.
The child's physical energy level has been estimated by calculating the covered distance between two consecutive frames of the mocap file, and is given by (7.3), with $n = 1, 2, ..., N$ where N is the total number of frames.

$$energy_n = \sqrt{(x_{Cn} - x_{Cn-1})^2 + (y_{Cn} - y_{Cn-1})^2 + (z_{Cn} - z_{Cn-1})^2} \tag{7.3}$$

In (7.1), x_C , y_C and z_C refers to the x, y and z coordinates of the child's right wrist.

Chapter 8

Combining Modalities

The uniqueness of the data provided by Babylab lies in the fact that three different data modalities have been used to measure the interaction between the mother and her child. In continuation of the speaker identification task in chapter 5 as well as the emotional classification in chapter 6 that focused only on features extracted from sound, the chapter presented here seeks to solve the same problems just now by incorporating the information from the motion capture. The first part of this chapter therefore discusses the combination of the sound and motion capture for the speaker identification task and the emotional classification, respectively. The chapter is rounded off by a brief discussion on the third available data modality; video and how this modality could contribute in the two problems considered.

8.1 Combining Sound and Motion Capture

The automatic annotations from mocap are, as previously stated, the head orientation and the physical energy level of the child as well as the distance between the mother and the child. These are all possible candidates for improvement of the classification error rates for both the speaker identification and the emotion

recognition tasks.
The following provides a combined discussion on the possible improvement in both classification tasks for all three motion capture annotations. This discussion includes acknowledged psychological theory, Babylab's ideas and thoughts on the interactions, as well as the intuitive assumptions and expectations of the authors of this thesis that have emerged during the study.
Regarding the head orientation of the child with respect to the mother, the psychologists at Babylab believe in a connection between the child's utterance and head orientation. In combination with this is the distance between the mother and child. At Babylab the concept of *chase and dodge* as explained in chapter 7 is incorporated in their analyses of the mother-child interaction.
With this theory it would be reasonable to assume that the head movement is assisted by an emotional utterance. It is possible to assume that when the child makes a negative utterance, it could be prone to turn its head away from the reference position, that is, the child facing the mother. This also induces the expectation that for the child's positive utterances, the head orientation is most likely in the direction of the mother.
Another, very likely scenario, is that if the child is bored it might turn its head to examine whether more interesting things are occurring in the vicinity. The expected immediate response of the mother is that she begins to speak and perhaps leans forward to capture the child's attention again.
Are these theories in fact true, the angular feature and the head distance feature could be of assistance to both the speaker and the emotion classification, and thereby improve the ratio of correct automatic labels in both tasks.
For a substantiation to the above mentioned theories, selected video recordings have been visually inspected by the authors of this thesis, but since much of the stated interactions occur simultaneously it is very difficult to validate the presumptions.

The results for the two problems of speaker identification and emotion recognition are discussed in sections 9.4.1 and 9.4.2.
The child's physical energy feature could likewise, potentially, improve both classification tasks. In the recorded videos it is often seen that the child responds to the mother's actions (vocal or physical) by either moving or making a sound or sometimes both. The physical energy level of the child could therefore possibly contribute in a more confident assigning of the label *child speaking* in the speaker identification task. An issue with this interaction between sound and movement is that it is not always instantaneous. A delay between the child's sounds and the movement is observed, which could result in a deterioration of the classification compared to excluding this feature in the speaker identification problem. The result of including this feature can be seen in section 9.4.1.

Regarding emotion recognition no apparent connection is present with respect to the child's physical level from the video recordings, but a somewhat far-fetched possibility is that the child could combine positive utterances with movement to show the enthusiasm. The physical energy of the child is thus included in the emotion recognition task, all though no improvement is expected. The results are discussed in section 9.4.2.

8.2 Information from Video

The third data modality available is as mentioned the video recordings. The video enables the visual interaction between the mother and her child, recall figures 3.3(a) and 3.3(b). Thus, the video carry information that can not be extracted from the two additional recording modalities sound and motion capture, respectively.
Regarding the problems already considered concerning the speaker identification and emotion recognition, it could be very beneficial to extract the child's facial expressions from the video as well as information about the mouth movement of the child. These informations could contribute as a support to the information already extracted from the sound and motion capture files in the two respective problems and thereby improve the precision of the classifications.
In addition to the presumed improvement of the classification problems approached in this thesis, the information on the child's facial expressions is applied by Babylab in the analyses of many psychological interaction patterns between mother and child. One of their teams focus on coding the facial expressions of the child to analyse the emotions of the child, inspired from [44], as well as the interaction pattern between this and the mother's actions. As mentioned in 1, the problems with manual codings are that there can be large differences in the labelling from one coder to another and that the coding is very time consuming. Therefore, it is of great interest for Babylab to obtain automatic annotations.
To extract the mentioned information from the video modality the believe is that the Active Appearance Model can be applied, [45]. Due to the scope of this thesis, the Active Appearance Model has only been investigated briefly and the description as well as discussion of this can therefore be found in appendix B.

Chapter 9

Results and Discussion

In this chapter the results obtained in this thesis are presented and discussed. The chapter is divided into four sections, where the first, section 9.1, presents the results obtained in the speaker identification problem, exclusively based on features from the sound modality. Section 9.2 focus on the results obtained from the emotion classification, again exclusively based on the sound modality. The subsequent section, 9.3, discuss the annotations obtained from the motion capture modality whereas the section hereafter, 9.4, provides the results for the speaker identification problem as well as the emotional classification problem when the features from motion capture are included.

9.1 Speaker Identification

In this section the results from the speaker identification task will be shown and discussed. Since the intention of the speaker identification task is to generate a generalized model across dyads which can be used by Babylab, the results shown in this section are for 14 dyads, unless otherwise mentioned. One dyad is used as test set to see how the generalized model performs in classifying the data from an unknown dyad. The model is thereby tested in this thesis using the same approach as is intended for Babylab. It is to be noted that for all tests performed regarding speaker identification, except of course for the combining

of channels, only the mother's channel is applied (channel 2). This is because of the much more noise-filled channel belonging to the child (channel 1), due to the many movements the child makes during the recording sessions.
It should be noted that due to the large data set and the time-consuming machine processing, all tests in this section on speaker identification have been run only once, that is for dyad 001 as test set. Therefore no error bars will appear on the resulting figures.
Various aspects regarding the performance of the classifiers have been discussed and augmented before running the actual tests of the performances. Among these are the optimal window size, parameter optimization for the five respective classifiers, combination of features, combination of channels and so on. Due to the many possible combinations of the above mentioned, it is not possible to optimize all of these aspects concurrently. Augmented decisions are therefore to be made and from this, the best possible combination of parameters is to be found so as to obtain the best possible model performance.

It is in this study decided that the first evaluation is performed as a function of the window sizes, as explained in section 5.2, for each of the five classifiers described in section 5.4. Before performing this test, the parameters of each classifier are to be estimated. These results are shown in section 9.1.1 and from this the performance of the classifiers can be compared as well as their individual performance for each of the window sizes can be determined.
Section 9.1.1 on parameter estimation and window size, is followed by a section regarding the confusion matrices, section 9.1.2. Here the types of errors that each of the classifiers make are evaluated, as well as a discussion on the reason for these errors. A brief description on how the true class labels are coded, as well as the reliability of these manual codings, is also included in this section.
The optimal composition of features is determined by an evaluation of the model performance as a function of different feature combinations in section 9.1.3, providing the optimal features in the speaker identification problem. It is furthermore investigated in this section if the combination of classifiers with different features can be combined to boost the performance of the classifier. Again the types of errors made by each classifier are discussed.

As mentioned, the performance of the classifiers for different window sizes are tested. In section 9.1.4 it is investigated if the use of sub windows to predict a given window size can contribute in a boosting of performance of that given window size. In continuation of this, the time-related errors that the classifiers make are investigated through plots of the class labels as a function of time. This will contribute in the understanding of the prediction performance using sub windows.
While the already mentioned tests are carried out using only one channel it is

in section 9.1.5 tested whether a combination of the outcome of the classifier for each the two available channels can contribute in a boosting of the performance. The chapter is rounded off by a brief summary of the results obtained so far in the speaker identification problem.

9.1.1 Parameter Estimation

As mentioned, the first evaluation of the model performance regards the six window sizes, in order to obtain the size of the sound segment that has the best predictive ability as well as to determine which of the classifiers that show the best performance. Before this test is possible, the parameters for the five respective classifiers, as mentioned in section 5.4, are to be decided.
For the sake of overview, table 9.1 presents each classifier and the corresponding parameters that are optimized in this thesis.
In the following, the approach of the parameter estimation is described for each

Classifier	Parameter
GMM	Components, K
KNN	Nearest neighbour, K
TREE	Size of leaf, κ + split criteria
MNR	None
ANN	Hidden units, H

Table 9.1: Each of the five classifiers and the corresponding parameters to be optimized.

of the five classifiers.

9.1.1.1 Gaussian Mixture Model

The GMM, as explained in section 5.4.1, models each speaker in the speaker identification problem separately. Thus, for each class the optimal number of components, K, in the Gaussian Mixture is to be decided. Before deciding the number of components, different decisions about the covariance matrix Σ should be made where the number of covariance matrices is one of them.
Different opportunities arises; the model can have a covariance matrix for each component K, one single covariance matrix for all components in one speaker

model and finally shared covariance matrix between the models and thereby speakers. Furthermore the covariance matrix can be chosen to be full or diagonal. If K is the number of components and F the number of features, the full covariance matrix would result in the estimation of $K(F + F(F + 1)/2) + K$ parameters whereas for the diagonal covariance matrix this reduces to the estimation of $K(F + F) + K$ parameters. In the expression of the number of parameters, the term KF represents the number of means to be estimated, the term $K(F(F+1)/2)$ presents the number of parameters in the covariance to be estimated and finally the last term K presents the number of probabilities for each component, π_k, to be estimated.
[57] uses the approach where a covariance matrix for each component K is used and further these covariance matrices is chosen to be diagonal. The choice was based on initial experiments where this composition showed the best identification results. It is therefore in this thesis chosen to use the same approach as in [57].
As mentioned the number of components to model each speaker is to be found. One way to do this is to calculate the Bayesian information criterion (BIC) which is given by equation (5.17), from section 5.4.1. In figure 9.1, the BIC is shown for class 1 as a function of number of components K. As already explained, BIC only allows the number of components to increase if the model complexity does not overcome the increase in likelihood. The curve is therefore expected to have a minimum at a given K that thereby represents the best trade-off between model complexity and how well the model fits the data.

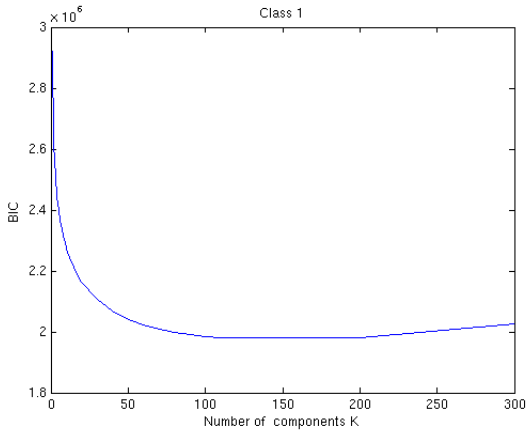


Figure 9.1: BIC as a function of number of components, K, for the GMM here shown for class 1, window size 150 ms, channel 2 (mother) and with all features included as shown in table 5.3.

By looking at figure 9.1 it is seen that the BIC decays as a function of number of components K, until approximately $K = 100$. An increase in BIC is seen when the number of components reaches 200. This means that the increase in model complexity at the point of $K = 200$ overcomes the increase in likelihood and thereby how well the model fits the data. From this it can be assumed that the data investigated in this study has a really complex structure, since the Gaussian Mixture for class 1 alone should include between 100 and 200 components to model and thereby fit the data.
The time it takes to train the model should also be taken into account in the evaluation of the number of components used in the GMM. The fitting of a model with 100 components and with the number of observations as in the scope of table 5.2, with the window size 150 ms, takes about 7 hours compared to approximately 7 minutes for 10 components, when the calculation is performed on the cluster facilities at IMM. Therefore, because of the results obtained from figure 9.1 it is decided to seek another method for finding the optimal K.
Another way to calculate the optimal K is by evaluating the classification error rates as a function of the number of components, where these are assumed equal for each class, i.e. $K_1 = K_2 = K_3 = K_4$. This is carried out for each window size, and the result for 150 ms is shown in figure 9.2. The results are obtained using all the features presented in table 5.3. Results for the remaining five window sizes are shown in appendix D.1.1.

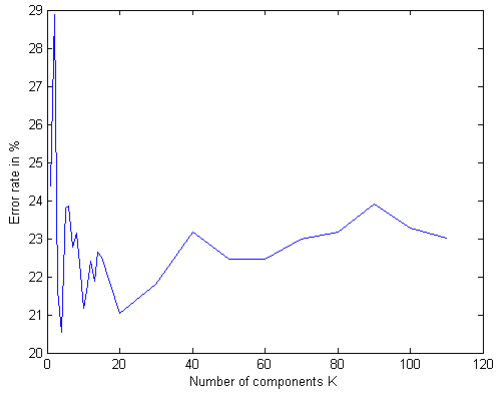


Figure 9.2: The error rate as a function of number of components K for GMM, when K is assumed equal for all classes. Here shown for the window size 150 ms.

By looking at figure 9.2 it can be seen that for $K = 4$ the error rate reaches its minimum. This result and the results derived from the figures in appendix

D.1.1 regarding the remaining five window sizes are summarized in table 9.2.

Window size	Optimal K
10 ms	10
50 ms	30
100 ms	11
150 ms	4
200 ms	4
250 ms	7

Table 9.2: The optimal number of components, K, for GMM for each window size, when the number of components are assumed equal across classes and is decided as the K with the best resulting error rate. Results are for channel 2 (mother) and with all features included as shown in table 5.3.

9.1.1.2 K-Nearest Neighbour

In the K-nearest neighbour algorithm, presented in section 5.4.2, the number of neighbours included in the classification can be varied and the optimal number of neighbours must therefore be found. This is carried out by calculating the error rate as a function of the number of neighbours. The results are shown in figure 9.3 for the window size of 150 ms with all features included. See table 5.3 for a recollection of these features.

The results in figure 9.3 show that when the number of neighbours reaches 15, the error rate is at its minimum. The figures showing the results for the remaining window sizes can be found in appendix D.1.2 whereas a summing up of the results for each window size can be seen in table 9.3.
From table 9.3 it is seen that the optimal number of neighbours lies in the range 8-18 for the six respective window sizes. These results indicate that the observations are positioned in smaller clusters in the 61-dimensional feature space, since only a smaller number of neighbours are required to obtain the best model fit. If the optimal neighbours on the other hand had shown to be 50 the clusters would probably have been larger in the 61-dimensional feature space. The optimal number of neighbours given in table 9.3 are used in the following regarding

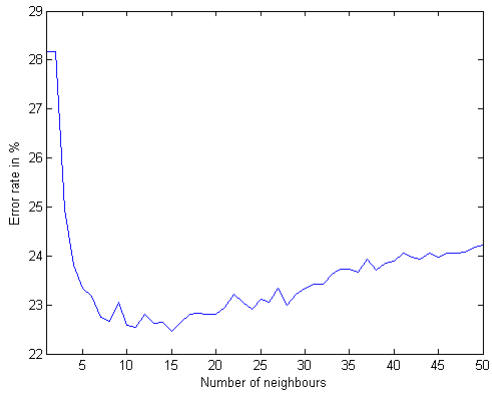


Figure 9.3: The error rate as a function of number of neighbours when using KNN. Here shown for the window size 150 ms, channel 2 (mother) and with all features included as shown in table 5.3.

Window size	Optimal neighbours
10 ms	18
50 ms	10
100 ms	8
150 ms	15
200 ms	8
250 ms	8

Table 9.3: The optimal number of neighbours when using the KNN shown for each window size. Results are for channel 2 (mother) and with all features included as shown in table 5.3.

the KNN classifier.

9.1.1.3 Decision Tree

Regarding the Decision Tree, as described in section 5.4.3, different parameters can be varied to optimize the fitting of the tree. The size of each leaf in the decision tree is varied, to investigate the performance of the tree as a function

of the number of κ. Here κ is the number of observations that each impure node must at least have to undergo a split. Recall from section 5.4.3 that an impure node is a node where observations belonging to more than one class is present. The result of varying κ can be seen in figure 9.4 for the window size of 150 ms, where the entropy, as given by equation (5.26), was used as the impurity measure. The results for the remaining window sizes can be found in appendix D.1.3.

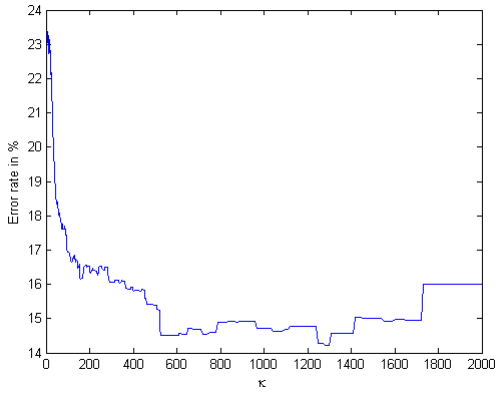


Figure 9.4: The error rate as a function of κ in the decision tree classifier. κ is the number that decides the minimum number of observations before the impure node undergoes a split. Here shown for the window size 150 ms, channel 2 (mother) and with all features included as shown in table 5.3. The measure of impurity used for these results is the entropy measure given by equation (5.26).

As seen from figure 9.4 the error rate reaches its minimum when κ is equal to approximately 1300 for the window size of 150 ms. Off-hand, 1300 observations seems of many, but in comparison to the size of the data set of 150 ms, as seen in table 5.2, 1300 only accounts for approximately 2 % of the entire data set and 22 % of the smallest class. The results for the remaining window sizes are shown in table 9.4.

As mentioned in the section about decision trees, section 5.4.3, another way of determining the split stop is by pruning the tree. Recall that pruning is the process where smaller parts of the tree can be removed if these parts only contribute minimally in the final outcome of the classifier. Because pruning is another method of determining when the splitting should stop, it is decided to

Window size	Optimal κ
10 ms	800
50 ms	1520
100 ms	1060
150 ms	1300
200 ms	300
250 ms	1900

Table 9.4: The optimal number of κ in the decision tree classifier where κ is the number that decides the minimum number of observations before the impure node undergoes a split. Results are for channel 2 (mother) and with all features included as shown in table 5.3.

omit a test of this.
As mentioned in section 5.4.3 the two measures of impurity as split criteria are the entropy and the Gini measure. The test for deciding the optimal number of κ above was carried out using the entropy measure. Therefore it is also tested how the Gini impurity measure acts on this data. The results for this can be seen in appendix D.1.3 in the figures D.16, D.17, D.18, D.19 and D.20. In looking at the results for the Gini impurity measure, the error rates seem to be insignificantly different in comparison to the entropy measure. It is therefore decided that the remaining tests for the decision tree set-up is carried out using the entropy measure.
In figure D.31 in appendix D.6 a TREE is shown. Due to the very complex structure of the TREE's obtained in this study a simplified version has been made. The TREE is fitted using only 1300 observations and only the features energy, zcr and the cross-correlation has been used.

9.1.1.4 Multinomial Regression

As mentioned under the classification methods in section 5.4, the MNR classifier has also been applied to the speaker identification problem. There are no obvious parameters to be optimized with regard to this classifier and the MNR is therefore tested with default settings, see appendix D.2 for these settings.

9.1.1.5 Artificial Neural Network

The ANN requires the number of hidden units as an input. There is no general rule describing how many hidden units is needed to model a given data set and in the case considered in this study, not much is known about the data set. The number of hidden units is therefore to be decided by testing the performance of the ANN, with the number of hidden units in the range from 1 to 15. The maximum of 15 is based on the computational cost of the calculations, which increase drastically when increasing the number of hidden units. The results, based on the error rates of the classification, of varying the number of hidden units from 1 to 15 can be seen in figure 9.5 for the window size of 150 ms.
In figure 9.5 in can be observed that the number of hidden units that results in

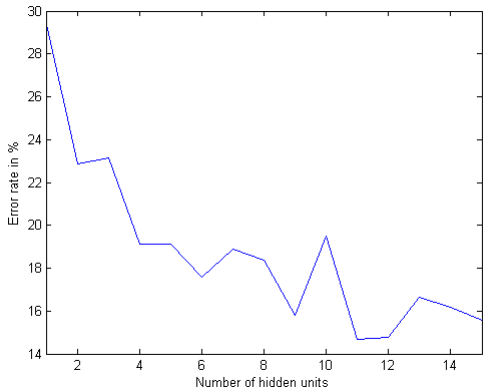


Figure 9.5: The error rate as a function of number of hidden units in ANN. Here shown for the window size 150 ms, channel 2 (mother) and with all features included as shown in table 5.3.

the lowest error rate for the window size of 150 ms is 11. Table 9.5 summarizes the results for the remaining window sizes, where the corresponding figures can be seen in appendix D.1.4.

The results in table 9.5 show that the number of hidden units needed to model the data are in the range of 5-11 for the six respective window sizes. The number of hidden units provides a kind of information on how complex the problem to be modelled is. Because of the relatively high number of hidden units needed to model the data, it seems that the data has a highly non-linear structure in the feature space.

Window size	Optimal hidden units
10 ms	5
50 ms	9
100 ms	8
150 ms	11
200 ms	9
250 ms	10

Table 9.5: The optimal number of hidden units when using the ANN shown for each window size. Results are for the window size 150 ms, channel 2 (mother) and with all features included as shown in table 5.3.

9.1.1.6 Test of Window Size

With the optimal parameters presented in tables 9.2, 9.3, 9.4 and 9.5, the error rate as a function of window size can be calculated for all respective classifiers. The result of this can be seen in figure 9.6.

In figure 9.6 the first thing that should be noticed is that every classifier performs better than the by-chance error rate described in equation (5.56) of 67 % for the imbalanced data set. This indicates that there is a clear signal, meaning that the classifiers are capable of separating the four respective classes from each other in the feature space. Furthermore it is seen from the figure that the MNR classifier clearly has the worst performance. As mentioned, the number of hidden units needed to model the data indicates that the problem dealt with has a highly non-linear structure in the feature space. The same conclusion was drawn when the BIC was plotted against the number of components, as shown for 150 ms in figure 9.1. This non-linearity of the feature space as well as the fact that the MNR classifier is only capable of making linear decision boundaries, explains the much worse performance of the MNR compared to the four other classifiers.

Observing the different window sizes in figure 9.6, and excluding the MNR, the classifiers of 10 ms windows has the worst predictive ability in comparison to the classifiers of the larger window sizes. This implies that the information obtained from a 10 ms sound signal holds a smaller amount of information than the other window sizes. This should be compared to the fact that the data set of 10 ms

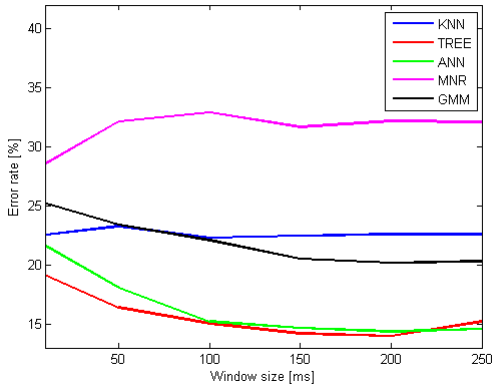


Figure 9.6: Error rate of the five classifiers (KNN, decision tree, ANN MNR and GMM) each represented by different colours as a function of window size. The results are for channel 2 (mother) using all the features in table 5.3.

windows by far has the highest amount of observations, as seen in table 5.2. It seems understandable that the classifiers' predictability of the 10 ms windows are worse than the others, because 10 ms of sound is very little for distinguishing between the four classes. A task that not even the human ear would be able to handle.

From the figure of window sizes, another noticeable fact is that ANN and TREE show really good performances in comparison to the three remaining classifiers. Generally, the performance with respect to error rate is in the order of 15 %, which clearly deviates from the error rates of the others. In the window range of 100 ms - 250 ms the error rates are almost stable for ANN and TREE. Thus, not one particular window size appear more capable of prediction than another. Based on this as well as on the claim made by [24], as mentioned in section 5.2, that the concept of stationarity in sound signals holds for window sizes up to 200 ms, the choice of 150 ms windows in the further tests thereby seems reasonable. An advantage of the 150 ms window, is that the 50 ms window can be used in the test of predictability, which is carried out in section 9.1.4.

9.1.2 Confusion Matrix

Before making a final conclusion on which of the classifiers that in the best possible way models the four-class speaker identification problem, the confusion matrices should be taken into consideration. These illustrate how many of each class that have been classified correctly and how many that have been misclassified into each of the three other classes, see 5.5.1 for an explanation on confusion matrices. Since ANN and TREE seem to be the best performing classifiers from figure 9.6, the confusion matrices for these are shown in figure 9.7, whereas the confusion matrices for MNR and KNN can be found in appendix D.3. A window size of 150 ms is used, as decided in the previous section, section 9.1.1.

The error rates for ANN and TREE are observed to be 15% and 14%, re-

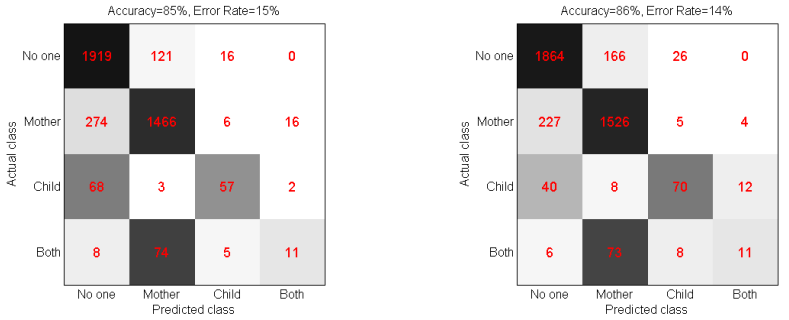


Figure 9.7: Confusion matrices shown for (a) ANN and (b) TREE, both for the window size 150 ms.

spectively. Based on this, the performances of these two classifiers appear quite equal. By looking at the confusion matrix for ANN in figure 9.7(a) it is seen that the errors occur in the classification of *mother* versus *no one* (and vice versa), *child* versus *no one*, and *both* versus *mother*. Here the true classes are indicated by the former of the two and the predicted classes as the latter. From figure 9.7(b) it is seen that the misclassifications made by the TREE are similar to ANN, only with a little lower frequency.

It is seen from the confusion matrices that both the ANN and the TREE models the classes *no one* and *mother* quite well. These are the classes with the highest number of observations as seen in table 5.2. Another noticeable thing about the two confusion matrices is that when the actual class is *no one*, then no observations are classified as the class *both*. Furthermore, when the actual class is *no one* then 121 and 166 observations are classified as the class *mother* by the ANN and TREE, respectively. These two cases might be the result of

the imbalanced data set and thereby unequal prior probabilities for each class, recall table 5.4 that showed the prior probabilities for each class in the training set and the test set.
From table 5.4 is can be seen that the prior probability of the classes *both* and *mother* are 13 % and 32 %, respectively. This, as mentioned, might be the reason why no observations from the class *no one* are classified as the *both* class. That the two classes *mother* and *no one* have very large priors is probably the explanation of the 121 and 166 misclassified observations for the ANN and TREE, respectively. The difference between these two classes is presumed to be large in the feature space and thereby few misclassifications between the two is expected. Clearly, this is an example of the effect of the prior probabilities in classification tasks.

Because GMM is the most applied classifier in the speaker identification problem [57], [31], [33], it is interesting to analyse the confusion matrix for GMM as well. This is shown in figure 9.8. From this it can be seen that the confusion is largest when the child speaks. Furthermore it can be observed that the misclassifications of the GMM are somewhat similar to the mistakes made by ANN and TREE, only with a higher frequency.

In continuation of the discussion of the confusion matrices shown in this sec-

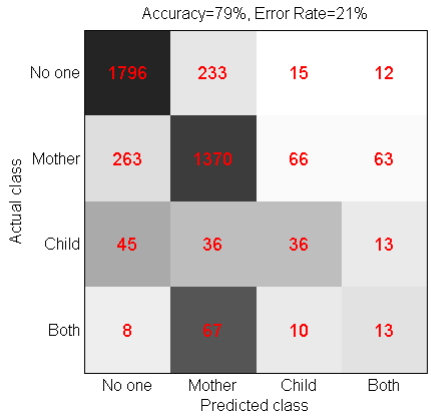


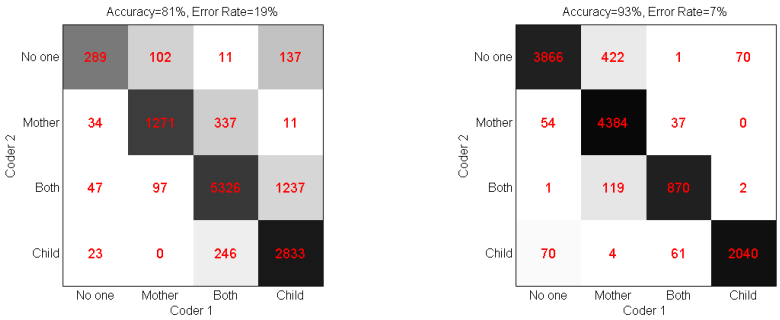
Figure 9.8: Confusion matrix for GMM.

tion, it is interesting to move deeper into the discussion of the errors that the classifiers make. The confusion matrices shows that 274, 227 and 263 observations belonging to the class *mother* are classified as the class *no one* by the tree classifiers ANN, TREE and GMM, respectively. The manually annotated labels made at Babylab are, as already stated, used as the true classes and naturally human errors will occur in the coding process. In continuation hereof, certain

guidelines are made at Babylab for these codings. One of the instructions at Babylab for the manual annotations is that when the mother whispers, the true class label is set to the class *mother*.
Due to the low power in the signal at time instances where the mother whispers, this could be the reason why the 274, 227 and 263 observations are classified as the class no talks. Of course it should be kept in mind that this class, *no one*, as seen from table 5.4 has a prior probability of 45 %, but because the class *mother* also has a high prior probability (32 %) this fact is probably not capable of explaining but a few of the misclassifications of the *mother* to the *no one* group. In continuation hereof, the intuition is that the classifier would normally be able to distinguish between these two groups, due to their presumed dissimilarity in the feature space. It is therefore assumed that by far the majority of the misclassifications of the class *mother* to the class *no one* is due to the manually annotated labels, where whisperings of the mother is assigned to the class *mother*.

Another example of these guidelines of the manual codings is that the child's burpings and hiccups are not included in the class *child* with the argument that this is not to be used in the further analysis of the labels. This could also cause a confusion in the classification of speaker identity due to the fact that the signal segments of these occurrences contain energy as well as spectral content.
When the annotations are carried out at Babylab one coder annotates the full 10 minutes of the recordings while another coder annotates 2 minutes of the same recoding, for the sake of reliability testing. The confusion between two coders can be seen for two different dyads, 018 and 012, in the confusion matrices in figure 9.9. The confusion between two coders for dyad 006 and 020 can be found in appendix D.3.

The confusion matrix in figure 9.9(a) and 9.9(b) between two coders show that the error rate between their labels are 19% and 7%, respectively, whereas the ones shown in appendix are 8 % and 31 %, respectively. This gives rise to the question; what are the true labels?
The striving after an as small as possible error rate should of course be held up against the fact that no exact definition of the true labels is available. This means that if an error rate of 3 % is obtained when comparing the automatic estimated labels with the annotations of one coder, an error rate of 15 % might be obtained if the same automatic estimated labels were compared to the annotations made by another coder.
It should be mentioned that the confusion of 31 % is being re-annotated by Babylab due to this very bad reliability. In continuation of the discussion about the true labels it should be noticed that the precision with which Babylab performs the annotations in Praat is 10 ms. It could be doubted that a precision of



(a) (b)

Figure 9.9: The human confusion between two coders at Babylab shown for (a) dyad 018 and (b) dyad 012.

this size will always result in the true class labels because the human ear simply cannot validate the reliability of these labels.
A note should be made that only these four data sets were available from Babylab with double codings for reliability. If double codings were available for all dyads, a probability density function over all the human error rates could be calculated. Assuming this probability density function over human error rates was available, it would be possible to see if the error rates obtained with the machine learning approach in this thesis, would belong to this probability density function. If this was the case, the classifier performance would be just as good as the manual annotations. Since this is not possible, it can only be assumed that the classifier confusions are similar in size to the human confusions.
As a conclusion on this section, it is chosen to exclude the MNR and KNN classifier in the subsequent sections due to their higher error rates compared to ANN and TREE. This also explains why the ANN and TREE are used in the following experiments. It is furthermore chosen to investigate the GMM as well because this classifier, as mentioned, is the most commonly used in the literature regarding speaker identification.

9.1.3 Test of Features

Until now, the full feature combination has been used, as it appears in table 5.3. The composition and choice of these features is as mentioned due to the literature findings, see section 5.3. Naturally, the composition of features that results in a good error rate depends on the specific problem, data and classification method. It is therefore, in this thesis, decided to perform a test of the

features to investigate the influence of the different features on the performance of the classifier.
Furthermore, the problem of speaker identity in this thesis differs from the usual speaker identification problems because of the availability of two microphones as well as the position of these microphones. Also the case of four classes with two of them not just consisting of one speaker, but either both or none, differentiates this problem from the problems in the literature.

To test the influence of different feature combinations, the performance of the classifiers TREE, ANN and GMM is evaluated as a function of variable feature compositions. This is shown in figure 9.10, with the numbers from 0 to 9 representing these combinations. Table 9.6 shows the feature combinations that corresponds to the numbers. In relation to this it should be noted that either all MFCC's are included or none. Combinations of different coefficients of the Mel-frequency cepstrum are thereby not tested.

As seen in figure 9.10, the performance of the classifiers varies with the different combinations of features. The lowest error rates are obtained with the feature composition 9 for GMM and ANN and with feature composition 8 for TREE. These two feature compositions include only very simple features of the sound signal, namely the energy of the signal and the cross-correlation between channel 1 and channel 2 for both compositions and the zero crossings included

Composition	Feature Composition
0	MFCC, delta MFCC, delta-delta MFCC, energy, zcr, cross-correlation
1	MFCC, delta MFCC, energy, zcr, cross-correlation
2	MFCC, energy, zcr, cross-correlation
3	MFCC, energy, zcr
4	MFCC, zcr, cross-correlation
5	MFCC, energy, cross-correlation
6	MFCC, cross-correlation
7	MFCC
8	Energy, zcr, cross-correlation
9	Energy, cross-correlation

Table 9.6: The tested feature compositions.

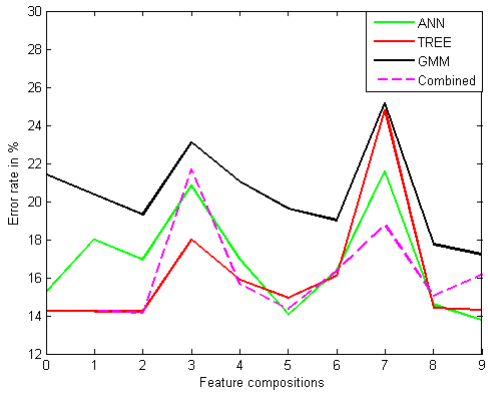


Figure 9.10: The performance of the classifiers as a function of feature combinations. The combination of the three classifiers at each feature composition is shown as well.

in composition 8 as well. Thus, the feature compositions that result in the best error rates do not include the MFCC features.
Due to the results from the literature mentioned on MFCC, [24], [55], [46] and [57] where the MFCC's are the primary features used, the outcome of figure 9.10 was not as expected. But as mentioned, the problem considered here is not identical to that of the literature, which might be the reason why the MFCC features does not have the same impact on the performance of the classifiers. Another possible scenario is that the number of coefficients extracted from the Mel-frequency cepstrum is not optimal with respect to this problem. The number of coefficients used in this study was based on the literature, and although interesting, this aspect will not be further investigated here.
The feature compositions that performs the worst are composition 3 and 7, respectively. From table 9.6 it is seen that these two compositions are the only ones that do not include the cross-correlation features. The cross-correlation between the two channels provides the information of the identity of the speaker is the mother or the child, see the explanation of the cross-correlation in section 5.3.1. It therefore appears reasonable that this feature contributes in an increase of the performance of the classifiers.

As mentioned in section 5.5.3.1, the outcome of each classifier can be combined by a majority voting. If, as mentioned, the errors made by the individual classifiers are independent of each other a gain in the performance would be seen when combining these. In figure 9.10 the combination of the TREE, ANN and GMM at each feature composition can be seen as the dotted magenta curve.

What is seen is that the combination of the three different classifiers only show a gain at feature composition 7. Thus an overall gain in the performance is not seen when combining the classifiers.
Figure 9.10 illustrates how influential the individual features and the combination of these are to the classification (ranges from 14 % to 25 % for TREE, 14 % to 21 % for ANN and 17 % to 25 % for GMM). Based on this observation it would be interesting to investigate if the classifiers for different feature combinations can contribute in a boosting of the performance, as described under model evaluation in section 5.5. The prerequisite for obtaining a lower error rate by combining classifiers is that the errors made by the classifiers should be independent of each other. This means that when one classifier fails in ascribing an observation correctly, another classifier perhaps correctly ascribes the same observation, which could result in a boosting of the performance if these classifiers were combined. To make a majority voting between the classifications from several classifiers an odd number of classifiers must be combined.
This test is performed for the TREE classifier and therefore the types of errors that this classifier makes are analysed. For this, the confusion matrices are useful. The confusion matrices for all the feature compositions, are shown in appendix D.3. From these it is observed that the confusion matrices of feature composition 0 and 1 are identical, i.e. they make the exact same errors.
It can also be seen that all the classifiers to some degree are capable of modelling the two classes *no one* and *mother*. Where the classifiers differentiate from each other, is in the modelling of the class *child*. The classifiers of the feature compositions 0, 1, 2, 5, 8 and 9 show the highest precision of this class.
Of interest would be to combine one of these classifiers with one that has a higher precision in one or more of the other classes, to benefit from the different classification errors. From the confusion matrices it is seen that the feature compositions 4 and 6 models the class *no one* somewhat better than the 0,1,2,5,8 and 9, mentioned before. Based on this, and on the error rates of the classifiers, it is investigated if the combination of 6, 8 and 9, that obtain the lowest error rates individually, results in an improvement of the error rate.
The resulting error rate is 14 % and is shown in table 9.7. The error rates for composition 6, 8 and 9 are 16%, 14% and 14%, respectively, meaning that no overall gain in the performance is obtained when combining these classifiers.
In table 9.7 the results of combining other feature compositions is also shown. Here the individually obtained error rate is shown as well in the parenthesis next to each of the feature compositions. The approach is the same as the previous: the combinations chosen show different kinds of errors in the confusion matrices.

As seen from table 9.7 the error rates obtained when combining 3 TREE classifiers with different feature compositions, are no lower than the smallest error rate for each individual classifier. If a final conclusion was to be made, all combinations of the feature compositions should be tested. This is not carried out due

Composition A	Composition B	Composition C	Combined
6 (16 %)	8 (14 %)	9 (14 %)	**14 %**
2 (14 %)	5 (15 %)	9 (14 %)	**14 %**
1 (14 %)	4 (16 %)	9 (14 %)	15 %
1 (14 %)	5 (15 %)	8 (14 %)	**14 %**
7 (25 %)	8 (14 %)	9 (14 %)	**14 %**
0 (14 %)	4 (16 %)	8 (14 %)	15 %

Table 9.7: The error-rates obtained when combining 3 TREE classifiers with different feature compositions. The individually obtained error rates are shown as well.

to numerous combinations. But from the results shown, it seems that no boosting in the performance can be obtained by applying the method of combining classifiers with different feature compositions.

9.1.4 Test of Predictability: Windows versus Sub-Windows

The section presented here tests the predictability of a given window size. The window size used is that of 150 ms. The predictability of the 150 ms window is tested from the outcomes of the 50 ms windows. The error rate obtained here is then to be compared with the error rate from the classifier of the 150 ms windows. The details on window predictability were given in section 5.5.3.2. The majority voting of the 50 ms windows is carried out for the classifiers TREE, ANN and GMM. The optimal parameters for the window size 50 ms, shown in tables 9.2, 9.4 and 9.5, are used. The results can be seen in table 9.8 where the feature composition 8 is used for TREE and feature composition 9 for ANN and GMM (recall that these were the feature compositions that resulted in the lowest error rate for the respective classifiers in section 9.1.3).

As seen from table 9.8 the only classifier that have been boosted in performance by the window predictability is the GMM. The analysis of error types made by the classifier has been done until now through the confusion matrix. The disadvantage of the confusion matrix is that it does not provide any information on the time-related errors. With this is meant that it is not known if the errors made by the classifiers are randomly placed throughout the classified data set

Classifier	50 ms	150 ms	Sub - windows
TREE	17 %	**14 %**	15 %
ANN	17 %	**14 %**	16 %
GMM	17 %	17 %	**15 %**

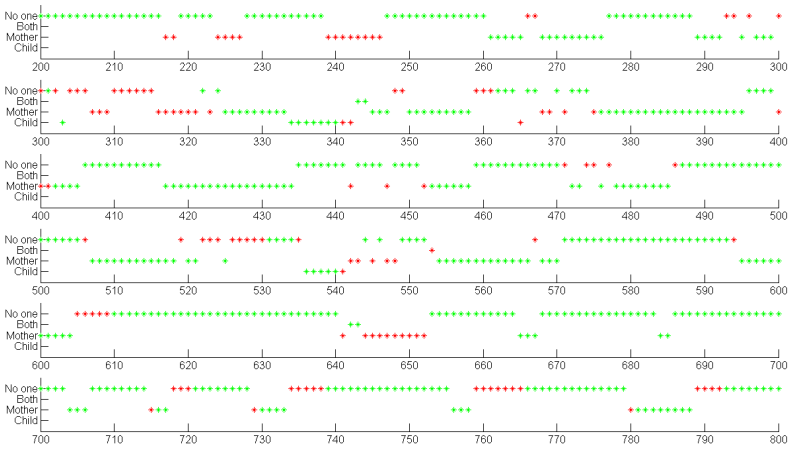
Table 9.8: The error rates for windows of 50 ms and 150 ms are shown. The column *Sub - windows* presents the results obtained when using the 50 ms windows to predict the outcome of the 150 ms window. The feature composition 8 is used for TREE and feature composition 9 is used for ANN and GMM.

or if for example 5-10 errors in a row occurs. If the last situation is valid, then the majority of 3 consecutive 50 ms window would not contribute in a boosting of the performance, but on the other hand, if the first statement is valid, then the majority voting of the 3 consecutive window will act as a kind of smoothing and thereby possibly contribute in a boosting of the performance. It is therefore interesting to investigate the class labels as a function of time for GMM where the result in table 9.8 showed a performance boosting (decrease in error rate of 2 %) and for example the ANN where the result in table 9.8 had the opposite effect on the performance. Figure 9.11 shows the estimated class labels for 600 consecutive observations when using ANN and GMM respectively with the window size 50 ms. The misclassified observations are presented by red dots whereas the correct classified observations are presented with green dots.

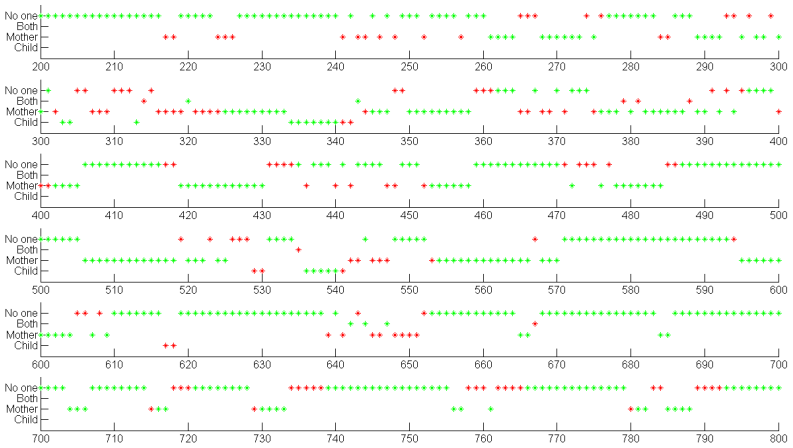
The estimated class labels for ANN in figure 9.11(a) show that when ANN makes one misclassification, it is often directly followed by 3 or more extra misclassifications. Comparing this to the misclassifications of the GMM, as seen in figure 9.11(b), the GMM makes a lot of single misclassification, i.e. misclassifications that are not followed directly by new misclassifications. This explains the difference in performance for the two classifiers, GMM and ANN.
For the TREE classifier in table 9.8, no boosting was found either. The estimated class labels for 600 observations are shown in appendix D.4 and the same argumentation as for the ANN can be used to clarify why no boosting in performance is obtained for the TREE when using sub windows.

9.1.5 Combining Channels

As mentioned in 5.5, the outcome of the classifiers for the two respective channels can be combined in a majority voting with the purpose of boosting the classifier performance. In this way the information from both channels, which to some



(a)



(b)

Figure 9.11: (a) The estimated class labels for 600 observations when using ANN at 50 ms. Red indicates a misclassified observation and green a correct classified observation. (b) The estimated class labels for 600 observations when using GMM at 50 ms.

degree is assumed identical, is combined. Section 5.5 gave the details on the approach used for this. From figure 5.12 it was shown that if the class label for the same 150 ms window of the two channels was not the same, the majority of the 3 corresponding 50 ms windows for each channel (6 in total) determined the class label of the 150 ms window. The test is carried out for each of the three classifiers, TREE, ANN and GMM. Feature composition 8 is used for TREE, table 9.6 since this was the feature composition that gave the best result for the TREE classifier, see figure 9.10 whereas feature composition 9 is used for ANN and GMM with the same argumentation.
In table 9.9 the error rates for the TREE classifiers at 50 ms and 150 ms window are shown for each of the two channels individually.

Window	**Channel 1**	**Channel 2**
50 ms	18 %	17%
150 ms	16 %	**14 %**

Table 9.9: The error rates for the two windows 50 ms and 150 ms for each of the two channels. The TREE classifier has been used where the combined error rate gives 16 % .

The table shows the error rates of the classifiers for the two channels and the two window sizes used in combining the channels. What can be observed from the table is that channel 2 has the lowest error rate of the two. Channel 1 represents the child's microphone and this channel is quite noisy due to the child's many movements. The higher error rate of channel 1 might therefore stem from this noise.
Combining the channels, in the way shown in figure 5.12, results in an error rate of **16 %**. When comparing to the error rates given in table 9.9, no gain in the performance was obtained by the combination of channels, which is assumed to be because of the difference between the error rates of channel 1 and 2, as seen in table 9.9. Also the larger error rate for the 50 ms window classifiers should be taken into considerations, because these windows are used in the majority voting. The results for the ANN and GMM can be seen in appendix D.5 where the results for the ANN showed, as with the TREE, that no gain in performance was obtained when combining the channels. When combining the channels for the GMM on the other hand a gain of 1 % was observed.

9.1.6 Summary

Many tests have been conducted for the speaker identification task with the purpose of identifying the best possible classifier. The original setting consisted of the five classifiers GMM, KNN, TREE, MNR and ANN, and the six window sizes 10 ms, 50 ms, 100 ms, 150 ms, 200 ms and 250 ms. The feature vector consisted of MFCC, delta-MFCC, delta-delta-MFCC, energy, zcr and the cross-correlation between the two channels. The data set consisted of 15 dyads, from where 14 was used as training set and 1 as test set, i.e. the hold-out method was applied.
With the first test of window sizes, all five classifiers were optimized according to their model parameters. The window sizes of 100 ms, 150 ms and 200 ms all showed good results, and the window of 150 ms was chosen. All following tests were therefore run using this particular window size. Furthermore the classifiers KNN and MNR were excluded from further testing because they showed poor results. The TREE and ANN performed the best, but because of the overweight of GMM classifiers for speaker identification in the literature, this classifier was kept for further testing as well.
Following the performance optimization for the individual classifiers using all the features, was a discussion on the types of errors that the two best classifiers made. This also included a discussion on the reliability of the manual codings made at Babylab, that had error rates in the range of 7 % to 31 %. These findings raised the question of what the true labels are and thereby making it more easily acceptable with a classifier error rate of 14 - 15 %, which was the lowest error rates obtained so far.
The following tests conducted was for finding the optimal feature composition. Several combinations were tested and the best ones for the three remaining classifiers; TREE, ANN and GMM, were actually the most simple features. For GMM and ANN the best error rate was obtained with the energy and cross-correlation as features, whereas the zero crossings were included as well for the TREE. An important conclusion that was made following these results was that the cross-correlation features contributed with a large amount of information to the classifications.
To take advantage of the many classifiers all of different feature compositions, it was tested if the performance could be boosted by combining several of these through a majority voting. For TREE one of these combinations consisted of composition 6 (16 %), 8 (14 %) and 9 (14 %), based on their individual types of errors, and the combined error rate was obtained to be 14 %. This meant that no performance improvement was obtained, which was the general image throughout the combinations.

With the many window sizes available, the question of predictability arose: was

it possible to predict the 150 ms windows from the three corresponding sub-windows more accurately than by classifying with the 150 ms windows? The test showed that the time-aspect of the classifications could explain the contradicting results. Improvement was seen when using sub-windows for prediction only for the GMM. Here the misclassifications were observed to be spread out in time, whereas the misclassifications were more grouped for ANN and TREE, where a deterioration of the error rates was observed when using sub-windows.

Finally it was tested if it would improve the classification performance to combine the classified labels of the mother's and the child's channels. If the time-corresponding windows for the two channels differed in label, the three sub-windows of 50 ms each for both channels were used in a majority vote to estimate the label. A slight worsening in the result was observed when using the TREE: 14 % error rate for the 150 ms window for the mother's channel alone and 16 % for the 150 ms window for the combined channels. The assumption for this outcome was that the child's microphone was more filled with noise, and thereby more prone to errors, as well as the fact that the classifiers of 50 ms had higher error rates than those of 150 ms. The same conclusion was drawn when using the ANN but for GMM a gain in performance of 1 % was obtained when combining the channels.

The final constellation of the three classifiers for the speaker identification task is thereby as shown table 9.10.
It should be noticed at this point that the best performance of 14 % was actually obtained in the very first test; the test of window size. The remaining tests only showed a gain in performance of the GMM classifier. A reason for this could be that the GMM as a starting point showed a higher error rate than that of the TREE and the ANN. As mentioned, the manually annotated labels are used as the ground truth. Due to already mentioned issues concerning these labels, the reason why no further gain in performance was obtained could possible be explained by these issues. It is therefore assumed that the performance of 14 % is just as good as the manual annotations carried out at Babylab.

9.2 Emotion Classification

The results of the emotion classification of the child's utterances are presented in this section. The first results are based on the HMM parameter optimization, which will be discussed in section 9.2.1. The model's performance evaluated on the basis of the confusion matrix will be discussed in the subsequent section,

	GMM	TREE	ANN
Windows	150 ms	150 ms	150 ms
Parameters	$K = 4$	$\kappa = 1300$	$H = 11$
Features	energy, cross-corr	energy, cross-corr, zcr	energy, cross-corr
Sub-windows	yes	no	no
Combine channels	yes	no	no
Error rate	15 %	14 %	14 %

Table 9.10: The three best classifiers for the speaker identification problem of this thesis. The rows *Sub-windows* and *Combine channels* are marked *yes* if this execution resulted in a gain of the performance and vice versa if set to *no*, i.e. no gain in performance was observed. The error rate shown in the last row is the best obtained error rate for the given classifiers.

9.2.2. Here the human error rate, i.e. the error rate between two manual codings performed at Babylab, is included and discussed as well. The last section, 9.2.3 focus on obtaining the feature combination that provides the lowest error rate.

9.2.1 Parameter Estimation

The size of the codebook of the HMM, corresponding to the total number of clusters K along with the number of states S must be chosen prior to the classification using the two HMMs (one for *protest* and one *for no protest*). The method chosen here is to fix S and vary K to obtain the best codebook size. With the optimal K fixed, S is varied and optimized.
As explained in section 6.3, the distortion measure can be used to determine K, where the wish is to minimize J. In [51] the distortion measure was analysed, where it was found that when the number of clusters exceed 32, the decrease of J per increase of K is limited. This fact is shown for the data in this thesis in figure 9.12.

The figure clearly illustrates the claim of [51] that after around a codebook size of around 30, the reduction in distortion is only very small.

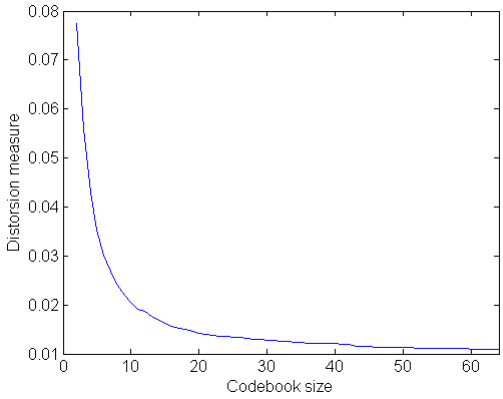


Figure 9.12: The distortion measure for a codebook size varied from 2:64.

With this knowledge at hand, it is of interest to investigate whether a smaller number of K than 32 is able to describe the emotional data set better, in spite of the higher distortion measure. This is obtained by evaluating the mean error rate across 15 replicates of the entire set-up for each K. This means that for each $K = [2 : 32]$ the following process is performed 15 times: all feature vectors of the training set are quantized into a common codebook, the HMM for each emotion is trained and each sequence in the test set is classified as being one of the two emotions. The number of states is here fixed at $S = 5$, which was randomly chosen.
The mean error rates of the emotion classification task with the number of states fixed at 5, while varying the codebook size as described above, are illustrated in figure 9.13. In the figure the red lines indicate the standard deviation of the mean.
From the figure it is clear that the mean error rate varies quite a lot depending on the codebook size. The lowest error rates are for $K = 10$ and $K = 29$, which are both 12 %. It could be argued that the distortion measure should be included in this discussion and thereby that $K = 29$ should be used because of the reduction in distortion with an increase from $K = 10$ to $K = 29$, recall figure 9.12. Since the mean error rates show equal results for the two codebook sizes, the higher distortion measure is not an issue, and a codebook size of 10 is therefore chosen to represent all the features through vector quantization because of the simpler feature representation.

The number of states to include in the HMM can be set based on various thoughts [51]. The first option is to set the number of states according to the phonemes in the word to be characterised. Since the emotions worked with

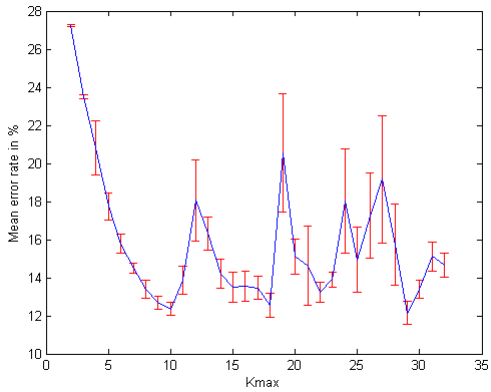


Figure 9.13: The mean error rate estimated across 15 replicates for each codebook size of $K = [2 : 32]$. S is held fixed at 5 states. The red vertical lines indicate the standard deviation of the mean.

in this thesis are not expressed through words, but rather through sounds, this option is probably not appropriate. A second option is to have as many states as segments in each sequence, which would be 10 in this case. Although a rational choice, the approach chosen is a third option - to estimate the number of states based on the mean error rate of 15 replicates. Furthermore, it is chosen to apply the same number of states to both HMMs. This means that just one S should be estimated instead of two, which could be a simplification of the problem that is not truly correct. Despite the interesting aspect of investigating the number of states for the two classes individually, the limited time prospect of this thesis, means that it is left for future work.
The optimal K, found as described above, is fixed while the number of states is varied. The optimal number of states is as K investigated for values up to 32. Thus the total interval is $S = [1 : 32]$.
With $K = 10$ fixed, the number of states is estimated as well from the mean error rates based on 15 replicates each. Figure 9.14 illustrate the results. In the figure it can be observed that the lowest mean error rate is obtained at $S = 11$ with a mean error rate of 11 %. This is therefore chosen as the number of states to be applied in the two HMMs. Likewise it is observed that the difference in mean error rate varies only very little when the number of states exceed 5.

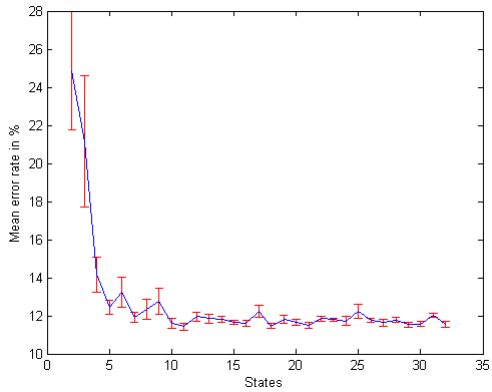


Figure 9.14: The mean error rate estimated across 15 replicates for each number of states of $S = [1 : 32]$. K is fixed at a codebook size of 10. The red vertical lines indicate the standard deviation of the mean.

9.2.2 Confusion Matrix

With the combination of a codebook size of 10 and a number of states of 11, the types of errors that the HMM makes can be investigated. For this the confusion matrix is used, which is shown in figure 9.15. It should be noted that because the optimal parameters of the HMM are found by replicating the classifications 15 times, the confusion matrix representing the lowest error rate of the 15 replicates is illustrated.

The confusion matrix illustrates that out of 124 observations belonging to the class *no protest* in total, only 1 of these is misclassified as *protest*, corresponding to less than 1 % of the total number of *no protest*s. On the other hand, more than 20 % of the class *protest* are misclassified as *no protest*.
A possible explanation is that the misclassifications of the class *protest* are the outcome of misinterpretations at Babylab for when the child is in protest or not. Since the task of interpreting an infant's sounds is very subjective, it is possible that where the child is actually not protesting it is classified by the coders at Babylab as if the infant was protesting.
Following the discussion on the true labels from Babylab, the reliability of the codings carried out at Babylab is tested. This has only been possible to investigate for one dyad, since this is the only recoding provided by Babylab where reliability of the emotional state of the child has been carried out. The human

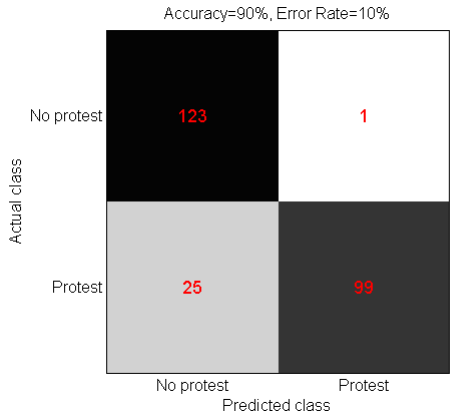


Figure 9.15: The confusion matrix with the lowest error rate of the 15 replicates for the best combination of codebook size and number of states, $K = 10$ and $S = 11$.

confusion matrix illustrating the errors made between the two coders for dyad 018 is shown in figure 9.16. The error rate of the human labelling between the

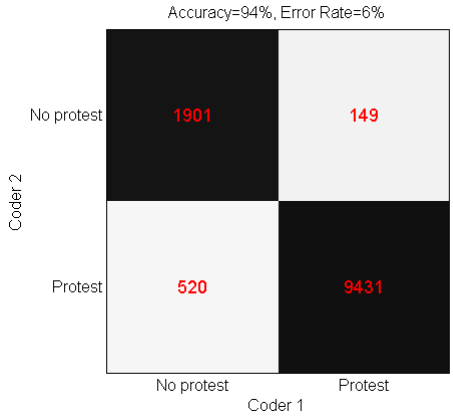


Figure 9.16: The human confusion matrix between two coders at Babylab shown for dyad 018.

two coders is seen to be 6 %. Since this is the only reliability coding available, it is difficult to determine if this is the general image. Including the reliability testings performed for the speaker identification task, section 9.1.2, it was seen that the error rates of these varied from 7 % to 31%. A reliable assumption is that the manual annotation task of determining the speaker, is more objective

than that of determining the emotion. Since the emotion recognition requires the coder to interpret on the utterances, it could be argued that the general reliability between coders in this annotation task thereby is worse or as a minimum the same as for the speaker identification task.

9.2.3 Test of Features

To optimize the classifier, different feature compositions are tested. Some of the included features, see table 6.3 for a recollection of these, may not be appropriate for the emotion classification task considered in this thesis. Table 9.11 illustrate the tested combinations. Note that feature composition 0 corresponds to the full feature vector.
Figure 9.17 visualizes the error rates obtained for the 10 tested feature compositions. The x-axis in the figure corresponds to the combinations shown in table 9.11.

Composition	Feature Composition
0	MFCC, delta-MFCC, energy, zcr
1	MFCC, energy, zcr
2	MFCC, energy
3	MFCC, zcr
4	MFCC
5	energy, zcr
6	MFCC, delta-MFCC, energy
7	MFCC, delta-MFCC
8	delta-MFCC, energy
9	delta-MFCC, energy, zcr

Table 9.11: The tested feature compositions for the emotion recognition task.

From the figure it can be observed that feature compositions 3, 4 and 7 are by far the worst. In common for these three compositions is that the energy is not included, which it is in all other feature combinations, see table 9.11. The figure therefore clearly shows that the energy feature is of the uttermost importance to the emotion classification.

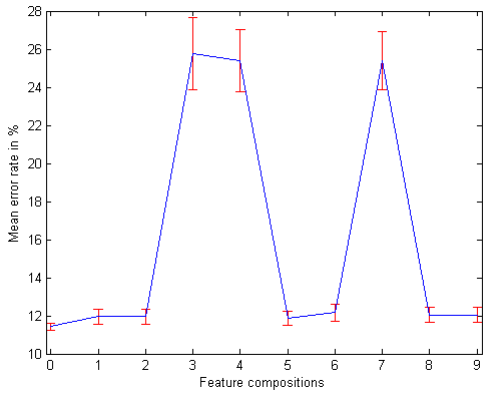


Figure 9.17: The obtained error rates as a function of the tested feature compositions for the emotion recognition task. Each number refers to a composition which can be seen in table 9.11.

In the figure it can also be seen that the best feature combination, although not convincingly, is actually the original one, with MFCC, delta-MFCC, energy and zcr included. The best error rate is therefore still the 11 % from before.
To follow the procedure of speaker identification, section 9.1.3, the combining of several classifiers of different feature compositions is tested here as well. The thought is, as has already been explained, that if these classifiers make errors that are independent of each other, the combination of these, could boost the performance.
In table 9.12 the classifiers of different feature compositions (three classifiers at a time) have been combined through majority votings, to obtain a common error rate. By looking at the confusion matrix of the best feature composition, figure 9.15, as well as the confusion matrices for the remaining 8 feature compositions, shown in appendix E, the same pattern is observed for all of them. The class of *no protest* is for all 9 classifiers the one of the two classes with the least errors. Therefore, the choices of combinations in table 9.12 are exclusively based on the observed error rates.
For each feature composition in table 9.11 the classifications were as mentioned run 15 times each, providing 15 class labels. For the combining of classifiers, a random replicate was therefore chosen to make the test.

From the table it is clear that the improvements in combining classifiers are limited. In fact, the only combination where an actual decrease is seen for the error rate is for the last tested combination. These all performed very good individually (12 % for all of them) and combining them gave an error rate of

Composition A	Composition B	Composition C	Combined
0 (11 %)	1 (12 %)	5 (12 %)	**11** %
3 (26 %)	4 (25 %)	7 (25 %)	27 %
0 (11 %)	3 (26 %)	5 (12 %)	**11** %
4 (25 %)	5 (12 %)	7 (25 %)	22 %
7 (25 %)	8 (12 %)	9 (12 %)	12 %
5 (12 %)	6 (12 %)	8 (12 %)	**11 %**

Table 9.12: The error-rates obtained when combining 3 HMM classifiers with different feature compositions. The individually obtained error rates are shown as well.

11 %. This improvement in spite, the best individual feature composition also provided an error rate of this size. It must therefore be concluded that the combination of classifiers of different feature compositions do not outdo that of the individual classifications.

9.2.4 Summary

The emotion recognition task carried out in this thesis have showed promising results based on the sound features alone. Tests were carried out to obtain the best fitted HMM, for the purpose of classifying the emotional states *no protest* and *protest*.
The parameters of the model were first to be estimated. The relatively small codebook size of $K = 10$ turned out to be very representative for the data applied in the thesis, in spite of the larger distortion measure. The number of states were tested as well, and the optimal number was $S = 11$.
With these parameters chosen, the resulting confusion matrix was discussed, where the reliability of the true labelling was included in this. Based on the argument that the coding of the emotional states of the child is a somewhat subjective task, the misclassifications can therefore, at least partly, be due to human labelling errors.

Different feature compositions were tested to investigate if an improvement of the error rate was possible. With this survey, it became clear that the energy in the signal segment was of great importance to the classifier. Furthermore it was

determined that the feature composition that gave the lowest error rate was in fact the primary one, i.e. the composition with all the features included.
Finally it was tested if a combining of classifiers of different feature compositions would improve the classifier performance. Here only very small decreases in error rates were observed, but not lower than the already lowest obtained error rate of 11 %.
The final constellation of the emotion recognition classifier is thereby the following, shown in table 9.13.

	HMM
Parameters	$K = 10$, $S = 11$
Features	MFCC, delta-MFCC, energy, zcr
Combine classifiers	no
Error rate	11 %

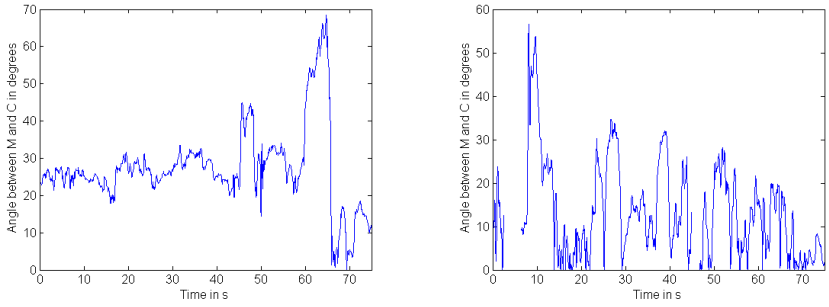
Table 9.13: The best HMM for the emotion recognition task of this thesis. The row *Combine classifiers* is marked *no* because the combinations of the classifiers of different feature compositions, provided no error rates lower than already obtained. The error rate shown in the last row is the best obtained error rate in the speech emotion recognition.

9.3 Motion Capture Annotations

As mentioned in section 7, the annotations made from the motion capture modality includes the head position of the child, the distance between the faces of the mother and the child and finally the child's physical energy level. All these annotations have been executed on a frame-by-frame basis for each dyad. The following three sections presents the profiles obtained using the three automatic methods for the mocap annotations.

9.3.1 Child's Head Position

The angular profile between the mother and the child, as a representation of the child's head orientation, is shown for dyad 002 in figure 9.18(a), and for dyad 010 in figure 9.18(b) for the time interval 0-75 seconds.

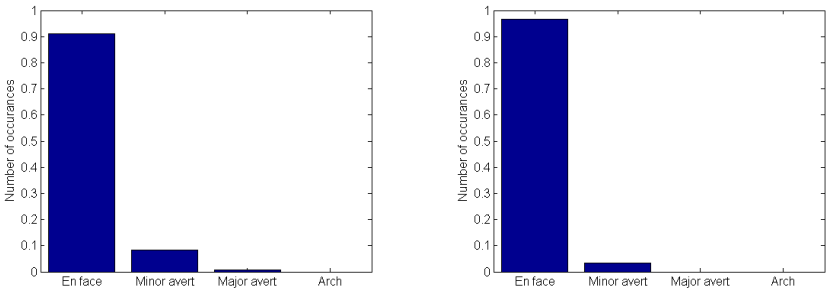


(a) (b)

Figure 9.18: Angular profiles for the first 75 seconds of (a) dyad 002, and (b) dyad 010. It is seen that for dyad 010 there are several missing angles around 5 seconds, reflecting the not-identified mocap markers in Qualisys. Note that the y-axis is not the same in the two figures.

What is seen from figure 9.18, showing the angle between the mother and the child, is that in the interval 0-75 seconds, the angle is equal to 0 only once in (a), at around 70 seconds, and several times in (b). The zero angle, represents the time instants where the mother and the child are facing each other.
To relate the raw angles to the scheme of categories introduced in table 7.1, the following figure 9.19 shows the above illustrated angle profiles in a histogram where each bar represents one of the four categories.

From the figures it is clear that by far the most of the angles belong to the *En face* category. This represents the angles of 0-30 degrees. In only few of the frames the child's head is averted more than 30 degrees from the mother's which is clear from both 9.19(a) and 9.19(b). This is also the general image when analysing the angle profiles.
As explained in chapter 7, the manual codings of the angles determined at Babylab are carried out using a given reference point in the room and not with



(a) (b)

Figure 9.19: Histogram representing the distribution of the 36000 (frame-by-frame) calculated angles between the mother and child with respect to the four categories, *En face*, *Minor avert*, *Major avert* and *arch*. (a) illustrate that of dyad 002, and (b) that of dyad 010. Note that the histogram has been normalized.

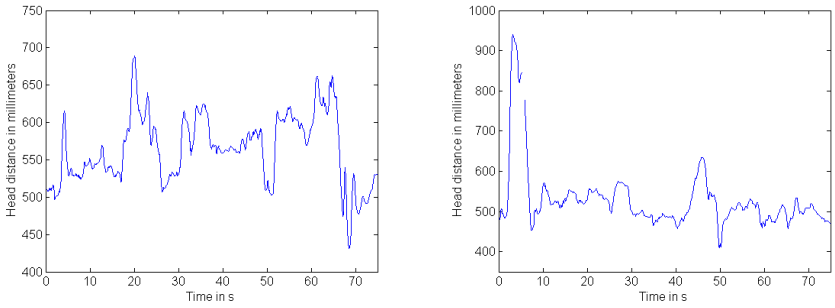
respect to the position of the mother as the approach used in this thesis. It has therefore not been possible to validate the angles obtained automatically with the manually coded angles.

9.3.2 Distance Between Faces

The distance between the mother's and child's faces for dyad 002 and 010 are illustrated in figure 9.20 for 0-75 seconds.

As for the angle profile, it can be observed for dyad 010, in 9.20(b), that some not-identified markers are present around 5 seconds.

As mentioned in the introduction of chapter 7 the distance between the faces of the mother and the child is calculated in Excel by the psychologists at Babylab. No validation of the distance profiles obtained in this thesis with respect to the ones calculated by Babylab has been carried out. The reason for this is that the distance profiles are unique, meaning that they only have one solution. The only difference is the actual distance which will differ from the ones in this thesis to those of Babylab, because they at Babylab use the back markers MheadB and CheadB, see figure 3.1, for the calculations. In this study the MheadM and CheadM are used, see figure 7.2, since this is thought of as a good representation of the head positions. Despite this difference, the relation between the distance of each frame remains constant.



(a) (b)

Figure 9.20: Distance between the head of the mother and child for the first 75 seconds of (a) dyad 002, and (b) dyad 010. It is seen that for dyad 010 there are several missing distances around 5 seconds, reflecting the not-identified mocap markers in Qualisys. Note again the difference of the y-axis between the two figures.

9.3.3 Child's Physical Energy Level

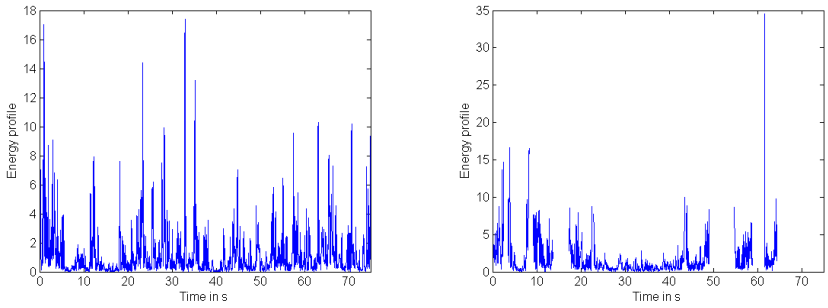
Figure 9.21 shows the distance profile for dyads 002 and 010 for 0-75 seconds, based on the method of calculation shown in section 7.3.

It can be observed for dyad 010, in 9.20(b), that many not-identified markers are present around 15 seconds, from 50-55 seconds, around 60 seconds and again from 65 seconds and up to 75 seconds.

As was the case for the calculations of the head distance above, it was mentioned in the introduction of chapter 7 that the child's physical energy level is calculated in Excel by the psychologists at Babylab. The energy profiles obtained in this thesis has not been validated with respect to the ones calculated by Babylab with the same reasoning as above.

9.3.4 Summary

The motion capture annotations of the head position of the child, the distance between the faces of the mother and the child and finally the child's physical



(a) (b)

Figure 9.21: The child's physical energy level for the first 75 seconds of (a) dyad 002, and (b) dyad 010. It is seen that for dyad 010 there are several missing distances around 15 seconds, from 50-55 seconds, around 60 seconds and again from 65 seconds and up to 75 seconds. These time intervals reflect the not-identified mocap markers in Qualisys. Note again the difference of the y-axis between the two figures. This reflects the difference in physical energy level of the child in the two recording sessions.

energy level have all been calculated and illustrated in this minor section. Despite their own calculation methods, the psychologists at Babylab have shown interest in the annotation methods obtained in this thesis, which could be because of the more automated, and thereby less time-consuming, approach used here.
The child's head position is calculated with respect to the mother, as explained in 7.1, as opposed to a reference point in the room as in the approach of Babylab. Therefore it has not seemed appropriate to compare the resulting annotations of the two methods.
The resulting distance profiles as well as those of the child's physical energy level have not been validated either, due to the unique solutions of these when calculating them from the marker coordinates.

9.4 Combing Modalities

The results for the speaker identification and the emotion recognition tasks so far, has exclusively been obtained using features extracted from the sound modality.

In order of being able to combine the features from sound and motion capture in the tasks of speaker identification and emotion recognition, certain modifications must be carried out. For the speaker identification problem, the features from motion capture are to be fitted to the window sizes used in the classification, that is the window sizes 10 ms, 50 ms, 100 ms, 150 ms, 200 ms and 250 ms. The approach to this is to take the mean of the angles from the frames that corresponds to the particular window size. This approach is also applied for the distance features, whereas features for the child's physical energy level is carried out by summing over the frames, to obtain the covered distance in each window. The motion capture files has, as mentioned in chapter 3, a sampling frequency of 60 Hz, i.e. a sample every 16.667 ms. The corresponding number of frames for each of the six window sizes can be seen in table 9.14.

Window	Frames
10 ms	0.6
50 ms	3
100 ms	6
150 ms	9
200 ms	12
250 ms	15

Table 9.14: The window sizes with the corresponding number of frames in mocap.

As can be seen in table 9.14, the fitting of the mocap features to the window size of 10 ms gives rise to a problem since 0.6 is not an integer. In the speaker identification this is not a problem, since the 150 ms window showed to be the best.
For the emotional classification segments of 10 ms are used to predict the outcome of the HMM at 100 ms. To obtain the mocap features for this window size, interpolation is therefore carried out. The interpolation is performed such that 3 mocap frames are used to obtain the values of five 10 ms segments (3 mocap frames equals 50 ms, table 9.14). The first of the five segments is set to the value of the first of the three frames. The second segment is set to the mean value of the first and the second frame. The third segment is set to the value of the second frame and so on. The procedure can be seen in table 9.15, where the value of the frames in this case refers to either angle, head distance or physical energy in that given frame.

In the following the combining of sound features and motion capture features

Window	Frames
First 10 ms	The value of the first frame
Second 10 ms	The mean of the first and second frame
Third 10 ms	The value of the second frame
Fourth 10 ms	The mean of the second and third frame
Fifth 10 ms	The value of the third frame

Table 9.15: The interpolation procedure in order to fit the mocap features to the 10 ms segments as used in the emotion recognition.

for the tasks of speaker identification, section 9.4.1, and emotion recognition, section 9.4.2, is conducted and the resulting classifiers are discussed.

9.4.1 Speaker Identification

With respect to the speaker identification task it is in this section investigated if features from the data modality motion capture can contribute in the task of identifying the speaker, see also the description in section 8.1 about combining modalities.
To be able to use the features from the motion capture data, the synchronization between the sound modality and the motion capture must be found, which is carried out in chapter 4. The delay is only known for 10 full data sets, and the motion capture feature can therefore only be included for these 10 dyads to ensure synchronization. The different issues regarding the number of available dyads are discussed in chapter 4 about synchronization.
Two models must be generated from the dyads where the synchronization is known: one that is based exclusively on the sound features and one where the features from motion capture are included as well. This makes it possible to obtain an expression of the performance of the motion capture features.
It is decided to run this test for the TREE classifier only. The new TREE is fitted to nine dyads, and the remaining one is used as the test set, as with the model generated from 14 dyads.
For the model with sound-based features only, the same parameters as found for the larger data set is chosen, i.e. $\kappa = 1300$ and a window of 150 ms. The features correspond to the feature composition 8 from table 9.6, that showed the best result for the TREE with the larger data set. These are energy, cross-correlation and zcr. The resulting error rate for this particular TREE classifier

is shown in table 9.16.

Feature Composition	Error rate
Energy, zcr, correlation	**14 %**

Table 9.16: The error rates of the TREE generated from 9 dyads based exclusively on features from the sound modality.

The same model is then generated only now with the features from motion capture included as well, in different combinations. The results of these tests are shown in table 9.17.
As seen from table 9.17 the best result obtained is 14 % when including the

Feature Composition	Error rate
Energy, zcr, correlation, mocap-energy, mocap-distance, mocap-angle	16 %
Energy, zcr, correlation, mocap-energy	16 %
Energy, zcr, correlation, mocap-angle	15 %
Energy, zcr, correlation, mocap-dist	**14 %**
Energy, zcr, correlation, mocap-dist, mocap-angle	15 %
Energy, zcr, correlation, mocap-energy, mocap-angle	16 %
Energy, zcr, correlation, mocap-energy, mocap-dist	16 %

Table 9.17: The error rates for the TREE generated from 9 dyads based on features from both sound and motion caption. The features with prefix mocap are from the motion capture modality whereas the ones with no prefix are the features from the sound modality.

motions capture feature. When comparing with table 9.16 where the sound features exclusively have been used, the error rate obtained also showed a minimum of 14 %. The inclusion of the features from motion capture therefore appears to have no effect on the performance of the speaker identification problem.

9.4.2 Emotion Recognition

For the emotion recognition task it is likewise investigated if the motion capture features affects the performance of the classifier in a positively way.
The number of dyads for which the synchronization difference has been extracted as well as where the manual annotations of the child's emotional state have been executed confines to 6 dyads. This means that 5 are used to fit the HMM and one as test set.

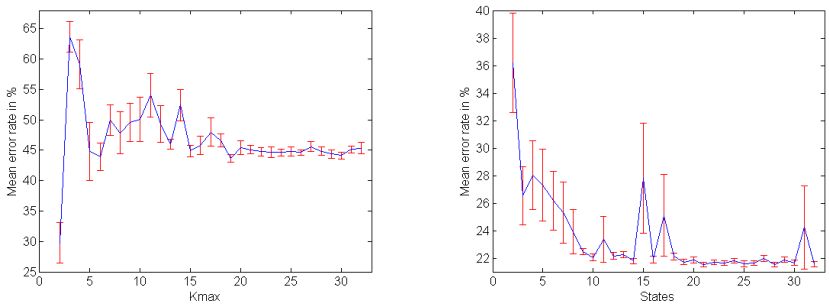
9.4.2.1 Parameter Optimization

After the synchronization has been performed, two models are again constructed: one for the sound-based features only, referred to as the sound-based HMM, and one that combines the sound features with the motion capture features, the sound/mocap-based HMM. Since the data set is reduced by half from the original set-up, see section 6.1, it is decided to estimate the optimal codebook size and the number of states again for each model with full feature vectors. Here the same approach is used as for the original model, which is to fix the number of states at $S = 5$ and estimate K and then fix K at this value whilst varying S to determine the optimal number of this.
The feature vectors of the two models can be seen in table 9.18.
For the sound-based HMM, the estimation of optimal K and S is shown in figures 9.22(a) and 9.22(b), respectively. It is to be noted again that the choice of number of states is based on the best codebook size.

Model	Features
Sound-based HMM	MFCC, delta-MFCC, energy, zcr
sound/mocap-based HMM	MFCC, delta-MFCC, energy, zcr, mocap-energy, mocap-distance, mocap-angle

Table 9.18: The feature compositions for the two models that exclude and include mocap features, respectively.

Figure 9.22(a) illustrates that the optimal codebook size clearly is $K = 2$. For $K = 3$ the mean error rate increases heavily, whereupon it stabilizes around 45 % for $K > 3$. This course of error rate as a function of codebook size, is very different from that of the full data set from section 9.2. This must be due to the much smaller data set size, which, as can be seen, has a large impact on the

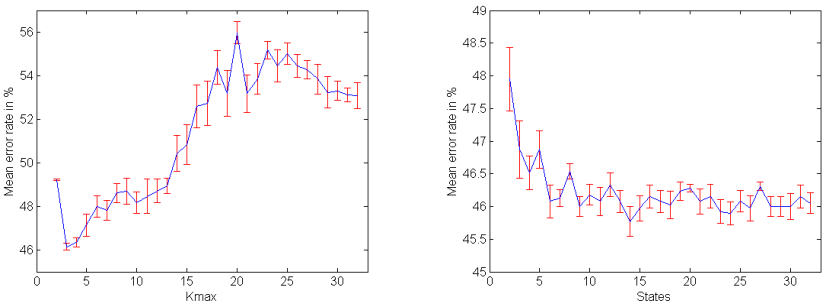


(a) (b)

Figure 9.22: The choice of parameters for the sound-based HMM. (a) shows the estimation of the size of the codebook, and (b) the estimation of number of states. The y-axis on both figures are the obtained mean error rates of 15 replications. The red vertical lines indicate the standard deviation of the mean.

parameter estimation.
With $K = 2$ fixed, S is varied to find the optimal number of states. From figure 9.22(b) it is observed that for increasing S the error rate stabilizes around 22 %. The first time the error rate reaches 22 % is when the number of states S equals 10, where also a small standard deviation of the mean can be seen. $S = 10$ should therefore be chosen. Thus the optimal error rate obtained with this sound-based HMM on 5 dyads is therefore 22 %.
For the sound/mocap-based HMM, the estimation of K and S can be seen from the following figures, 9.23(a) and 9.23(b), respectively.

In figure 9.23(a) it can be observed that the best error rate is obtained for a codebook size of $K = 3$. The error rate here is 46 %, which is observed to increase heavily with increasing K. Again the surprisingly bad error rates must be caused by the much smaller data set as well as the inclusion of the motion capture features.
With this $K = 3$, the S is varied to extract the most optimal number of states. Figure 9.23(b) illustrates the stabilization of error rate with increasing S. The optimal number of states is observed to be $S = 14$. The best error rate for this feature combination is thereby 46 %.



(a) (b)

Figure 9.23: The choice of parameters for the sound/mocap-based HMM. (a) shows the estimation of the size of the codebook, and (b) the estimation of number of states. The y-axis on both figures are the obtained mean error rates of 15 replications. The red vertical lines indicate the standard deviation of the mean.

9.4.2.2 Test of Features

Since the optimal combination of sound-based features was tested for the original set-up with full data set, this is assumed to be valid for the sound-based HMM investigated here for the smaller data set as well.
Different feature compositions are, on the other hand, tested for the sound/mocap-based HMM. The full set-up of the sound-based features are included in all compositions since this composition showed the best performance in the model for the larger data set. The feature combinations for the mocap-based features are on the other hand varied. The following table, 9.19, illustrates the compositions of features. It can be observed in the table that the optimal feature composition is found using the mocap-energy feature only in combination with the sound-based features. Although the best, the error rate obtained with the same data set, but for sound-based features only, was observed to reach its minimum at an error rate of 22 %. From this it must be concluded that the motion capture features are deteriorating for the classifier of the emotion recognition task. If more synchronized files were available, and thereby a larger data set was at hand, it is possible that the inclusion of motion capture features could have a positive effect on the classifier's performance. Or at least be indifferent to the classification, as was the case in the previous section on including mocap features in the speaker identification task.

Feature Composition	Error rate
MFCC, delta-MFCC, energy, zcr, mocap-energy, mocap-distance, mocap-angle	46 %
MFCC, delta-MFCC, energy, zcr, mocap-energy, mocap-distance	46 %
MFCC, delta-MFCC, energy, zcr, mocap-energy, mocap-angle	65 %
MFCC, delta-MFCC, energy, zcr, mocap-distance, mocap-angle	46 %
MFCC, delta-MFCC, energy, zcr, mocap-energy	**36 %**
MFCC, delta-MFCC, energy, zcr, mocap-distance	46 %
MFCC, delta-MFCC, energy, zcr, mocap-angle	66 %

Table 9.19: The error rates for the sound/mocap-based HMM from 5 dyads based on features from both sound and motion caption. The features with prefix mocap are from the motion capture modality whereas the ones with no prefix are the features from the sound modality.

9.4.3 Summary

To make use of the fact that three data modalities are available, it has been tested whether the inclusion of the three motion capture features, child's head position, head distance between mother and child and child's physical energy level, could improve the classifier performance.
In the speaker identification task it was observed that the combining of information from both sound and motion capture did not change the error rate obtained with the pure sound-based TREE classifier, neither in a positive or negative direction. The best error rate is therefore still 14 %.
In the emotion recognition task, on the other hand, the combination of sound-based features and mocap features showed a clear deterioration of the classifier performance. It was here argued that the reason could be the much smaller data set. If a larger data set was available, it would be interesting to see if a gain in performance could be obtained by including the mocap features. The best error rate obtained in the emotion recognition task is therefore 11 %.

Chapter 10

Conclusion and Perspectives

During the course of this thesis, automatic approaches for the re-labelling of Babylab's manually extracted labels, have been investigated. The results obtained in the sound-based tasks of speaker identification as well as emotion recognition showed very promising results.
For the speaker identification task different classifiers were tested and the lowest error rate obtained was 14 %. This was obtained for TREE and ANN with only the simple features of energy, cross-correlation and zero crossings included. Despite the many initiatives for performance boosting, including the incorporation of features from mocap, the results remained steady at 14 % as the lowest, implying that the human labelling perhaps were constraining to the improvement in performance.
The same scenario was observed for the emotion recognition task. The lowest averaged error rate obtained, of 11 %, was the outcome of the HMM with all features included. Combining of classifiers or incorporation of mocap features did not improve the error rate. Here, the human confusion was discussed as well as a probable candidate for the lack in performance improvement.

From these results, it should be beneficial for Babylab to incorporate the, in this thesis, developed methods into their future annotation tasks. Since the obtained error rates are seen to be within the acceptance area of the manual annotations and a massive workload reduction would be the output at Babylab,

the arguments for keeping the approach of conducting manual codings are few. If manual labellings were available for other dyads, the models could be expanded for the sake of generalizability, which could induce even more reliable classification results. Furthermore, the availability of several age groups (7, 10 and 13 months) could be exploited as well. First of all, it should be tested if the models at hand are capable of classifying these other age groups with the same performance rate as that of the 4 months. If the models were capable of this, the generalizability of the models would definitely be established.
If, on the other hand, the models were not applicable, new models should be fitted to the new data sets. I.e. models for speaker identification and emotion recognition, respectively, should be fitted to the training data for each of the available age groups on the basis of the manually annotated labels.
Another, very interesting, annotation that could be re-labelled automatically is the vocalization of the mother. If she is speaking or singing is of importance to the psychologists at Babylab, since their assumption is that the child's emotional state is more likely to change from negative to positive when the mother start singing.
This automatic labelling could be pursued by using the already applied classifiers and the already extracted sound features and is therefore one of the more easy approachable tasks for future work.

Facial expression recognition through the use of Active Appearance Models has been superficially examined for the purpose of acquiring a basic understanding of the possibilities within this area for the psychologists at Babylab. The manual annotations of the child's facial expressions made at Babylab are extremely time-consuming, which is why an automatic approach, here as well, would be beneficial.
The results of the small study on automatic facial expression recognition showed that especially the shape recognition seemed promising. It should be kept in mind that the training set used for this test is very small; only four images of the same child has been used to obtain these results.
It would be very interesting to investigate this area further. If more images were used and if several of the dyads were included, the hope is that a completely generalized model would be obtained that could be used to make the annotations at Babylab in the future.

The annotations at Babylab are used for the analysis of the interaction between mother and child. These analyses are based on connections and patterns between the annotations and between modalities. Amongst many, the onset of vocalizations of mother and child and the pattern between these onsets: for example, does the child begin speaking when the mother speaks or the other way around? This is thereby a sound-based analysis.

Also the pattern between the mother's leaning behaviour in relation to the child's head turning is of interest. Here, the idea is that if the mother leans forward the child avoids her by turning its head away. In this, the velocity of the mother's movement could be included. This would therefore be a mocap-based analysis. A relationship between the child's physical energy level and its vocalizations, as well as the facial expressions of the child and its emotional utterances, are interesting multimodal analyses that are possible to solve by machine learning.

The unique data provided by Babylab has the potential of providing the basis of numerous analyses within the area of mother-child interaction. For many of these analyses, the machine learning approach is very applicable, in that it provides generalizability and in an elegant manner is able to account for the large variety in practical data set.

Appendix A

Facial Expression Scheme

The following scheme shows the the categories of the manual facial expression annotations that are conducted by Babylab.

Code	Infant Facial Affect	Criteria/definition	Mouth Widen[1] (MW)	Mouth Open[2] (MO)
6PO	Positive (medium high/high positive)	Forehead smooth, cheeks raised, mouth corners drawn back and curved up in full display, mouth fully open	MW 2	MO 3 (4)
5LOPO	Low /medium positive	Forehead smooth, eyes open, mouth corners curved up, mouth open or closed	MW 1	MO 1 (2)
4NEUIN	Neutral-interest	Forehead smooth, eyes open, mouth relaxed open/closed, or slightly pursed (*let sammenknebne/spidsede*) Focus on mother, object or shows visuel explorstion	MW 0 (1)	MO 0 (1)
3NEUDIS	Neutral-disinterest	Forehead smooth, eyes open, mouth relaxed open/closed, or slightly pursed (*let sammenknebne/spidsede*) Does not focus on mother or object and does not show visual exploration	MW 0 (1)	MO 0 (1)
2MINE	Mild negative	Inner corners of eyebrows raised, eyes open or squinting (*sammenknebne*), mouth corners down (grimace), or lips squeezed tightly together ('line mouth') *NB. Hvis der kodes på baggrund af rynke i panden, må der ikke være tvivl om den er tilstede eller ej.*		MO 0 (1) [and/or frown]
1NE	Negative/high negative	Pre cry/cry-face (partial/full display) Eyebrows drawn together (classical frown), eyes squinting, mouth angry, open and squarish (*"firkantet vred mund"*)		MO 2 (3) [and/or frown]
0NC	Non-codeable	Face hidden more than 250 mile sec/7 frames **NB! UNDGÅ SÅ VIDT MULIGT AT BRUGE DENNE KATEGORI!**		

1) Two degreese of Mouth Widen *(dvs. at munden kan være meget eller lidt bred)*

MW1 = sideways lip stretch (without zygomaticus retraction)
MW2 = lip-corner raise (zygomaticus retraction)

2) Four degreese of Mouth Open
MO1 = lips slightly parted
MO2 = Mouth slightly open
MO3 = Mouth (medium) open
MO4 = Mouth fully open

Figure A.1: The scheme followed by Babylab in order to obtain the manual annotations on facial expressions.

Appendix B

Active Appearance Model

B.1 Information from Video

In section 8.2 it was outlined that the third data modality, video, carries information that can not be extracted from either sound or motion capture. An interesting aspect of the automatic annotation process evaluated in this thesis is to investigate if the facial expressions of the child can be extracted from the video by machine learning methods. This could automate the complex manual annotation task, for which the coding scheme is shown in appendix A, as well as support the results already found in the speaker identification and emotion recognition tasks.
To extract this information from the video modality, the believe is that the Active Appearance Model (AAM) can be applied, [44]. A brief study on the AAM and an expansion of this, the Elastic Appearance Model (EAM), has therefore been carried out in order to see if this model is capable of extracting the facial expressions of the child.

B.2 The Model

The AAM is a statistical model describing both the shape and the grey-level appearance of a given object of interest, in this case the child's face. In the following, the shape model and the appearance model is described separately, starting with the shape model.
The shape model is build up of a mesh that is believed to describe the shape of the object of interest. This mesh or shape is defined as the coordinates that build up the mesh, see B.1.

$$\mathbf{s} = (x_1, y_1, x_2, y_2, ..., x_v, y_v)^T \tag{B.1}$$

In (B.1), the subscript v defines the number of coordinates with which the mesh is build. In order for the model to be able to capture the changes in the shape of the object, the shape is allowed to vary linearly. This is incorporated in the model by describing the shape as a base shape plus a linear combination of a given number n of shape vectors, see B.2.

$$\mathbf{s} = \mathbf{s}_0 + \sum_{i=1}^{n} p_i \mathbf{s}_i \tag{B.2}$$

Here, the base shape s_0 represents the mean shape of a given number of training meshes, whereas the p_i represents the shape parameters belonging to the shape vectors s_i.
The particular model considered in this study is the EAM, which is an expansion of the AAM. The term elastic in the name EAM refers to the fact that the shape can vary not only linearly but also non-linearly. For this, the Riemann elasticity framework has been applied, which is capable of capturing complex deformations, see [22] for further details.
In order to obtain the n shape vectors, principal component analysis (PCA) is applied on the training meshes, [43].
PCA is a method to project data onto a principal subspace to maximize the variance of the projected data. Furthermore, PCA is usually applied to reduce the dimensionality of the data, such that the principal subspace has a lower dimension than the original data.
[13] shows that if the observations in the data, $\mathbf{x}_n$ where $n = 1, 2, ..., N$ and N is the total number of observations in the data set, is projected onto a vector $\mathbf{u}_1$ then the variance is represented by (B.3) where $\mathbf{S}$ is the data covariance matrix, $\mathbf{u}_1$ is an eigenvector of $\mathbf{S}$ with the corresponding eigenvalue λ_1.

$$\mathbf{u}_1^T \mathbf{S} \mathbf{u}_1 = \lambda_1 \tag{B.3}$$

Maximizing the variance in (B.3) corresponds to setting $\mathbf{u}_1$ equal to the eigenvector that belongs to the largest eigenvalue, λ_1 of $\mathbf{S}$.
To relate PCA to the shape model, the n shape vectors represents the eigenvectors belonging to the n largest eigenvectors of the training meshes.
The training meshes are labelled by hand on a given number of images. In this thesis only a small study on the EAM has been carried out as already mentioned and the training of the model is based on 4 images of the child's face.

The meshes are annotated by a 73-point mesh system.
The training images of the child's face only constitute a small section of the original images, corresponding to (70×70) pixels. The extracted images describing the child facial expressions belong to dyad 011, thus only one child is used in this study.
The appearance model is defined in a way similar to the shape model where the appearance is defined as a base appearance image $A_0(x)$ and a linear combination of m appearance images $A_i(x)$, as given in (B.4).

$$A(x) = A_0 + \sum_{i=1}^{m} \lambda_i A_i(x) \qquad \forall \mathbf{x} \in \mathbf{s}_0 \tag{B.4}$$

As indicated in (B.4), the image $A(x)$ is defined over the pixels in the mesh of the shape model $\mathbf{s}_0$. λ_i are the appearance parameters. Again $A_0(x)$ is set to be the mean image of the training images and the $A_i(x)$ to be the eigen-images corresponding to the m largest eigenvalues. [43].
Above, the two parts of the model, shape and appearance, was described separately. The AMM/EAM combines the two parts to one model. From (B.2) and (B.4) it is seen that the shape and the appearance can be obtained if the shape parameters $\mathbf{p} = (p_1, p_2, ..., p_n)^T$, as well as the appearance parameters $\lambda = (\lambda_1, \lambda_2, ..., \lambda_m)^T$ are given. The model is then created by warping the texture from the base mesh to the shape. With warping is meant that points are mapped to points without changing the grey levels.
The way the warping is carried out is by triangulating the mean shape $\mathbf{s}_0$, whereupon linear interpolation is carried out in order to fit the triangles in $\mathbf{s}$. [43] has visually illustrated this in a very expressive way, see figure B.1.
The warping is in the following represented as $\mathbf{W}(\mathbf{x}; \mathbf{p})$ and the model is rep-

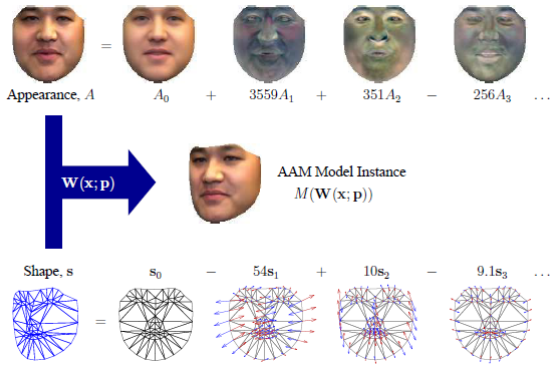


Figure B.1: Illustration of the warping from $\mathbf{s}$ to $\mathbf{s}_0$. Figure taken from [43].

resented as $M(\mathbf{W}(\mathbf{x}; \mathbf{p}))$. With the goal of fitting the model, $M(\mathbf{W}(\mathbf{x}; \mathbf{p}))$, to an input image, $I(x)$, in the best possible way, the approach is to minimize the error between the estimated $M(\mathbf{W}(\mathbf{x}; \mathbf{p}))$ and $I(x)$.
To ensure that the coordinates of the model and of the input image are the

same, the coordinate system for the model is used. The term that needs to be minimized is, for each given pixel, the difference between the appearance of the model in that pixel and the intensity of the input image in that pixel. This is formally described by B.5.

$$\sum_{\mathbf{x}\in\mathbf{s}_0} \Big(A_0(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i A_i(\mathbf{x}) - I(\mathbf{W}(\mathbf{x};\mathbf{p})) \Big)^2 \tag{B.5}$$

As indicated in the subscript of the sum, the minimization takes place in every pixel inside the mesh defining the shape. Where B.5 is minimized the optimal shape and appearance parameters $\mathbf{p}$ and λ are found. For a more detailed description of the Active Appearance Model see [22], [43].

B.3 Results

The results of the AAM includes both a shape and the appearance of the test images where the appearance can be perceived as the texture or grey level appearance in the image. The shape output is more interesting in the case of automating the facial expressions of the child, because it represents the coordinates of different key points of the child's face. Due to the publication of this thesis the images showing the results from the Active Appearance Model will not be shown.

To sum up the study on the AAM/EAM, it has shown promising results. The model in this study is as mentioned trained using only 4 frames and tested using 6 frames, all of which showing the same child. If a more thorough study should be carried out, the training set should consist of a number of frames from several dyads (only the face of the baby) to generalize the model completely. This model should be tested on the remaining frames of all the dyads to evaluate the performance. The hope for this is that the optimal model, in an acceptable way, represents the child's facial expressions. These could then be applied as features in a classifier that could group the facial expressions according to the scheme in A and thereby to be used at Babylab. Furthermore the key points could be used as features in the classifiers in speaker recognition and emotion classification to improve the performance.
The Matlab code used in this study is from [8].

Appendix C

Synchronization

C.1 Sound versus Motion Capture

To extract the syncronization information between the external sound files and the motion capture files, one idea was to extract the distance profile between the mother and child from each of the two modalities and then correlate the two. The distance between the mother and child can be calculated, for each frame, from the head marker coordinates extracted from the Qualisys files. In practice, the forehead coordinates for both the mother and child are estimated by taking the mean of the two front head markers. The distance between these two estimates are then assumed to represent the head distance between the dyad. An example of a distance profile is shown in figure C.1.

Likewise, the distance profile can be estimated from the time delay between the mother's and child's mouth microphones. Since the mother's utterances are registered in the child's microphone and the other way around, the time delay between the two channels can be estimated through the cross-correlation function, see equation 4.1. By applying the speed of sound measure, i.e. sound travels with a speed of around 343 m/s, the distance between the two microphones (corresponding to the distance between the mother and child) can be estimated. A cross-correlation function is calculated for every interval of 800 samples in the sound signal, which is equal to one mocap frame, in which the location of the maximum value represents the time delay. A common represen-

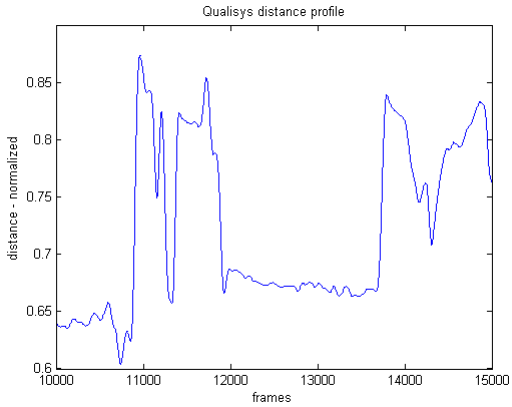


Figure C.1: Distance profile of dyad 001 calculated from the head markers of the mother and child.

tation of the two distance profiles is thereby obtained.
The estimated distance profile for dyad 001 is shown in figure C.2. The two dis-

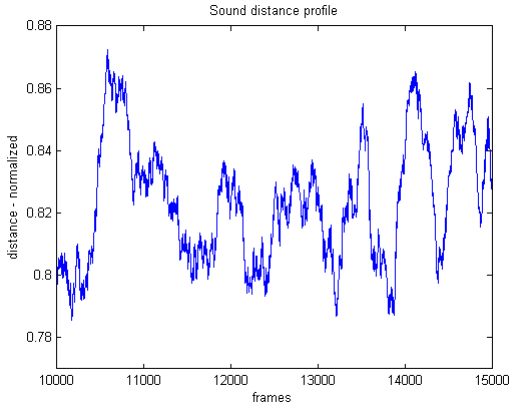


Figure C.2: Distance profile of dyad 001 estimated using the cross-correlation of the signals from the two mouth microphones.

tance measures can then be compared by correlating the distance development over time for each of the two recording modalities. As is observed from the two filtered and normalized distance profiles, they are very dissimilar. The cross-correlation between the two profiles is calculated to extract the synchronization difference. This function is shown in figure C.3.

As can be seen, no clear maximum peak is observed, making this approach

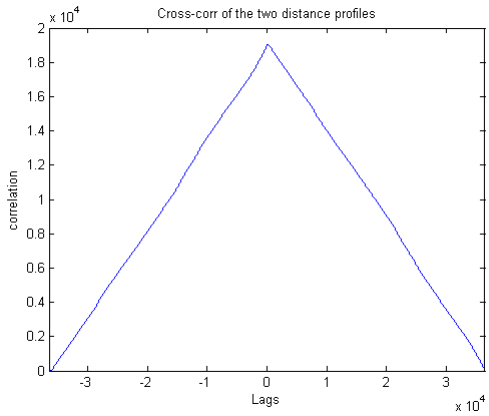


Figure C.3: Cross-correlation of the two distance profiles.

rather uncertain. The maximum peak is shown in figure C.4. Here can be seen that there is an actual max value, but that it is very uncertain. This maximum peak corresponds to a time delay between the two distance profiles, and thereby between the external sound file and the motion capture file, of 1.45 seconds. Further more, if analysed closer, the entire flat area of the maximum values exactly shows the position of the two signals where they overlap completely.

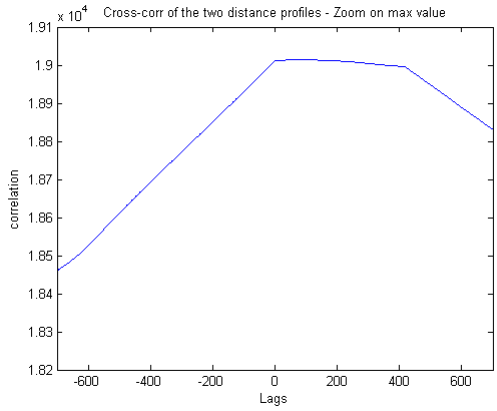


Figure C.4: Cross-correlation of the two distance profiles, zoomed in on the maximum peak.

Appendix D

Results - Speaker Identification

This appendix assists the results obtained in the speaker identification problem in section 9.1. The appendix is divided into sections corresponding to the ones in section 9.1.

D.1 Parameter Estimation

Before the test of window size can be carried out, the optimal parameters is to be decided for each of the five respective classifiers. In the following results used to decide these optimal parameters for each of the five classifiers are shown.

D.1.1 Gaussian Mixture Model

The figures presented here are figures D.1, D.2, D.3, D.4 and D.5 showing the error rate of the GMM classifier as a function of number of components, K, for the window sizes 10 ms, 50 ms, 100 ms, 200 ms and 250 ms respectively.

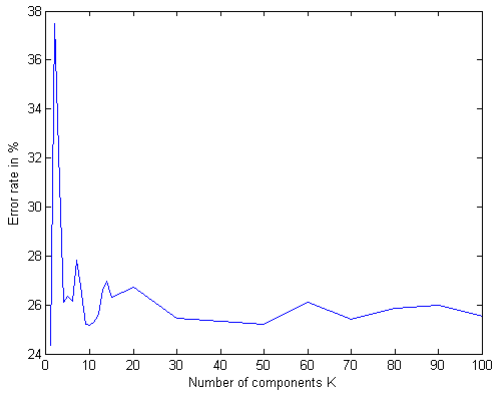


Figure D.1: The error rate as a function of number of components, K, for GMM when K is assumed equal for all classes. Here shown for the window size 10 ms.

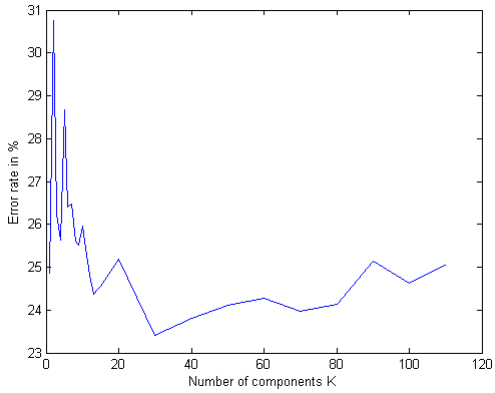


Figure D.2: The error rate as a function of number of components, K, for GMM when K is assumed equal for all classes. Here shown for the window size 50 ms.

D.1.2 K-Nearest Neighbour

This section presents figures that shows the error rate for the KNN classifier as a function of the number of neighbours. The results are shown in figure D.6, D.7, D.8, D.9, D.10 for the window sizes 10 ms, 50 ms, 100 ms, 200 ms and 250 ms respectively. The result for the last window size of 150 ms is shown in figure

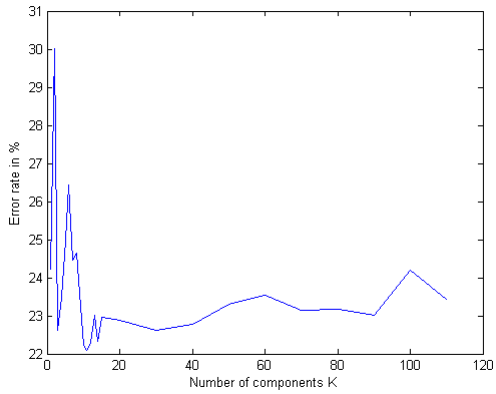


Figure D.3: The error rate as a function of number of components, K, for GMM when K is assumed equal for all classes. Here shown for the window size 100 ms.

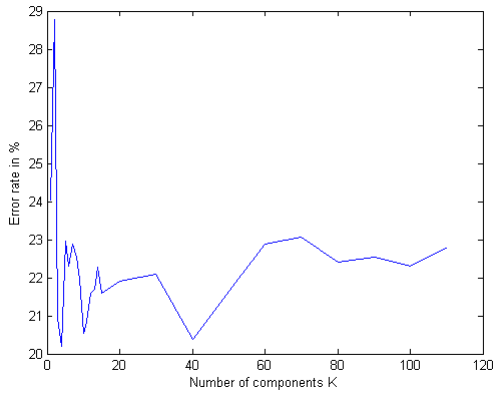


Figure D.4: The error rate as a function of number of components, K, for GMM when K is assumed equal for all classes. Here shown for the window size 200 ms.

9.3 in section 9.1.1. The summing up of the results shown in this section can be seen in table 9.3 in section 9.1.1.

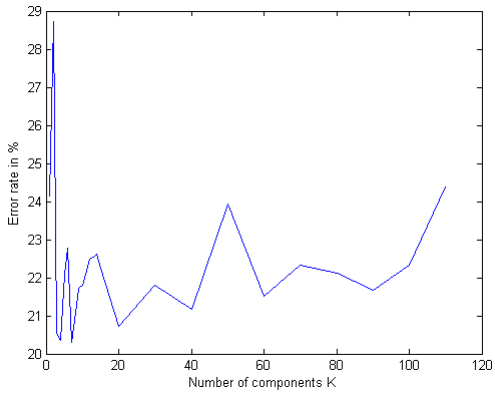


Figure D.5: The error rate as a function of number of components, K, for GMM when K is assumed equal for all classes. Here shown for the window size 250 ms.

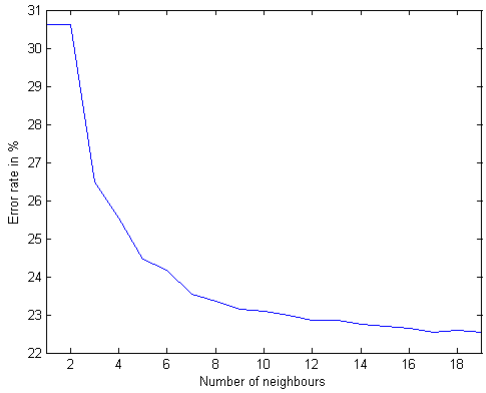


Figure D.6: The error rate as a function of number of neighbours in KNN. Here shown for the window size 10 ms. It should be noted that the number of neighbours only goes up to 19, due to computational time.

D.1.3 Decision Tree

This section shows the results needed for deciding the minimum number, κ, of observations in each node before the impure node undergoes a split. The results are shown both when using the entropy as a split criteria in section D.1.3.1 as

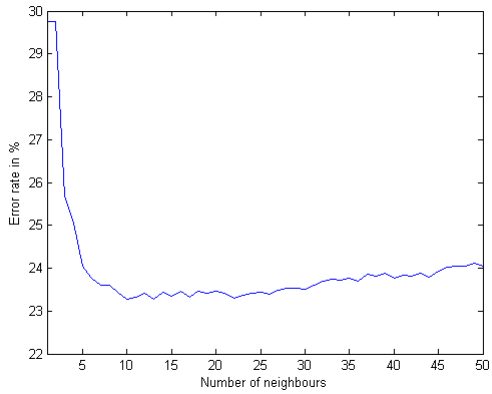


Figure D.7: The error rate as a function of number of neighbours in KNN. Here shown for the window size 50 ms.

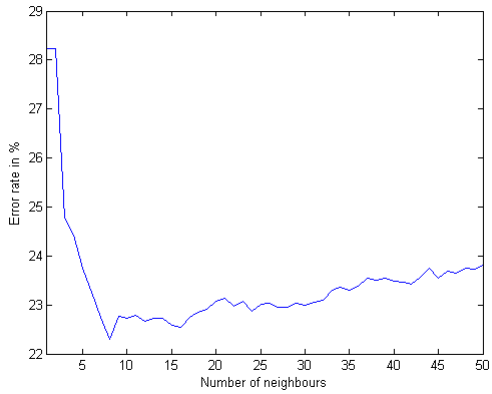


Figure D.8: The error rate as a function of number of neighbours in KNN. Here shown for the window size 100 ms.

well as using the Gini impurity measure as a split criteria in section D.1.3.2.

D.1.3.1 Split Criteria - Entropy

This section presents figures that shows the error rate for the decision tree classifier as a function of the number, κ, that decides the minimum number of observations before the impure node undergoes a split. The split criteria used

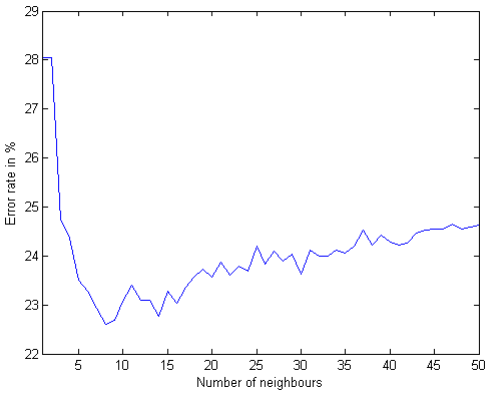


Figure D.9: The error rate as a function of number of neighbours in KNN. Here shown for the window size 200 ms.

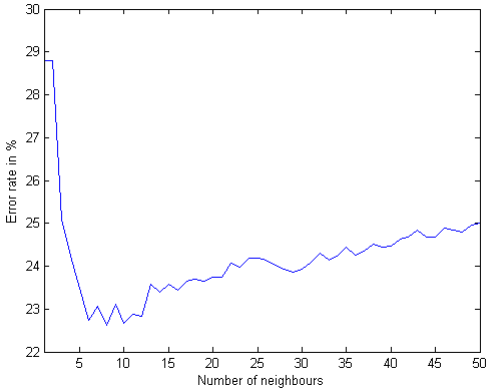


Figure D.10: The error rate as a function of number of neighbours in KNN. Here shown for the window size 250 ms.

in this section is the entropy measure, as given in equation 5.26 in section 5.4.3. The results are shown in figure D.11, D.12, D.13, D.14 and D.15 for the window sizes 10ms, 50ms, 100ms, 200ms and 250ms respectively. The result for the last window size of 150 ms is shown in figure 9.4 in section 9.1.1.

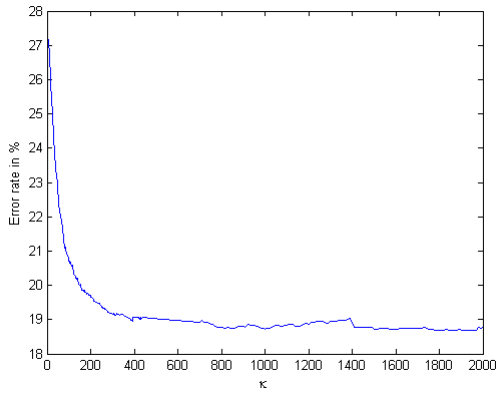


Figure D.11: The error rate as a function of κ in the decision tree classifier. κ is the number that decides the minimum number of observations before the impure node undergoes a split. Here shown for the window size 10 ms, channel 2 (mother) and with all features included as shown in table 5.3. The measure of impurity used for these result is the entropy measure given by equation 5.26.

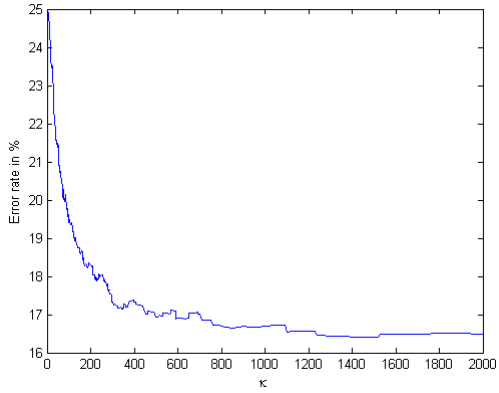


Figure D.12: The error rate as a function of κ in the decision tree classifier. κ is the number that decides the minimum number of observations before the impure node undergoes a split. Here shown for the window size 50 ms, channel 2 (mother) and with all features included as shown in table 5.3. The measure of impurity used for these result is the entropy measure given by equation 5.26.

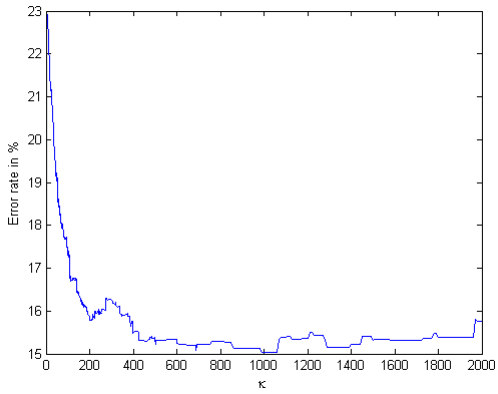


Figure D.13: The error rate as a function of κ in the decision tree classifier. κ is the number that decides the minimum number of observations before the impure node undergoes a split. Here shown for the window size 100 ms, channel 2 (mother) and with all features included as shown in table 5.3. The measure of impurity used for these result is the entropy measure given by equation 5.26.

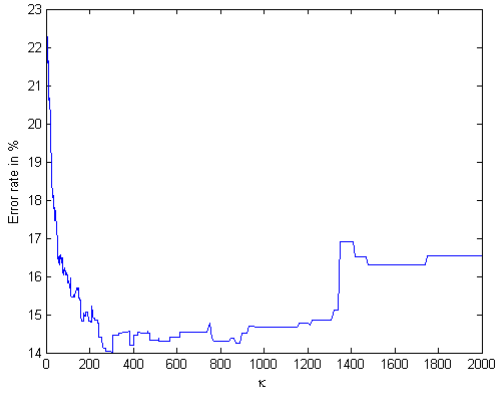


Figure D.14: The error rate as a function of κ in the decision tree classifier. κ is the number that decides the minimum number of observations before the impure node undergoes a split. Here shown for the window size 200 ms, channel 2 (mother) and with all features included as shown in table 5.3. The measure of impurity used for these result is the entropy measure given by equation 5.26.

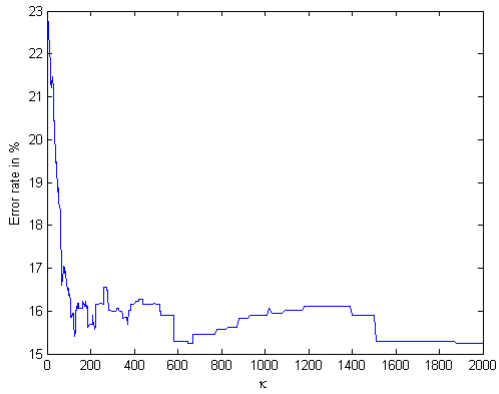


Figure D.15: The error rate as a function of κ in the decision tree classifier. κ is the number that decides the minimum number of observations before the impure node undergoes a split. Here shown for the window size 250 ms, channel 2 (mother) and with all features included as shown in table 5.3. The measure of impurity used for these result is the entropy measure given by equation 5.26.

D.1.3.2 Split Criteria - Gini

This section presents figures that shows the error rate for the decision tree classifier as a function of the number, κ, that decides the minimum number of observations before the impure node undergoes a split. The split criteria used in this section is the Gini impurity measure, as given in equation 5.27 in section 5.4.3. The results are shown in figure D.16, D.17, D.18, D.19 and D.20 for the window sizes 50ms, 100ms, 150ms, 200ms and 250ms respectively.

D.1.4 Artificial Neural Network

This section presents figures that shows the error rate for the ANN classifier as a function of the number of hidden units. The results are shown in figure D.21, D.22, D.23, D.24, D.25 for the window sizes 10ms, 50ms, 100ms, 200ms and 250ms respectively. The result for the last window size of 150 ms is shown in figure 9.5 in section 9.1.1.

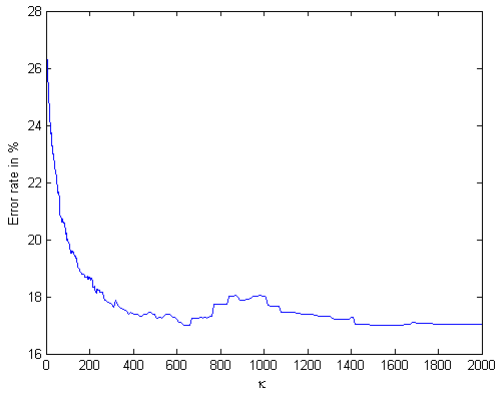


Figure D.16: The error rate as a function of κ in the decision tree classifier. κ is the number that decides the minimum number of observations before the impure node undergoes a split. Here shown for the window size 50 ms, channel 2 (mother) and with all features included as shown in table 5.3. The measure of impurity used for these result is the Gini measure given by equation 5.27.

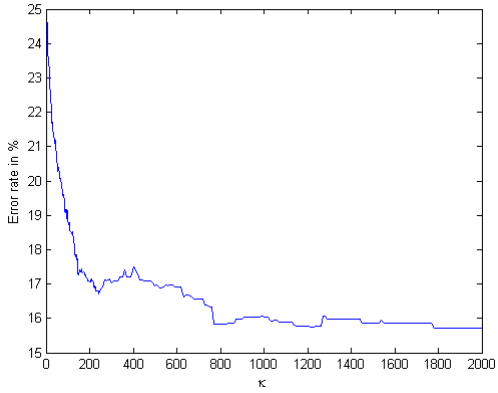


Figure D.17: The error rate as a function of κ in the decision tree classifier. κ is the number that decides the minimum number of observations before the impure node undergoes a split. Here shown for the window size 100 ms, channel 2 (mother) and with all features included as shown in table 5.3. The measure of impurity used for these result is the Gini measure given by equation 5.27.

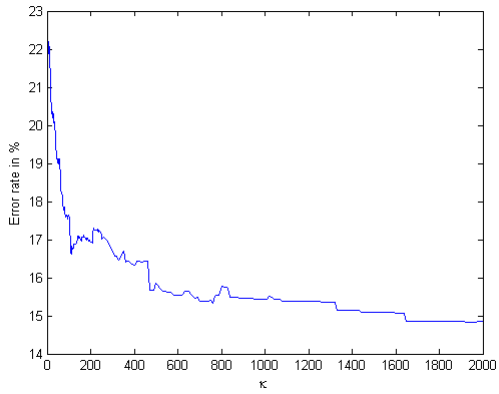


Figure D.18: The error rate as a function of κ in the decision tree classifier. κ is the number that decides the minimum number of observations before the impure node undergoes a split. Here shown for the window size 150 ms, channel 2 (mother) and with all features included as shown in table 5.3. The measure of impurity used for these result is the Gini measure given by equation 5.27.

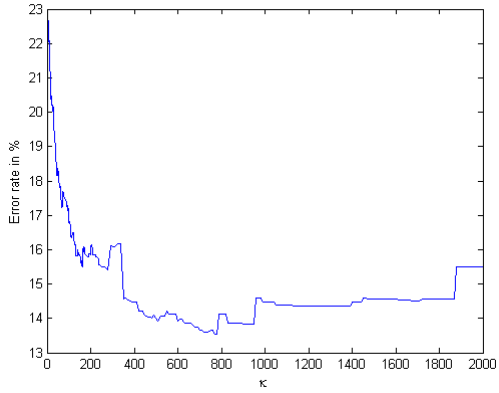


Figure D.19: The error rate as a function of κ in the decision tree classifier. κ is the number that decides the minimum number of observations before the impure node undergoes a split. Here shown for the window size 200 ms, channel 2 (mother) and with all features included as shown in table 5.3. The measure of impurity used for these result is the Gini measure given by equation 5.27.

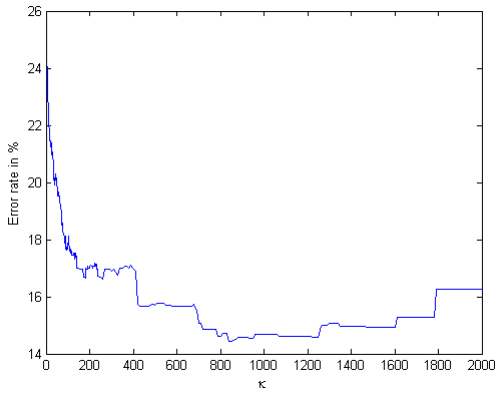


Figure D.20: The error rate as a function of κ in the decision tree classifier. κ is the number that decides the minimum number of observations before the impure node undergoes a split. Here shown for the window size 250 ms, channel 2 (mother) and with all features included as shown in table 5.3. The measure of impurity used for these result is the Gini measure given by equation 5.27.

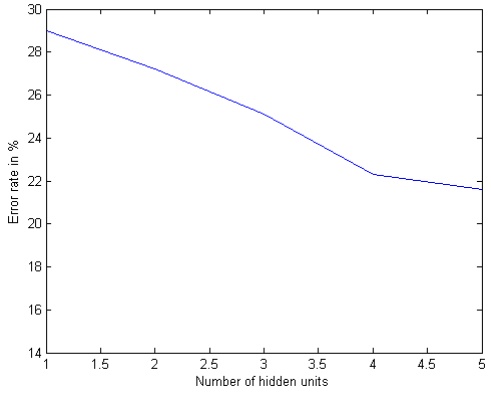


Figure D.21: The error rate as a function of number of hidden units in ANN. Here shown for the window size 10 ms, channel 2 (mother) and with all features included as shown in table 5.3. Note that this figure only shows the error rate op to 5 hidden units. Due to the size of the data for the window size of 10 ms it is not computationally possible to go higher up in number of hidden units.

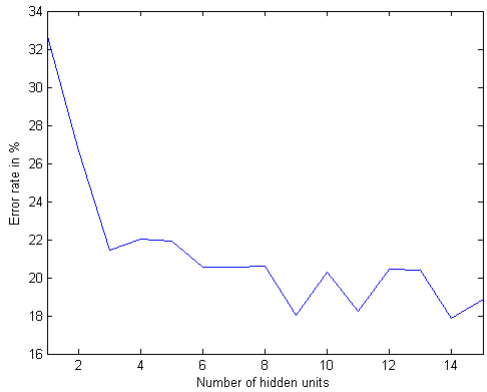


Figure D.22: The error rate as a function of number of hidden units in ANN. Here shown for the window size 50 ms, channel 2 (mother) and with all features included as shown in table 5.3.

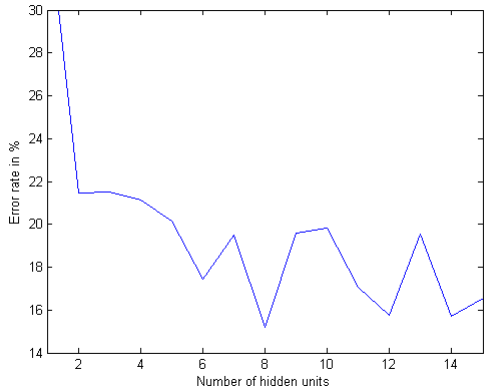


Figure D.23: The error rate as a function of number of hidden units in ANN. Here shown for the window size 100 ms, channel 2 (mother) and with all features included as shown in table 5.3.

D.2 Other Optional Parameters

This section includes tables regarding the five respective classifiers and their setting values in Matlab.

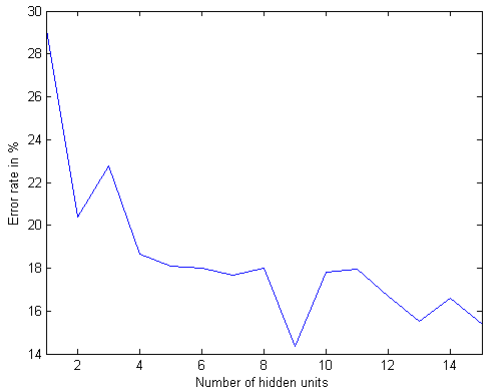


Figure D.24: The error rate as a function of number of hidden units in ANN. Here shown for the window size 200 ms, channel 2 (mother) and with all features included as shown in table 5.3.

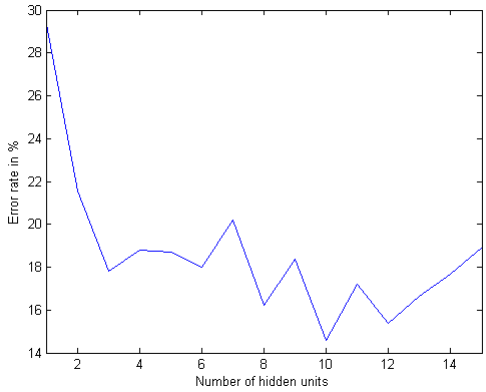


Figure D.25: The error rate as a function of number of hidden units in ANN. Here shown for the window size 250 ms, channel 2 (mother) and with all features included as shown in table 5.3.

D.3 Confusion Matrices

In this chapter the confusion matrices for the remaining classifiers (KNN and MNR) can be found in figure D.26.

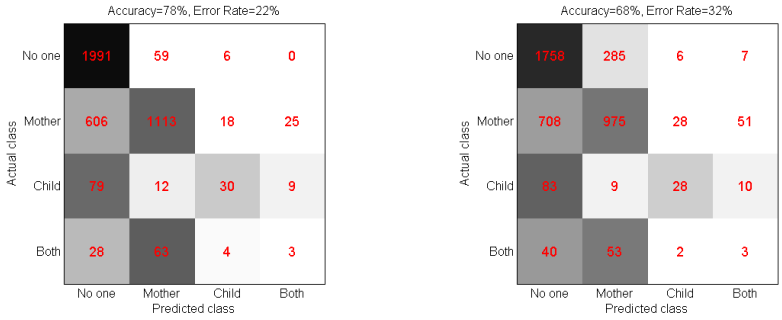
In figure D.27 the human confusion between two coder at Babylab can be seen

Parameter	Value
'distance'	'euclidean'
'rule'	'nearest'

Table D.1: The setup for the KNN classifier in Matlab.

Parameter	Value
'Nh'	Variable

Table D.2: The setup for the ANN classifier in Matlab.



(a) (b)

Figure D.26: The confusion matrices for the (a) KNN classifier and (b) MNR classifier both for the window size 150 ms.

for dyad 006 and 020 respectively. It should be mentioned that the data for dyad 020 are re-annotated at the moment at Babylab due to the high confusion of 31 % and thereby low reliability.

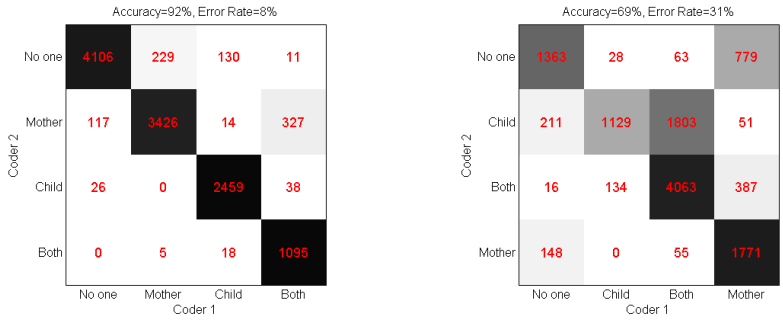
Figure D.28 shows the confusion matrices for the TREE classifier, the window size 150 ms and for each of the five first feature compositions in table 9.6. Figure D.29 shows in continuation to figure D.28 the confusion matrices for the last five feature compositions from table 9.6.

Parameter	Value
'model'	'nominal'
'interactions'	'off'
'link'	'logit'
'estdisp'	'off'

Table D.3: The setup for the MNR classifier in Matlab.

Parameter	Value
'start'	'randSample'
'replicates'	30
'CovType'	'diagonal'
'SharedCov'	'false'
'regularize'	0

Table D.4: The setup for the GMM classifier in Matlab.



(a) (b)

Figure D.27: The confusion matrices for the human coders at Babylab. (a) Dyad 006. (b) Dyad 020.

Parameter	Value
'prune'	'on'
'minparent'	Variable
'weights'	1
'splitcriterion'	Deviance/Gini

Table D.5: The setup for the TREE classifier in Matlab.

D.4 Test of Predictability: Windows versus Sub-Windows

Figure D.30 shows the estimated class labels for 600 consecutive observations when using the TREE classifier with the window size 50 ms. The misclassified observations are presented by red dots whereas the correct classified observations are presented with green dots. The estimated class labels for TREE shows that when TREE makes one misclassification, it is often directly followed by 3 or more extra misclassifications.

D.5 Combining Channels

This section provides the tables for the ANN and the GMM classifier when combining the channels. The result when combining are shows in the caption to the tables. Table D.6 shows the results for ANN whereas table D.7 shows the results for GMM.

Window	Channel 1	Channel 2
50 ms	20 %	17%
150 ms	17 %	**14 %**

Table D.6: The error rates for the two windows 50 ms and 150 ms for each of the two channels. The ANN classifier has been used where the combined error rate gives 15 % .

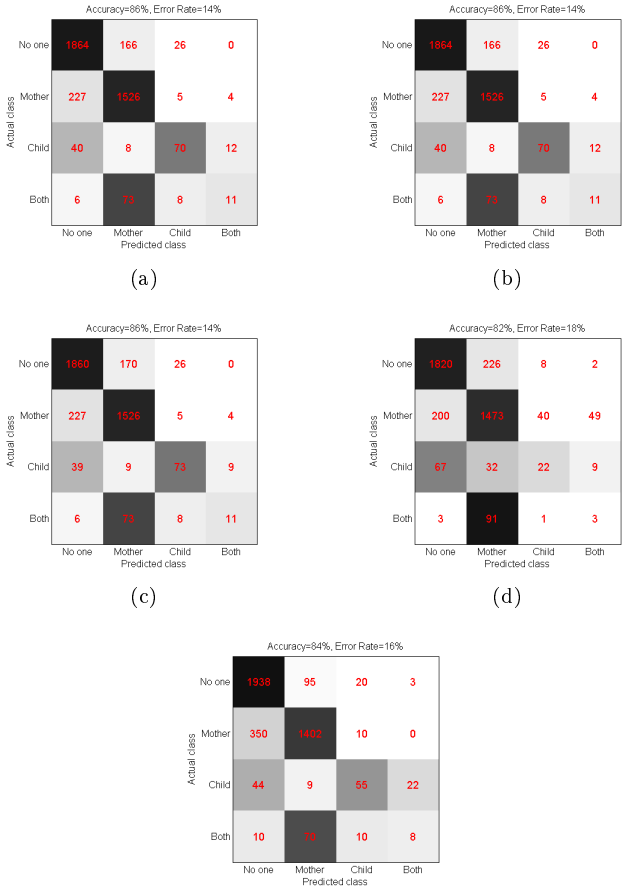


Figure D.28: The confusion matrices for the TREE classifier for the first five feature compositions as shown in table 9.6. (a) Composition 0. (b) Composition 1. (c) Composition 2. (d) Composition 3. (e) Composition 4.

D.6 Example of a TREE

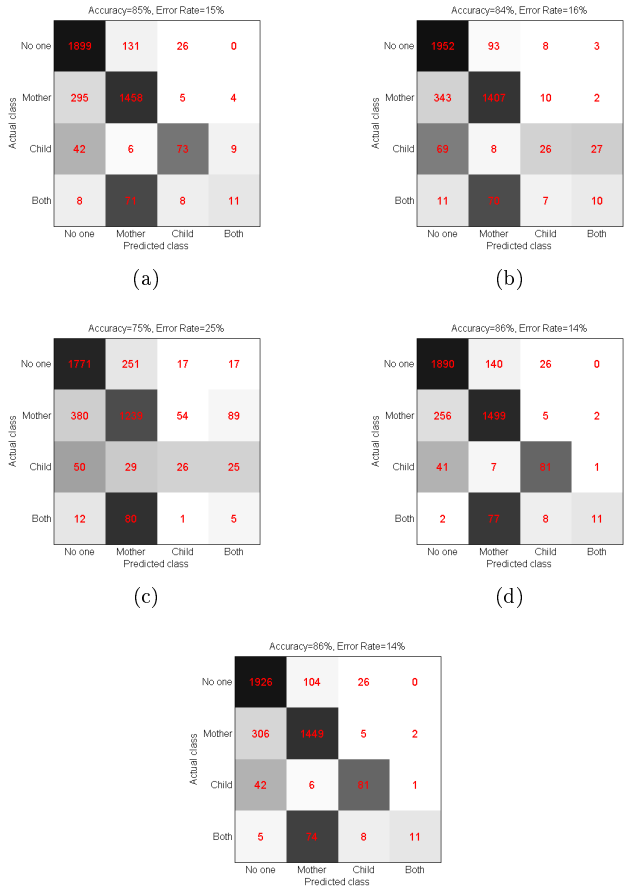


Figure D.29: The confusion matrices for the TREE classifier for the last five feature compositions as shown in table 9.6. (a) Composition 5. (b) Composition 6. (c) Composition 7. (d) Composition 8. (e) Composition 9.

Window	Channel 1	Channel 2
50 ms	18 %	**17 %**
150 ms	17 %	**17 %**

Table D.7: The error rates for the two windows 50 ms and 150 ms for each of the two channels. The GMM classifier has been used where the combined error rate gives 16 % .

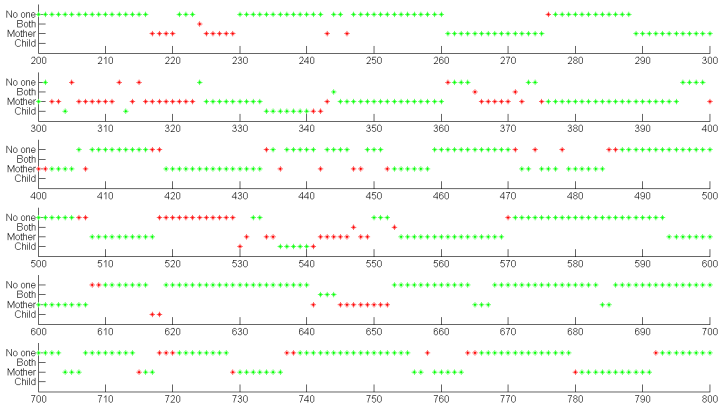


Figure D.30: The estimated class labels for 600 observations when using TREE at 50 ms. Red indicates a misclassified observation and green a correct classified observation.

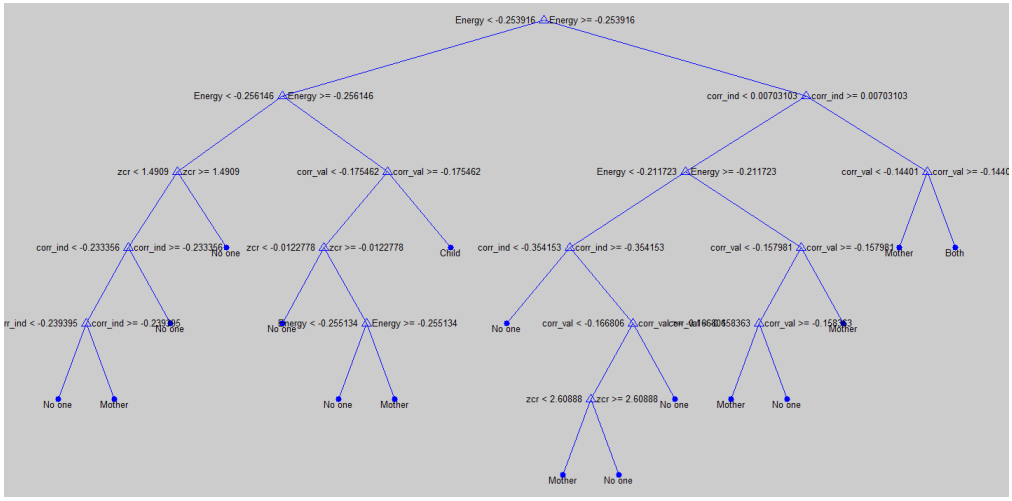


Figure D.31: An example of a TREE where only the features, energy, zcr and the cross correlation has been used. Only 1300 observations has been used to fit the TREE due to overview.

Appendix E

Results - Emotion Recognition

In this chapter the best confusion matrices for the emotion recognition task are shown. All feature combinations are included, except for composition 0 (all features included) that was discussed in section 9.2.2.

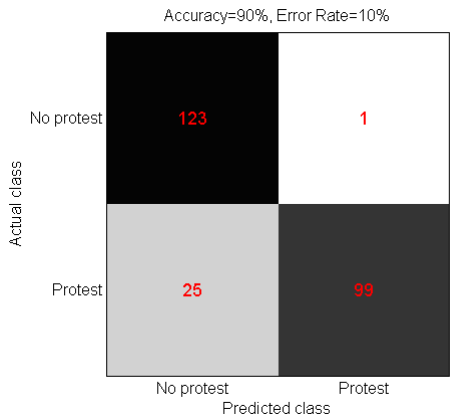


Figure E.1: Confusion matrix for HMM, feature composition 1: MFCC, energy, zcr.

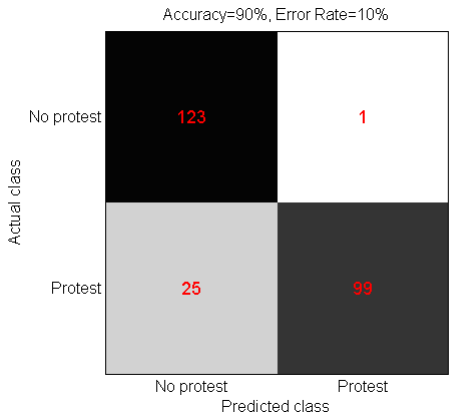


Figure E.2: Confusion matrix for HMM, feature composition 2: MFCC, energy.

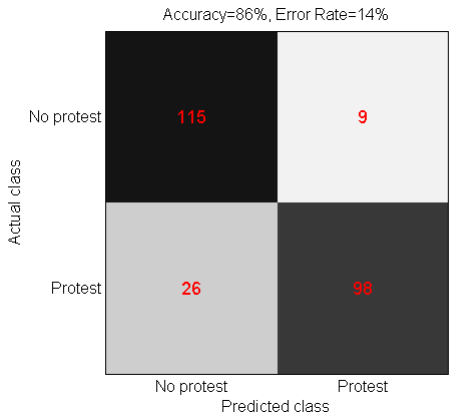


Figure E.3: Confusion matrix for HMM, feature composition 3: MFCC, zcr.

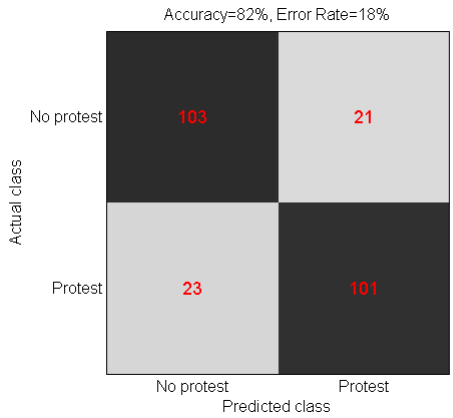


Figure E.4: Confusion matrix for HMM, feature composition 4: MFCC.

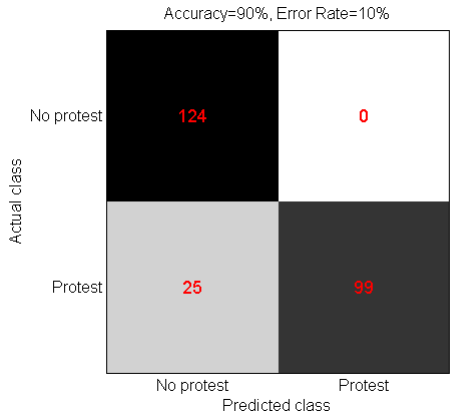


Figure E.5: Confusion matrix for HMM, feature composition 5: energy, zcr.

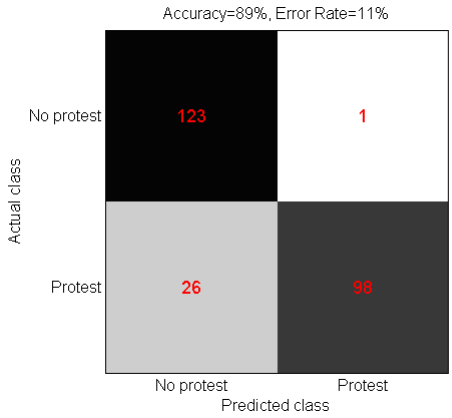


Figure E.6: Confusion matrix for HMM, feature composition 6: MFCC, delta-MFCC, energy.

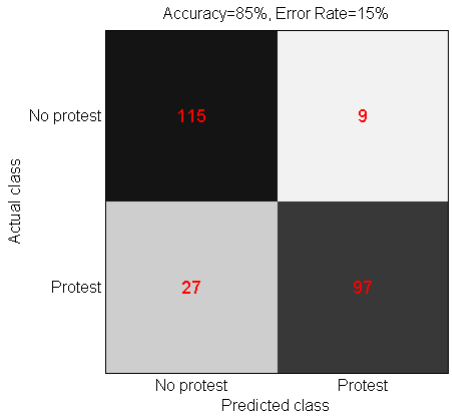


Figure E.7: Confusion matrix for HMM, feature composition 7: MFCC, delta-MFCC.

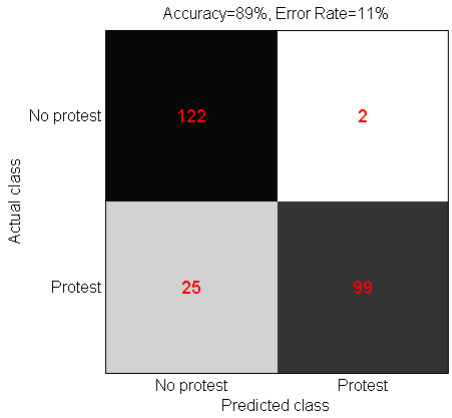


Figure E.8: Confusion matrix for HMM, feature composition 8: delta-MFCC, energy.

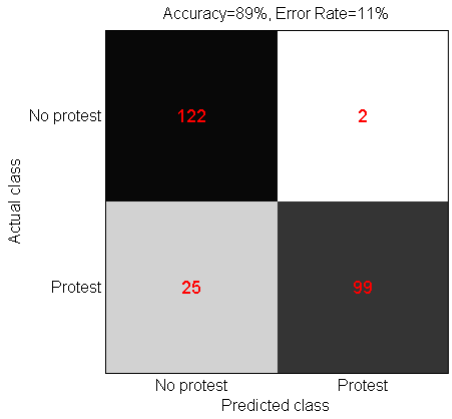


Figure E.9: Confusion matrix for HMM, feature composition 9: MFCC, energy, zcr.

Bibliography

[1] http://ieipi.wordpress.com/2011/04/10/.

[2] http://www.personal.rdg.ac.uk/ llsroach/phon2/artic-basics.htm.

[3] http://www.hitl.washington.edu/publications/hollander/2.html.

[4] http://www.dcs.shef.ac.uk/ ning/resources/gammatone/.

[5] http://en.wikipedia.org/wiki/Discrete_cosine_transform.

[6] http://www.isip.piconepress.com /projects/speech/software/legacy/decision_tree/index.html.

[7] http://en.wikipedia.org/wiki/Hidden_Markov_model.

[8] http://svn.imm.dtu.dk/AAMLab/svn/AAMLab/trunk/.

[9] B.J. Anderson, P. Vietze, and P.R. Dokecki. Reciprocity in vocal interactions of mothers and infants. *Child Development*, pages 1676–1681, 1977.

[10] B. Beebe, J. Jaffe, S. Markese, K. Buck, H. Chen, P. Cohen, L. Bahrick, H. Andrews, and S. Feldstein. The origins of 12-month attachment: A microanalysis of 4-month mother–infant interaction. *Attachment & human development*, 12(1-2):3–141, 2010.

[11] B. Beebe, F. Lachmann, and J. Jaffe. Mother?infant interaction structures and presymbolic self-and object representations. *Psychoanalytic dialogues*, 7(2):133–182, 1997.

[12] H. Beigi. *Fundamentals of speaker recognition*. Springer US, 2011.

[13] C.M. Bishop and SpringerLink (Service en ligne). *Pattern recognition and machine learning*, volume 4. springer New York, 2006.

[14] D.A. Cairns and J.H.L. Hansen. Nonlinear analysis and classification of speech under stressed conditions. *J. Acoust. Soc. Am*, 96(6), 1994.

[15] N.V. Chawla, K.W. Bowyer, L.O. Hall, and W.P. Kegelmeyer. Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002.

[16] L. Devillers, L. Vidrascu, and L. Lamel. Challenges in real-life emotion annotation and machine learning based detection. *Neural Networks*, 18(4):407–422, 2005.

[17] M. El Ayadi, M.S. Kamel, and F. Karray. Survey on speech emotion recognition: Features, classification schemes, and databases. *Pattern Recognition*, 44(3):572–587, 2011.

[18] A.J. Eronen, V.T. Peltonen, J.T. Tuomi, A.P. Klapuri, S. Fagerlund, T. Sorsa, G. Lorho, and J. Huopaniemi. Audio-based context recognition. *Audio, Speech, and Language Processing, IEEE Transactions on*, 14(1):321–329, 2006.

[19] R. Feldman. Parent–infant synchrony and the construction of shared timing; physiological precursors, developmental outcomes, and risk conditions. *Journal of Child Psychology and Psychiatry*, 48(3-4):329–354, 2007.

[20] D.J. France, R.G. Shiavi, S. Silverman, M. Silverman, and M. Wilkes. Acoustical properties of speech as indicators of depression and suicidal risk. *Biomedical Engineering, IEEE Transactions on*, 47(7):829–837, 2000.

[21] T. Ganchev. Contemporary methods for speech parameterization. *Contemporary Methods for Speech Parameterization*, pages 1–106, 2011.

[22] M.F. Hansen, J. Fagertun, R. Larsen, and DTU Informatic. Elastic appearance models. In *Jesse Hoey, Stephen McKenna and Emanuele Trucco, Proceedings of the British Machine Vision Conference, pages*, pages 91–1.

[23] P. S. K. Hansen and L. K. Hansen. Exercise from the course 02457.

[24] M.R. Hasan, M. Jamil, and M.G.R.M.S. Rahman. Speaker identification using mel frequency cepstral coefficients. *variations*, 1:4, 2004.

[25] S. Hayakawa and F. Itakura. Text-dependent speaker recognition using the information in the higher frequency band. In *Acoustics, Speech, and Signal Processing, 1994. ICASSP-94., 1994 IEEE International Conference on*, volume 1, pages I–137. IEEE, 1994.

[26] H. He and E.A. Garcia. Learning from imbalanced data. *Knowledge and Data Engineering, IEEE Transactions on*, 21(9):1263–1284, 2009.

[27] M.A. Hossan, S. Memon, and M.A. Gregory. A novel approach for mfcc feature extraction. In *Signal Processing and Communication Systems (ICSPCS), 2010 4th International Conference on*, pages 1–5. IEEE, 2010.

[28] H.C. Hsu, A. Fogel, and D.S. Messinger. Infant non-distress vocalization during mother-infant face-to-face interaction: Factors associated with quantitative and qualitative differences7. *Infant behavior and development*, 24(1):107–128, 2001.

[29] J. Jaffe. *Rhythms of dialogue in infancy: Coordinated timing in development*. Wiley-Blackwell, 2001.

[30] J.H. Jensen. Isound. Matlab Toolbox.

[31] M. Ji, S. Kim, H. Kim, K.C. Kwak, and Y.J. Cho. Reliable speaker identification using multiple microphones in ubiquitous robot companion environment. In *Robot and Human interactive Communication, 2007. RO-MAN 2007. The 16th IEEE International Symposium on*, pages 673–677. IEEE, 2007.

[32] B.H. Juang and L.R. Rabiner. Hidden markov models for speech recognition. *Technometrics*, pages 251–272, 1991.

[33] T. Kinnunen and H. Li. An overview of text-independent speaker recognition: From features to supervectors. *Speech Communication*, 52(1):12–40, 2010.

[34] O.J. Kirk. Udvikling af vaerktoejer til at hjaelpe babylabs psykologer med at analysere 3d motion capture data fra mor/barn-interaktioner. Master's thesis, University of Copenhagen, 2011.

[35] M. Koulomzin, B. Beebe, S. Anderson, J. Jaffe, S. Feldstein, and C. Crown. Infant gaze, head, face and self-touch at 4 months differentiate secure vs. avoidant attachment at 1 year: A microanalytic approach. *Attachment & human development*, 4(1):3–24, 2002.

[36] C.M. Lee, S. Narayanan, and R. Pieraccini. Recognition of negative emotions from the speech signal. In *Automatic Speech Recognition and Understanding, 2001. ASRU'01. IEEE Workshop on*, pages 240–243. IEEE, 2001.

[37] J. Luque and J. Hernando. Robust speaker identification for meetings: Upc clear?07 meeting room evaluation system. *Multimodal Technologies for Perception of Humans*, pages 266–275, 2008.

[38] D. MacKay. The evidence framework applied to classification networks. *Neural Computation*, 4(5):720–736, 1992.

[39] David J. C. MacKay. A practical Bayesian framework for backpropagation networks. *Neural Computation*, 4:448–472, 1992.

[40] D.J.C. MacKay. Bayesian interpolation. *Neural computation*, 4(3):415–447, 1992.

[41] S.R. Madikeri and H.A. Murthy. Mel filter bank energy-based slope feature and its application to speaker recognition. In *Communications (NCC), 2011 National Conference on*, pages 1–4. IEEE, 2011.

[42] D.J. Mashao and M. Skosan. Combining classifier decisions for robust speaker identification. *Pattern Recognition*, 39(1):147–155, 2006.

[43] I. Matthews and S. Baker. Active appearance models revisited. *International Journal of Computer Vision*, 60(2):135–164, 2004.

[44] D.S. Messinger. Positive and negative: Infant facial expressions and emotions. *Current Directions in Psychological Science*, 11(1):1–6, 2002.

[45] D.S. Messinger, M.H. Mahoor, S.M. Chow, and J.F. Cohn. Automated measurement of facial expression in infant–mother interaction: A pilot study. *Infancy*, 14(3):285–305, 2009.

[46] K.S.R. Murty and B. Yegnanarayana. Combining evidence from residual phase and mfcc features for speaker recognition. *Signal Processing Letters, IEEE*, 13(1):52–55, 2006.

[47] K.S.R. Murty and B. Yegnanarayana. Combining evidence from residual phase and mfcc features for speaker recognition. *Signal Processing Letters, IEEE*, 13(1):52–55, 2006.

[48] T.L. Nwe, S.W. Foo, and L.C. De Silva. Speech emotion recognition using hidden markov models. *Speech communication*, 41(4):603–623, 2003.

[49] V. Petrushin. Emotion in speech: Recognition and application to call centers. *Artificial Neu. Net. In Engr.(ANNIE'99)*, pages 7–10, 1999.

[50] C.J. Plack. *The sense of hearing*. Lawrence Erlbaum Associates Publishers, 2005.

[51] L.R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[52] L.R. Rabiner and R.W. Schafer. *Digital processing of speech signals*, volume 100. Prentice-hall Englewood Cliffs, NJ, 1978.

[53] A. Rahman, R. Harrington, and J. Bunn. Can maternal depression increase infant risk of illness and growth impairment in developing countries? *Child: care, health and development*, 28(1):51–56, 2002.

[54] R.M. Rangayyan. *Biomedical signal analysis*. IEEE press, 2002.

[55] D.A. Reynolds. Experimental evaluation of features for robust speaker identification. *Speech and Audio Processing, IEEE Transactions on*, 2(4):639–643, 1994.

[56] D.A. Reynolds. Channel robust speaker verification via feature mapping. In *Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on*, volume 2, pages II–53. Ieee, 2003.

[57] D.A. Reynolds and R.C. Rose. Robust text-independent speaker identification using gaussian mixture speaker models. *Speech and Audio Processing, IEEE Transactions on*, 3(1):72–83, 1995.

[58] B. Schuller, G. Rigoll, and M. Lang. Hidden markov model-based speech emotion recognition. In *Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on*, volume 2, pages II–1. Ieee, 2003.

[59] I. Shafran, M. Riley, and M. Mohri. Voice signatures. In *Automatic Speech Recognition and Understanding, 2003. ASRU'03. 2003 IEEE Workshop on*, pages 31–36. IEEE.

[60] S. Sigurdsson, J. Larsen, L.K. Hansen, P.A. Philipsen, and H.C. Wulf. Outlier estimation and detection - application to skin lesion classification. In *IEEE International conference on acoustic speech and signal processing*, volume 1. IEEE; 1999, 2002.

[61] M. Slaney. Auditory toolbox. *Interval Research Corporation, Tech. Rep*, 10:1998, 1998.

[62] S. Steidl, M. Levit, A. Batliner, E. Nöth, and H. Niemann. Of all things the measure is man: Automatic classification of emotions and inter-labeler consistency. In *Proc. ICASSP*, volume 1, pages 317–320, 2005.

[63] P.N. Tan, M. Steinbach, V. Kumar, et al. *Introduction to data mining*. Pearson Addison Wesley Boston, 2006.

[64] M.E. Yale, D.S. Messinger, A.B. Cobo-Lewis, and C.F. Delgado. The temporal coordination of early infant communication. *Developmental Psychology*, 39(5):815, 2003.