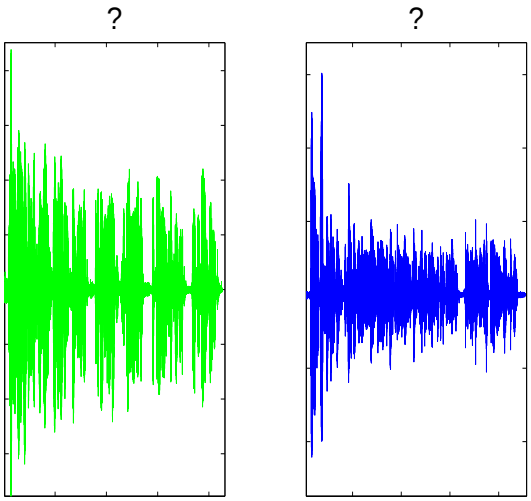
Speaker Identification for Hearing Instruments

Maïa E.M. Weddin

Master's Thesis



IMM, Denmarks Technical University

March 2005

Every day you may make progress. Every step may be fruitful. Yet there will stretch out before you an ever-lengthening, ever-ascending, ever-improving path. You know you will never get to the end of the journey. But this, so far from discouraging, only adds to the joy and glory of the climb.

Sir Winston Churchill
British politician (1874 - 1965)

Abstract

This thesis proposes a speaker identification system that can differentiate between members of a small set of speakers as well as being able to detect an impostor sound and classify it accordingly. The identification system is text-independent, so no specific words or sounds have to be uttered for the identification to work. In cooperation with GN ReSound, the ultimate implementation of this system would be in hearing aids, more specifically, those designed for children, as they have more difficulty adjusting a hearing instrument when such an adjustment becomes necessary. A variety of speech feature sets are extracted, including fundamental frequency estimates, LPCC, warped LPCC, PLPCC, MFCC and the LPC residual. Three classifiers are used to establish which combination of feature set and classifier is optimal. These classification methods are the Mixture of Gaussians models, k-Nearest Neighbour and the nonlinear Neural Network. The classification results are obtained for each frame of a test sentence and the performance of each system setup is measured both in identification rate of the small set of speakers, that is calculated using consensus over the individually classified frames for each sentence, and in the percentage of correctly classified frames. The Neural Network classifier proves to be more robust than the Mixture of Gaussians classifier and already results in a 100% correct identification rate for the 8MFCC feature set.

As the ultimate aim of this research is the implementation of a speaker identification system in a hearing instrument, a method for detecting impostors is implemented. This is done by using density modelling with the Mixture of Gaussians classifier and a rate of 90% impostor detection is obtained for the 12ΔMFCC feature set.

Finally, the small set of speakers is divided into a group of female speakers and a group of male speakers based on fundamental frequency estimates. A division of feature sets is implemented so that subsets based on whether a frame is voiced, unvoiced, voiced preceded by a voiced frame, or unvoiced preceded by a voiced frame, are formed. For the 12ΔMFCC feature set used with the Neural Network classifier, the correct identification of all speakers using a limited amount of data is only obtained when using the voiced preceded by unvoiced and the unvoiced preceded by voiced features subsets, and the correct frame rate using these subsets combined with gender separation is increased by up to 23%.

Keywords: Fundamental frequency estimation, MFCC, LPCC, PLPCC, Mixture of Gaussians, impostor detection, nonlinear neural network, voiced/unvoiced speech

Acknowledgements

This thesis would not have been possible without the constant technical advice, innovative thinking and endless enthusiasm of my supervisor Associate Professor Ole Winther, to whom I am grateful for not only providing guidance and but also for the avid interest that he showed in my work. I would also like to thank Brian Pedersen of GN ReSound for giving me this great opportunity and for the advice along the way. My thanks extend to Professor Steven Greenberg, who has generously provided invaluable knowledge and advice and with whom I have greatly enjoyed discussing aspects of my project. I am also grateful for the considerable time and thought that Thomas Beierholm put in his suggestions and comments on my work during the entire course of this thesis.

Thank-you to the staff, Ph.D students and other M.Sc. students at IMM for their help and for continuously providing a warm and stimulating working environment. In particular I am grateful to Ling Feng for kindly providing the ELSDSR database and for providing me with a starting point for my research.

Finally, I will be eternally grateful to the family and friends who have given me no end of love, support and understanding during the sometimes trying days, weeks and months that it took to complete this thesis.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Speaker Recognition

The possibilities that automatic speaker recognition systems provide are exciting, numerous and powerful. A lot of research has therefore been invested in the development of such systems, though a number of questions remain unanswered.

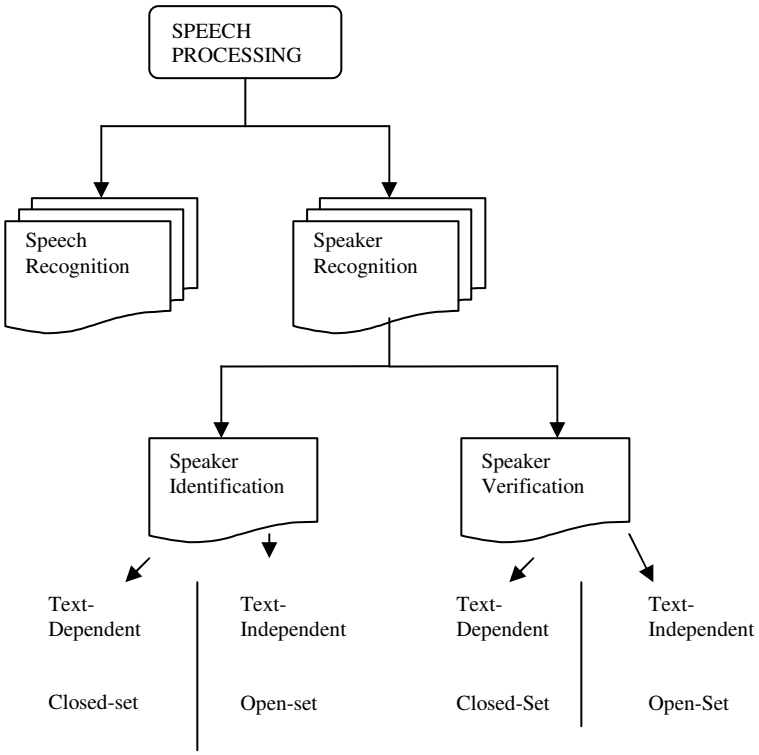


Figure 1.1: The Scope of Speaker Recognition

Speech processing techniques over the past few decades have developed to such an extent that it is now possible to construct both automatic speech recognition systems and automatic speaker recognition systems. Speech recognition is achieved when a system can reliably recognise a given word or other utterance regardless of the person who produced the sound. On the other hand, the aim of speaker recognition is to make a decision on which speaker made an utterance regardless of the speech content.
Speaker recognition can be divided into two parts: Speaker Identification (SID) and

Speaker Verification(SV), see Figure 1.1. For speaker identification, the aim is to answer the question: *Which speaker does this voice belong to?* The expected response is a choice of one speaker out of many possibilities. In SV, the query is: *Is the claimed speaker correctly identified?* The answer here is of a binary form; Accept or Reject the identity claim.

An SID system can be divided into two parts:

1. An *enrollment* phase
2. A *test* phase

During the enrollment phase, the voices of a set of reference speakers, i.e. speakers that the system will be expected to be able to identify, are recorded. This will be referred to as the training speech. The reference speakers provide speech both for training and testing purposes, though during the enrollment phase the training speech is used exclusively. The training speech undergoes some front-end processing that is described in Section 3.3. It is hereafter processed further by extracting certain *features* and creating feature vectors that are the input to the speaker modelling system. The optimal system parameters for the speaker models are obtained based on this training data. These speaker models are also referred to as voiceprints. If the SID task is *text dependent*, the training and test utterance must be identical. In the case where the identification process is *text independent*, the training and test utterances are different.
After the enrollment phase, the ability of the SID system to identify a speaker is evaluated during a test phase. In the test phase, the test speech is processed by the same front-end processing and feature extraction processes as were implemented to obtain the training data. This gives rise to test *patterns* that can be compared with the reference speaker patterns that were created during the enrollment phase. Given the test pattern, the reference model with the highest probability of having produced the test data is found and the test speech can be classified accordingly, using a predefined *decision logic*. The SID system thus identifies a speaker as speaker i if the probability of the i^{th} speaker model is the highest.

Speaker identification can therefore be said to consist of three parts that work interactively: Feature extraction, pattern matching, and classification. The identification process is divided into two phases, the enrollment phase and the test phase. A schematic representation of the process in the test phase is shown in Figure 1.2.

The details concerning feature selection and extraction methods will be presented in Chapter 3. One of the fundamental problems with feature extraction is the inevitable redundant data that is included in each feature set. This data is not useful for the identification of different speakers and can therefore be seen as noise within the feature set. It is not known which speech segments and which feature extraction methods are the ones that contain most of the highly speaker-dependent information content in the speech signal, which is why it is necessary to base the feature extraction methods on different criteria that are discussed in Chapter 3.

The input to the SID system can be further divided into either being *closed-set* or *open-set*. A closed-set problem is only expected to identify a speaker from the reference model database, while a system based on an open-set of input speakers must be able to identify a test sequence that does not match any of the reference speakers. This extra class

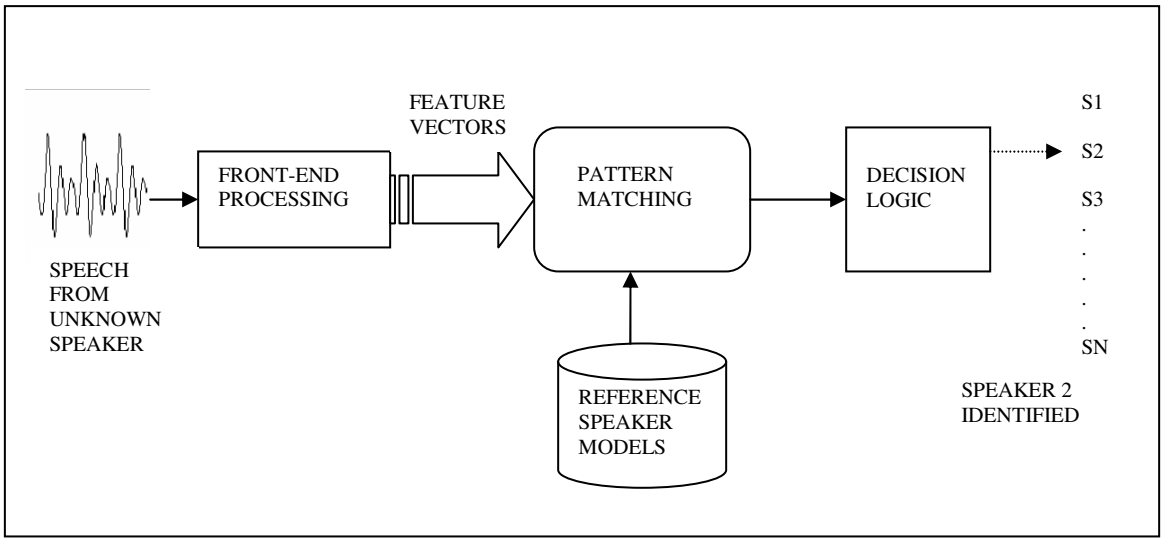


Figure 1.2: A basic Speaker Identification System, adapted from [9]

is referred to as the *impostor* class. The impostor class should be detected before the final pattern matching is implemented so as to spare computational time and minimize classification error in the classification process. The necessity of rooting out impostors is undoubtable given the amount of people (not to mention other sounds) that the wearer of a hearing instrument is exposed to everyday. Most of these would not be stored as reference speaker models. The impostor detection method is based on density estimation and is described fully in Chapter 5. A schematic representation of the process is shown in Figure 1.3, which is a modification of the basic outline shown in Figure 1.2.

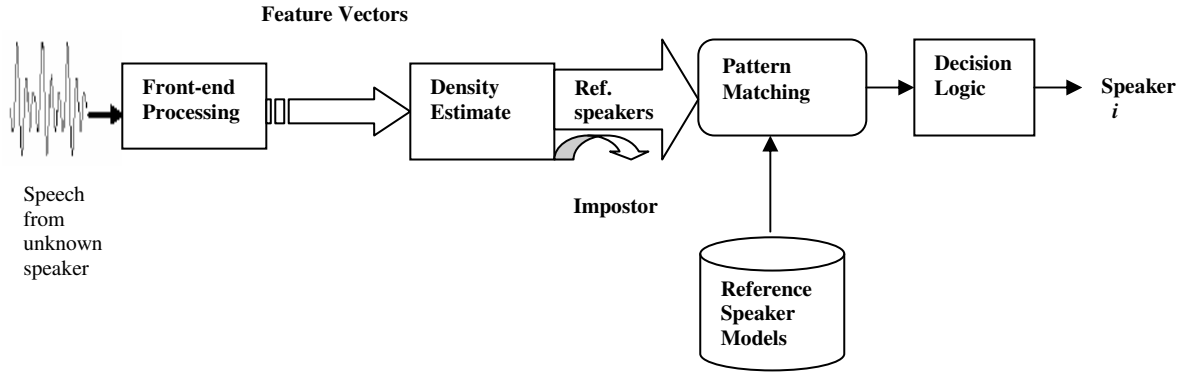


Figure 1.3: A Speaker Identification system with density estimation for impostor detection

In speaker verification, a decision has to be made between two hypotheses. The first

hypothesis (H_1) is that the voice is from the claimed speaker, the second is that the voice is from an impostor (H_2). Depending on a match score when comparing the test speech with the reference model, one of the two hypotheses is chosen. The decision is therefore either "Accept" (if H_1 is chosen) or "Reject"(if H_2 is chosen). The score matching can be done by implementing a usually empirically defined threshold value so that for threshold value Θ, the probability $p_i(\mathbf{y})$ that the test characteristics $\mathbf{y}$ belong to speaker i is used to classify speaker i as the correct speaker if $p_i(\mathbf{y}) > \Theta$, otherwise the claim is rejected. Two types of errors are thus associated with the SV system, the *false acceptance* rate that measures how often a speaker that should be rejected is accepted, and the *false rejection* rate that measures the amount of times a speaker that should be accepted is rejected. The threshold Θ can be adjusted according to the balance that is desired between these two types of error. Impostor detection is closely related to speaker verification as an impostor detector system rejects an impostor speaker for all reference speaker models in the system, thus implementing the binary decision making process several times for each test pattern.

The work that is presented in the remainder of this report is concerned with:

- A Speaker *Identification* system
- An *open-set* problem
- Input that is *text-independent*

1.2 Outline of Project

The application of automatic speaker identification in hearing instruments would enable the instrument to detect a certain speaker and adjust its speech processing setting accordingly, thus facilitating the use of such instruments. Although this is the long-term practical motivation for the work in this thesis, the actual implementation of such a system lies beyond the scope of this project.

Our work is first concerned with extracting certain *features* from speech signals. These features must reduce dimensionality and contain speaker-dependent information. As no standard feature has yet been found for the optimal solution of the SID problem, several possibilities will be explored. Several classifiers are also implemented and tested.
The report is divided into the following chapters:

Chapter 2 provides an introduction to the basics of speech production and speech modelling.

Chapter 3 goes into detail about the choice and extraction of feature sets. Explanations as to why certain features should provide good speaker-dependent representations of speech will be provided along with a description of how these features are obtained. Some of the features that are included are the Linear Prediction cepstral coefficients [9], the Perceptual Linear Prediction cepstral coefficients [62], the Mel-Frequency cepstral coefficients [5], pitch-related features [26] and the LPC residual [22].

Chapter 4 describes the concepts that are common for all the classifiers that are implemented. These include the decision rule, impostor detection and sentence classification using consensus over frame classification.

Chapter 5 provides a broad view on density modelling for speaker identification and a detailed description of the Mixture of Gaussians classifier [59] and its implementation for speaker identification and impostor detection.

Chapter 6 describes the structure and implementation of the k-Nearest Neighbour classifier [16].

Chapter 7 provides theory on the nonlinear neural network [15] and discusses its implementation.

Chapter 8 describes the ELSDSR database that is the source of all the speech data used in this thesis.

Chapter 9 provides the results of all the trials implemented with the different feature sets and classifiers, as well as an analysis of the effects on system performance of dividing feature sets into groups depending on speaker gender and on the voicing information of the frames.

Chapter 10 concludes on the findings of this thesis and gives suggestions for future work.

1.3 Use of the Database

The full description of the ELSDSR database that is used as a source of speech signals for this thesis is provided in Chapter 8. To facilitate understanding of the results that are already obtained in earlier chapters, a brief explanation is provided here. Of the 22 speakers that make up the database, 6 are used as the reference speaker set for most of the implementations presented in this report. Of these, there are 3 male speakers and 3 female speakers. The other speakers in the set can be used as impostors when the need to test for impostor detection arises. Each speaker has provided 7 training sentences. These are labelled as sentence a, b, c, d, e, f and g and are identical for all speakers in the database. Each speaker also provided 2 test sentences that are different for each speaker.

Chapter 2

Speech Signals

2.1 Speech Production

People are able to identify each other by listening to one another. Each person has a unique voice, but also a unique way of speaking that is not directly related to the actual quality of the voice. This is because speech is produced by a combination of the physiological traits and the learned characteristics such as intonation and language usage [17]. In the following we will examine the physiological aspects of speech production.

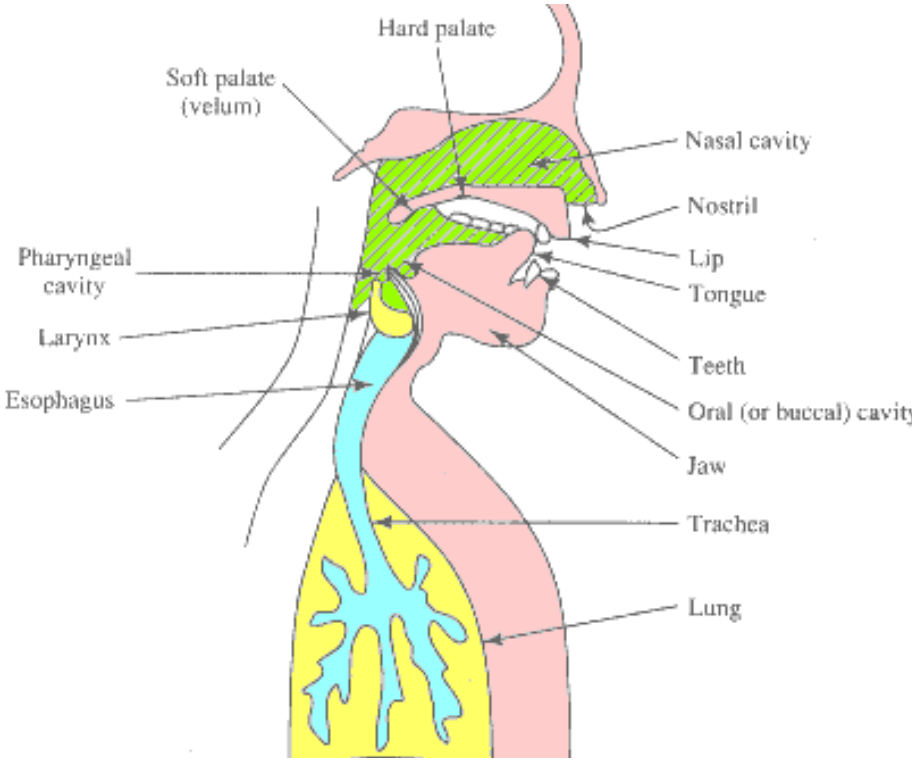


Figure 2.1: The human speech production mechanism, taken from [33]

Speech is produced by pushing air up from the lungs (see Figure 2.1) and through the

vocal cords (larynx), into the throat and the oral cavity to the lips. Sometimes the air flow is directed through the nasal cavity, too [33]. The vocal tract begins just after the vocal cords and ends at the input to the lips, see Figure 2.1. The nasal tract begins at the soft palate, or velum, which controls whether sounds are emitted through the oral cavity or the nasal cavity or both.

The air that is expelled from the lungs and pushed up through the trachea causes the vocal cords to vibrate. These resultant air pulses are the source of excitation of the vocal tract, and are often referred to as the glottal[1] pulses. The nature of the air flow through the glottis defines whether the speech is *voiced* or *unvoiced*. Voiced speech is produced by tensing the vocal cords periodically, causing the vibration of the air flow that passes through them and thus resulting in glottal pulses that are quasi-periodic [2]. The vibration rate of these glottal pulses is denoted as the *fundamental frequency*, F_0. The value of F_0 is dependent on the physical shape and positioning of the vocal cords. Voiced sounds that are produced by the periodic glottal pulses include all the vowels as well as the nasal consonants such as /m/ and /n/ [8].

The acoustic wave formed by the air flow from the lungs and past the glottis is altered by the resonances of the vocal tract and by the lip radiation. The vocal tract resonances depend on the length and shape of the throat and the position of the jaw, tongue and velum, ie. the physical attributes of the speaker. The vocal tract resonances are called formants [14]. The formant frequencies in voiced speech vary when different vowels are produced. This means that in voiced speech, the resulting waveform is not only dependent on the fundamental frequency, but also on the formant frequencies, where the former is a result of the physical attributes of the vocal cords and the latter a representation of the physical characteristics of the vocal tract.

When the vocal cords are relaxed and air is pushed through them, a constriction at some point along the vocal tract results in turbulence and the unvoiced sounds are produced. In this case the sound can be modelled as a stochastic process such as white noise. As the glottis does not vibrate to create these sounds, they do not contain fundamental frequency information though they do contain information pertaining to the vocal tract characteristics. The unvoiced sounds include virtually all consonants. One group of consonants that are produced in this way are the fricatives, produced by a turbulent flow of air which results in such sounds as 'sh' and 'f', while another group contains the stop consonants referred to as plosives, such as 'b' and 'p' [9].

2.2 Speech Modelling

The way that speech is modelled is often referred to as the source-filter model [2]. This is because the speech that is ultimately produced by the process that is described in Section 2.1 depends on two factors: The *source* characteristics of the speaker and the *system* characteristics. The system comprises of the vocal tract and lip radiation, i.e. physical attributes, while the source factors are the pulses produced by the air flow through the vocal cords and include such information as the fundamental frequency. The process by

[1] Glottis = vocal cords and the space between them

which the vocal tract causes changes to the glottal waveform can be modelled as a filtering of the source (glottal pulse) spectrum by the system (vocal tract) characteristics. This model is represented in Figure 2.2. The resulting speech signal thus has an output energy spectrum that is the product of the source function and the system transfer function. The source function is periodic in the time domain, and therefore has a discrete spectrum in the frequency domain [13]. This spectrum decreases with the square of the frequency, see Figure 2.2. The system filter function is approximately periodic and its peaks indicate the formant frequencies [2]. The resultant output spectrum has peaks that represent these formant frequencies formed by the vocal tract system characteristics. The vocal tract can be modelled as a cylindrical tube and it is the resonant frequencies of this tube that are the formants [39]. By changing the shape of such a tube, f.ex. by movement of the tongue, the positions of the resonant frequencies are shifted, thus allowing different sounds to be produced.

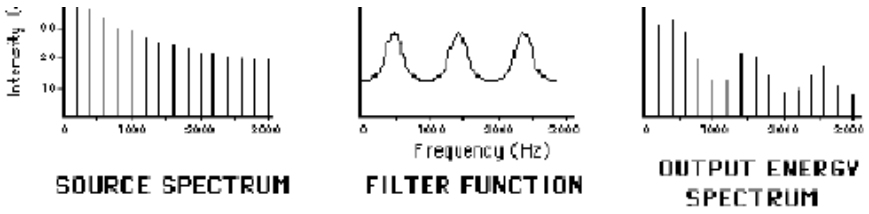


Figure 2.2: Source Spectrum, System Filter Function and Output Spectrum, taken from [11]

At the core of the source-system speech model is the fact that the source and filter spectra are independent of one another. The power of this model is therefore that it opens the possibility of separating the spectra and modelling just the filter function which can reliably be found in most speech segments, as will be discussed in Chapter 3. The complete speech production model is shown schematically in Figure 2.3.

The source-system model can be represented mathematically by referring to Figure 2.3. In discrete time, we let $u(n)$ represent the excitation signal, which can be the glottal waveform or turbulence or both, depending on the sound being produced. For voiced speech, the excitation signal is quasi-periodic with fundamental period T_0. (The corresponding rate of vibration is the fundamental frequency, $F_0 = \frac{1}{T_0}$). For unvoiced speech the excitation signal is modelled as noise [2]. The vocal tract is represented by the filter function $H(z)$ while the effect of lip radiation on the speech signal is denoted as $R(z)$. In the time domain, this leads to the following simplified mathematical model for speech production:

$$s(n) = u(n) \otimes h(n) \otimes r(n) \tag{2.1}$$

In the frequency domain, this can be written as:

$$S(z) = U(z) \cdot H(z) \cdot R(z) \tag{2.2}$$

$U(z)$ is the excitation spectrum, $H(z)$ is the vocal tract spectrum and the impedance caused by the lips is approximated by $R(z)$ [1]. The transformation to the frequency

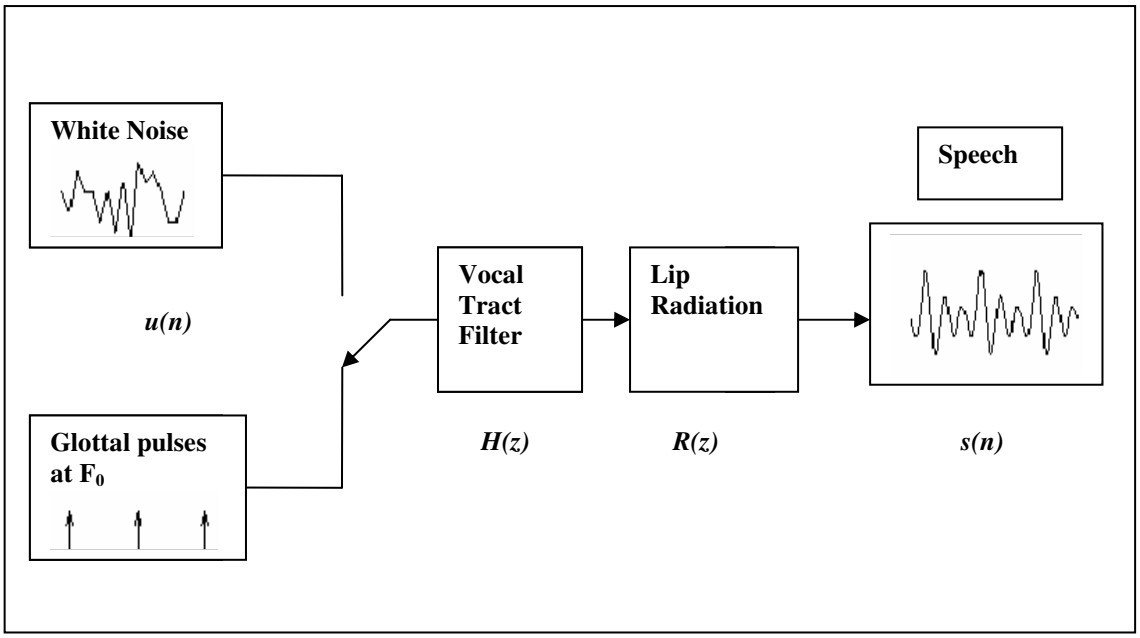


Figure 2.3: Source-Filter Model of Speech Production, adapted from [38]

domain is defined by the Fourier transform [13], given by:

$$X(z) \equiv \sum_{n=0}^{N-1} x(n) z^{-n}, \qquad z = e^{j\frac{2\pi}{N}} \tag{2.3}$$

By using the source-filter model we can derive several different types of features, either in the time domain or in the frequency domain. This means that for some features (such as those involving the fundamental frequency), it is possible to analyze the speech signal in the time domain, while it is necessary to transform the signal to the frequency domain in order to enable the extraction of other features, f.ex. the Mel-Frequency cepstral coefficients. The choice of feature sets also depends on whether the aim is to model the excitation signal (the source) or the vocal tract filter (the system).

Chapter 3

Choosing and Extracting Feature Sets

3.1 Representing Speech

The question of interest when speech is to be processed for the purpose of speaker identification is: *What is it in a speech signal that conveys the speaker's identity?* The attempt to answer this question forms the basis of the first part of the speaker identification task - the selection of certain *features* from the speech signal. These features are grouped into feature vectors that serve the purpose of reducing dimensionality and redundancy in the input to the SID system, while retaining ample speaker-specific information. As the presence of irrelevant information with regards to speaker discrimination is a common problem for all feature sets, it is the topic of ongoing research that strives to determine feature sets of reduced complexity that can be applied to speaker identification.

This research is significant as the performance of a speaker identification system depends heavily on the selection of the feature sets. Apart from being unique for each individual speaker, attributes that make features desirable are [2]:

- Frequent and natural occurence in speech
- Simple to measure
- Not varying over time, ie. robust against ageing effects
- Not sensitive to illness that may affect speech, e.g. a cold
- Independent of specific transmission characteristics and background noise, e.g. microphone characteristics
- Difficult to imitate

To date, there is no feature set that satisfies all of the above conditions, so it is necessary to extract several feature sets and observe how well the classification can be performed for each one. A feature extraction method is based on certain criteria, though. Firstly, it is of vital importance that the features can be extracted reliably. This is a common factor for all feature extraction methods.

The exact nature of the feature set depends on what part of a speech signal the features are

expected to represent and thus what type of information is to be extracted. This is why feature sets can be grouped as being *source* based features or *system* based features. In Chapter 2, the source is described as being the actual sound wave that is transmitted from the diaphragm through the glottis and so these features are concerned with determining the characteristics of the vocal cords, where this waveform is shaped. The particularities of an individual's speech in the form of linguistic information [17](behavioural style of speaking) contain a high level of speaker-specific information and are known as the high-level features. These features are difficult to extract automatically from the speech signal and lack reliability, especially when there is not a lot of training and test material available as they are calculated from relatively long segments of speech. In this thesis, the features representing the source characteritics are mostly limited to estimating the fundamental frequency. This is a basic measurement that defines the time between the series of vocal fold openings that are executed when a voiced word or sound is being produced, and can be extracted from short segments of speech.

The extraction of system based features, or low-level features, has an intrinsic advantage over the source feature extraction methods. They can be extracted through simple acoustic measurements and where the glottal pulse is exclusively present in voiced speech, the system characteristics are also present in unvoiced segments of speech. This means that low-level features can be extracted easily and reliably, especially when using speech from the ELSDSR database as these signals are not contaminated by noise and no mismatch between training and testing material exists. The system characteristics can be extracted for the vocal tract, the nasal cavity and the lip radiation, though it is common to focus on the formant frequencies (see Section 2.1) of the vocal tract.

For each feature extraction method, it is therefore necessary to know exactly what is being extracted so as to avoid imprecisions and ambiguity. As phase information in a speech signal is not significant for discrimination between speakers, it can be omitted in order to simplify calculations, i.e. the magnitude of the spectrum of the speech signal is used. Additionally, knowledge of the filtering of speech in the ear can also be applied in the derivation of features. The use of these techniques are mentioned when they are used in conjunction with a particular feature set.

The features that will be extracted are divided into two groups:

Source Features -
Features that are concerned with modelling the original sound wave that passes through the glottis. The most feasible parameter that can be determined is F_0. In [3], the values of F_0 are given as approximately:

- 125Hz for men
- 250Hz for women
- 300Hz for children

System/Filter Features -
These features model the filter characteristics of the vocal tract that can be derived from

information contained in voiced and unvoiced speech. This information includes the formant frequencies that are predominantly present in vowels. The system features reflect the physiology of the speaker.

The feature sets that will be extracted in this thesis and their grouping are listed in Table 3.1.

Source based features	*System based features*
Fundamental Frequency	Linear Prediction Cepstral Coefficients
LPC Residual	warped Linear Prediction Cepstral Coefficients
	Perceptual Linear Prediction Cepstral Coefficients
	Mel-Frequency Cepstral Coefficients

Table 3.1: List of source- and system-based features

The traditional and to date most reliable way to represent speech for recognition purposes is by modelling the system characteristics. In the source-filter model, this means that the source features are not used to identify the speaker. The most commonly used system-based features are the *cepstral coefficients*. The two types of cepstral coefficients that are widely applied are:

1. Linear Predictive Cepstral Coefficients (LPCC) [5]
2. Mel-frequency Cepstral Coefficients (MFCC) [21]

The derivations of these coefficients are presented in Sections 3.5 and 3.8, respectively.

As it is assumed that the system and source characteristics are uncorrelated, it is worthwhile to study the influence each kind of feature set has on the SID system's performance. An analysis into the possibility of classifying speakers based on only selected frames that contain a high level of speaker dependent information is commenced in Section 3.10 and is completed in Chapter 9. The remainder of this chapter is concerned with the selection and extraction of the features listed in Table 3.1.

3.2 Spectrographic Analysis

Before describing the extraction of the feature sets, a spectrograhic analysis is carried out. A spectrogram is a short-time Fourier transform (see Eq.(2.3)) that shows the energy of a signal as a function of positive time and frequency [25], thus allowing us to locate areas of energy in the speech signal. It only represents the amplitude of the speech signal, as no phase information is retained. This is not perceived as a problem, though, as phase information is not necessary for speaker identification purposes [1]. The short-time Fourier transform is computed for each window of a speech signal that has a preset length corresponding to N samples. As time and frequency are inversely proportional, a longer window in the time domain yields a narrowband spectrogram in the frequency domain, and a short time window results in a wideband frequency analysis. In Figure 3.1, the

wideband and narrowband spectrograms for a female speaker for training sentence *a* are shown, while in Figure 3.2 the waveform and spectrograms for a male speaker are shown for the same sentence.

The fundamental frequency is the zero'th harmonic and contains the highest level of energy, to be followed by a few harmonics that represent the first formant, second formant, and so on. In the narrowband spectrograms (bottom plots of Figures 3.1 and 3.2), the fundamental frequency and its harmonics are easily observable. The wideband spectrogram is seen to have a poor frequency resolution and the fundamental and formant frequencies cannot be discerned here. Notice the increased speech activity that can be observed in the higher frequency area of the spectrogram for the female speaker in Figure 3.1. These show a tendency to be gender specific, as they are for the most part missing in Figure 3.2, where the energy level above 4kHz is almost non-existant. The spectrographic analysis leads to the conclusion that when using a feature extraction method in the frequency domain, the fundamental frequency information must be extracted using a time frame that cannot be chosen arbitrarily.

3.3 Preprocessing

Prior to the feature extraction phase, the speech signal that is used either as training or as test input data to the SID system is preprocessed. The preprocessing steps are described here and are implemented as the initial step in all the feature extraction methods that follow.

- **Preprocessing step 1: ADC**
  An analog-to-digital converter converts the analog speech signal to a digital signal at a sampling frequency of F_s. All the speech signals in the ELSDSR database are sampled at $F_0 = 16$kHz.

- **Preprocessing step 2: Pre-emphasis**
  A FIR high-pass filter with the transfer function shown in Eq.(3.1) is used to flatten the signal spectrum.
  $$H(z) = 1 - az^{-1} \tag{3.1}$$
  where a usually lies in the interval $0.9 \leq a \leq 1.0$ [8]. The high frequencies of the speech signal formed in the vocal tract are attenuated as the sound passes through the lips [1]. By dampening some of the low-frequency information in the resultant speech signal a more equal balance between high- and low- frequency information is achieved in the spectrum.

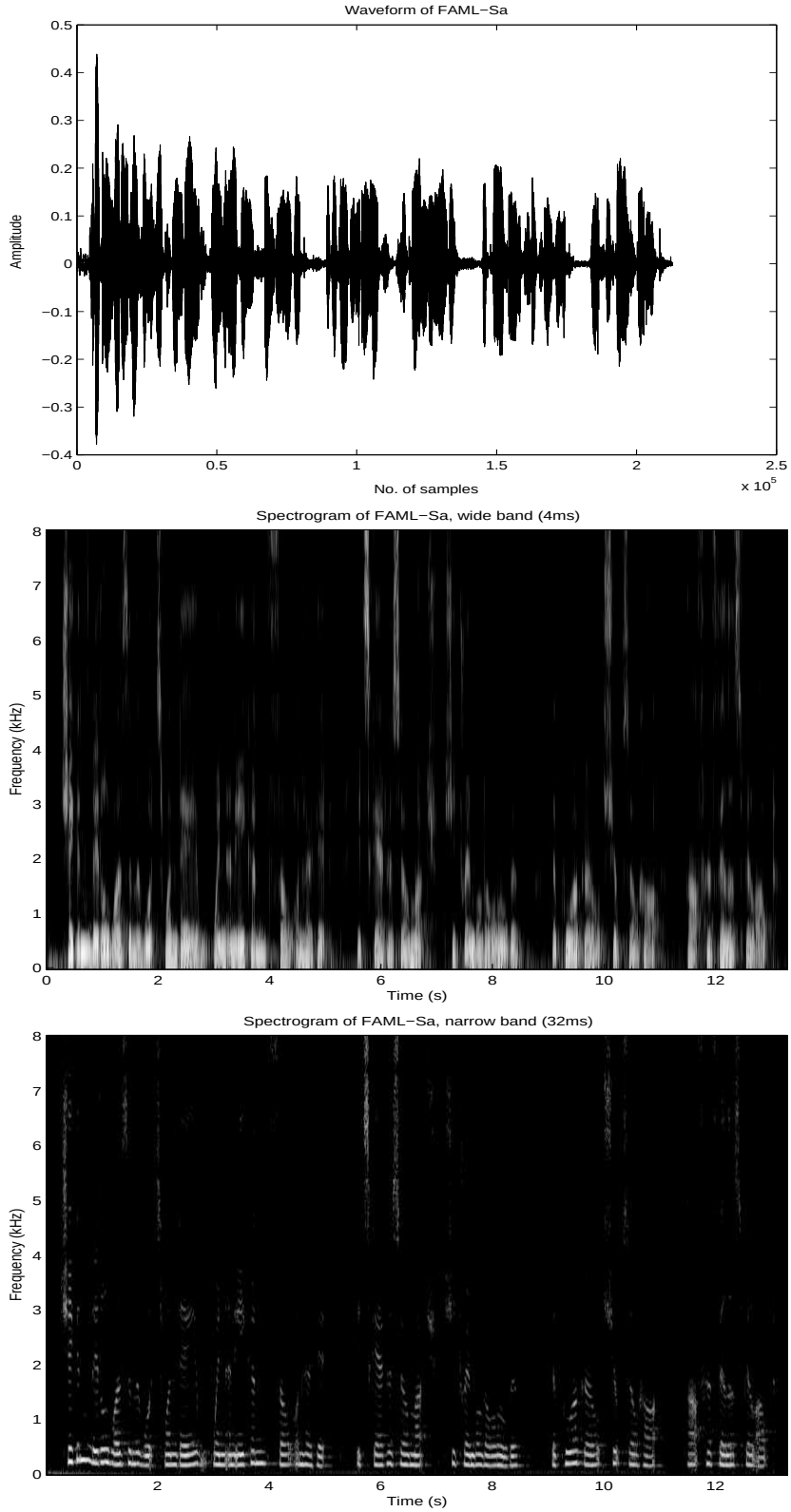


Figure 3.1: The waveform and spectrograms of FAML_Sa: Wideband spectrogram uses a window length of 4ms and at a sampling rate of 16kHz that corresponds to a window of 64 samples, while the narrowband spectrogram uses a window length of 32ms, ie. 512 samples for F_s=16kHz

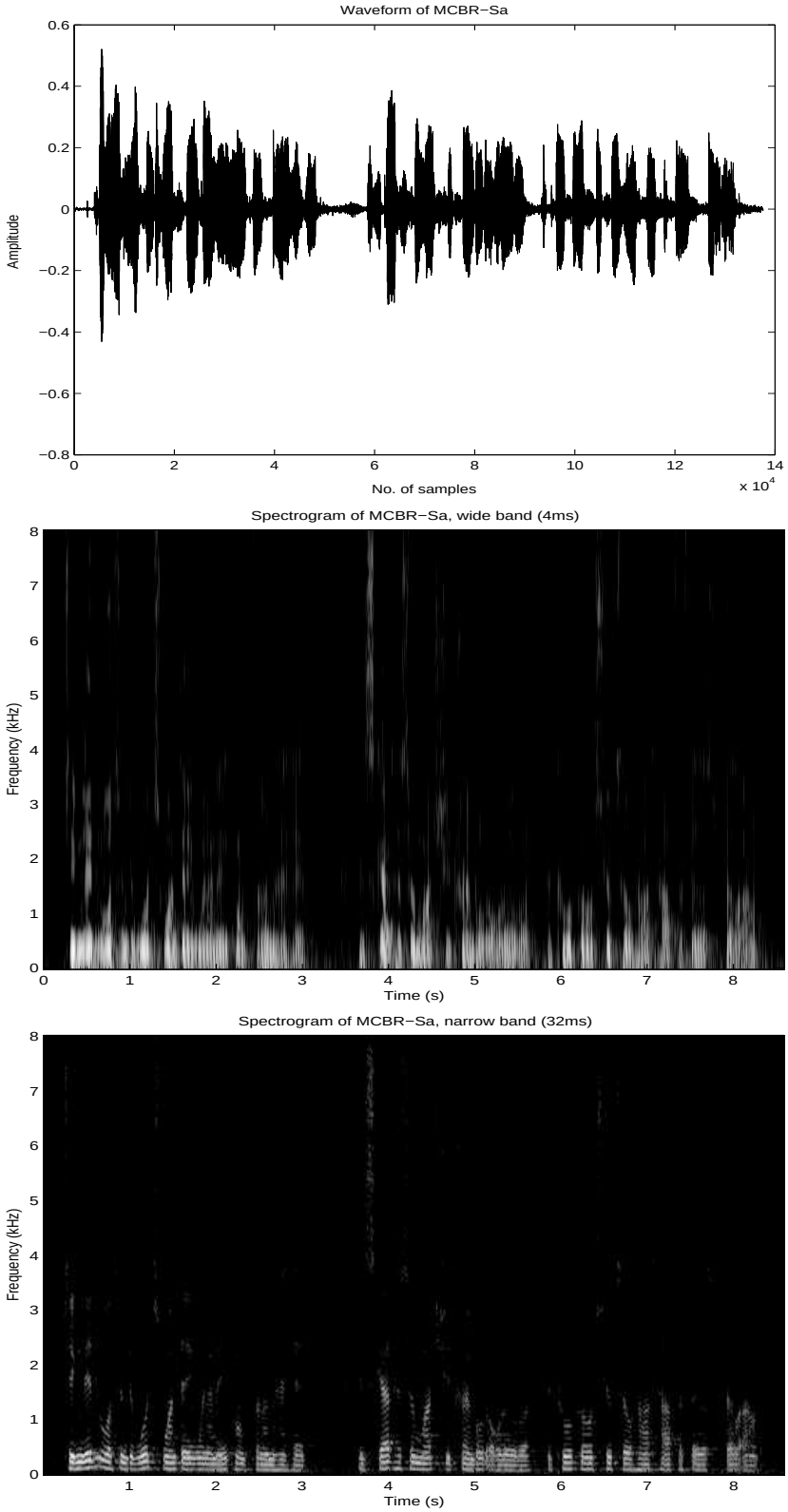


Figure 3.2: The waveform and spectrograms of MCBR_Sa: Wideband spectrogram uses a window length of 4ms and at a sampling rate of 16kHz that corresponds to a window of 64 samples, while the narrowband spectrogram uses a window length of 32ms, ie. 512 samples for F_s=16kHz

- **Preprocessing step 3: Windowing**
  The pre-emphasized signal is divided into short frame blocks, and a window is applied to these frames. The frame length can vary, but based on empirical results, is often chosen from 20 to 30ms [5]. This length depends on the specific feature extraction method that is applied. For the speech signals in the ELSDSR database, a frame length of 30ms corresponds to frames containing 480 samples. Framing using this length and an overlap of 10ms (160 samples) is implemented. The window function that is applied is preferably not rectangular, as this can lead to distortion due to vertical frame boundaries [8]. The windowed speech waveform for frame j is defined as:

$$s(n) = w(n) \cdot s_j(n), \qquad n = 0, 1, 2, ...N - 1 \tag{3.2}$$

  where $w(n)$ is the window function.

  A common choice for the non-rectangular window is the Hamming window [1]. The mathematical function of the Hamming window is shown in Eq.(3.3) and the Hamming waveform is shown in Figure 3.3.

$$w(n) = 0.54 - 0.46 \cos \frac{2\pi n}{N-1}, \qquad n = 0, 1, 2, ...N - 1 \tag{3.3}$$

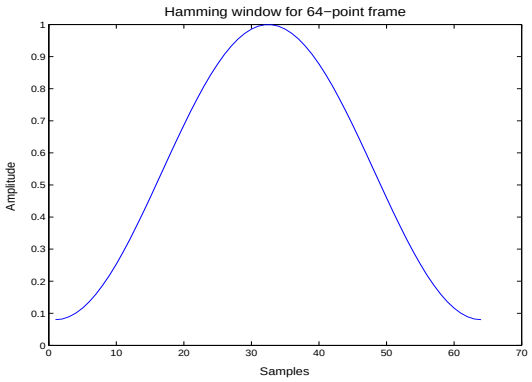


Figure 3.3: Hamming window

3.4 Fundamental Frequency Estimation

One of the source based features that are extracted is the fundamental frequency, F_0. As described in Section 2.1, F_0 represents the periodicity of the voiced sounds, these being predominantly vowels. Although pitch and fundamental frequency are often assumed to mean the same thing, it must be pointed out that this is not the case. It has been established that pitch is the human ear's perception of a sound's fundamental frequency, which is not identical to the actual fundamental frequency of the sound being produced [1]. The methods of fundamental frequency extraction that will be presented in the following are all concerned with the true fundamental frequency value and not the perceived pitch value. A number of different F_0 estimators have been developed to date and extensive work is ongoing in this field [50]. The challenge for all these estimators lies in the imperfect nature of the periodicity of a segment of a speech signal. In addition to the fact that only certain, voiced, sounds are periodic, even these waveforms are only quasi-periodic, causing estimation of the periodicity to be difficult. The formant frequencies may also confuse the F_0 estimation process.

To illustrate the difference between the periodic and stochastic segments of a speech signal, two frames of length 30ms are extracted from a training sentence for Speaker 1. One frame contains a voiced, quasi-periodic, segment of speech, another a low-energy, unvoiced segment of speech. These two frames can be seen in Figure 3.4.

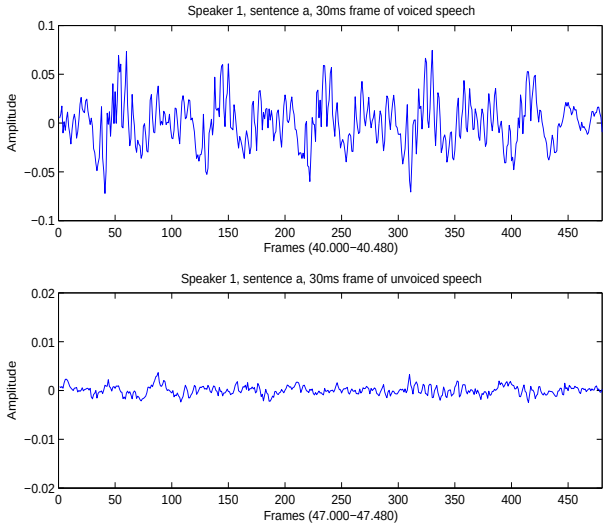


Figure 3.4: Voiced and unvoiced segments of speech from Speaker 1

Alternative methods of finding the fundamental frequency can be divided into two groups: the Time-Domain methods and the Frequency-Domain methods.

3.4.1 Time-Domain methods: The Autocorrelation Method

F_0 can be extracted by using the autocorrelation method [36]. The autocorrelation function of a signal is a representation of the amount of overlap contained within the signal,

at different time lags. At a time lag of zero, the maximum of the autocorrelation function is found. The estimated autocorrelation function of a speech signal $s(n)$ is shown in Eq.(3.4):

$$R_{ss}(\tau) = \frac{1}{N} \sum_{n=0}^{N-\tau-1} s(n)s(n+\tau) \tag{3.4}$$

The autocorrelation function of a periodic signal is also periodic [13]. For a perfectly periodic waveform, this is because the signal is repeated at a certain time lag at which the autocorrelation function has its maximum peaks. The R_{ss} function thus has a periodicity P that results in peaks at samples 0, $\pm P$, $\pm 2P$, ... For the analysis of a speech signal, the first peak of the autocorrelation function, found at the smallest non-zero time lag, indicates the fundamental period of the speech waveform.

In Figure 3.5, the autocorrelation function of the segment of speech shown in the upper plot of Figure 3.4 is shown.

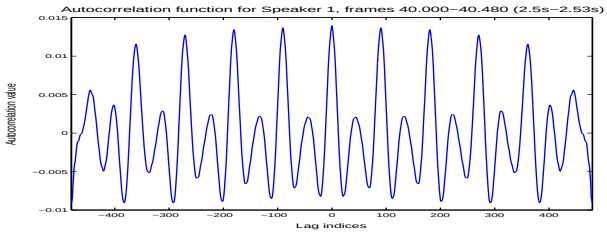


Figure 3.5: The autocorrelation function of the voiced segment from Speaker 1

From Figure 3.5, the smallest lag index that yields a considerable peak is found at roughly $\tau = 90$, corresponding to a periodicity of 178Hz for $F_s = 16$kHz. As Speaker 1 is a woman, this is a possibility. A lower bound on the range of τ indices to be included in the search for the maximum peak is necessary to avoid the risk of always finding this peak at $\tau = 0$. The lower bound is set as the τ index for the first dip after the maximum peak at the origin, while the upper bound is the length of the autocorrelation function for a frame. As the function is symmetric, only the positive indices need to be searched.

A number of factors can reduce the ability of the autocorrelation method to determine F_0. The quasi-periodic nature of the waveform may cause the higher order harmonics of the fundamental period to form additional, smaller, peaks in the autocorrelation function. The larger peaks must thus be differentiated from these. One procedure that attempts to do away with eventual ambiguity due to the formant frequencies is the center-clipping autocorrelation method [36]. The first and last third of the signal segment are analyzed so that the smallest of the peak amplitudes sets a threshold value. The clipping factor is set to 60% of this threshold. The parts of the speech segment that fall below this value are removed, thus flattening the speech spectrum and reducing the complexity of the resulting autocorrelation function.

The autocorrelation clipping algorithm can be extended to include a *voiced/unvoiced* decision-making functionality. Each block of the speech signal is labelled as being voiced or unvoiced speech. The value of the autocorrelation function is compared to a pre-specified threshold so that all frames that do not yield a value above the threshold are

classified as being unvoiced. Although this cannot be used as a feature set for speaker identification, the interest here lies in establishing whether the classification of a frame shows a dependency on whether the frame is voiced or unvoiced.

To facilitate the implementation of an automatic method that chooses the correct peak, the blocks of speech that are used to extract F_0 must be long enough for the zero'th harmonic to be found, i.e. two cycles of the fundamental period must be present. As the range of some of the formant frequencies overlap that of the fundamental frequency, it is not possible to implement filtering that eliminates the possibility of estimating a formant frequency instead of F_0.

The dependency of the F_0 estimate on the length of the blocks of speech segments used is analyzed and the results are listed in Table 3.2. The clipping value is set at 0.6 and the 7 training sentences from each speaker in the reference speaker set were used in order to obtain the median values of the F_0 estimates, given in Hz, over all the voiced frames in the sentence. The labels FAML, FDHH, and so forth identify each speaker, the first letter "F" denoting women and "M" denoting men, as explained in Chapter 8.

frame length	FAML	FDHH	FEAB	MASM	MCBR	MFKC
64ms	190	188	195	131	107	119
32ms	188	188	195	131	105	116
16ms	188	188	192	132	97	104

Table 3.2: F_0 for varying frame lengths and clipping factor 0.6

The reduction of frame length in the time domain corresponds to an increase in the range of frequencies that are included in the F_0 estimation analysis. When the frame length is decreased to 16ms, the estimates for the last two male speakers deviate from the previously found values. This may be attributed to the short length of the time frame, which does not allow the completion of two full cycles of the periodic waveform and so results in a less precise estimation of the the fundamental frequency. The frame length must thus be set to at least 32ms in accordance with these results and those obtained from the spectrographic analysis in Section 3.2.

From Table 3.2, it is clear that there is a significant difference between the estimates for the female and the male speakers. This could be useful for gender separation of speakers, which could then greatly simplify the classification process as the number of speakers to identify would be reduced. This possibility is studied in Chapters 6 and 9.

3.4.2 Time-Domain methods: The YIN Estimator

The YIN estimator [48] was developed by Alain de Cheveigné and Hideki Kawahara in 2001. It is based on the autocorrelation method of fundamental frequency estimation, but introduces a number of modifications to circumvent many of the weaknesses that alternative autocorrelation methods, including the center-clipping autocorrelation method, suffer from, thus making the YIN estimator more precise than these.

The first step in implementing these modifications is the replacement of the autocorrelation function of Eq.(3.4) by a difference function. The speech signal $s(n)$ is modelled

as a periodic function with period T, so that the difference between the signal at time n and at time $n+T$ is zero for all n. The square of this difference is thus also zero and so a function, $d_n(\tau)$, can be defined as being the average of the square of the aforementioned difference:

$$d_n(\tau) = \frac{1}{N}\sum_{n=1}^{N}(s(n) - s(n+\tau))^2 \tag{3.5}$$

This difference between the waveform at $s(n)$ and the delayed waveform at $s(n+\tau)$ must be *minimized* in order to determine eventual periodicity in the signal. This is in opposition to what is done when using the autocorrelation function, as in the latter case the product of the original and delayed waveform must be *maximized* in order to establish periodicity. Otherwise, the difference between $d_n(\tau)$ and $R_n(\tau)$ is not significant. The vital improvement on the autocorrelation method is described below.

With the difference function, a problem that remains is that the voiced parts of the speech signal are quasi-periodic as opposed to perfectly periodic and thus $d_n(\tau)$ is only zero for $\tau = 0$. The average of the difference function is therefore evaluated so that each new value of $d_n(\tau)$ is compared to its average over smaller-lag values. Where this decrease is considerable, causing a dip, the period is assumed to have been found. The new, averaged difference function is denoted as $\tilde{d}_n(\tau)$ and is called the *cumulative mean normalized difference function*:

$$\tilde{d}_n(\tau) = \begin{cases} 1, & \tau = 0 \\ \frac{d_n(\tau)}{\frac{1}{\tau}\sum_{j=1}^{\tau}(d_n(j))}, & \tau \neq 0 \end{cases} \tag{3.6}$$

One of the advantages of using $\tilde{d}_n(\tau)$ is that this function starts at 1 and not zero. This effectively removes the need to set a lower bound on the range of admissible lag values, as there no longer exists the risk that the difference function is minimized at zero lag. There is thus no upper limit for the fundamental frequency search range. This makes the YIN estimator effective especially when working with music, where higher frequencies than those that are predominant in speech may occur. The advantage of using YIN for speaker identification is that it may provide more precise estimations of the fundamental frequency than many other time domain algorithms are capable of.

At the core of this higher level of precision is the cumulative mean normalized difference function of Eq.(3.6). With its implementation, a threshold is set so that the smallest time lag for which the dip in $\tilde{d}_n(\tau)$ that falls below this threshold is accepted as being the dip that denotes the signal segment periodicity. In the absence of any values falling below the threshold, the global minimum of $\tilde{d}_n(\tau)$ is chosen. The YIN estimator also makes use of parabolic interpolation and a best estimate method in order to refine the period estimation process. The YIN estimator article (de Cheveigné and Kawahara,[48]) provides a detailed description of this sequence of modifications to the original autocorrelation method as well as derivations of additional measures that counter the effects of amplitude variation, frequency variation, and the presence of various types of noise.

In [48], it is recorded that the YIN F_0 estimation is substantially more precise than a variety of other autocorrelation-based estimators, so the YIN estimator will be used as

one of the methods that estimate the fundamental frequency for each speaker in the reference set. As the YIN algorithm does not make voiced/unvoiced decisions, these will be obtained from the autocorrelation with clipping algorithm.

3.4.3 Frequency-Domain methods: Real Cepstrum Method

The speech in the ELSDSR database was recorded in conditions that were largely free of noise and thus the speech data has a high signal-to-noise ratio. This, however, will not be the case when a hearing instrument is exposed to daily sounds in all kinds of environments. The time domain fundamental frequency estimation methods risk not to be robust for low signal-to-noise conditions, meaning that the autocorrelation method and even the YIN estimator may lack reliability. In order to eventually obtain more reliable estimations of F_0 a frequency-domain method for F_0 estimation is implemented. The selected method is the Real Cepstrum method [1].

The following steps are implemented in order to extract an estimate for F_0 in the frequency domain: first, the frequency spectrum of a speech segment is calculated using the Fourier transform of Eq.(2.3). As described in Section 2.2, the convolution of the excitation signal with the filter response becomes a multiplication in the frequency domain. By taking the logarithm of this function, an additive (linear) relation is obtained instead of a multiplicative (nonlinear) one:

$$S(z) = U(z) \cdot H(z) \tag{3.7}$$

$$log(S(z)) = log(U(z)) + log(H(z)) \tag{3.8}$$

$U(z)$ is the excitation spectrum and $H(z)$ is the simplified system filter response. The resultant $log(S(z))$ is reduced to a more usable scale than the original spectrum is, while maintaining periodicity in the frequency domain if the original speech segment is periodic. This periodicity indicates the fundamental frequency of the speech segment. By taking the inverse Fourier transform of the $log(S(z))$, the result is referred to as the *cepstrum* of the signal and is measured as a function of *quefrency*. The word "*cepstrum*" is a play on the word "*spectrum*", and "*quefrency*" on "*frequency*". The fast variations that are due to the excitation from glottal pulses are represented at high quefrency values, while the slower variations that are attributed to the vocal system resonances are found at the lower end of the quefrency scale. In association with this, a separation of the fast variations from the slow variations can be implemented by a filtering technique referred to as *liftering*, a corresponding play on the word "*filtering*". Low-time liftering is analogous to low-pass filtering, and where in the latter the higher frequencies can be sorted from a spectrum, in the former the variations at higher quefrencies can be sorted. Precise separation is only possible in ideal conditions, though, that cannot be assumed to prevail in practical applications, where overlap often arises between the fast glottal variations and the slow system variations on the quefrency axis.
The quefrency scale is very closely related to the time scale, and its unit is seconds. The fundamental frequency is extracted from the *real cepstrum*, where the periodicity of the original waveform is indicated by a dominant peak. The complex cepstrum is not used because phase information can be discarded for F_0 estimation, thus reducing computational complexity. To summarize, the real cepstrals are derived as the inverse

Discrete Time Fourier Transform(DTFT) [1] of the logarithm of the real DTFT of the speech signal:

$$c(n) = F_{DTFT}^{-1}\{log|F_{DTFT}\{s(n)\}|\} \tag{3.9}$$

In Figure 3.6, the real cepstrum of a section of sentence a from Speaker 1 is shown as a function of quefrencies. The search range has a lower bound set at 40ms on the quefrency scale, so that the frequency range is kept below 400Hz. The lower bound in the frequency range is set at 50Hz.

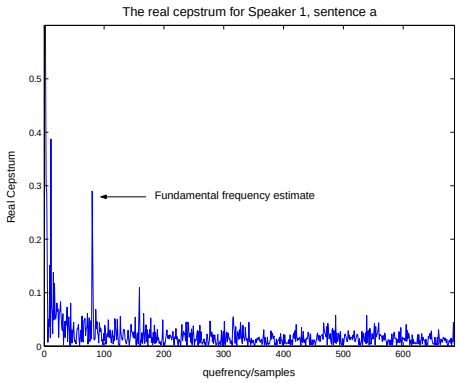


Figure 3.6: The Real Cepstrum and F_0 estimate for Speaker 1, sentence a

From Figure 3.6, the maximum peak is seen to be situated at the quefrency at sample index of approximately 85, which corresponds to an estimate of $F_0 = 188$Hz for Speaker 1.

The length and type of the window used to create blocks of speech signal to be analyzed by the real cepstrum method is significant. As with the time-domain methods, it is important that the block be long enough to allow two entire cycles of the periodic waveform. Once the window meets the necessary requirements, it is relatively easy to extract the peak that indicates F_0.

3.4.4 Comparison of Fundamental Frequency Estimators

Using each of the three fundamental frequency estimators that are discussed in Sections 3.4.1-3.4.3, an average F_0 for each speaker in the reference set is obtained. The estimation of the fundamental frequencies of all six reference speakers is implemented by first estimating a value for each sentence - all 9 sentences from each speaker are used, including both training and test data. A median value calculated over the estimate for every frame in each sentence is used for the real cepstrum and autocorrelation methods, while the output of the YIN estimator yields a "best" estimate of F_0 for the entire sentence. This estimate is determined at the dip in the cumulative mean nomalized difference function discussed in Section 3.4.2 that is found at the minimum lag value. As the other two F_0 estimators return an estimate for F_0 for each frame, the median must be calculated to provide one estimate for the entire sentence. For each speaker, the average F_0 is found as the mean of the estimates over all 9 sentences. The results for all three estimators are shown in Figure 3.7.

The YIN estimator was implemented with default parameters, as numerous trials with varying threshold values and frame lengths yielded no significant change in the results.

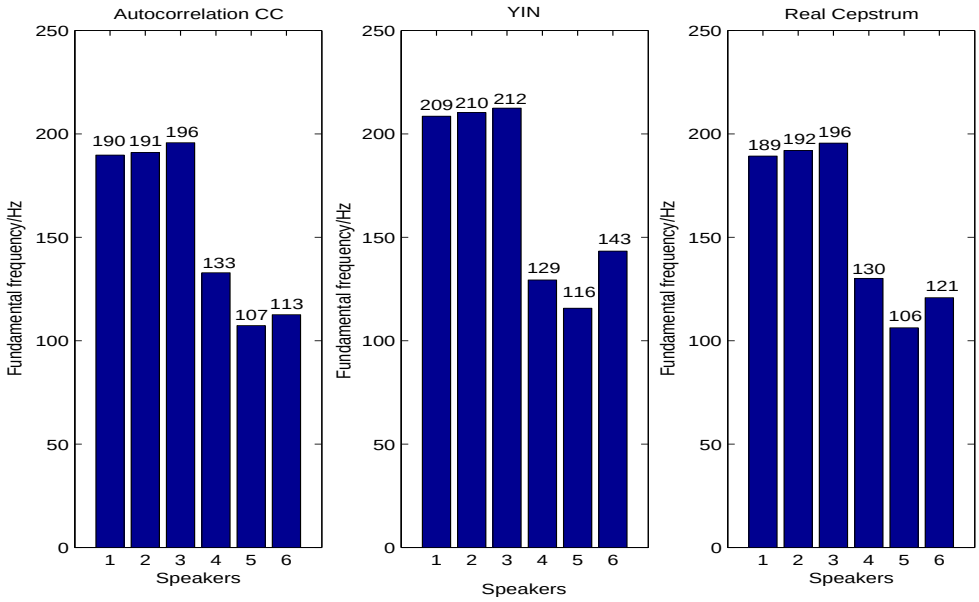


Figure 3.7: Fundamental frequency estimation for Autocorrelation CC, YIN and Real Cepstrum methods

The lower frequency bound is set at $F_{0,min} = 30$Hz and the window length set to the sampling frequency divided by this value, see Eq.(3.10), as this is assumed to be enough to determine the signal periodicity. For the speakers in the ELSDSR database, this gives a window length of $W = 33ms$.

$$W = \frac{F_s}{F_{0,min}} \tag{3.10}$$

The optimal frame lengths for the other F_0 estimators were determined by trial and error: 30ms for the autocorrelation with center clipping method, and 64ms for the real cepstrum method.

Figure 3.7 shows that the YIN estimator has a tendency to produce higher estimates of the fundamental frequency than the other two estimators. The results from all three estimators, however, show that while the differences between gender groups are large - this can be seen as the first 3 speakers are women, the last 3 men - the variation within each gender group is very small, especially for the women, and it is thus unlikely that this feature is well suited for the general speaker identification task. According to the documentation in [48], the deviance between the fundamental frequency estimates are larger between the YIN estimates and the other two sets of data because YIN is more precise.

Results based on all feature sets and an analysis to determine whether the voiced/unvoiced decisions influence system performance will be discussed in Chapter 9. The time required by each method to return a fundamental frequency estimate is considered here. Averaged over all 7 training sentences and both test sentences for each speaker, these times are shown in Figure 3.8. The training and testing data sets are kept separate because of the difference in length of the sentences contained in each set. The results are averaged over

all 6 reference speakers.

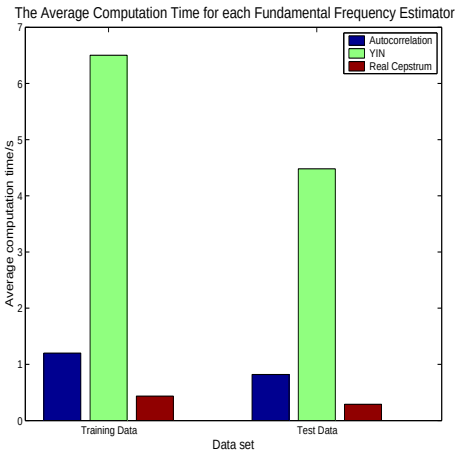


Figure 3.8: The average computation time for each fundamental frequency estimator

From Figure 3.8 F_0 estimation is seen to be most rapid using the real cepstrum method, while the YIN estimator requires a significant increase in computational time when compared to the other two methods. The choice of estimator, however, will be left until further trials in Chapter 9 have been completed.

The fundamental frequency that has been determined so far has been a single value, averaged over the sequence of frames that combined constitute sentences from each speaker. The way that the fundamental frequency changes as a function of time when a speaker is talking is not represented in this analysis, though this may prove to be interesting as a possible feature for speaker identification. In Figure 3.9, the trajectories of fundamental frequency estimates for entire sequences of frames are shown. The two top speech sequences are of women's voices and the two bottom plots are of male voices. The sentence used was arbitrarily chosen, though identical for all speakers to ensure that the trajectories depicted are comparable. Sentence d is used. The original scaling has not been modified in order to achieve a uniformity that would facilitate the comparison of these plots, as the differences are in some places so significant that this was not feasible. It is precisely these differences, though, that lead to the observation that the range of each speaker's fundamental frequency varies considerably, f.ex. the F_0 values for Speaker 1 have a range of roughly 300Hz, while for Speaker 6 they vary within a range of only approximately 130Hz. The number of frames is not equal for all speakers and this shows that the speed with which each speaker utters sentence d is speaker dependent. Despite these differences, it is easily seen that a large amount of each speaker's fundamental frequency estimates lie within the intervals that are defined by the fundamental frequencies of the other speakers. This leads to the assumption that there is little possibility that the trajectory of the fundamental frequency will prove efficient in discriminating between speakers.

In Figure 3.10, more evidence is found that supports the assumption that the sequence of F_0 estimates may not be an effective feature vector in speaker identification. The top plot shows the F_0 estimates for two training sentences from Speaker 1 and the bottom plot shows a corresponding analysis for two different speakers, but for the same sentence.

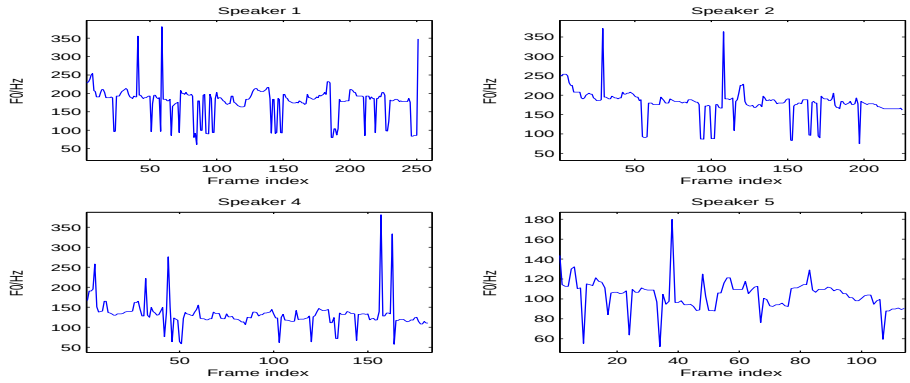


Figure 3.9: Fundamental frequency trajectories for different speakers

The number of frames for each sentence is as follows listed below:

- Speaker 1, sentence *a*: 169
- Speaker 1, sentence *b*: 251
- Speaker 2, sentence *a*: 145

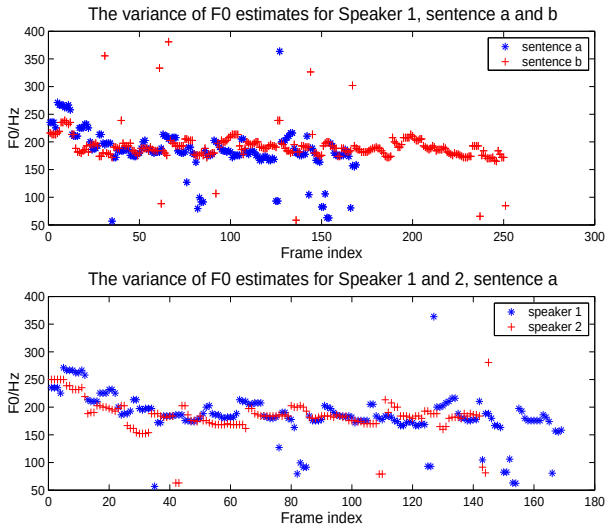


Figure 3.10: Pitch trajectory data, for different speakers and sentences

Although Figure 3.10 reveals slightly more overlap between the two sets of points in the top plot, the difference is not significant and it is difficult to see how a classifier would differentiate between the speakers if the pitch trajectories were used as features for SID. This feature will be tested, however, as there may be enough variance between some speakers to allow a degree of separation that is greater than seen here.

The feature sets that have been derived in Section 3.4 are representative of the source information in a speech signal and will be tested with different classifiers in Chapter 9. The next few sections are dedicated to describing the derivation of other feature sets.

3.5 Linear Prediction Coding

The focus is now shifted onto the system based features. One method of representing the system filter characteristics is Linear Prediction Coding (LPC). The results of the LPC analysis are converted to cepstral coefficients that are used as a feature set.

In Section 2.2, speech is modelled as the product of the models for the excitation source, the vocal tract and the lip radiation. To enable the implementation of LPC analysis, it is necessary to have an all-pole model for the filter characteristics of the speech model. An all-pole model is implemented because it enables a computationally simple way to derive the coefficients that define it, i.e. by solving a system of linear equations. Only the spectral magnitude is predicted in the all-pole model, while phase information is lost [1]. As the latter is known not to be necessary in order to be able to discriminate between speakers' voices, this is of limited importance. The loss of phase information corresponds to listening to someone talking while they are moving, and whereas the human ear can perceive the shift in phase, the information gathered from the speech about speaker identity is not dependent on this movement. An illustration of the all-pole model is shown in Figure 3.11, which has been adapted from [38].

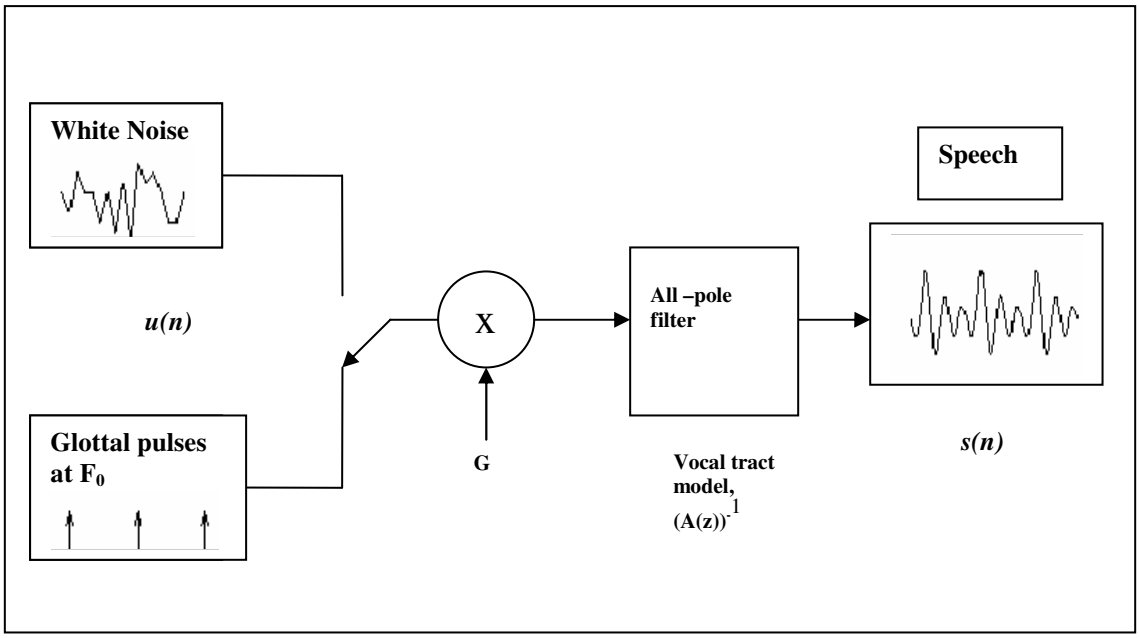


Figure 3.11: All-pole source-filter model of speech production

The all-pole model shows that the input from the excitation of the speech signal can either be glottal pulses or stochastic processes that can be modelled as white noise. This excitation is filtered by the all-pole filter that corresponds to the physical characteristics of the vocal tract and the radiation through the lips.

The lip radiation digital filter model $R(z)$ can be defined as:

$$R(z) = 1 - z^{-1} \tag{3.11}$$

The right-hand side of Eq.(3.11) contains a zero, which is assumed to be cancelled out

by one of the poles in the vocal tract filter [1], rendering it possible to define the vocal tract (*system*) model by means of the LPC coefficients a_i in the following digital filter model:

$$H(z) = \frac{S(z)}{U(z)} = \frac{G}{1 - \sum_{i=0}^{p} a_i z^{-i}} \tag{3.12}$$

where G is a gain factor, $U(z)$ is the excitation spectrum, $H(z)$ is the filter response of the vocal tract and $S(z)$ is the resultant speech spectrum. The value of p is the order of the LPC analysis that determines how well the LPC resultant all-pole spectrum models the short-term spectrum of the speech signal [20].

The LPC analysis is implemented as follows: in the discrete time domain, the LPC model of a signal is a linear combination of past values and a scaled value of the present input [9]:

$$s(n) = \sum_{i=1}^{p} a_i s(n-i) + G \cdot u(n) \tag{3.13}$$

This is an autoregressive process dependent on the values of a_i. For each fragment of speech, the prediction coefficients (a_i) are representative of the system characteristics of the vocal tract. $u(n)$ is the input excitation signal at time n and G is the gain factor. When the vocal tract resonance is sufficiently high, the first term is the dominant one as it depends on these system characteristics. The LPC analysis exclusively estimates these characteristics, so that

$$\hat{s}(n) = \sum_{i=1}^{p} a_i s(n-i) \tag{3.14}$$

is the LPC estimate that does not include a representation of the nonlinearities of the source signal. A prediction error, $e(n)$, is the difference between the actual speech signal and the LPC estimate.

$$e(n) = s(n) - \hat{s}(n) = s(n) - \sum_{i=1}^{p} a_i s(n-i) \tag{3.15}$$

LPC coefficients are obtained when the mean square of the prediction error is minimized. The mean squared error, E, is determined by the following equation:

$$E = \sum_{n} e^2(n) = \sum_{n} \Big[s(n) - \sum_{i=1}^{p} a_i s(n-i) \Big]^2 \tag{3.16}$$

In order to determine the a_i coefficients, Eq.(3.16) is differentiated with respect to a_j and set to zero:

$$\frac{\partial E}{\partial a_j} = 0 \tag{3.17}$$

This differentiation gives the following:

$$\sum_{n} s(n)s(n-j) = \sum_{n} \Big[\sum_{i=1}^{p} a_i s(n-i) \Big] \cdot s(n-j) \tag{3.18}$$

$$\tag{3.19}$$

The function $\phi_n(j,i)$ is defined as:

$$\phi_n(j,i) = \sum_n s(n-j)s(n-i) \tag{3.20}$$

So that Eq.(3.18) can be rewritten as:

$$\sum_{i=1}^{p} a_i \phi(j,i) = \phi_n(j,0) \tag{3.21}$$

as $\phi(j,i)$ is a symmetric function.

The mean square prediction error of Eq.(3.16) can also be rewritten as:

$$E = \sum_n \left[s(n) \cdot s(n) - \sum_{i=1}^{p} a_i \cdot s(n)s(n-i) + \left(\sum_i^p a_i \cdot s(n-i) \right)^2 \right] \tag{3.22}$$

$$E = \phi_n(0,0) - \sum_{i=1}^{p} a_i \phi_n(0,i) \tag{3.23}$$

as the first two terms of Eq.(3.22) are assumed to be dominant. The solution to Eq.(3.21), known as the Yule-Walker equation [9], yields the values of the linear prediction coefficients.

In order to solve Eq.(3.21), a constraint is placed on the interval used for the evaluation, so the upper bound is set to $N-1$. We begin by defining the autocorrelation function for the speech segments $s(n)$:

$$R(i) = \frac{1}{N} \sum_{n=0}^{N+i-1} s(n)s(n+i) \quad i = 0, \ldots, p \tag{3.24}$$

where the limit is set to $N+i-1$ as it is assumed that values outside the interval $N-1$ are zero. By defining an auxiliary variable $q = N-1-(j-i)$, Eq.(3.20) becomes:

$$\phi_n(j,i) = \sum_{n=0}^{q} s(n-j)s(n+j-i) \tag{3.25}$$

From Eq.(3.24), it can be seen that this corresponds to the autocorrelation function for $s(n)$, so that:

$$\phi(j,i) = R_n(j-i) \tag{3.26}$$

The linear prediction Yule-Walker equation shown in Eq.(3.21) can thus be expressed as follows:

$$\sum_{i=1}^{p} a_i R_n(|j-i|) = R_n(j) \tag{3.27}$$

This method of solving for the linear prediction coefficients is therefore referred to as the autocorrelation method [24].

The system of equations that are derived using Eq.(3.27) can be written in matrix form as:

$$\vec{r} = \mathbf{R}\vec{a} \tag{3.28}$$

where the elements of $\mathbf{R}$ and $\vec{r}$ are the autocorrelation values and the elements of $\vec{a}$ are the desired LPC coefficients, see Eq.(3.29).

$$\begin{bmatrix} R(1) \\ R(2) \\ R(3) \\ \vdots \\ R(p) \end{bmatrix} = \begin{bmatrix} R(0) & R(1) & R(2) & \dots & R(p-1) \\ R(1) & R(0) & R(1) & \dots & R(p-2) \\ R(2) & R(1) & R(0) & \dots & R(p-3) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ R(p-1) & R(p-2) & R(p-3) & \dots & R(0) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_p \end{bmatrix} \tag{3.29}$$

As $\mathbf{R}$ is a symmetric Toeplitz matrix, the recursive Durbin algorithm can be used to calculate the LPC parameters [9]. The Durbin algorithm is derived in [9] and [24] and is shown below

The Durbin algorithm
$\mathtt{E}^{0} = \mathtt{R}(0)$
$\mathtt{k}_{\mathtt{i}} = [\mathtt{R}(\mathtt{i}) - \sum_{\mathtt{j}=1}^{\mathtt{i}-1} \mathtt{a}_{\mathtt{j}}^{\mathtt{i}-1}\mathtt{R}(\mathtt{i}-\mathtt{j})]/\mathtt{E}^{\mathtt{i}-1}$
$\mathtt{a}_{\mathtt{i}}^{(\mathtt{i})} = \mathtt{k}_{\mathtt{i}}$
$\mathtt{a}_{\mathtt{j}}^{(\mathtt{i})} = \mathtt{a}_{\mathtt{j}}^{(\mathtt{i}-1)} + \mathtt{k}_{\mathtt{i}}\mathtt{a}_{\mathtt{i}-\mathtt{j}}^{\mathtt{i}-1}$
$\mathtt{E}^{(\mathtt{i})} = (1 - \mathtt{k}_{\mathtt{i}}^{2})\mathtt{E}^{\mathtt{i}-1}$

From Eq.(3.12), the LPC transfer function can be written as:

$$H(z) = \frac{S(z)}{U(z)} = \frac{G}{1 - \sum_{i=1}^{p} a_i z^{-i}} = \frac{G}{A(z)} \tag{3.30}$$

where $A(z)$ is an inverse filter of the all-pole model [9] that represents the vocal tract resonances and is defined by the LPC coefficients a_i.

These coefficients are called the LPC autoregressive (AR) coefficients and are used in the computation of the Linear Prediction cepstral coefficients (LPCC). A high order LPC analysis allows an extended search range for formant frequencies, which are what the analysis models. If the order is too high, F_0 may be found instead, while formant peaks may be missed altogether if the analysis is of too low an order. Determining an optimal value for p must be done empirically, though some values are known to work better than others. The initial LPC order in this thesis is chosen on the basis of common use in speaker identification applications.

3.5.1 Linear Prediction Cepstral Coefficients

The cepstral coefficients, c_m, are calculated using the AR coefficients from the LPC analysis [5]. Calculating the cepstral coefficients implements further smoothing to the speech spectrum that has already been smoothed in the sense that the excitation signal has been

removed in the LPC analysis. Results obtained in [40] and some other works show that the use of cepstral coefficients can lead to better speech classification than results based on the LPC a_i coefficients can. The p LPC AR coefficients are converted to M cepstral coefficients by the following recursion formulae:

$$c_0 = ln(G) \tag{3.31}$$

$$c_m = \frac{-ma_m + \sum_{k=1}^{m-1} a_k c_{m-k}(m-k)}{m}, \quad 1 \leq m \leq p \tag{3.32}$$

$$c_m = \frac{\sum_{k=1}^{m-1} a_k c_{m-k}(m-k)}{m}, \quad p \leq m \leq M \tag{3.33}$$

Each cepstral coefficient contains different information about the speech signal, and speaker identification success can vary depending on the order of the LPCC analysis. The higher cepstrals contain information about the finer detail of the vocal tract characteristics.

3.5.2 The LPC Residual

We recall from Section 3.5 that the LPC coefficients only represent the system characteristics of the speech system as a linear model. It can thus not describe the nonlinearities that might be present in the speech signal, i.e. the source signal. The LPC *residual* may therefore be assumed to contain some additional, complementary information that can be of use to identify a speaker. The residual error of the linear prediction analysis is obtained by subtracting the LPC estimate from the original speech signal, see Eq.(3.15). The residual error is thus equal to the input signal $G \cdot u(n)$ from Eq.(3.13). For convenience, these equations are rewritten here and the equality made explicit:

$$s(n) = -\sum_{i=1}^{p} a_i s(n-i) + G \cdot u(n) \tag{3.34}$$

$$\hat{s}(n) = -\sum_{i=1}^{p} a_i s(n-i) \tag{3.35}$$

$$e(n) = s(n) - \hat{s}(n) = s(n) - \sum_{i=1}^{p} a_i s(n-i) \tag{3.36}$$

$$e(n) = G \cdot u(n) \tag{3.37}$$

Some source information in the input signal frame $u(n)$ is thus present in the LP residual, and so the residual energy is selected as a possible feature. In Figure 3.12, the speech waveform of a female speaker for training sentence c is shown with the corresponding LPC residual energy, the 2nd LPC coefficient and the 2nd LPC cepstral coefficient. The LPC analysis implemented here is of order $p = 12$.

3.6 Warped LPCC

An alternative way of performing feature extraction is to attempt an approximation to the frequency analysis that is executed within the human ear. This approximation places weight on the perceptually significant parts of speech, and this may have a positive influence on the SID task. The Bark scale [47] is one that models the bank of filters that is

used as an approximation to the frequency analysis processes that take place in the inner ear. A detailed description of the Bark scale and this filter bank is provided in Appendix A. The Bark scale has a linear relation to the frequencies of incoming sounds up to 500Hz; after this, the relation is logarithmic, as can be seen in Figure 3.13, where the Bark values are shown as a function of the logarithmic frequency scale.

The linear prediction analysis models the vocal tract that influences the formation of a

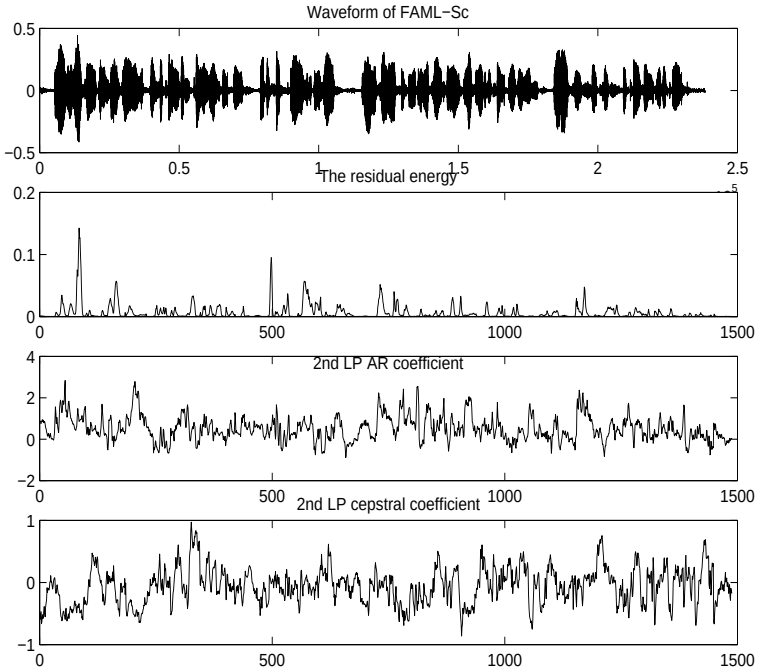


Figure 3.12: The waveform, LPC residual, LPC 2nd coefficient and 2nd LPC cepstral coefficient for FAML_Sc

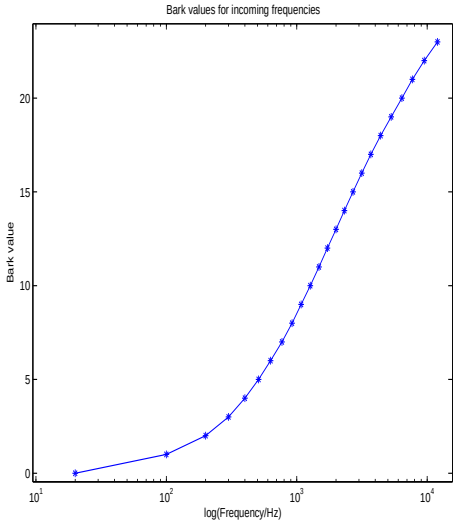


Figure 3.13: The Bark values for the logarithm of incoming frequencies

speech sound. In order to approximate this sound as it is heard in a human ear, the LPC coefficients, a_i, are warped to the Bark scale. The warping is implemented by shifting the poles of the all-pole filter by a variable λ that is determined by the Bark mapping of the sampling frequency, F_s. This mapping is given in [49] as:

$$\lambda_{F_s} \equiv 1.0674 \left(\frac{2}{\pi} \arctan \left(\frac{0.06583 F_s}{1000} \right) \right)^{1/2} - 0.1916 \tag{3.38}$$

The poles are shifted by using a first-order all-pass filter that depends on this variable. The transfer function, $G(z)$, of the all-pass filter is given by Eq.(3.39).

$$G(z) = \frac{z^{-1} - \lambda}{1 - \lambda z^{-1}} \tag{3.39}$$

In Eq.(3.30), the inverse filter polynomial $A(z)$ is given as

$$A(z) = 1 - \sum_{i=1}^{p} a_i z^{-i} \tag{3.40}$$

The process of warping the linear prediction coefficients a_i is achieved by replacing the delay term, z^{-i}, with the all-pass filter defined by the transfer function $G(z)$:

$$\hat{A}(z) = 1 - \sum_{i=1}^{p} a_i G(z)^i \tag{3.41}$$

By applying the warping transformation on the LPC coefficients, the resonant frequencies estimated by the LPC model are modified to approximate the frequency analysis in the human auditory system. Once the warping process is completed, the cepstral recursive formulae of Eq.(3.31)-Eq.(3.33) are used in order to obtain the warped LPCC feature set.

3.7 Perceptual Linear Prediction

An additional feature set that implements an approximation to the human auditory system is the Perceptual Linear Prediction (PLP) analysis [62]. As the name implies, Perceptual Linear Prediction is a combination between spectral analysis and linear prediction analysis. The PLP analysis gives rise to modified autocorrelation coefficients, $\tilde{a}_i$, that correspond to the LPC analysis a_i coefficients, and once again the cepstral recursion formulae (Eq.(3.31)-Eq.(3.33)) are used in order to calculate the PLPCC coefficients. The difference from the warped LPCC feature lies in that the PLP analysis consists of a preprocessing that not only warps the speech segments power spectrum to the Bark scale, but also applies other auditory approximations to obtain a more precise modelling of the processes in the ear. In addition, the warping implemented here is done prior to the derivation of the AR coefficients and thus the input to the linear prediction analysis is speech that is already modified so that it contains perceptually significant information.

The speech data that is to be analyzed using PLP is divided into framed blocks as described in Section 3.3. The power spectrum of each segment is then calculated using the discrete Fourier Transform. This spectrum is then warped to the Bark scale. The warped spectrum is convolved with the critical band masking curve, see Appendix A. This corresponds to multiplying the spectrum with the critical band transfer functions:

$$\tilde{S}(b) = \sum_{n=0}^{N-1} |H_b(n)|^2 |X(n)|^2 \tag{3.42}$$

where $X(n)$ is the frequency representation of the signal segment and $H_b(n)$ is the transfer function of the critical band filter b that is uniformly spaced on the Bark scale.

The resultant warped power spectrum is then pre-emphasized with the equal-loudness curve, which is given in Eq.(3.43).

$$E(b) = \frac{(b^2 + 56.8 \cdot 10^6) b^4}{(b^2 + 6.3 \cdot 10^6)^2 (b^2 + 0.38 \cdot 10^9)} \tag{3.43}$$

where b denotes the frequencies warped to the Bark scale and $E(b)$ is the transfer function that represents the human ear's sensitivity to different frequencies at roughly 40dB.

The equal-loudness pre-emphasis is used in order to approximate the sensitivity of the human ear to certain frequencies. It has been established that some frequencies are emphasized more than others. Not surprisingly, the frequencies in the area of human speech are among those that the ear shows heightened sensitivity to.

By multiplying the equal-loudness transfer function with the warped power spectrum, the perceptual intensity, $I(b)$, of the sound in each speech segment is obtained, see Eq.(3.44). Further modification to approximate human auditory perception is implemented by determining the *perceived loudness*, $Y(b)$, of this intensity. The perceived loudness of a tone is approximately proportional to the cubic root of the tone's intensity, thus reducing the amplitude of the spectrum. This auditory compression that simulates the relationship between intensity and perceived loudness is defined in Eq.(3.45).

$$I(b) = E(b) \cdot \tilde{S}(b) \tag{3.44}$$

$$Y(b) \approx \sqrt[3]{I(b)} \tag{3.45}$$

Loudness is measured in the unit Son, and as can be derived from Eq.(3.45), a doubling of loudness requires approximately a 10dB increase in intensity.

The next step in the process of extracting PLP coefficients is executed by taking the inverse Fourier transform of the spectrum of Eq.(3.45) and the perceptual autocorrelation, $\tilde{R}(m)$ is obtained so that:

$$\tilde{R}(m) = \sum_{i=1}^{p} \tilde{a}_i \tilde{R}(|m-k|) \tag{3.46}$$

The perceptual autocorrelation is analogous to the function of Eq.(3.28) and can thus be written in matrix form:

$$\tilde{\vec{r}} = \tilde{\mathbf{R}} \tilde{\vec{a}} \tag{3.47}$$

The Durbin algorithm from Section 3.5 is used to determine the PLP coefficients, $\tilde{a}_i$, which are then transformed into cepstral coefficients by applying the cepstral recursion formulae of Eq.(3.31)-Eq.(3.33). The results are the perceptual linear prediction cepstral coefficients, the PLPCC, with coefficients $\tilde{c}(m)$.

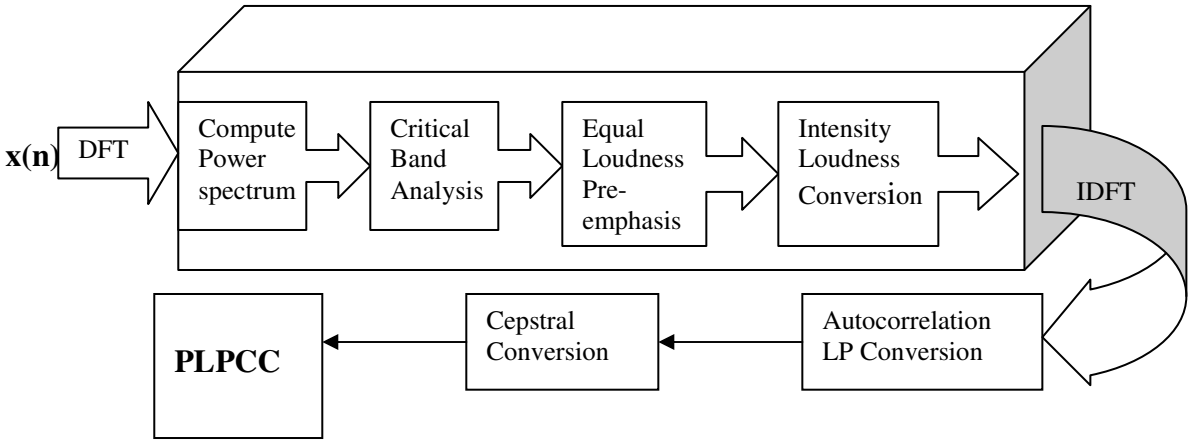


Figure 3.14: The derivation of the PLPCC feature set

The PLP analysis is illustrated by a block diagram in Figure 3.14.

As described above, the derivation of PLPCC coefficients requires a preprocessing that is performed in the frequency domain prior to the LPC analysis for the purpose of adding weight to the perceptually significant portions of the speech signal's spectrum. It is hoped that this approximation to the biological processes that are executed in the human ear and the consequent smoothing of the spectrum will prove helpful in the effective discernment between different speakers by the SID system.

3.8 Mel Frequency Cepstral Coefficients

Another feature set that represents the filter characteristics of the source-filter model is the mel-frequency Cepstral Coefficient feature set. Here, the mel frequency scale is used in order to mimic the cochlear filtering processes in the ear which places more emphasis on certain frequencies [2]. The reference point of the mel scale is at a tone of 1000Hz which is set equal to a pitch of 1000mels. Hereafter the mel intervals become logarithmically distributed. The mel scale was experimentally derived by measuring the difference between a linear frequency scale and the perceived pitch that human listeners registered during a series of tests [1].

In order to warp the frequency spectrum to the mel-frequency spectrum, the following calculation is required [21]:

$$f_{mel}(f) = 2595 \cdot log\left(1 + \frac{f}{700\text{Hz}}\right) \tag{3.48}$$

The mel scale defines a mel filter bank. Each filter's center frequency follows the mel scale in such a way as to imitate the audiological critical band, see Appendix A. The mel filters are triangular and spaced about 150mels apart, each triangle being 300mels wide. The filter bank consists of 20 filters in total.

The Mel-Frequency cepstral coefficients are derived by the following procedure:

1. The signal is frame-blocked and windowed, as described in Section 3.3
   $s(n) = u(n) * h(n)$
2. The FFT of the signal is taken
   $S(z) = U(z) \cdot H(z)$
3. The magnitude is taken, thus discarding the phase information
   $|S(z)| = |U(z) \cdot H(z)|$
4. The spectrum is warped using a mel-filterbank
   $\tilde{S}(k) = \sum_{z=0}^{N/2} S(z) \cdot M_k(z)$ where M_k is the k^{th} filter from the filter bank.
5. The logarithm is taken
   $\log(\tilde{S}(k)) = \log(\tilde{U}(k)) + \log(\tilde{H}(k))$
6. A Discrete Cosine Transform (DCT) is used to derive the MFCC's
   $c_u(n) + c_h(n)$

This process is shown in Figure 3.15

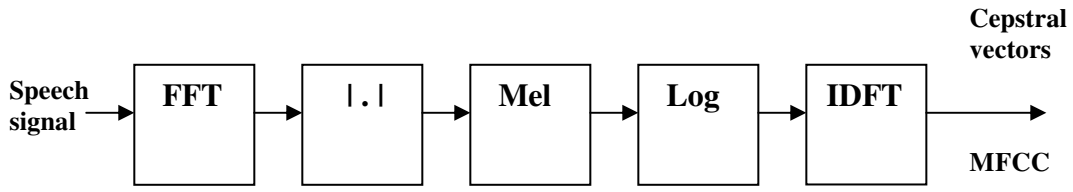


Figure 3.15: Derivation of MFCC

The logarithm of step 5 performs a deconvolution of the source and system features. In step 6, let S_k define the logarithm of the k^{th} filter of the signal. In all there are K log-spectral coefficients. The Q cepstral coefficients are then derived according to the following DCT transform [5]:

$$c_n = \sum_{k=1}^{K} S_k \cdot cos\Big[n(k - \frac{1}{2})\frac{\pi}{K}\Big], \quad n = 1, \ldots, Q \tag{3.49}$$

The DCT has the advantage of being able to decorrelate the statistically dependent spectral coefficients into independent cepstral coefficients. The zeroth cepstral coefficient, c_0, describes the overall energy in the spectrum, while c_1 measures how the energy is distributed between the high and low frequencies [23]. The remaining coefficients show the finer detail of the spectrum, and are thus used as features.

3.9 The Temporal Derivatives of Cepstral Coefficients

The temporal derivatives (denoted Δ) of all the cepstral coefficient feature sets (LPCC, warped LPCC, PLPCC and MFCC) can be determined using the function shown in

Eq.(3.50). To include dynamic information about how the speech signal and thus the cepstral vectors vary with time, the temporal derivatives of these vectors are calculated. The performance of a speaker identification system might be improved when the first and second order cepstral derivatives are included so that the final feature set is $3M$ long [5], where M is the length of the original feature set, i.e. the order of the LPCC or PLPCC or MFCC extraction. Although these derivatives can be obtained from a computation over only two frames, subtracting the previous cepstral vector from the present one, this method is not very representative and thus the computation is based on a window of length θ cepstral vectors (one vector per frame), where $\theta > 2$ [23]. The computation of the first order derivatives is thus:

$$\Delta c_m(n) = \frac{1}{\theta}\Big(c(n+\theta) - c(n-\theta)\Big) \tag{3.50}$$

for the m^{th} frame.
The second-order derivatives ($\Delta\Delta$) are computed in the same way from the first order derivatives. The value of θ is set to 16 for the feature extraction methods implemented for the SID task of this thesis.

To observe the difference between a cepstral coefficient and its temporal derivatives, the waveform of the speech signal of training sentence c from Speaker 1 is once again shown in Figure 3.16 and below it, the 2nd LPCC coefficient is depicted, followed by the first and second time derivatives of this coefficient.

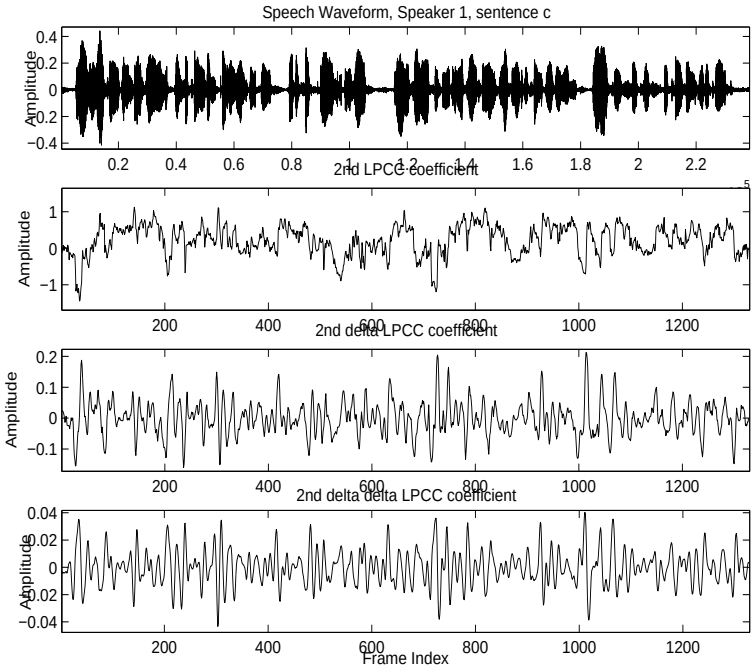


Figure 3.16: The waveform, LPCC 2nd coefficient, 2nd ΔLPCC, 2nd $\Delta\Delta$LPCC for FAML_Sc

By observing the plots of Figure 3.16, it is possible to see that the temporal derivatives of the LPC cepstral coefficients vary in time in closer accordance with the original waveform than the LPCC coefficient does. The first and second temporal derivatives of the

cepstral coefficient feature sets are implemented and tested in Chapter 9 for the different classifiers in order to establish whether these aid the speaker recognition process.

3.10 Principal Component Analysis of Cepstral Coefficients

As different sets of cepstral coefficients have been derived, it is of interest to conduct a preliminary test on each feature set's ability to separate speakers. This is done by implementing a Principal Component Analysis (PCA) [15] for the 12MFCC+12ΔMFCC, 12LPCC+12ΔLPCC, 12 warped LPCC+12Δ warped LPCC and 13PLPCC+13ΔPLPCC feature sets. The orders of these feature extraction methods are chosen because they are commonly used for speech processing applications [1]. The feature sets are extracted from training sentence *a* for Speaker 1 and Speaker 2, both women. The choice of the same training sentence ensures that the speech uttered is identical for both speakers and so it is each speaker's physiological characteristics that are modelled by the system based feature sets that are the only source of difference between the sentences. This allows an analysis that can highlight which feature sets separate speakers effectively.

The PCA results in the projection of the data in the feature matrices in the directions that provide most variance. The projection of data in the direction of the first two principal components is shown in Figure 3.17. It is desirable that the variance between two speakers ensures a good separation of the two speech signals by being larger than the variance within a speaker's feature set.

The MFCC feature set seems to have a lot of overlap between the two speakers, while the PCA on the LPCC yields a cluster of overlapping data points, but also two groups of data that exclusively belong to one or the other speaker. The warped LPCC and PLP coefficients result in data points that are grouped in less dense clusters than those for MFCC and are thus subject to a lesser of overlap between different speaker data.

To establish whether the fact that a frame of speech is voiced or unvoiced has an effect on the separation of different speakers for feature sets, another PCA analysis is implemented. This is of interest as the possible reduction of a feature set while preserving the majority of speaker-dependent information would significantly improve the speed and performance of a SID system. The first and simplest step in this direction is to use the results gained from the autocorrelation with center clipping algorithm to divide each feature set into a voiced feature set and an unvoiced one. The voiced/unvoiced decisions are made for frames that are 30ms in length, with an overlap of 10ms. The number of voiced frames in each case is roughly 4 times that of unvoiced frames, hence the difference in the number of frames used in each analysis. The number of frames is divided as shown in Table 3.3. All numbers are for frames of 10ms.

Speech signal	Total	Voiced	Unvoiced
Speaker 1,sentence *a*	1328	1076	252
Speaker 2, sentence *a*	1118	864	254

Table 3.3: Number of voiced and unvoiced frames in training sentence *a*

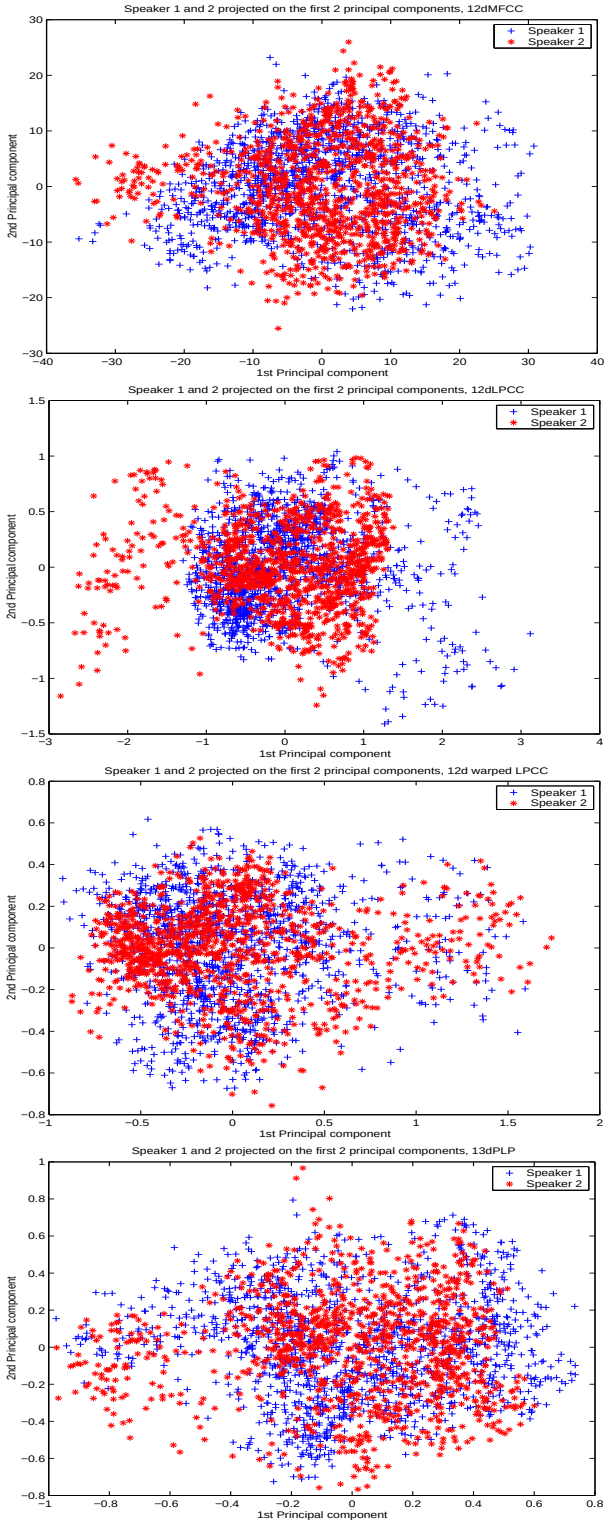


Figure 3.17: PCA on 12ΔMFCC, 12ΔLPCC, 12Δ warped LPCC and 13 ΔPLPCC

Figure 3.18 shows the results of PCA on the voiced frames of sentence *a* from Speakers 1 and 2, while the corresponding results for the unvoiced frames are presented in Figure 3.19.

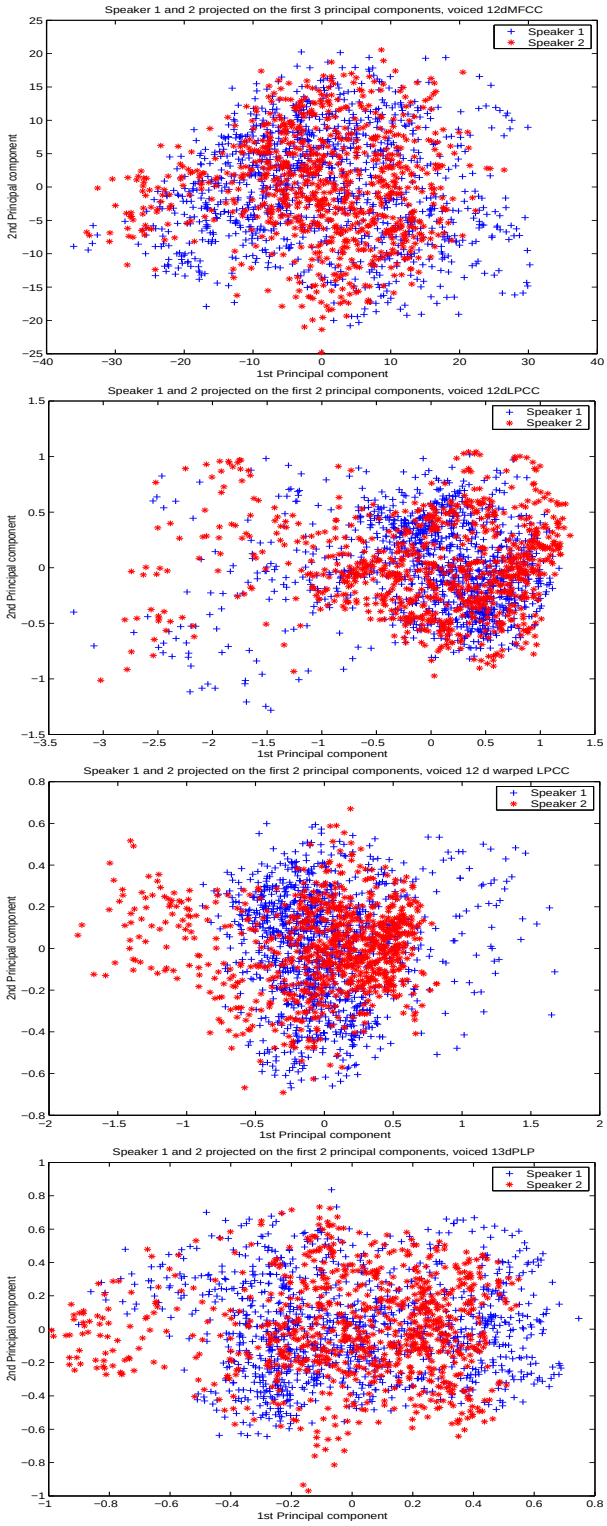


Figure 3.18: PCA on Voiced frames of 12ΔMFCC, 12ΔLPCC, 12Δ warped LPCC and 13Δ PLPCC

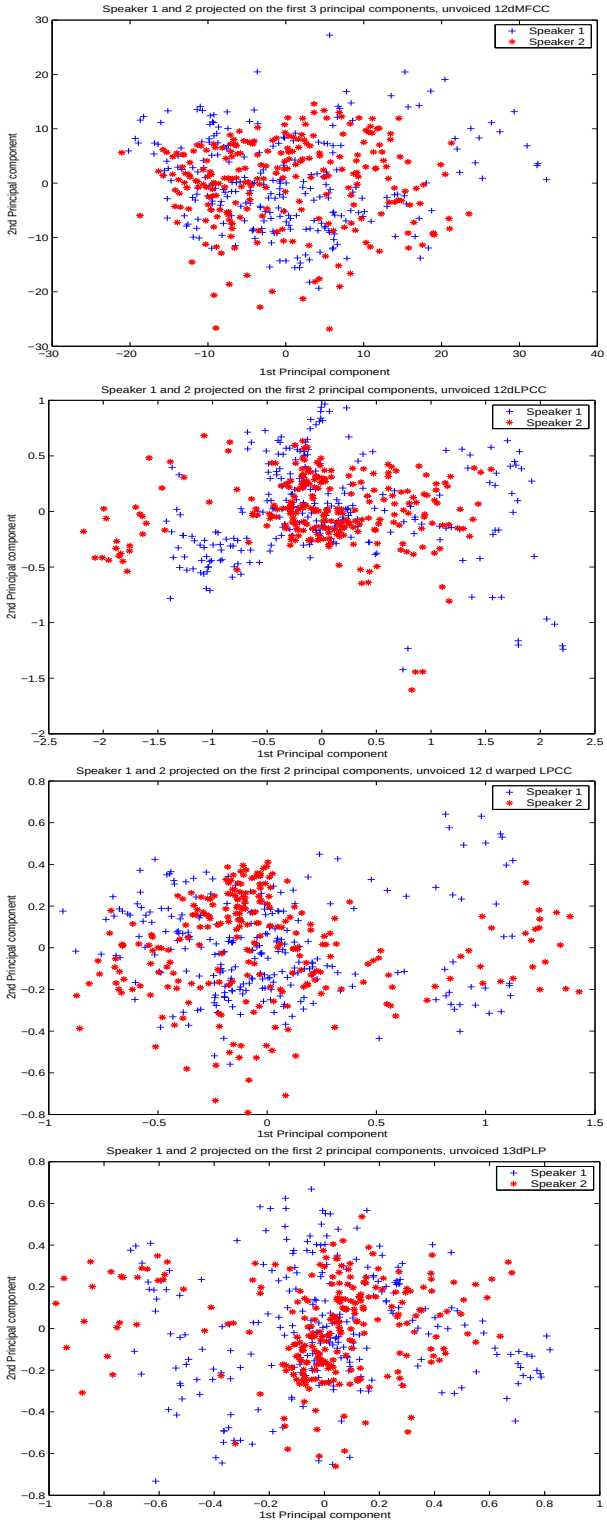


Figure 3.19: PCA on Unvoiced frames 12ΔMFCC, Δ12LPCC, 12Δ warped LPCC and 13 ΔPLPCC

In Figure 3.18 the separation of features in the directions of most variance for voiced frames do not show any general improvement compared to the results for all frames shown in Figure 3.17, though a few changes are visible. There is more overlap between data points using the LPCC feature extraction method and less overlap for the warped LPCC features. The plots for the MFCC features and PLPCC features remain almost unchanged. The corresponding analysis for the unvoiced frames that is shown in Figure 3.19 does not lead to good separation of the data from Speakers 1 and 2. Although the reduced amount of data in these sets make it superficially seem like there is less overlap, there is no clear division of the points into two groups for any of the feature sets and so there is a high degree of overlap here.

Regardless of these early observations, all feature sets will be used in the trials that are executed in Chapter 9, as by using different classifiers the distribution of data in feature space may prove suitable for speaker identification depending on the classification method applied. This preliminary analysis may, however, prove useful in understanding some of the results that will be recorded at a later stage.

3.11 Discussion of Feature Sets

The derivation of the cepstral coefficients is computationally effective and reliable, and they generally perform well for speaker recognition tasks. For these reasons it is common practice to implement speaker recognition systems using cepstral coefficients as the selected speaker-dependent features. There are, however, a couple of problems associated with this solution. One of these is that cepstral coefficients are not robust against channel distortion and background noise, though this has limited significance for this project as there is no mismatch between the training and the test set data in the ELSDSR database. Additionally, some speaker-dependent information is invariably lost when only the power spectral envelope of the system characteristics is used [10]. There have been numerous attempts at exploiting source and prosodic information[1] for speaker recognition. Most experiments show that these feature sets used in isolation perform poorly in comparison with the cepstral features. They do, however, provide complementary information about the speech signal, which means that combining the two types of features can lead to increased performance when compared to performance based exclusively on the use of cepstral coefficients. These conclusions are drawn on the basis of the work presented in [22], [19], [6], [26], [17] and [18]. In [29], [30] and [35], feature selection procedures are implemented and it is shown that a combination of system-based features can also lead to a reduction in error. In this thesis the feature sets will be implemented individually and the performance of the SID system for each classifier noted. In this way it will be possible to assess which feature set is optimal for the speaker identification task involving a small group of speakers that provide a limited amount of data that is free of noise and mismatch. These results are presented in Chapter 9.

[1] Prosody is the pattern of stress and intonation in a language and features that can be extracted that contain this information are discussed in [17]

Chapter 4

Fundamentals of Classification

Once the feature sets of speaker data have been extracted, a classifier must be implemented. The classifier uses the training and test data sets as input data sets and it produces an output of classification labels for each test data set, identifying the speaker who uttered the speech contained within the set. This corresponds to the "Pattern Matching" and "Decision Logic" steps of Figure 1.2. The structure of a speaker identification system classifier can vary, as can the decision rule that is implemented to make the final identification.

In Chapter 3 different feature sets are discussed as an optimal feature extraction method for speaker identification does not exist. A similar situation affects the choice of classifiers for SID, as each classifier has its share of trade-offs. The performance of the entire SID system is heavily dependent on the type of features that are extracted, but it is also significantly affected by the type of classifier that is implemented. There is no absolute answer as to which classifier is most suited for the speaker identification task. Three different types of classifiers are therefore implemented in order to establish which one is optimal for the SID task of this thesis. The implementation of different classifiers also enables a more thorough analysis of the suitability of the different feature sets for speaker identification.

The three classifiers that are implemented are:

- Mixture of Gaussians Models (MoG)
- k-Nearest Neighbour (k-NN)
- nonlinear Neural Network (NN)

The specific details concerning each of the three classifiers are presented in Chapters 5, 6 and 7. Despite the various different ways that classifiers are structured, a number of concepts are relevant for all of them and will therefore be described here.

4.1 The Decision Rule

The decision rule is vital in the classification process, as it effectively decides which class a test data sample for the n^{th} frame of feature data, $\mathbf{x}^n$, belongs to after matching it

to the training data or parameters adjusted by the training data during the enrollment stage. The n^{th} data sample contains a feature vector of dimension d that depends on which feature set is used. An optimal decision rule minimizes the risk of an incorrect classification. Although each classifier has a unique structure to process data, the decision rule for all three classifiers that are implemented can be described using a probabilistic interpretation. In order to obtain a decision rule from probability distributions, *Bayes' Theorem* [15] is used. Bayes' theorem determines the *posterior probability* $P(C_i|\mathbf{x}^n)$ for a speaker represented by the class C_i, $i = 1, \ldots, S$, where S is the number of speakers, given that the test frame $\mathbf{x}^n$ is observed, and is derived as

$$P(C_i|\mathbf{x}^n) = \frac{p(\mathbf{x}^n|C_i)P(C_i)}{p(\mathbf{x}^n)} \tag{4.1}$$

where $p(\mathbf{x}^n|C_i)$ is the *class-conditional* probability density function that evaluates the probability of $\mathbf{x}^n$ having been generated for the given class C_i. Details of the estimation of the class-conditional density function are discussed in Chapter 5. $P(C_i)$ is the prior probability for the speaker class i, and $p(\mathbf{x}^n)$ is the unconditional density function for $\mathbf{x}^n$. The purpose of having $p(\mathbf{x}^n)$ as the denominator is to provide a scaling factor that ensures that the posterior probabilities sum to unity, i.e. $\sum_{i=1}^{S} P(C_i|\mathbf{x}^n) = 1$. The unconditional density is computed in Eq.(4.2).

$$p(\mathbf{x}^n) = \sum_{i=1}^{S} p(\mathbf{x}^n|C_i)P(C_i) \tag{4.2}$$

The unconditional probability of $\mathbf{x}^n$ is thus not dependent on the different classes as it simply defines the probability density function of the test feature set for frame n. From Eq.(4.1), it can be deduced that the posterior probability derived from Baye's theorem is proportional to the class-conditional density function and the prior probabilities, as seen in Eq.(4.3).

$$P(C_i|\mathbf{x}^n) \propto p(\mathbf{x}^n|C_i)P(C_i) \tag{4.3}$$

For this speaker identification task, the prior probability for the different reference speakers is not known. The prior probability $P(C_i)$ is therefore set to being equal for all speakers. For S speakers in total, each speaker's prior probability is thus assumed to be $P(C_i) = \frac{1}{S}$. From Eq.(4.3) the proportionality factor leads to the conclusion that the functions that ultimately discriminate between speakers are the class-conditional probability density functions, $p(\mathbf{x}^n|C_i)$.

In Chapter 5, a method that estimates the class-conditional probability density functions and then applies them to Bayes' Theorem is described. Density estimation with the k-nearest neighbour classifier is briefly discussed in Chapter 6, while in Chapter 7 it is shown that the neural network yields results in the form of posterior probabilities.

Common for all these methods of classification is that the decision of which speaker a test frame is assigned to corresponds to maximizing the posterior probability for that speaker. The advantage of applying Bayes' Theorem in many cases is that while the posterior probability in itself may be difficult to calculate, the probability functions that

it depends on can be estimated and then used to derive the posterior probability as seen in Eq.(4.1).

4.2 The Curse of Dimensionality

The curse of dimensionality plays a central role in affecting the performance of different classifiers. It is closely connected to the probability density functions discussed in Section 4.1. A probability density function estimates the distribution of data points in feature space by mapping this distribution with a number of parameters. If P is the number of parameters needed to estimate a distribution for the 1-dimensional point x^n, then P^d parameter values must be determined for the d-dimensional vector $\mathbf{x}^n$, where $\mathbf{x}^n = \mathbf{x}_1^n, \mathbf{x}_2^n, \ldots, \mathbf{x}_d^n$. As the number of parameters to be estimated increases exponentially, so should the number of frames used to estimate the probability density function. For a large number of dimensions, this means that the required data set becomes exponentially large, but as a limited amount of data is available for the speakers in the ELSDSR database, this increase cannot be provided. The data sets used to estimate distributions of high dimensionality are thus sparse and the resulting probability density estimation becomes a poor representative of the underlying distribution of input data. This provides motivation to seek a way in which to limit the dimensionality of the input data set without decreasing the performance of classifiers. As will be discussed in Chapters 5, 6 and 7, the curse of dimensionality affects some classifier types worse than others.

4.3 Impostor detection

The reason that imposter detection must be implemented is that the speaker identification task is open-set. The implementation of impostor detection can also be described using a probabilistic approach. The class-conditional density, $p(\mathbf{x}^n|C_i)$, if estimated reliably, will yield a far higher density value for class i, if speaker i uttered the speech segment in $\mathbf{x}^n$, than for any other class. It can therefore be assumed that

$$p(\mathbf{x}_i^n|C_i) \gg p(\mathbf{x}_j^n|C_i),\ j \neq i$$

As the class C_i can only be one of the 6 reference speaker classes that are used to provide training data for each classifier, the impostor test frame $\mathbf{x}_{Imp}^n$ should yield a low class conditional probability density for all S reference classes. Whether this always holds true depends on the accuracy of the probability estimation as well as on the eventual overlap of data points in the feature sets of different speakers. The process of detecting an impostor is more reminiscent of speaker verification than speaker identification, as instead of selecting the speaker class that yields the maximum posterior probability for a given test frame, the criteria for detecting an impostor is that the test frame is rejected as being one of the reference speakers for all speakers in the reference set. This requires the determination of speaker specific thresholds that correspond to Θ. Each threshold must be high enough to prevent impostors from being accepted and low enough for test frames from the correct speaker to be accepted. As density estimates are not always reliable and because it is not certain that the test frame contains a high level of speaker-specific information, it is not possible to determine Θ so that errors never occur. A balance must be struck

between the amount of false rejections and false acceptances that are desirable and Θ set accordingly. A detailed description of the implementation of an impostor detection method is provided in Section 5.6. The structures of all three classifiers are described for use in a closed-set speaker identification task, as the impostor detector is implemented prior to the commencement of the SID systems classification stage, as seen in Figure 1.3.

4.4 Consensus

As the principle of classification by consensus is used repetitively throughout the next few chapters, it will be described free of any case-specific references here. Consensus in itself means the reaching of an agreement by a group as a whole, and is therefore commonly also referred to as majority voting. For the classifiers that will be presented in Chapters 5, 6 and 7, each test data frame $\mathbf{x}^n$ is classified as belonging to a particular class. In our case, these classes can be Sp1, Sp2 ... Sp6 for the 6 reference speakers, or an impostor class.

Let us assume that a test speech sequence consists of a sentence that is divided into N frames. The feature vectors extracted for each frame are used as input to a classifier, one at a time, so that the classification is executed N times. A very simplified representation of the classification of one frame is shown in Figure 4.1, where the classifier is unspecified and therefore represented by a "black" box.

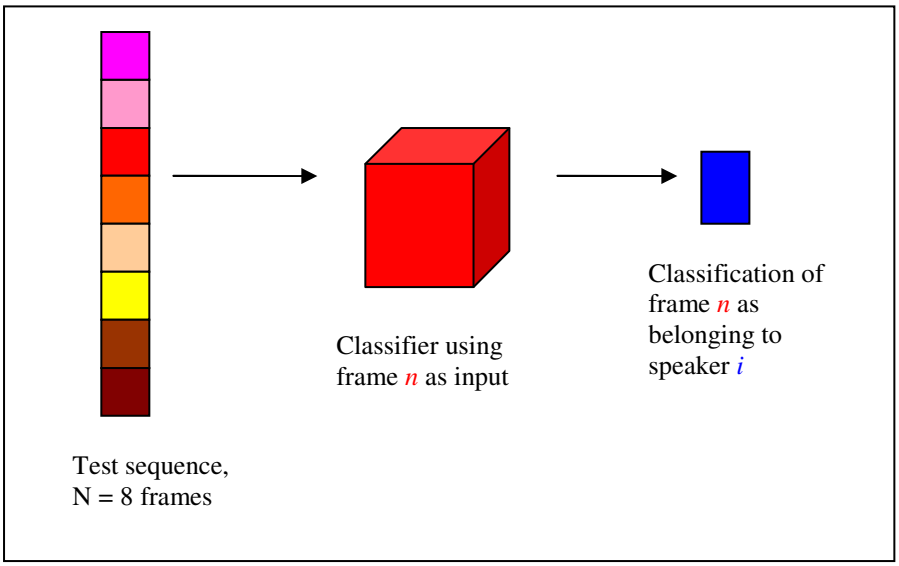


Figure 4.1: Classification of one frame of a test sequence

The sequence of frames is thus transformed into a sequence of N labels, each indicating class membership. The correct class for the entire test sentence $\mathbf{x} = \mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^N$ is then chosen as the one that is present in the relative majority of these classified frames, when all class scores are compared. Classification of a sequence of frames into different classes is shown in Figure 4.2.

In Figure 4.2, it is assumed that an impostor can be classified as an additional class,

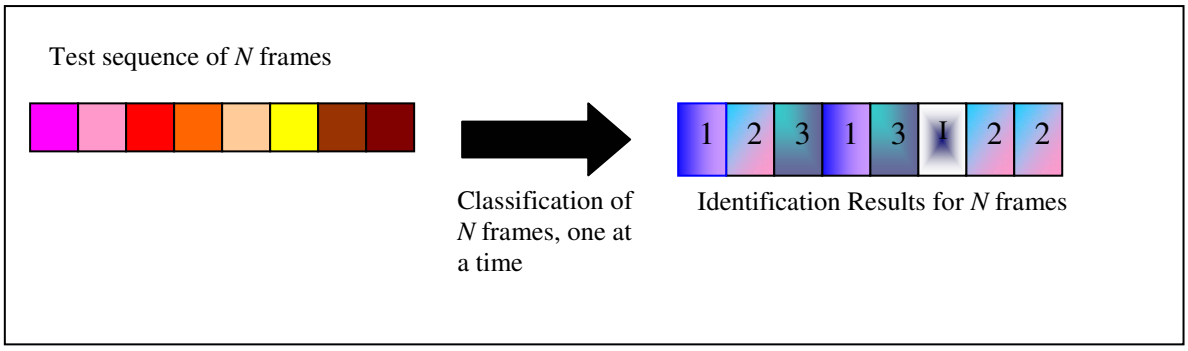


Figure 4.2: Classification of N frames into S classes

hence the classification of one of the frames as belonging to I, meaning that the classifier has detected an impostor frame. As Speaker 2 is the class that $\frac{3}{8}$ of the frames belong to, and all the other speaker classes claim a lesser share of the classified frames, by consensus, this test sequence would be classified as Speaker 2. There is an advantage when finding the correct speaker by using majority voting in this way as there is uncertainty as to which frames contain truly speaker dependent information so it is not possible to exclusively select "usable" frames as input to the classifier. By implementing classification by consensus, a probability is obtained, based on the frequency of classification of test frames. The speaker is thus identified on the basis that it has the highest probability of being the correct speaker. It is possible to implement speaker identification using other methods than consensus, however the latter is used in this thesis as it provides a means by which to analyze classification results on a frame-by-frame basis. This enables an investigation of what frames are usable for speaker identification, a process that would not be possible if just one class label was returned for an entire test sentence. The frame-by-frame analysis is discussed in Chapter 9.

4.5 Confusion Matrices

The confusion matrix is a good measure of performance for each classifier implemented as part of the SID system. It contains information about the actual labels of data and the corresponding estimated labels of the same data. Each row in the confusion matrix represents a reference speaker and each column represents an estimated reference speaker.

A small example, using a set of just three hypothetical speakers, illustrates the use of the confusion matrix. These speakers are denoted as reference speakers **A**, **B**, and **C** with the corresponding estimated reference speakers denoted as A,B and C. The results of classification are in percentage. If the classifier assigns all test frames to the correct speakers, then all the frames in the confusion matrix are located in the diagonal, as all the frames for reference speaker **A** are estimated as belonging to Speaker A, and so forth, as seen in Figure 4.3.

In the more realistic case where only a certain amount of frames are correctly classified, values will be observed outside the diagonal of the confusion matrix. For the case where

A	100	0	0
B	0	100	0
C	0	0	100
	A	*B*	*C*

Figure 4.3: The confusion matrix for all frames classified correctly

as an example the test frames from Speaker **A** are classified as belonging to estimated speakers A,B and C at a rate of 59%, 12% and 29% respectively, the confusion matrix is shown in Figure 4.4, where similar situations apply for reference speakers **B** and **C**.

A	59	12	29
B	23	72	5
C	18	34	48
	A	*B*	*C*

Figure 4.4: The confusion matrix using for majority fraction of frames classified correctly

In the case shown in Figure 4.4 the identification of speakers is still correct in each case as the largest fraction of frames is found in the diagonal for each speaker, but there is less certainty as to which speaker is correct as a certain amount of frames are assigned to incorrect speakers.

Summing up the number of frames in the diagonal of a confusion matrix and then dividing this with the total amount of frames in the matrix provides a measure of how many frames are correctly classified in total. It can also be practical to use a confusion matrix in order to establish which speakers the wrongly classified frames are assigned to and thus detect eventual bias towards one speaker in a set. Confusion matrices are used in Chapters 5, 6 and 7 to display performance results for all three classifiers.

Chapter 5

Speaker Density Models

5.1 Introduction

The topic of this chapter is the creation of stochastic models that can be used for speaker identification. Each speaker i is represented by a model, λ_i. Based on these models, the class-conditional probability of a speaker who has uttered a test utterance $\mathbf{X} = (\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^N)$, where N is the total number of frames, can be computed for each frame of the observed test sequence. $\mathbf{x}^n$ is a feature vector extracted from the speech segment in frame n using one of the feature extraction methods discussed in Chapter 3.

The reference speaker models, $\lambda_i, i = 1 \ldots S$, are created during the enrollment phase of the SID system, using data from the training sentences uttered by each speaker. When the enrollment phase is completed, each speaker is represented by a model that has unique, speaker dependent, parameter values.

During the test phase, the test utterance is classified frame by frame, so that the input to the classifier is $\mathbf{x}^n$. The class-conditional probability density function that is evaluated using the reference density models is denoted as $p(\mathbf{x}^n|\lambda_i)$ and represents the probability that the speaker model λ_i generated the test frame data sample $\mathbf{x}^n$. The speaker identification is executed by determining the speaker model that maximizes this class-conditional probability density, as according to Section 4.1 this in turn corresponds to maximizing the posterior probability for the i^{th} speaker model when $\mathbf{x}^n$ is given, $P(\lambda_i|\mathbf{x})$. In the case of a sequence of independent feature vectors, the overall class-conditional density function is defined as the product of the density function for each test frame, as shown in Eq.(5.1), and the principle behind the identification of a speaker is shown in Eq.(5.2).

$$p(\mathbf{X}|\lambda_i) = \prod_{n=1}^{N} p(\mathbf{x}^n|\lambda_i) \tag{5.1}$$

$$i^* = \arg \max_{1 \leq i \leq S} \; p(\mathbf{X}|\lambda_i) \tag{5.2}$$

The identification rule of Eq.(5.2) does not apply to the implementation of the density models described in this chapter, as the density estimate of interest is that of each frame and not of an entire sequence of frames. Entire sequences will be classified according to

consensus over the classification results of all frames. Eq.(5.2) can be applied to one test frame instead of the entire sequence:

$$i^* = \arg \max_{1 \leq i \leq S} \; p(\mathbf{x}^n | \lambda_i) \tag{5.3}$$

Using Bayes' Theorem, the posterior probability for speaker model λ_i given the test vector $\mathbf{x}^n$ is calculated as:

$$P(\lambda_i | \mathbf{x}^n) = \frac{p(\mathbf{x}^n | \lambda_i) P(\lambda_i)}{p(\mathbf{x}^n)} \tag{5.4}$$

Each frame is classified as a member of the speaker model class that maximizes this posterior probability.

As the class models in this chapter are defined as functions of a set of parameters, the class-conditional densities are referred to as *likelihood* functions and they define the likelihood that, given λ_i, the test data point $\mathbf{x}^n$ is generated. These likelihood functions and the parameter set that they are dependent on are described in the next section.

Speaker density modelling is implemented for two purposes:

- speaker identification

- impostor detection

The speaker identification task is executed as described above, using the rule described in Eq.(5.3). Speaker identification using density modelling is discussed in detail in Sections 5.5. Although impostor detection can be included as a part of speaker identification in an open-set case, here it is implemented in a pre-classification phase. In Section 5.6, the impostor detection implementation using probability density functions is described. When an impostor is detected, the corresponding test data is excluded from the final classification phase. This alleviates the data load that is used as input to the classifier and should help in optimizing the SID system's performance.

5.2 Gaussian Mixture Models

The choice of a stochastic model for speaker identification has to be made with certain criteria in mind. Density models are used to describe the distribution of a data set, meaning that the model that is chosen must be able to fit the training data. The model must also be able to recognize test data that has a distribution similar to that of the training data. This ability is referred to as the generalization ability of the model. It deteriorates if the model is too finely tuned to the training data, as this means that test data cannot be recognized if it deviates a little from the training data.

There are two main subsets of density models; parametric and non-parametric. The non-parametric method does not have a pre-specified form and depends entirely on the data itself with no prior assumptions made. This leads to the ability to estimate the real density probability very closely, though for data sets of large dimensionality the problems of inadequate storage space and lengthy computational time may arise. In cases where there may be missing data points, the non-parametric model does not provide a good representation of the data.

The parametric methods, on the other hand, have a pre-specified functional form that depends on a number of parameters that can be adjusted. These adjustments are made when the parametric model is fitted to the data set during the training, or enrollment, phase. When data is sparse, the model retains to a certain level its ability to represent the input data. The disadvantage of these methods is that the density model may be unable to provide a good representation of the true input data density, as the latter may deviate substantially from the model's basic form. A third alternative to these methods are the semi-parametric methods [15].

The advantage of using semi-parametric methods is that they allow many degrees of freedom, making them more flexible and sensitive to the true density function of the input data than the parametric density models. The structure and parameters within the semi-parametric model, however, ensure that the density function has a known way of behaving and is thus more robust when dealing with sparse data than the non-parametric methods are, though they are subject to the curse of dimensionality explained in Section 4.2.

Semi-parametric distributions can be realized as *mixture* distributions [15]. The density model that is implemented here is the Mixture of Gaussians (MoG) model [41]. MoG models are chosen as they are known to be able to approximate any density with arbitrary precision, and because they have been proven to be very well suited for speech modelling tasks and subsequent text-independent speaker identification [59].

A MoG model consists of, as the name implies, a mixture of Gaussian distributions. A Gaussian distribution is defined by two parameters: μ, the mean, and σ^2, the variance. These parameters are defined in d-dimensional space as the mean vector μ of dimension $d \times 1$ and the covariance matrix $\mathbf{\Sigma}$ of dimension $d \times d$. The Gaussian density model, $\mathcal{N}(\mathbf{x}; \mu, \mathbf{\Sigma})$, is defined in Eq.(5.5). The dimensionality is determined by the dimension of the feature sets that are modelled. The frame of input data $\mathbf{x}$ is used free of frame index n here so as to simplify the initial derivations.

$$\mathcal{N}(\mathbf{x}; \mu, \mathbf{\Sigma}) = \frac{1}{(2\pi)^{d/2}|\mathbf{\Sigma}|^{1/2}} \exp\Big\{ -\frac{1}{2}(\mathbf{x}-\mu)^T \mathbf{\Sigma}^{-1}(\mathbf{x}-\mu)\Big\} \tag{5.5}$$

A third parameter defines the Mixture of Gaussians model. This is the mixing weight vector of dimensionality $M \times 1$, where M is the number of Gaussian components in the model. The MoG model is thus defined as a weighted sum of Gaussian density functions that is dependent on M Gaussian components and their corresponding mixing weights, denoted as $P(j), j = 1, \ldots, M$. The mixing weights are all positive and sum to unity. The MoG is defined in Eq.(5.6).

$$p(\mathbf{x}) = \sum_{j=1}^{M} P(j)\mathcal{N}(\mathbf{x}; \mu_j, \mathbf{\Sigma}_j) \tag{5.6}$$

where M is the number of mixture components, $\mathcal{N}(\mathbf{x}; \mu_\mathbf{j}, \mathbf{\Sigma_j})$ is the j^{th} Gaussian component density function, $P(j)$ is the probability for the j^{th} component and $p(\mathbf{x})$ is the MoG model for the feature vector of an observation sequence. The constraints that apply to

the probabilities that contribute to the mixture model are listed below:

$$\sum_{j=1}^{M} P(j) = 1, \qquad 0 \leq P(j) \leq 1, \qquad \int p(\mathbf{x}|j)d\mathbf{x} = 1$$

The number, M, of Gaussian components in the model has to be prespecified. Initially, a common value for M will be determined, whereafter separate values for each speaker model, M_i, will be implemented to determine whether this leads to an increase in overall classification performance. Apart from the number of components, the mixture model is flexible and not dependent on any prior knowledge of the distribution of data points in the feature vectors that are used as input for training or for testing. More specifically, this means that the MoG model is suitable for the text-independent task, as no predefined sequence of words has to be used as input to the model.

Each speaker is represented by a MoG model that is defined by a parameter set θ_i, so that $p(\mathbf{x})$ of Eq.(5.6) can be denoted as $p(\mathbf{x};\theta_i)$. The speaker-specific parameter set consists of the parameters $P_i(j)$, $\mu_{i,j}$ and $\mathbf{\Sigma}_{i,j}$, for $1 \leq i \leq S$ and $1 \leq j \leq M$.

To illustrate a basic MoG, a simple 1-dimensional MoG is derived. The data used to estimate the model is 100 frames from the training sentence a for Speaker 1, and the feature vector used is the 5^{th} MFCC. The distribution of these data points is shown in Figure 5.1.

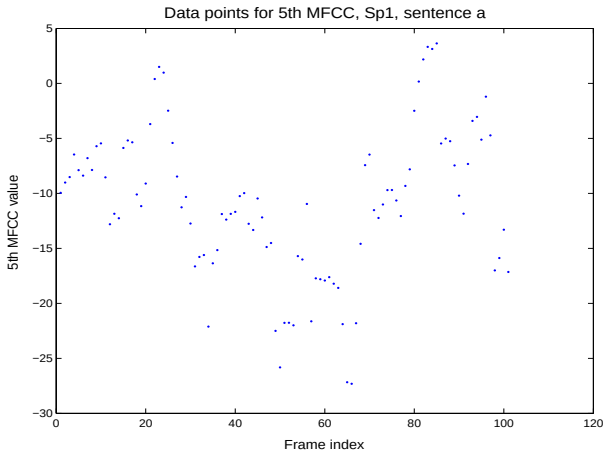


Figure 5.1: The values of the 5^{th} MFCC for 100 frames of Sp1, sentence a

The MoG model is implemented with $M = 3$ components. In Figure 5.2, the three Gaussian components are shown and the resulting overall model is drawn, based on the weights of each of the mixture components.

The means of the three Gaussians in the MoG model vary from -19 to 0, which roughly corresponds to the region that the data points in Firgure 5.1 occupy, as can be seen from the values of these points along the y-axis.

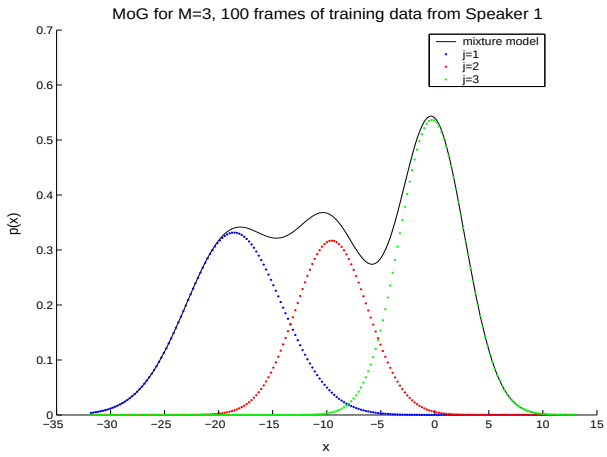


Figure 5.2: The 1-dimensional Mixture of Gaussians model for $M = 3$, the 5^{th} MFCC for 100 frames from Sp1, sentence a

5.3 The EM Algorithm

The parameters of the MoG model are estimated using the iterative Expectation-Maximization (EM) algorithm [60], [15], [43]. This algorithm uses the given data - the training sequence feature vectors - to determine the unknown parameters of the MoG model.

The training data points are denoted as $\mathbf{x}_n$, so that for each time frame n there is a feature vector $\mathbf{x}_n$. The subscript is used for n to avoid confusion with the test data points $\mathbf{x}^n$. The problem that must be solved is: given an observed data sample $\mathbf{x}_n$ that is generated by a MoG, estimate the means, variances and weights of the M mixtures of this MoG model. Basically, the optimal parameter set is obtained by maximizing the likelihood that the given training data is generated by the mixture model defined by this parameter set.

For the likelihood problem to be feasible, however, the data set must be *complete*, meaning that for each feature vector $\mathbf{x}_n$, there is a corresponding class label, z_n. In this case, each class label represents which Gaussian mixture component j is responsible for having generated the data point $\mathbf{x}_n$. The class labels correspond therefore to the mixture weights $P(j)$, meaning that $P(j) = 1$ if the j^{th} Gaussian component is responsible for having generated the data point $\mathbf{x}_n$. When the class label information is not available, the data set is *incomplete* and the estimation of the model parameters is referred to as *unsupervised* learning.

The EM-algorithm is able to determine a solution for the unsupervised learning problem based on incomplete data by substituting the labels z_n with the posterior probability for each component. This is done using the following iterative procedure:

1. An initial parameter set for the MoG model is chosen

2. Expectation (E) step: Using Bayes' theorem, the *old* parameters are used to determine the posterior probability for each class (Gaussian component), given the

training data sequence:

$$P(j|\mathbf{x}_n) = \frac{p(\mathbf{x}_n|j)P(j)}{p(\mathbf{x}_n)}$$

and the expectation function is determined.

3. Maximization (M) step: By maximizing the expectation likelihood function found in step 2, the *new* parameter set is determined.

4. If convergence is not reached, return to step 2.

The initialization of the EM algorithm and the detailed descriptions of the E-step and M-step to estimate parameters are provided in Appendix B.

Convergence of the EM algorithm is obtained when the new parameter set is equal to the old one, or if the difference between these parameter values becomes smaller than a certain tolerance threshold. With each new estimate of the parameter values, the likelihood that the resulting model generated the test sequence increases, while the relative improvement of the likelihood decreases.

The speed of convergence for the EM-algorithm that is implemented using [44] is shown in Figure 5.3. This EM algorithm has to meet two criteria to carry on iterating back to the E-step once the M-step is completed: the relative likelihood improvement is higher than a small, predetermined threshold and the amount of iterations does not exceed a preset maximum value. The likelihood improvement is shown as a function of iterations of the EM algorithm where $M = 16$ for 3 feature sets of differing dimensionality: the 12MFCC ($d = 12$), the 12ΔMFCC ($d = 24$) and the 12$\Delta\Delta$MFCC ($d = 36$) feature sets. The number of training points (frames from sentence a) is 1328.

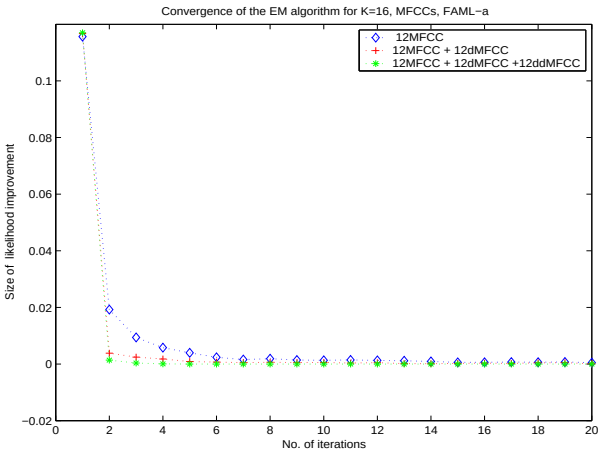


Figure 5.3: Convergence of the EM algorithm for all 6 speakers, sentence a, 12 MFCC feature sets, MoG of order 16

From Figure 5.3, it can be seen that the EM-algorithm converges rapidly, almost reaching convergence within the first 8 iterations for all 3 feature sets. From the discussion in B concerning the initialization of parameters when implementing the EM algorithm, this is not necessarily a measure of good performance, as rapid convergence does not ensure

rapid convergence to an extremum. Under the assumption that the drastic decrease in log-likelihood improvement does correspond to the approach towards a minimum (as the minimum of the negative log-likelihood is what is sought after here), the best results (in the form of the most rapid convergence) are achieved for the $12\Delta\Delta$ feature set. This could be due to the fact that more information is available in the feature set that includes the first and second temporal derivatives when compared to the other sets, though a large number of additional computations are required because of the increased dimensionality. The EM algorithm is implemented with full covariance matrices that are computationally heavy and could have been approximated with diagonal covariance matrices instead. A discussion on the convergence issues of the EM algorithm is not included here but can be found by referring to [61].

5.4 Reference Density Models

Using the concepts of Chapter 4 and the theory of the previous sections of this chapter, density modelling for speaker identification is now put into perspective. Each reference speaker is represented by a corresponding MoG density model. The input data to these models is the training set that consists of d-dimensional feature vectors, one for each of N frames. The EM algorithm is implemented to estimate the parameters of the d-dimensional Mixture of M Gaussians model. Figure 5.4 is identical to Figure 1.3, and shows the significance of density estimation in the SID system.

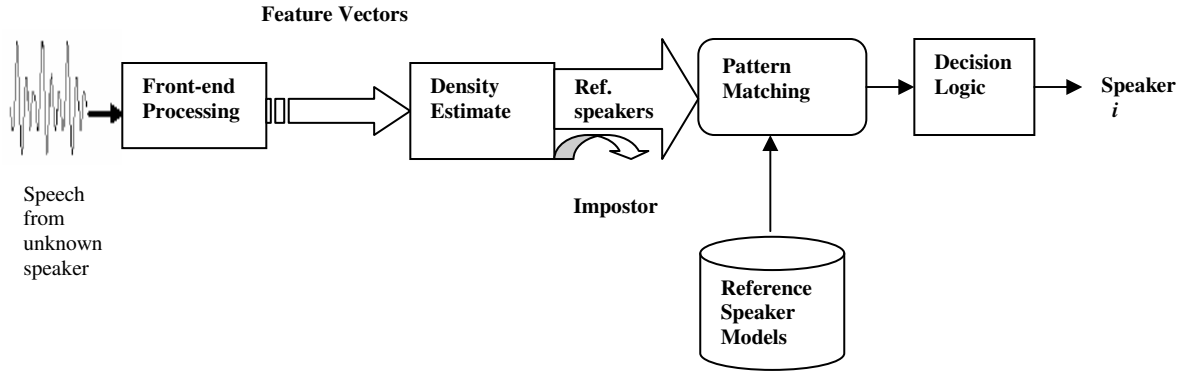


Figure 5.4: A Speaker Identification system with density estimation for impostor detection

Mathematically, the probability estimate of each training data sequence generated by a speaker reference model can be represented by the density function shown in Eq.(5.7).

$$p(\mathbf{x}_{i,n}|\lambda_i) = \sum_{j=1}^{M} P_i(j) \cdot p_i(\mathbf{x}_{i,n}|j) \tag{5.7}$$

where $\mathbf{x}_{i,n}$ is a sequence of frames from the training set for speaker i.

The reference speaker models are created in the enrollment phase and used in the test phase as a template with which to match patterns, as described in Section 1.1. An unknown sequence of test frames, $\mathbf{X^{test}} = \{\mathbf{x}^1, \mathbf{x}^2 \ldots, \mathbf{x}^N\}$ is classified, frame by frame, based on the value of the likelihood evaluation $p(\mathbf{x}^n|\lambda_i)$. The likelihood is calculated for each speaker model $\lambda_i,\ i = 1, \ldots, S$. This likelihood for test frame $\mathbf{x}^n$, given the model for speaker i, is derived in Eq.(5.8).

$$p(\mathbf{x}^n|\lambda_i) = \sum_{j=1}^{M} P_i(j) \cdot \frac{1}{(2\pi)^{d/2}|\Sigma_{i,j}|^{1/2}} exp\Big\{ -\frac{1}{2}(\mathbf{x}^n - \mu_{\mathbf{i,j}})^T \mathbf{\Sigma_{i,j}}^{-1}(\mathbf{x}^n - \mu_{\mathbf{i,j}})\Big\} \tag{5.8}$$

where $P_i(j)$ is a scalar that represents the weight of the j^{th} component for speaker i.

The process of calculating the class-conditional density of a test data point $\mathbf{x}^n$ using a MoG model is depicted in Figure 5.5.

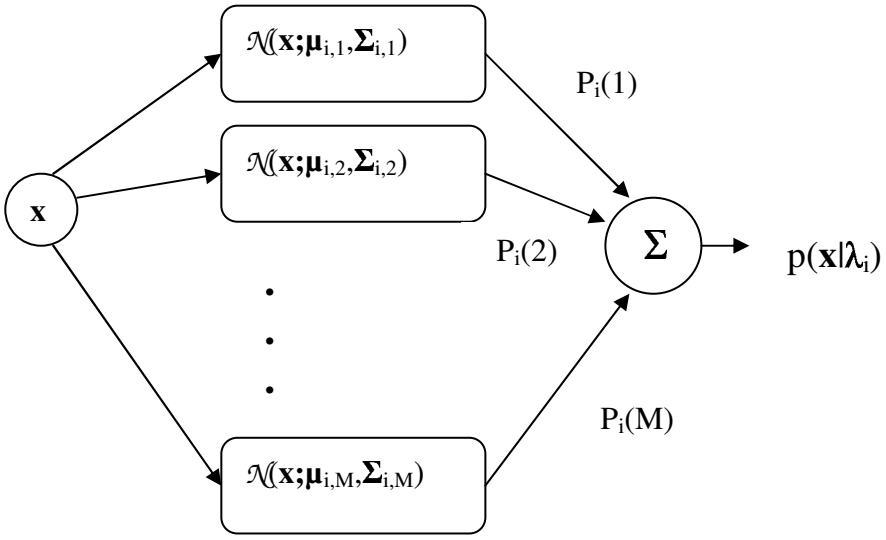


Figure 5.5: The process of probability estimation using a MoG model

The implementation of the density evaluation procedure is executed by first taking the natural logarithm of the right-hand side of Eq.(5.8). This is done to ensure a higher level of precision and more numerical stability, esp. in the case where data points deviate significantly from the average distribution and thus cause very large differences in the exponent of Eq.(5.8). The final results are obtained by transforming the results back to the original domain by using the inverse of the natural logarithm.

5.5 Speaker Identification using MoG Models

Once the probability density function of a test frame data sample for each reference speaker model is determined, decision logic in the form of Bayes' theorem is implemented.

Depending on the relative values of the posterior probabilities obtained (in order to determine the maximum posterior probability), each frame of a given test sequence is classified as belonging to Speaker 1 - S, where $S = 6$ in this case. When an entire test sequence of frames has been classified, the speaker identification is based on consensus. In this section, the closed-set identification task is analyzed, to be followed by the implementation of an impostor detection method that is capable of providing a pre-classification solution to the open-set problem.

In Figure 5.6, one frame, $\mathbf{x}^{39}$, of a test sequence is used as input to the MoG classifier and the density function for this test frame is evaluated for each reference model. As the maximum density estimates for one speaker model can differ from the remaining density estimates by a factor 10 or more, the natural logarithm of these likelihoods is taken so that the values are restricted to a more useable scale. The results of taking the logarithm of the likelihood evaluation for test frame $\mathbf{x}^{39}$ are shown in Figure 5.6. The six subplots each represent test speech from one of the six speakers. In each subplot the x-axis shows what speaker model is used and the y-axis the resultant density estimation after taking the logarithm.

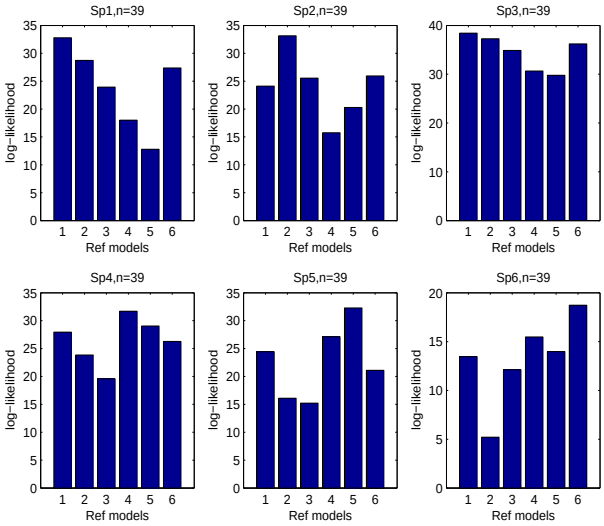


Figure 5.6: The log-likelihood evaluation for each reference speaker for one frame

From the log-likelihood values in Figure 5.6, it is possible to see that for all speakers excluding Speaker 3, the maximum log-likelihood of the correct speaker is only approached by one or two of the other likelihood values for the remaining speaker models, while for Speaker 3 there exists a lot more ambiguity as to which speaker is the correct one. Although this analysis is based on one frame only, it does show the tendencies that are observable when entire test sequences of frames are considered.

In Chapter 9, different feature sets will be used to evaluate the classifier's performance. It is therefore not convenient to allow too many other variable parameters in the classifier. As a preliminary measure to allow the initial implementation to be executed, the values of a few parameters are determined here. These parameters include M, the number of mixtures in the MoG model, and N, the number of test frames needed to enable iden-

tification. The feature set comprised of 12MFCC + 12ΔMFCC coefficients is used as a yardstick, as this feature set is commonly used in speaker recognition tasks and so is assumed to be reliable. However, for the SID system presented in this thesis, this feature set has not been proven to outperform the alternative feature sets at this point in time. For future reference, this 24-dimensional feature set is called the reference feature set.

During the preliminary trials, it was observed that the parameter set $\left[P_i, \mu_i \text{ and } \mathbf{\Sigma_i}\right]$ varies with each run of the EM-algorithm. At times a tendency to classify all test sentences as belonging to one reference speaker was noted. This means that no one model reflects an absolute speaker model parameter set for a particular training set and this is a source of unreliability in the classification process. Although this problem remains untreated for the testing implemented in what follows, it must be considered as a possible reason for the inability of the MoG classifier to perform well in some cases. The instability of the MoG model is due to the high dimensionality of the reference speaker set that leads to the sparse training data problem that is the direct result of the curse of dimensionality. F.ex., there are 9896 training frames for Speaker 1 and the dimensionality of the covariance matrices for each Gaussian component j is $24 \times 24 = 576$. As there is no additional data available for the reference speakers, the MoG classifier is implemented as is and the testing commenced, using in each case the reference speaker model that yields the best performance for classifying test frames, chosen after training is executed a number of times with the same training set.

Once the speaker models have been estimated, the preliminary testing to determine certain variables is implemented. An important variable parameter in the MoG model that needs to be determined is the number of mixture components, M. It can be expected that the higher the number of Gaussian components, the better the density model can fit to the real training set distribution as the model is more flexible. However, the model must not be too complex either, as this would increase computing time and the model would risk fitting the training data too accurately. Over-fitting the training data set leads to a decrease in robustness in the general case, and the ability to classify test data is therefore decreased. In order to observe how the number of components affects the rate of correct classification of frames in the system, M is varied from $M = 2$ to $M = 48$ Gaussian components and the percent of correctly classified frames is recorded for each different value of M. This is done for $N = 800$ frames, corresponding to 8s, of test data from each of the reference speakers. The training set contains all 7 training sentences for each speaker. This corresponds to between 68.4s and 93.6s of speech from each reference speaker (see Table 8.1).

The results are shown in Figure 5.7.

The dotted line in Figure 5.7 represents the total percentage of correctly classified frames, divided by the number of speakers. This is done because the results for different speakers vary so much for each value of M that the average over the entire set of reference speakers must be used to establish which model has the best overall performance. From $M = 2$ to $M = 12$, the average is quite stable and the best result is obtained for $M = 12$, though by a small margin. As the number of Gaussian components is increased, the amount of correctly classified frames for individual speakers increases significantly, yet as the other speakers' results drop considerably, the average is decreased. It is interesting to note that for $M = 16$, it is possible to identify Speaker 3, who in this case is identified

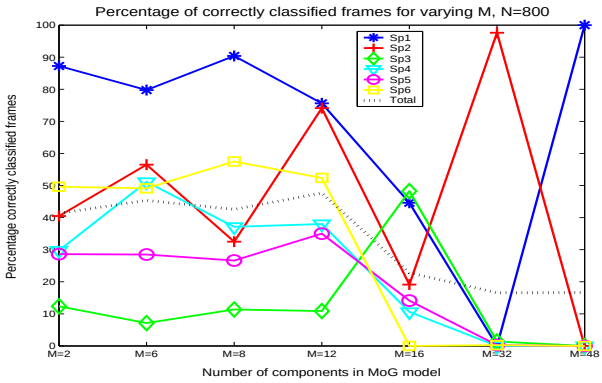


Figure 5.7: The percentage of correctly classified frames for $N = 800$ and varying number of components

correctly in almost 50% of the frames. Yet as the number of correctly classified frames for the other speakers is greatly reduced, the number of Gaussian components to be used is thus set to $M = 12$, despite the low performance for Speaker 3. A reevaluation of the effect of the number of components in the MoG model on the correct frame classification rate must be executed for the different feature sets that are implemented.

As the number of mixtures can now be set to a constant value of 12 for the reference feature set, the parameter N can be determined. N is the number of frames that must be included in the consensus to ensure a reliable classification result. This number can also vary for different speakers and for different feature sets. A basic idea of how the number of frames affects the ability of the classifier to make a reliable identification is established by using the reference feature set. In Figure 5.8 it is observed that as the number of frames in the test sequence is increased, the total percentage of frames that are correctly classified is also increased. This holds true for all 6 reference speakers, although the increase in percentage is minimal for Speaker 3 when compared to the significant and almost linear increase recorded for Speakers 1 and 2.

The classification of all $N = 800$ frames from each reference speaker's test data is shown in Figures 5.9 and 5.10. The colourbars on the right-hand side of each classified sequence of frames shows which colour indicates the corresponding reference speaker. F.ex. Speaker 1 is represented by a dark brown colour, thus every frame that is coloured dark brown for the test data from Speaker 1 is correctly classified.

The total classification based on consensus over all 800 frames is a correct identification of Speakers 1,2,4,5 and 6. The number of frames that are correctly classified for the speech utterance made by Speaker 3 is so small that it is obvious why the system fails to identify this speaker, see Figure 5.9. The majority of frames here are classified as belonging to Speaker 1. This is in accordance with the various results that are recorded and displayed in Figures 5.6, 5.7 and 5.8.

Based on Figure 5.8, a larger number of frames yields a better identification rate. However, a small number of frames would decrease the time needed to decide on a class so it is interesting to determine how many frames are sufficient in order for the identification to be reliable. This number is different for each of the different speakers, as can be seen in Figure 5.11. Classification by consensus is implemented for a varying total number of

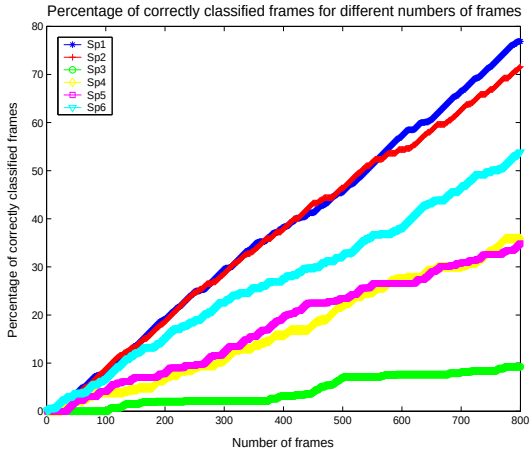


Figure 5.8: The percentage of correctly classified frames as a function of the number of frames

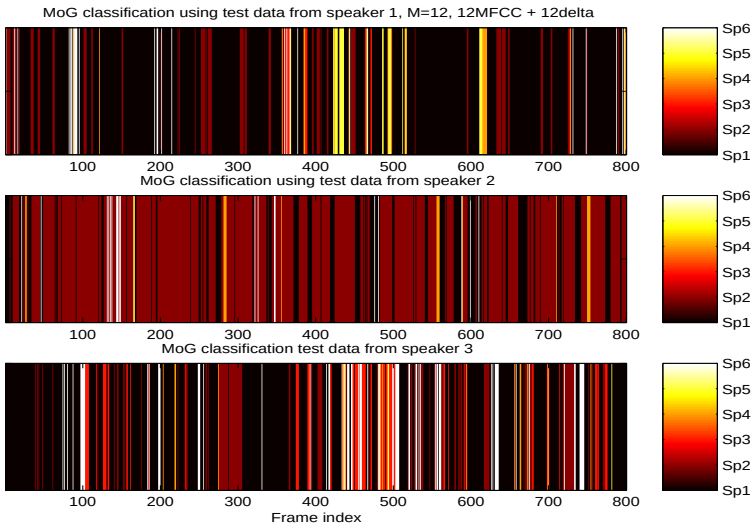


Figure 5.9: Classification of $N = 800$ frames for the female speakers, $M = 12$

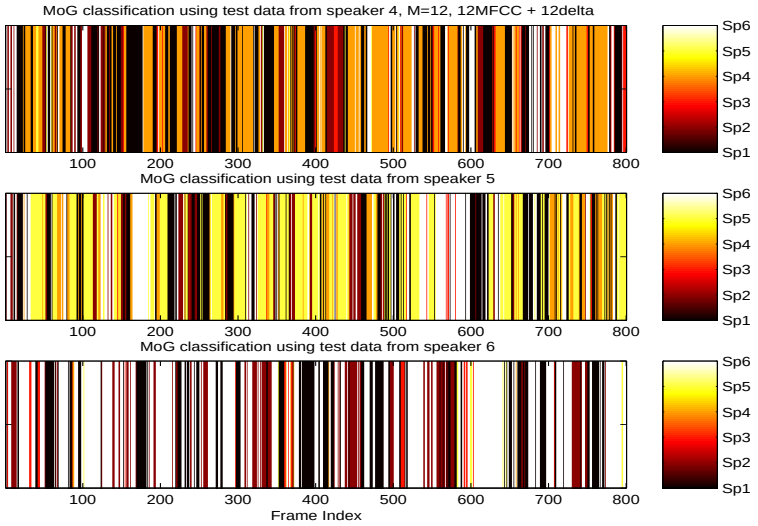


Figure 5.10: Classification of $N = 800$ frames for the male speakers, $M = 12$

frames $N = 1 \ldots 800$.

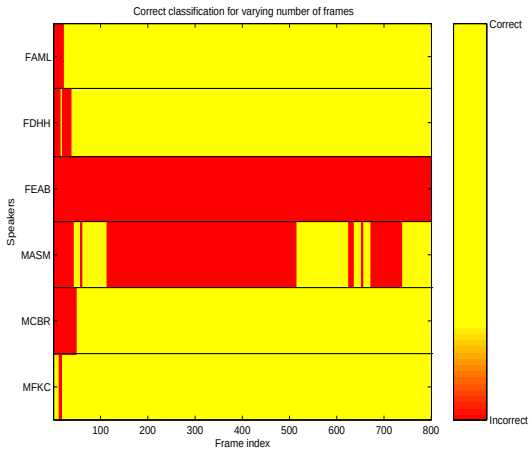


Figure 5.11: The correct classification of each speaker for varying number of frames

For each N the classification of the test sequence frames is labelled as being *correct* (yellow) if the classification matches the identity of the speaker that uttered the test sentence, or *incorrect* (red) if this is not the case.

While the identification of Speakers 1, 2, 4, 5 and 6 is successful for a relatively small number of frames (correct classification is achieved for all these speakers at just above $\frac{1}{2}$s of test speech), it is interesting to note that for Speaker 4 this classification seems coincidental until the number of frames is greatly increased, at which time the classification becomes more reliable. This stability is already achieved at a much lower total frame count for Speakers 1,2,5 and 6, where practically the entire test sequence is correctly classified. From Figure 5.11 it can be seen that Speaker 3 is not correctly identified for any length of test data speech, up to $N = 800$. Here, increasing N is of no significance, as the majority of frames are continually classified as belonging to Speaker 1. This may be due

to an imprecise modelling of Speaker 3's training data, a very plausible possibility when the high dimensionality of the reference set is taken into consideration with the effects of curse of dimensionality in mind. Other feature sets may prove more suitable for MoG model classification of Speaker 3.

In order to get a better idea as to how many frames are allocated to each reference speaker and to establish the possible existence of bias for a certain speaker, the confusion matrix for the identification using the MoG model classifier is shown below.

$$\mathbf{C_{MoG}} = \begin{pmatrix} \mathbf{76.88} & 12.00 & 1.88 & 2.38 & 3.13 & 3.75 \\ 23.50 & \mathbf{71.63} & 0 & 2.00 & 0.38 & 2.50 \\ \mathbf{64.50} & 14.63 & 9.25 & 1.25 & 1.00 & 9.38 \\ 33.88 & 10.88 & 4.63 & \mathbf{35.88} & 1.38 & 13.38 \\ 23.00 & 8.75 & 1.38 & 10.50 & \mathbf{34.88} & 21.50 \\ 24.75 & 16.00 & 3.13 & 0.88 & 1.38 & \mathbf{53.88} \end{pmatrix}$$

From the confusion matrix it can be seen that there is a bias towards Speaker 1, as this is the speaker that claims the most frames for Speakers 1 and 3, and the second most frames for the remaining reference speakers. As Speaker 1 is identified on the basis of a very large percentage of classified frames, a method of removing bias can be implemented by setting a minimum threshold for the number of frames classified as Speaker 1 before a speaker is estimated as being Speaker 1. This does not, however, remedy the misclassification of Speaker 3, as this speaker would then be classified as Speaker 2. As this speaker is also identified by a substantial amount of frames, a threshold for removing bias towards Speaker 2 can also be implemented. This would, however, result in that Speaker 3 is classified as Speaker 6. As Speaker 6 is not classified with a large percentage of correct frames, it is not feasible to also remove bias here. Removing bias towards Speaker 1 is thus not implemented as it does not enable the system to recognize speech from Speaker 3. It can be used, if desired, to remove ambiguity within the classifier for the other speakers, in this case notably for Speaker 4.

Up to now, the identification of a speaker has been based on majority voting implemented by simply taking all available classified frames and deciding on the speaker that claims the majority of frames. Alternatively, a rule could be implemented that if the amount of frames belonging to one speaker is higher than a pre-specified threshold, then the test sequence was uttered by this speaker.

The threshold is denoted as η and an attempt to derive it for the reference feature set is made. The rate of correct classification is measured for each increase in the value of η. It is found that $\eta > 50\%$ gives the optimal results. The value of η is obviously dependent on the amount of test data available, as an increase in the length of the speech segment leads to a smaller η being needed, based on the results shown in Figure 5.11. As the results in the confusion matrix $\mathbf{C_{MoG}}$ reveal that both Speakers 4 and 5 were correctly classified even though the fraction of frames correctly classified here was below 50% sheds doubt as to how practical such a thresholding technique is. It may require a very large amount of frames to obtain a 50% correct classification rate for one speaker, while simply determining the maximum fraction of classified frames might prove more efficient.

Although the number of mixtures used for all the preceding preliminary trials was set at $M = 12$, it could result in a computational advantage if this number could be reduced without adversely affecting system performance. Once again, if we study Figure 5.7, it is observed that the difference between the average correct classification rate from $M = 2$-$M = 12$ mixtures does not vary much. In order to establish whether a number of mixtures lower than 12 can yield good performance, a number of runs were executed for differing numbers of components and over the set of all 22 speakers from the ELSDSR database in order to avoid dependency on specific speakers. Although the overall performance for this much larger set of speakers is decreased when compared to performance with the smaller set of the 6 reference speakers (only 50% of speakers could be identified) , it was possible to ascertain that the most recurring and best results were achieved for $M = 2$. The numbers of Gaussian mixtures were also made speaker specific but this yielded the same results, i.e. that little could be gained from using more than $M = 2$ for all speakers. Varying the number M_i for each speaker would be more beneficial if there was a greater difference between the amount of training data available for each speaker, as larger data sets are modelled more accurately with a larger number of Gaussian components than small data sets are. The number of Gaussian components is thus set to $M = 2$ and as many test frames as possible are included for the tests that are implemented using MoG classification of other feature sets in Chapter 9.

5.6 Impostor Detection using MoG Models

As a person wearing a hearing aid is unavoidably in contact with numerous unfamiliar people (and other sources of sounds) in the course of a single day, closed-set identification limits the optimal use of the instrument. Every single voice and sound that is registered is classified as being one of the reference speakers and in doing so the settings for that reference speaker are chosen. These settings risk not being appropriate for the impostor, leading to an experience of decreased performance by the wearer of the hearing instrument. The purpose of detecting an impostor is therefore to prevent this from happening, and to enable the eventual implementation of a separate, general, setting that is more suitable for impostors. Here, a method of detecting impostors based on probability density estimation is described.

From Section 4.3, impostor detection is based on the estimation of class-conditional density functions, where the assumption that the likelihood of a test frame from the correct speaker model is much larger than that of an incorrect speaker can be written as:

$$p(\mathbf{x}_i^n|\lambda_i) \gg p(\mathbf{x}_j^n|\lambda_i),\ j \neq i \tag{5.9}$$

Through extension of this observation, impostor detection can be implemented: It is assumed that an impostor will have a relatively low likelihood score for all the reference models. A method of exploiting this in order to detect impostors is to determine a threshold for each reference density model. This threshold defines the boundary between the likelihood value of a reference speaker and that of an impostor. For a reference speaker model λ_i, all speakers other than speaker i are viewed as impostors, irrespective of whether it is another reference speaker or a complete outsider.

The speaker-specific threshold value is related to the Θ threshold of speaker verification, only here as many thresholds there are reference speakers must be determined. These thresholds are denoted as τ_i. When deriving the optimal value of τ_i, certain considerations must be taken into account. The challenge is to determine a value for τ_i that is small enough to ensure that the highest possible number of frames that do actually belong to speaker i get classified as such, while making it large enough that the fewest possible impostor frames are accepted as being from speaker i.

The trade-off between the two conditions that must be satisfied when determining a value for τ_i is shown in Figure 5.12. Here speaker 1's reference density model, λ_1, is used. A small number of frames, $N = 5$, is taken from Speakers 1's test data as well as from the other reference speakers and some speakers from outside the reference set, 9 speakers in total. Two threshold values are found; one that is relatively large ($\frac{1}{2}$ of the average reference density for all training frames of reference Speaker 1), and one that is smaller ($\frac{1}{4}$ of the average reference density). The results for these two values are shown in Figure 5.12.

The second row of images in Figure 5.12 shows the true class membership of the frames.

For a larger τ_1 (top left-hand corner of Figure 5.12), only one of the five frames from Speaker 1 is correctly identified. When the threshold is made smaller (top right-hand corner), an additional two frames are correctly identified but now there is also an increase in the number of impostor frames that are incorrectly accepted and classified as Speaker

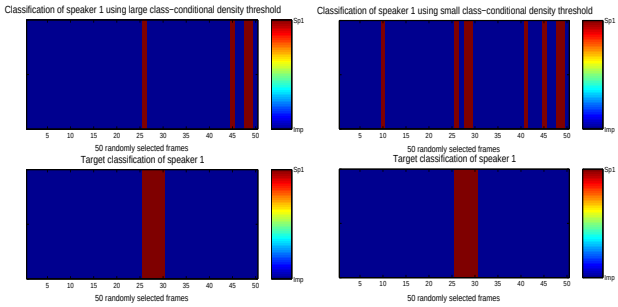


Figure 5.12: The detection of impostors using a large and a small value for τ_1

1 instead of impostors. A trade-off criteria must be established as it is not possible to completely eliminate one error rate without adversely affecting the other. This leads to the method for determining a value for τ_i for each reference speaker model, which will be described in the following.

The trade-off problem discussed in Section 4.3 means that in order to determine τ_i, a balance must be struck between two kinds of errors - the false acceptance and the false rejection error. The false acceptance error measures how often an impostor speaker is labelled as being reference speaker i. The false rejection error reflects how many times the test data from speaker i is classified as coming from an impostor. It is established by the results obtained in Figure 5.12 that for small values of τ_i, the false acceptance rate is high and the false rejection rate is low, while when τ_i is increased, the amount of false acceptances will fall while the opposite is true for false rejections. In order to find the optimal value for τ_i, the total error must be as small as possible. In the case of speaker identification for a hearing instrument, it is more critical that the false rejection error is very low, as this corresponds to minimizing the risk that a reference speaker is classified as an impostor, which is more serious than if an impostor is accepted as a reference speaker. Once again, final classification is based on consensus.

In order to derive a value for τ_i, the following procedure is implemented: the test data from each speaker is divided into two subsets. One set is used to determine an optimal value for τ_i, while the other set is used to test τ_i in order to establish how effective it is at separating impostors from reference speakers in a text-independent situation. The subset of data used to determine τ_i is referred to as the validation set, while the set used to test τ_i is referred to as the test set. A varying threshold value is tested for each frame of the validation set sentences. The threshold is initialized at a low value, and the false rejection and false acceptance errors are registered. For each increment of τ_i, the two errors are noted. The total error is based on the sum of the two errors in percentage. Two criteria for determining the optimal value of τ_i are tested: the minimum error rate and the equal error rate, denoted by the corresponding threshold values $\tau_{i,min}$ and $\tau_{i,eer}$. The minimum error rate is simply the minimum value of the total error. The equal error rate is the point where the false acceptance rate is equal to the false rejection rate, i.e. where as many impostors are classified as reference speakers (in percentage) as reference speakers are classified as impostors. The derivation of this error is discussed in more detail in [30]. Of importance here is to establish which type of error leads to better overall performance

in the impostor detection phase.

Once the optimal value for τ_i has been empirically determined using the validation set of likelihood estimates, the test set is used to establish the MoG impostor detectors ability to differentiate between reference and impostor speakers. This is done frame by frame, so that the choosing of a correct speaker can be written as:

$$p(\mathbf{x}^n|\lambda_i) > \tau_i \Rightarrow H_1 \quad (5.10)$$
$$p(\mathbf{x}^n|\lambda_i) \leq \tau_i \Rightarrow H_2 \quad (5.11)$$

where H_1 corresponds to the "Accept" decision of a test frame as belonging to speaker i and H_2 corresponds to the "Reject" option, i.e. the detection of an impostor, as explained in Section 1.1.

When all the test frame samples have been classified, majority voting is applied: if more than half the classified frames in the sequence are labelled as belonging to either a reference speaker or an impostor, this is the final result.

The reference feature set for a randomly selected reference speaker, Speaker 3, is used to test the impostor detection procedure. The validation and test sets are both comprised of $N = 300$ frames of data from Speaker 3's test data in the reference feature set. The sets do not overlap. This means that there is roughly 3s of speech available to determine τ_3 and 3s to test it. As impostor speakers, the remaining 5 reference set speakers and 10 other speakers are used. Validation and test sets of the same length as for Speaker 3 are also extracted for these speakers. The false rejection and false acceptance errors are recorded and the two errors are shown in Figure 5.13. As expected, the false rejection error increases as the threshold value gets larger, as more reference speaker frames are classified as impostors. The opposite holds true for the false acceptance rate, which decreases as τ_3 becomes larger.

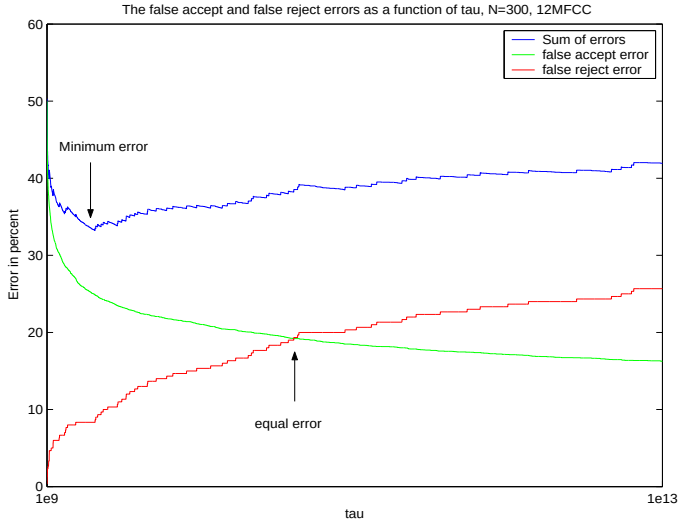


Figure 5.13: False rejection error and false acceptance error for the validation set

As can be seen in Figure 5.13, the minimum total error is found at a lower threshold value than for the equal error rate. This is due to the fact that, after a short while, the false rejection of reference speaker frames increases at a faster pace than the acceptance error rate decreases for each increase of τ_3. As the objective is to preferably accept too many impostors rather than risk rejecting a high number of reference speaker frames, the minimum error rate is a better choice, as it ensures that the false rejection rate is still quite low, while the false acceptance rate is not at its maximum.

The performance for each of these types of error is obtained by applying both $\tau_{3,min}$ and $\tau_{3,eer}$ to the test set. The results are listed in Table 5.1.

	Minimum Error Criteria	Equal Error Criteria
False acceptances	1231	911
False rejections	40	61
Overall test error	26.48%	20.25%
Correct id. of ref. speaker	Yes	Yes
Impostors classified as ref. speaker (out of 15)	4	3

Table 5.1: Results using the minimum and equal error rates

The overall test error is seen to be lowest for the equal error rate, and fewer impostors are accepted, as can be expected. It is clear, though, that the risk of rejecting a reference speaker test frame is much smaller for the minimum total error criteria. The minimum error was determined at a value that is factor 10^3 smaller than the average reference density for the training data of Speaker 3, while $\tau_{3,eer}$ is only a factor 10^2 smaller than this.

The impostor detection method is thus implemented for all reference speaker models trained on the 12ΔMFCC feature set by using $\tau_{i,min}$ as the threshold value. The classification of reference speakers and impostors is based on consensus when the density estimation of all the frames have resulted in a classification of each frame as a reference speaker or an impostor. The reference speakers are all correctly classified as such, while of the 10 impostors, 1 is classified as being a reference speaker. This gives an impostor detection rate of 90% and a reference speaker detection rate of 100%. Interestingly, for only 100ms of test speech available, the reference speakers are still detected but 40% of the impostors are classified as being reference speakers. Limiting test data length thus does not lead to inferior performance in the case of classifying reference speakers, but it has the undesirable effect of decreasing the number of impostors that are detected and this includes more irrelevant data in the closed-set classification phase.

Once an impostor has been detected, the relevant speech data can be excluded from the final classification phase. The density function estimates that are not rejected as impostors are used to determine the posterior probabilities of each reference speaker model. This procedure is identical to the closed-set case as the speakers that are not detected as impostors are assumed to be reference speakers. The results of using density modelling as a classification method and for impostor detection for different feature sets will be implemented are presented in Chapter 9.

Chapter 6

k-Nearest Neighbour

6.1 Introduction

A simple, straightforward and relatively flexible classifier, k-Nearest Neighbour (k-NN) is a non-parametric classification method that does not require a training stage in which system parameters are optimized. There is therefore no need for prior knowledge of the distribution of test and training data points in feature space. The k-NN classifier is simple to implement and practical for the purpose of comparing its results with those obtained from the more complex classifiers.

As the k-NN method is non-parametric, there is no separation of an enrollment phase from a test phase, and so the input to the classifier consists of both labelled training data and unlabelled test data. The labels associated with each frame of the training feature set indicates class membership of one of the six classes that correspond to the speakers in the reference data set. The labelled training points are used as reference points in the d-dimensional feature space of the classifier, where d is the number of data points in each training and test set feature vector. Each new test data point, $\mathbf{x}^n$, consisting of the feature vector for one frame of a test sentence, is then compared to these reference samples in order to establish the class to which it belongs. The comparison is implemented by using a distance metric that determines the test point's k nearest reference points (neighbours). The test point is hereafter assigned to the class that has a majority representation among these k nearest neighbours.

To illustrate the classification process in k-NN for $k = 3$ in 2-dimensional space, we refer to Figure 6.1, taken from [16].

In Figure 6.1, the point "**B?**" represents the unknown test data point. As it has two circles (corresponding to points from class **B**) and only one square (class **A**) as it's 3 nearest neighbours, the pattern is assigned to class B.

The k-NN classifier structure is, however, not entirely dependent on the input data, as two internal parameters must be defined prior to the commencement of the classification process: the distance metric and the value of k. A common distance metric used to calculate the distance between the labelled training points and the unknown test points is the *Euclidian distance*. An assumption is made that all the data points can be represented in Euclidean space, i.e. $\mathbf{x} \epsilon \Re^d$. Subsequently, it is possible to implement the Squared

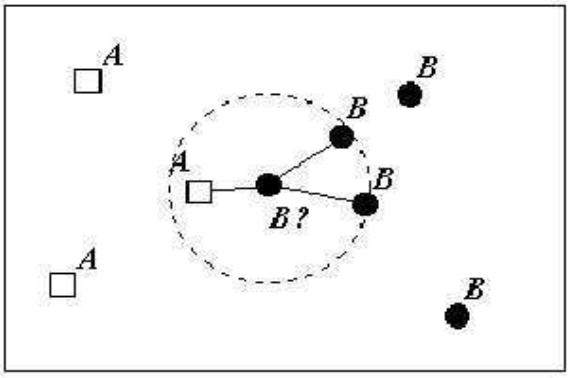


Figure 6.1: k-Nearest Neighbour selection for $k = 3$

Euclidean distance [37] that determines the distance between two points in the space $\Re^d$. k-NN classification is instance-based, so that each test feature vector is interpreted as one instance that has to be compared with another instance, in this case with a training set feature vector. A test feature vector is denoted as $\mathbf{x}^n = x^n(1), x^n(2), \ldots, x^n(d)$ and a training vector for frame n as $\mathbf{x}_n = x_n(1), x_n(2), \ldots, x_n(d)$. The squared Euclidean distance that measures the distance between two patterns $\mathbf{x}^n$ and $\mathbf{x}_n$ is shown in Eq.(6.1).

$$dist(\mathbf{x}^n, \mathbf{x}_n) = \sqrt{\sum_{z=1}^{d} (x^n(z) - x_n(z))^2} \tag{6.1}$$

Effectively computing the square root of the sum of squares of the difference between two instances.

The Euclidean distance will be used in the implementation of the k-NN classifier in all the trials that are conducted in Sections 6.2 and 6.3 and in Chapter 9. The optimal choice of the parameter k must be determined empirically, based on identification results. It is important that k not be too small, as in this case the classifier can become highly sensitive to each data point and the variance within the classifier becomes large, while if k is too large some class-specific information may be lost, as the k nearest neighbours may then encompass a merging between two or more classes. The optimal value of k is determined when a trade-off between these two extremities is found and the k-NN performance is optimal.

Once the distances between a test and training instance within the classifier have been determined, a decision rule is implemented to enable classification of the test vector. This is a relatively straightforward rule. In the case where there is an equal representation of two classes within the k nearest neighbour reference points, a random selection of one of the two classes is implemented. Otherwise, the class that is most heavily represented in the group of selected neighbouring reference data points is interpreted as being the one that has the highest probability of having generated the test data point $\mathbf{x}^n$. It is thus possible to write the k-NN decision rule as a special case of density modelling by viewing the class-conditional density estimate as being:

$$p(\mathbf{x}^n|C_i) = \frac{k_i}{N_i V^n} \tag{6.2}$$

where k_i is the number of nearest neighbour points that are members of the class C_i, N_i is the number of training points in class C_i and V^n denotes the volume of the D-dimensional hypersphere that contains the test point $\mathbf{x}^n$ and it's k nearest neighbours. This volume gives an indication of the density of data points in feature space: when the points are placed far apart, the volume is large and the density thus small, while the opposite is true for a high concentration of points contained within the hypersphere.

Using Bayes' Theorem, shown in Eq.(4.1), the decision rule can once again be defined here as classifying a given data point as belonging to the class that maximizes the posterior probability. The prior probability is given as the fraction of points that belong to class C_i, see Eq.(6.3). The unconditional probability density is defined as the estimation of all data points regardless of class membership, as shown in Eq.(6.4).

$$P(C_i) = \frac{N_i}{N} \tag{6.3}$$

$$p(\mathbf{x}^n) = \frac{k}{NV^n} \tag{6.4}$$

Using Bayes' Theorem, the posterior probability is computed as:

$$P(C_i|\mathbf{x}^n) = \frac{p(\mathbf{x}^n|C_i)P(C_i)}{p(\mathbf{x}^n)} \tag{6.5}$$

$$= \frac{k_i N_i N V^n}{N_i V^n N k} \tag{6.6}$$

$$= \frac{k_i}{k} \tag{6.7}$$

The test data point $\mathbf{x}^n$ is thus assigned to the class that yields the largest posterior probability, corresponding to selecting the class to which the largest fraction of total nearest neighbours belongs to. Analogous to the classification method for MoG model classification as described in Chapter 5, each data point $\mathbf{x}^n$ is classified and the classification of the entire test sequence is then based on consensus so that the correct class is the one with a majority representation amongst the identified frames.

As the k-NN classifier is interpreted above as a special case of density modelling, a form of impostor detection could be implemented here. The conditional density can be estimated from Eq.(6.2), and as discussed in Section 5.6, a suitable threshold could be set so that a test data point that has a density estimate below this threshold for all classes can be classified as an impostor. As the impostor detection method of Section 5.6 works satisfactorily, the subject will not be delved into with respect to the k-NN classifier.

The downside of k-NN classification is that it is necessary for the k-NN classifier to store all the training data points that it receives and this places considerable demands on computer storage capacity which risks to be limited, especially in a hearing instrument. As the dimensionality of the feature space grows with the dimensionality of the input data, the k-NN classifier can rapidly reach an unwieldy number of dimensions. Additionally, the lack of a training phase means that all computations take place at the time of classification and this can be extremely time-consuming in the case where d is very large. This makes it desirable, once more, to determine a way in which feature sets can be reduced while still allowing efficient classification, if k-NN classification is to be feasible.

On the other hand, an advantage over the MoG model classifier is that the k-NN classifier does not model a density distribution for the data points, and is therefore more robust in the case of sparse data, especially when working with data sets of high dimensions. Isolated data points have a limited impact on k-NN classification. For these reasons, as well as the fact that the k-NN classifier is simple to implement, it is included as one of the classifiers that will be used as part of the SID system.

6.2 Gender Classification

Prior to testing with the more high-dimensional feature sets, an initial attempt to test whether gender separation is indeed possible using the k-NN classifier is implemented, based on the observations made in Section 3.4.4. The dimensionality of the feature sets produced by the fundamental frequency estimators is much lower than that of the remainder of the feature sets, as instead of having $[N \times d]$ matrices, where N is the number of frames in a sequence and d is the dimensionality of the feature set, each speaker is represented by a $[1 \times 7]$ vector of training data and a test vector of dimension $[1 \times 2]$ (one estimate per sentence). Due to this low dimensionality, the F_0 feature sets should not place excessive computational demands on the k-NN classifier.

The k-NN classifier is implemented with $k = 2$, the number of nearest neighbours being kept at a minimum with the low dimensionality of the data set kept in mind. A limited number of Euclidean distance calculations that have to be executed also lead to less computation time being needed in order to classify each test point. The feature set used is the Real Cepstrum F_0 estimations, though it was observed that the results for similar gender classification trials with the autocorrelation with center clipping method and the YIN estimator were identical to those presented here. Using the Real Cepstrum feature set results in a complete separation of male and female speakers in a computation time of 0.70s. The gender classification can be seen in Figure 6.2.

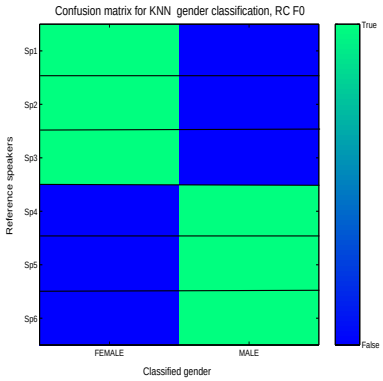


Figure 6.2: k-NN Gender classification using real cepstral F_0 estimates

This rapid discernment between genders might prove useful for various applications, as once it has been implemented, it halves the amount of speakers that have to be classified. How much influence this has on the SID task will be analyzed along with all other results in Chapter 9. Although the classification of different feature sets using the k-NN classifier will also be recorded in Chapter 9, a few initial trials are conducted here in order to determine a value for k and to observe basic differences between the Mixture of Gaussians classification and the k-NN classification.

6.3 Preliminary Trials

The same feature set as was used for all the preliminary testing of the MoG is used here, i.e. the 12MFCC + 12ΔMFCC feature set. The k-NN classifier placed too many demands on memory storage to permit training with all the data points available, as was done with the MoG classifier. Instead, 10s of training material is used from each speaker. Once again, 8s of test data from each speaker is used. The limitation of the training sequence length also ensures that an equal amount of reference data points is present for each speaker and prevents bias towards speakers who have larger amounts of training material available. Due to the creation of a separate MoG model for each speaker in Chapter 5, this equalization was not necessary.

The initial test is conducted by increasing the number of nearest neighbours that are included in the classification and observing this effect on the results. In all cases, from $k = 2$ to $k = 32$, the identification rate was the same: Speaker 4 could not be recognized and the total percentage of frames that are classified correctly is approximately 41%. A substantial increase in calculation time is only seen when the number of nearest neighbours is increased to 32, and was otherwise stable around 45s. The number of nearest neighbours is thus chosen to be $k = 2$.

The results of k-NN classification using the reference feature set and for $k = 2$, with 10s of training data and 8s of test data from each reference speaker, are shown in Figures 6.3 and 6.4.

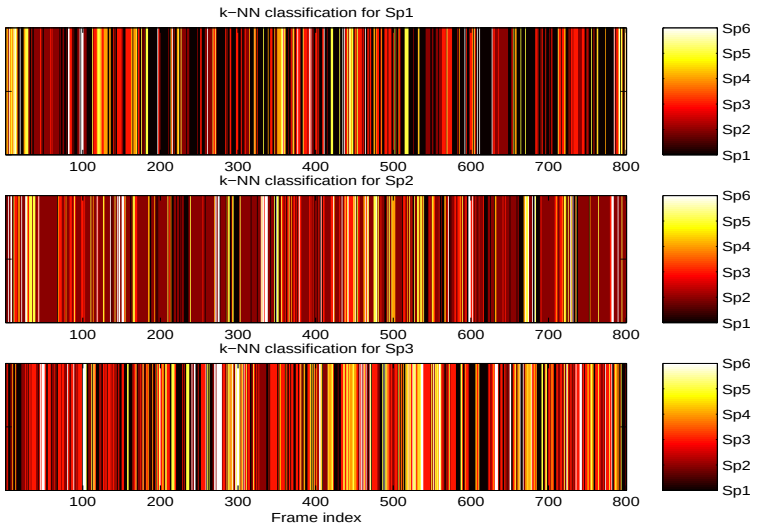


Figure 6.3: The k-NN classification of 800 test frames from Speakers 1-3, 12ΔMFCC feature set

Figures 6.3 and 6.4 show that although all but Speaker 4 were identified, the frames seem to be far less homogenous in their classification than was the case with MoG classification (see Figure 5.9 and Figure 5.10), so that each sequence here is a more varied combination of the six different classes. The actual percentage of correctly classified frames is almost as high as for the MoG classification, which leads to the conclusion that

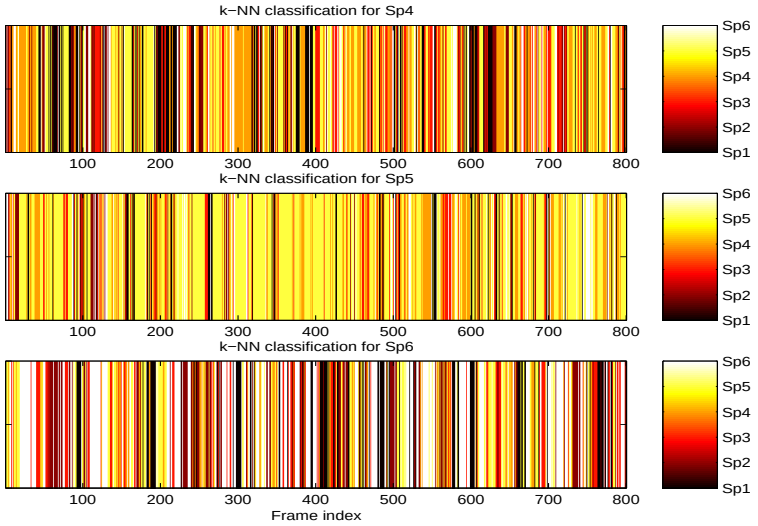


Figure 6.4: The k-NN classification of 800 test frames from Speakers 4-6, 12ΔMFCC feature set

the assignment of correct frames must be more evenly distributed over the six speakers, and that there is less bias towards one or two specific speakers here than was observed with the MoG models. This can be verified by comparing the confusion matrix of the k-NN classifier with the one given for the MoG classification in Section 5.5.

$$\mathbf{C_{k-NN}} = \begin{pmatrix} \mathbf{44.38} & 18.50 & 19.38 & 7.00 & 7.00 & 3.75 \\ 12.00 & \mathbf{56.63} & 12.25 & 5.25 & 7.50 & 6.38 \\ 23.88 & 17.63 & \mathbf{31.50} & 8.38 & 10.63 & 8.00 \\ 16.50 & 8.25 & 14.38 & 22.00 & \mathbf{27.38} & 11.50 \\ 7.00 & 4.13 & 8.75 & 18.00 & \mathbf{52.63} & 9.50 \\ 14.13 & 14.00 & 11.5 & 8.50 & 16.25 & \mathbf{35.63} \end{pmatrix}$$

The confusion matrix $\mathbf{C_{k-NN}}$ reveals that there is indeed a far more even assignment of correctly identified frames to each speaker than can be observed in $\mathbf{C_{MoG}}$. In contrast to the latter, $\mathbf{C_{k-NN}}$ reveals no overwhelming bias for Speakers 1 and 2, and except for the case of Speaker 4, the value in the diagonal is a good deal higher than those on either side of it meaning that there is little ambiguity as to which speaker is the correct one. It is interesting to note that the total amount of correctly classified frames is a mere 41%. Although this does not yield 100% correct identification, it does indicate that far from the majority of speech segments are needed in order to be able to identify a speaker. Including all frames thus corresponds to using noisy data as a lot of the information contained in the input data is obviously irrelevant for the SID task. The problem of *which* frames to keep and which can be excluded is one that remains to be solved, though, and will be treated in Chapter 9.

Chapter 7

Artificial Neural Network

7.1 Introduction

A powerful, highly flexible classifier, the Neural Network (NN) is suitable for non-linear classification tasks of varying degrees of complexity. It consists of a set of units whose interconnections represent a large number of degrees of freedom that lead to an adjustable total transfer function. It is this malleable quality of the NN structure that renders it capable of solving a large variety of classification problems. For highly non-linear classification problems in feature space of multiple dimensions the NNs flexibility gives it an advantage over the MoG classifier as it does not place constraints on the distribution of input data and it is less sensitive to the curse of dimensionality.

The units within the NN are referred to as *neurons*, after the nerve cells that constitute the biological neural network in the brain, which the NN classifier attempts to model. The brain has numerous advantages over digital computers: it can learn by experience and apply acquired knowledge to deal with associated problems. Functionality can be retained even when parts of the brain are destroyed and it is a far more powerful computing tool than the digital computer, capable of efficiently handling large and noisy input data sets. These reasons lie behind the motivation to create artificial neural networks that imitate the processes of the biological neural network. The functionality of the latter is briefly outlined in Appendix C.1.

In the artificial neural network, the synapses [1] of the biological neural network are replaced by *weights* that can be adjusted. The output of each neuron is weighted by multiplication with these values, thus causing the weights to influence the input to the following neuron. One neuron receives its input from several other neurons, but has only one output. The artificial neuron is described in Appendix C.2, which includes a diagram.

The combination of weights within a network is in effect the major deciding factor of the network's ability to model and recognize a data set. Weights have to be specifically adjusted in order to obtain the optimal mapping solution for a particular set of data. The determination of these weight values is what constitutes the training phase of the NN.

[1] The synapsis is the area between two neurons in the brain that transmits electrochemical impulses from one neuron to the other, explained in Appendix C.1

The versatility of the NN comes from its ability to *learn* during the above mentioned training phase. Learning from specific data leads to the network being an "expert" at solving a particular problem and this is the source of its increasing popularity. The use of neural networks for classification purposes started in the 1950's, but promising results gained especially in the last two decades have led to ongoing research into what type of network is optimal for the solution of a variety of problem definitions, including speaker identification.

7.2 The Multi-Layer Perceptron

The type of network chosen for this text-independent speaker identification task is a non-linear network capable of both forward and back propagation of data: the *Multi-Layered Perceptron*(MLP) [15]. Input data is fed to the MLP and at the output, the value of a predefined error function is calculated based on the difference between the network output and the correct answer that is provided for the input data. This error is then fanned backwards (back propagation) through the network so that each connection weight can be adjusted in order to reduce the error. This process is repeated until the network reaches convergence, meaning that the error rate becomes acceptable or that the network cannot yield better performance for the given data. This process is described in Section 7.3 and 7.4.

The MLP that is implemented consists of three layers of neurons. The layers are the *input*, *hidden* and *output* layers. The three layers are connected to one another by two sets of weights, W_{in} representing the weights connecting the input to the hidden layer and W_{out} defining the weights from the hidden to the output layer. This organization is shown in Figure 7.1.

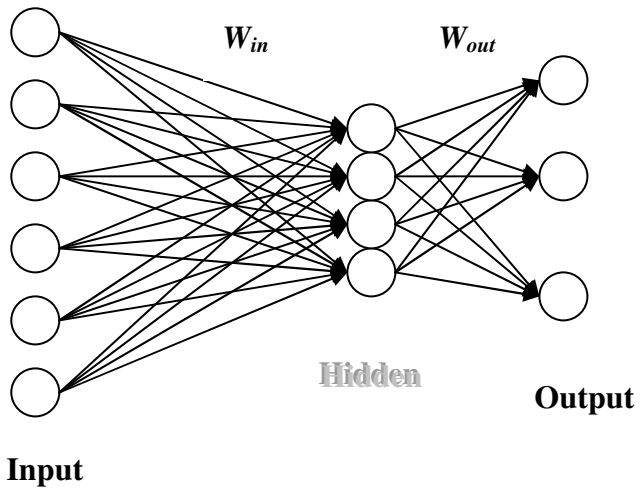


Figure 7.1: The input, hidden and output layers of a neural network

The number of units shown in each layer does not correspond to what is actually used,

but is reduced for schematic purposes in Figure 7.1.

The training data used as input to the MLP during the training stage are the frames containing feature vectors extracted from the training sentences of the reference speakers, as well as a corresponding class membership label for each frames feature vector. The training feature vectors are denoted as $\mathbf{x}_n$ and the associated class label as $\mathbf{t}_n$ (target values). The training data is normalized and then fed to the input layer of the MLP from where the data is fanned to the hidden layer, where some data processing is applied before sending the data to the output layer for final classification and where the network error is evaluated. The flow of data from the input to the hidden and from the hidden to the output layer is the forward propagation of data through the network. As will be seen in the following sections, forward propagation of data is implemented for both training and test data.

7.3 Design Details

Although training data is used in the notations of this section, the processes described here are relevant for both training and test data, except where noted otherwise. The input layer works with preprocessed data, $\bar{\mathbf{x}}_n$. The input feature vectors are preprocessed so as to scale all values to a uniform scale, preventing high variance within the set. Normalization is implemented by subtracting the mean $\tilde{x}_n$ of the feature vector from each element within the set and then dividing the result with the standard deviation $\tilde{\sigma}_n$:

$$\bar{\mathbf{x}}_n = \frac{\mathbf{x}_n - \tilde{x}_n}{\tilde{\sigma}_n} \tag{7.1}$$

The output of each input unit is multiplied with its corresponding weight before it is used as an input to a hidden unit. In the hidden units, the sum of the contributions from the input layer is transformed by an activation function. This function can be nonlinear when required for the solution of a nonlinear classification problem. The described process is shown in Eq.(7.2), where w_{hk} denotes the value of the weight for the connection from input unit k to hidden unit h, $\bar{\mathbf{x}}_n$ is the normalized feature vector for the n^{th} frame in the training sequence, and g is the activation function. This process is also described in Appendix C.2.

$$z_n(h) = g\left(\sum_{k=0}^{D} w_{hk}\bar{\mathbf{x}}_n(k)\right) \tag{7.2}$$

The activation function that is implemented is nonlinear in this case to allow for a smooth mapping of nonlinear feature data. The activation function must be differentiable to allow for back propagation of the error function, which will duly be explained. The *tanh* function is chosen as it meets these requirements and returns values within a restricted range of [-1,1]. It is defined in Eq.(7.3) and shown graphically in Figure 7.2.

$$g(a_h) \equiv \tanh(a_h) \equiv \frac{e^{a_h} - e^{-a_h}}{e^{a_h} + e^{-a_h}} \tag{7.3}$$

where a_h is the summed input, or activation, of hidden input h, as calculated in Appendix C.2.

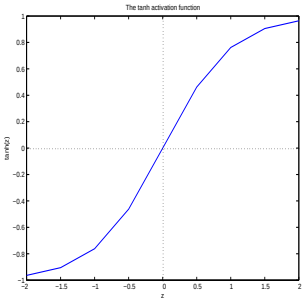


Figure 7.2: The tanh activation function

The outputs $z_n(h)$ from the hidden units are multiplied by the weights that connect the hidden and the output layers and these results are then used as input to the output layer. The output $o_n(j)$ from the output unit j is a linear transformation of the activation formed by the sum of the output from each hidden unit h multiplied with the weight w_{jh}:

$$o_n(j) = \sum_{h=0}^{N_h} w_{jh} z_n(h) \tag{7.4}$$

The number of hidden units starts at zero, as does the number of input units. This is because there is a bias parameter associated with each of these two layers, represented by a zero'th unit that has a fixed output of $z_k = z_h = 1$ for $k = h = 0$.

The NN must be able to classify input data as one of multiple classes. In order to obtain the network outputs as probabilities, the *softmax* function is applied to the values that are determined in Eq.(7.4). The softmax function is the normalized exponential of the output that returns all values within the range [0,1]. Large output values are assigned a value close to one, while the lower outputs are mapped closer to zero. The resultant set of values sum to unity and thus each output from the softmax transformation can be interpreted as a probability, more specifically the posterior probability $P(C_j|\mathbf{x}_n)$ of class j, as it is the probability that the class is the j^{th} speaker when the feature vector $\mathbf{x}_n$ is observed. Eq.(7.5) shows the softmax function.

$$y_j = \frac{\exp(o_j)}{\sum_{j'} \exp(o_{j'})} \tag{7.5}$$

The class with the largest posterior probability is then selected as being the correct class for the given training or test feature vector. The results of the classification are compared with the class membership labels, the target values $\mathbf{t}_n$, that were provided with the input feature vector. The difference between the network output and these target values is the network error that defines the value of a cost function. The network training consists of minimizing this cost function by adjusting the network's weight values. The set of optimal weight values should thus correspond to the cost function minimum. The cost function E_x that is implemented is the *cross-entropy* error [15], [58]. As the classes that must be recognized are independent of one another, the probability of observing the target values $\mathbf{t}_n$ given the input training pattern $\mathbf{x}_n$ is the product of all the classes' posterior probabilities given this pattern. From Eq.(7.5), these results are denoted as $y_{n,j}$ for the

n^{th} pattern. Each pattern has its associated target vector that is used in the evaluation of the probability of these patterns so that:

$$p(\mathbf{t}_n|\mathbf{x}_n) = \prod_{j=1}^{S}(y_{n,j})^{t_{n,j}} \tag{7.6}$$

The negative log-likelihood is obtained to define the cross-entropy cost function that has the form:

$$E^x = -\sum_{n=1}^{N}\sum_{j=1}^{S} t_{n,j}\log(y_{n,j}) \tag{7.7}$$

E^x is the total error function over all n training patterns. For the n^{th} training pattern, the cost function is denoted as E_n^x. In order to determine a minimum for the cost function by adjusting weight values, the former is differentiated with respect to the hidden-to-output weights and the input-to-hidden weights. First, we define the activation of a unit by referring to Appendix C.2, where the summed input from the input neurons to a hidden unit h is denoted as the activation a_h:

$$a_h = \sum_{k=0}^{d} w_{hk}\bar{\mathbf{x}}_n(k) \tag{7.8}$$

The corresponding activation in an output unit, a_j, is derived analogously.

In order to obtain the cost function's derivatives w.r.t. all the network weights, the chain rule is used. The cost function derivatives for the hidden-to-output weights and for the input-to-hidden weights are shown in Eq.(7.9) and Eq.(7.10), respectively:

$$\frac{\partial E_n^x}{\partial w_{jh}} = \frac{\partial E_n^x}{\partial a_j}\frac{\partial a_j}{\partial w_{jh}} \tag{7.9}$$

$$\frac{\partial E_n^x}{\partial w_{hk}} = \frac{\partial E_n^x}{\partial a_h}\frac{\partial a_h}{\partial w_{hk}} \tag{7.10}$$

where a_j and a_h are the summed and weighted input (activation) to an output and a hidden unit, respectively. The second term of the right-hand side of Eq.(7.9) and Eq.(7.10) is the derivative of the activation w.r.t. the input weights and is therefore the raw output of the previous unit, denoted as z in Appendix C.2. The first term in Eq.(7.9) and Eq.(7.10) is called the *back propagation error* [15] and for the output and hidden layer is denoted as δ_j and δ_h, respectively. The cost function derivatives can now be written as:

$$\frac{\partial E_n^x}{\partial w_{jh}} = \delta_j \cdot z_k \tag{7.11}$$

$$\frac{\partial E_n^x}{\partial w_{hk}} = \delta_h \cdot z_h \tag{7.12}$$

It is by the backwards flow of data in the form of the cost function and its derivatives that it becomes possible to assign "responsibility" for the size of the cost function to the weights within the network.

The cost function and the two sets of cost function weight derivatives are used as input to the network training algorithm that proceeds to determine the optimal weight values.

The algorithm that is implemented is the BFGS algorithm, described in Appendix D and in [46] and [45].

The results of the BFGS algorithm are the updated weight values and some updated hyperparameters that are used to check whether network convergence has been reached. If this is the case, the network is considered trained and ready for use as a classifier. In the event that convergence has not been reached, the cost function and its derivatives are reevaluated and once again propagated back through the network to be assigned as input to the BFGS weight optimizing algorithm. Convergence is checked again. The process is repeated until the convergence conditions, described in Section 7.4, are satisfied.

Once the network has converged, the training data is forward propagated once more through the network with no modifications being made to any of the weight values. The final training error is then obtained, indicating how well the network has modelled the training data feature set. The test error is found by using the above described methods for the forward flow of data through the network using the test feature vector $\mathbf{x}^n$ and the corresponding target vector, $\mathbf{t}^n$, as input to be able to calculate the test error. In this case the cost function is not differentiated with respect to weight values as back propagation of the error is exclusively used to train the network. During the testing phase, the performance of the network is established as its ability to recognize patterns that it has not been trained on, a vital performance measure for the text-independent speaker identification task. The performance is obtained as the amount of times that the class with the highest posterior probability corresponds to the correct target value. The identification process is implemented for each test frame $\mathbf{x}^n$ and as before, the final classification of an entire sequence of test frames from one speaker is based on consensus over these classified frames.

7.4 Generalization

In order to ensure that test data can be classified, a trade-off is associated with the learning process of the NN classifier. The ability of the NN to model the training set too accurately can prevent it from performing well when unknown (test) data samples are used as input. Correctly classifying data that the network has not trained on is a reflection of the network's *generalization* ability. The trade-off is between this ability and the ability to accurately model the given training data set. There is a risk that the network overfits the training data, meaning that too much information, including noise, is modelled and the generalization capability of the network becomes greatly decreased as the mapping of the test samples then bear little or no resemblance to the mapping of the patterns that were used to train the network. Several parameters can be adjusted in order to ensure a good trade-off.

One of these parameters is the number of hidden units, N_h. If this number is large, the network can approximate very complex distributions of training feature data but may become too specialized to allow for generalization. In this case, there exists a lot of variance in the network mapping of the input data. Excessesively restricting the size of the hidden layer, on the other hand, does not allow for a flexible mapping of the training data

and may result in high bias. N_h cannot be determined mathematically, and is therefore obtained through the observation of network performance using different numbers of hidden units, though the amount and complexity of available data can give an indication as to how many units should be implemented.

Furthermore, a cost function exclusively based on the training error of the network is clearly not suitable if the network must be able to generalize. This problem is addressed by introducing an additional parameter that ensures generalization is not sacrificed for the purpose of a very precise fit of training data. This is the regularization parameter, α. It is incorporated into the cost function so that it must also be minimized if the network is to converge. The direct purpose of the regularization parameter is to limit the variance in updated weight values and thereby prevent the formation of decision boundaries between the multiple classes that are too rough to allow for optimal classification of test patterns.

The regularization thus takes the form of a penalty that is implemented so that it grows larger for larger weights, and as the network cannot converge as long as α is too large, it forces the weight values to fall within a restricted range in order to achieve network convergence. There is one penalty term associated with input-to-hidden weights (α_{in}) and another for hidden-to-output weights (α_{out}).

The regularization term is multiplied with a decay constant γ that determines how much influence the former has on the cost function. For the actual implementation, $\gamma = 0.5$. The cross-entropy cost function of Eq.(7.7) with regularization becomes:

$$\hat{E}^x = E^x + \gamma \cdot \alpha_{in} \sum_{h=0}^{N_h} \sum_{k=0}^{d} (w_{hk})^2 + \gamma \cdot \alpha_{out} \sum_{j=1}^{S} \sum_{h=0}^{N_h} (w_{jh})^2 \tag{7.13}$$

The method used to estimate values for α_i and α_o is Mackay's evidence scheme [56].

An additional parameter, the outlier probability β [54] is implemented in the MLP but as shown in the following section, does not play a significant role in this speaker identification task.

The network training is completed when α_{in}, α_{out} and β fall below a preset low threshold. To provide an alternative, a maximum number of iterations is set so that if convergence is not obtainable, the training does eventually cease when this limit is reached. These are the convergence conditions that are associated with the iterative training process described in Section 7.3.

The neural network that is implemented is provided by [57], including the regularization functionality, the outlier probability evaluation and the BFGS weight optimization algorithm, leaving the only variable parameters being the number of hidden units and the length of training and test data to be used.

7.5 Preliminary Trials

Repeating the process for the MoG and k-NN classifiers, the NN classifier is tested in a preliminary round of trials in order to observe some initial results and determine some variable parameters, while the bulk of the testing with the NN is presented in Chapter 9.

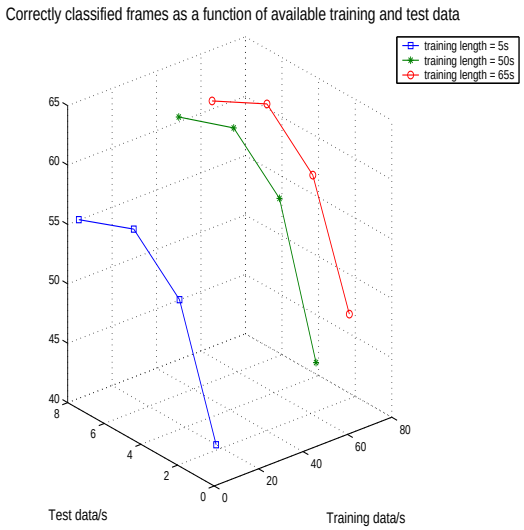


Figure 7.3: NN performance as a function of varying training and test sequence length

Once again, the 12MFCC + 12ΔMFCC feature set is used as a reference set. The number of input units corresponds to the dimensionality of the feature set, so that the entire feature vector can be contained by the input layer. For the reference feature set, this yields an input layer consisting of $d = 24$ units. As discussed above, the number of hidden units cannot be calculated and is thus initially set to $N_h = 15$. This number, being below that of the input units, should be able to model the main characteristics of the data without conforming too precisely to the input pattern. The number of output units depends on the number of different classes that are used as target labels, in this case corresponding to the number of speakers that the network must be able to differentiate from one another, and is so set to $S = 6$.

The weight values are initialized with random values chosen from a normal distribution with mean 0 and unit standard deviation.

There exists no absolute rule for how much training and test data must be available to the MLP for it to perform satisfactorily. Therefore, different lengths of training and test data sequences are used in order to establish the NN performance's dependency on the amount of both data sets. The trials are implemented by keeping the length of training data constant and varying the length of test sequences. When the different lengths of test data have been implemented, the length of the training data sequence is altered and once again a series of tests with test data of different lengths is implemented for a constant training set length. This is done for 3 different training set lengths and 4 different test set lengths.

As the amount of data for each speaker is different from one another, the upper bound for the training data is set to a common limit of $t_{train} = 65$s, so that a maximum of 65s of randomly selected training frames is used per speaker. Both test sentences are used for each speaker, allowing $t_{test} = 8$s as the maximum amount of test data available per speaker. The results are shown as 3-dimensional learning curves in Figure 7.3.

The curves in Figure 7.3 show that for increased training data length (along the x-axis), the performance for the classifier invariably also increases. The availability of

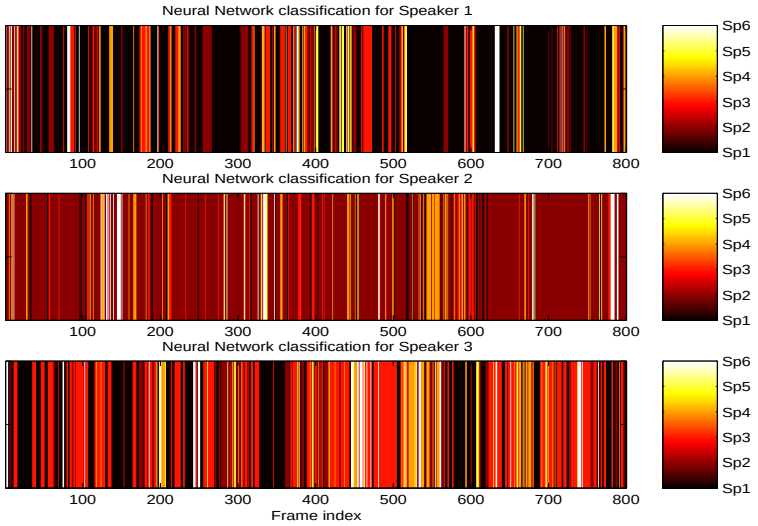


Figure 7.4: The NN classification of 800 test frames from Speakers 1-3, 12MFCC + 12ΔMFCC feature set

more training data would yield even higher correct classification rates but this cannot be confirmed empirically in this thesis due to the limited amount of speech in the ELSDSR database. The increased length of test data sequences also leads to improved performance until $t_{test} = 5$s. Hereafter, when all 8s of test material is included in the analysis, the performance drops in all cases. As each test sentence is different this does not show any conclusive evidence. It does suggest that the test data set for each speaker contains varying speaker-dependent information and so when the performance deviates from what is expected this does not necessarily indicate a fault that can be attributed to the classifier. Despite the drop in classification rate when additional test material is added, all of it is included in the first few tests of the network's performance, as it is generally better to use as much test data as is available and because it cannot be assumed that test data free of ambiguity can be obtained in real life circumstances.

It is encouraging, however, that with 5s of test material the performance of the NN for speaker identification of the 6 reference speakers is highly satisfactory. It is observed that for the trials using from $t_{train} = 50$s and upwards, the identification of all 6 speakers is 100% successful for both the 5s and the 8s test material sequences. This means that all 6 speakers are identified correctly by using consensus over all the test frame classifications for each speaker.

In Figures 7.4 and 7.5, the results of neural network classification for 8s of test speech from each speaker, using the reference feature set and 65s of training data per speaker, are shown.

When Figures 7.4 and 7.5 are compared with the corresponding Figures 6.3 and 6.4 for k-NN classification, it is instantly clear that more frames are identified correctly when using the neural network. This can be confirmed by observing the confusion matrix for the NN classification. All values in the confusion matrix are in %.

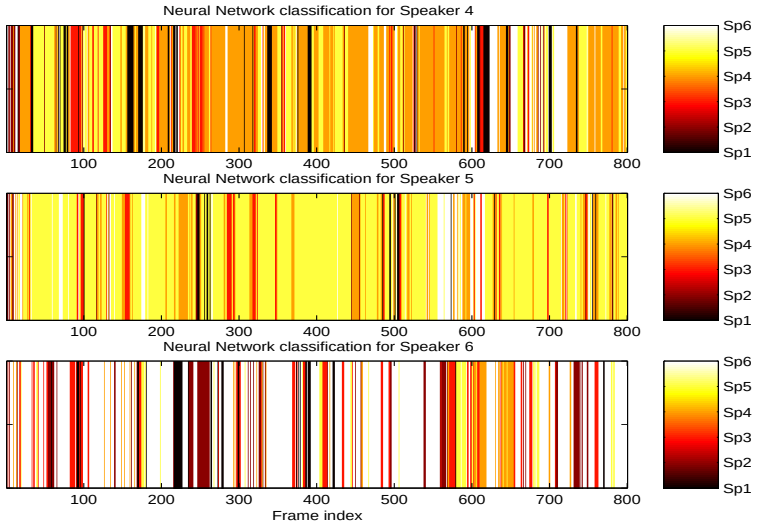


Figure 7.5: The NN classification of 800 test frames from Speakers 4-6, 12MFCC + 12ΔMFCC feature set

$$\mathbf{C_{NN}} = \begin{pmatrix} \mathbf{62.25} & 18.88 & 12.63 & 4.88 & 3.50 & 2.88 \\ 5.00 & \mathbf{75.88} & 5.63 & 8.38 & 1.00 & 4.13 \\ 32.38 & 13.75 & \mathbf{36.75} & 8.38 & 3.13 & 5.63 \\ 9.13 & 3.00 & 7.88 & \mathbf{44.25} & 23.00 & 12.75 \\ 2.50 & 1.50 & 5.75 & 14.38 & \mathbf{65.38} & 10.50 \\ 5.38 & 9.75 & 11.00 & 6.25 & 6.00 & \mathbf{61.63} \end{pmatrix}$$

Of significance when observing the confusion matrix for the neural network in comparison with those obtained for the same feature set with the MoG and k-NN classifiers is that all maximum values are situated in the diagonal, meaning that all six speakers are identified correctly. As was assumed, a larger amount of frames per speaker is assigned correctly here than in the case with k-NN. Additionally, the distribution of correctly classified frames is more evenly distributed between all 6 speakers here than in the case with MoG classification. There still exists a bias towards Speaker 1 in the case of Speaker 3, though not to the extent that misclassification of the latter speaker occurs. The total amount of correctly identified frames when using the neural network is 58%, which is 17% more than the k-NN classifier yielded and 11% more than was obtained with the MoG classifier, because the latter had such a high correct classification rate for a few speakers and a very low one for others. When the 12MFCC and their temporal derivatives are extracted as features, the optimal classifier to use would thus be the neural network.

In order to establish whether using 15 hidden units is suitable for use with the reference feature set, the network is tested with other values for N_h. Having more than 17 units in the hidden layer caused memory storage problems and so this was set as the maximum value. The NN performance, here presented as the percentage of correctly classified test frames, is shown for four different values for N_h in Table 7.1. All tests were implemented with $t_{train} = 65$s and $t_{test} = 8$s.

$\mathbf{N_h}$	10	14	15	17
Correct frame ID rate	57%	58%	63%	63%

Table 7.1: NN performance for different numbers of hidden units

The network's ability to correctly identify speakers is decreased (Speaker 3 is not identified) when the number of hidden units is limited to 10, though no significant improvement is observed when $N_h = 17$. The identification of all six speakers is still possible for $N_h = 14$, though a slight drop in correctly classified frames is observed. It can thus be assumed that the original value of 15 is satisfactory both with respect to modelling the training feature data and retaining the ability to generalize when test data is applied.

Before leaving the topic of the structure of the neural network, a note about the outlier probability, β, is made. It is mentioned in Section 7.4 that it has little influence on the implementation for the reference set of speakers. This is because it is known, before hand, which speaker uttered each sentence and thus there is very little chance that the training data frames are matched with the wrong label, which is what is detected using the outlier probability, as described in [54]. One of the convergence criteria of the NN is that $\beta < 1 \cdot 10^{-4}$. During the preliminary trials, the value of β never exceeded this threshold, as can be seen in Figure 7.6 for the analysis using the reference feature set.

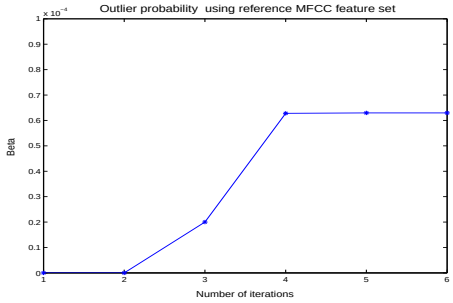


Figure 7.6: β for the NN classification of the 12ΔMFCC reference feature set

Chapter 8

The Database

The database that is used for all the experiments conducted in connection with this thesis is the English Language Speech Database for Speech Recognition (ELSDSR). It was created by Ling Feng during the course of and for use in her Master's Thesis [34]. The recording of training and test speech took place during the early summer of 2004 at the Department of Informatics and Mathematical Modelling (IMM) at the Technical University of Denmark (DTU). 22 speakers participated in the recording sessions. All recorded speech is in the English language. The training material consists of a set of 7 sentences that have been chosen to ensure that as many different parts of speech as possible are present for each speaker. The test sentences are randomly drawn from a text and here only 2 sentences are recorded for each speaker. All recordings were executed in the same room with the same equipment, so there is no mismatch between the training and test sequences. For more information pertaining to the recording setup and the backgrounds (such as age and nationality) of the speakers that partook in the recordings, refer to [34].

Each speaker is labelled as a male or female denoted as M and F, respectively. This letter is followed by three initials that identify the speaker. There are 10 women and 12 men, aged from 24 to 63. The length of total training speech uttered varies from speaker to speaker despite being produced from identical sentences, as each individual speaks with a unique speed and pause duration. As the test sentences are all different, the lengths of these vary to a greater extent. The test material differs not only between speakers but also from the training sentences, so that the identification problem becomes text independent. In Table 8.1, the length of training and test speech available from each speaker in the database is listed. The speakers are numbered as they are used in this thesis: The first six speakers are a combination of three women and three men used for the purpose of determining an optimal system for a small set of reference speakers. The remaining speakers aren't used in a particular order, so they appear as they do in the database, i.e. alphabetically. The 6 reference speakers are highlighted in bold print.

Speaker no.	Gender	Initials	Train/s	Test/s
1	**female**	**FAML**	**99.1**	**18.7**
2	**female**	**FDHH**	**77.3**	**12.7**
3	**female**	**FEAB**	**92.8**	**24.0**
4	**male**	**MASM**	**81.2**	**20.9**
5	**male**	**MCBR**	**68.4**	**13.1**
6	**male**	**MFKC**	**91.6**	**15.8**
7	female	FHRO	86.6	21.2
8	female	FJAZ	79.2	18.0
9	female	FMEL	76.3	18.2
10	female	FMEV	99.1	24.1
11	female	FSLJ	80.2	18.4
12	female	FTEJ	102.9	15.8
13	female	FUAN	89.5	25.1
14	male	MKBP	69.9	15.8
15	male	MLKH	76.8	14.7
16	male	MMLP	79.6	13.3
17	male	MMNA	73.1	10.9
18	male	MNHP	82.9	20.3
19	male	MOEW	88.0	23.4
20	male	MPRA	86.8	9.3
21	male	MREM	79.1	21.8
22	male	MTLS	66.2	14.05

Table 8.1: The length of training and test material for each speaker

Chapter 9

Experimental Results

In this chapter, the results of the extraction and implementation of the feature sets described in Chapter 3 used in conjunction with the classifiers of Chapter 5, 6 and 7 are presented. All of the testing is implemented using the small reference set of 6 speakers that is taken from the ELSDSR database as described in Chapter 8, unless the use of additional speakers is explicitly noted. When the computation time is provided, the times are registered for a Pentium 4.40GHz processor.

9.1 Preprocessing

The speech signals of the ELSDSR database that were used in all of the applications that are described in the following sections are preprocessed with a high-pass first order filter that has the transfer function

$$H(z) = 1 - 0.97z^{-1} \tag{9.1}$$

to attenuate the lower frequencies in the speech signal and thus emphasize the higher frequencies and thus create a balance between the low- and high- frequency representation in the speech spectrum. The pre-emphasized signals are then divided into frame blocks, each of them 30ms in length and with an overlap of 10ms for all feature sets, unless otherwise specified. The Hamming window was applied to each frame to ensure smooth transitions at frame boundaries.

9.2 Feature set extraction

9.2.1 F_0 Estimates

The three fundamental frequency estimators described in Sections 3.4.1, 3.4.2, and 3.4.3 and the implemented frame length for speech segments that are analyzed by each method are listed in Table 9.1.

There are three feature sets extracted using F_0 estimation:

- YIN F_0: contains one F_0 estimate for each sentence, calculated by the YIN estimator.
- RC F_0: contains an average F_0 estimate for each sentence, calculated by the Real Cepstrum method.

F_0 **Estimator**	**Frame length**
Autocorrelation CC	30ms
YIN	33ms
Real Cepstrum	64ms

Table 9.1: The frame lengths for each F_0 estimator

- F_0 Trajectories: contains the F_0 estimate of each frame in each sentence, calculated by the real cepstrum method.

The autocorrelation with center clipping method is used to return a voiced/unvoiced label for each frame in each sentence. It is not implemented as a feature set as the YIN estimator is assumed to be more precise in its estimations that its time-domain counterpart. Including the real cepstrum feature set ensures that both a time-domain and a frequency-domain method of extracting F_0 are represented.

9.2.2 LPCC, LPC Residual, Warped LPCC, PLPCC, MFCC

The LPCC, warped LPCC, PLPCC and MFCC feature sets are derived by the processes described in Sections 3.5, 3.6, 3.7 and 3.8, respectively. The LPC residual is derived from the LPC analysis as described in Section 3.5.2. The orders of the cepstral coefficient feature sets are chosen to be 12, except for the PLPCC set, where the initial order is 13. These values are selected on the grounds that they are commonly used with success in speech and speaker recognition applications and are therefore assumed to be suitable for the representation of the vocal tract characteristics of each speaker [1]. As a reduction of the dimensionality of the feature sets while preserving SID system performance is one of the aims of this thesis, the orders of the cepstral coefficient feature sets are changed to lower values and the effect of this on the speaker identification task is observed. The LPCC, warped LPCC and MFCC features are extracted for an analysis order of 8 and 10 as well as for the starting value of 12. The corresponding lower analysis orders for the PLPCC feature set are 9 and 11. The LPC residual is derived for each analysis order of the LPCC. The first temporal derivatives of the LPCC, warped LPCC, PLPCC and MFCC feature sets are calculated and these sets are denoted as f.ex. 12ΔMFCC that implicitly includes the original 12MFCC feature vectors, so that this feature set contains 24 feature data points for each frame. The labelling of the second order derivatives is analogous to this. The second order temporal derivatives of the LPCC, PLPCC and MFCC feature sets are implemented. When the cepstral coefficient feature sets are used, each frame of speech from the training and test sentences contains a feature vector of dimension d that depends on the order of the analysis. The LPC Residual returns one value for each frame, the prediction error of the analysis that is shown in Eq.(3.15). All of the cepstral coefficient feature sets are derived from framed segments of speech that are 30ms in length with a 10ms increment.

The F_0 estimates and the LPC residual are the source based features tested in this thesis while the system based features include the LPCC, warped LPCC, PLPCC and MFCC features sets.

9.3 Classifier settings

9.3.1 MoG Classifier

Six reference MoG models are trained with all of the training data available for each reference speaker. The sizes of these training data sets are listed in Table 8.1. As the optimal number of Gaussian components was established in Section 5.5 as being 2, this number was used in each case and then increased in order to observe the effect this had on the system performance for each feature set. In the majority of cases, $M = 2$ proved to be the optimal choice. An increase to $M = 4$, however, was necessary to improve classification performance in the case of the warped LPCC feature sets. The MoG model classifier was observed to be instable due to the unreliability of density estimation in high dimensions and so the results obtained from this method are not representative of an unambiguous classification process, but should rather be seen as one out of several possible outcomes. The preliminary trials with the MoG classifier are described in Section 5.6.

9.3.2 k-NN Classifier

The feature sets are all implemented with the k-NN classifier using $k = 2$ nearest neighbours and the Euclidean distance metric. The preliminary trials for the k-NN classifier are described in Section 6.3.

9.3.3 Neural Network

The neural network was tested with the number of hidden units set to $N_h = 15$. Although this is the number that has only been proven to be suitable for the 12ΔMFCC feature set, it is assumed that it is also appropriate for the other cepstral coefficient feature sets. The source based features, i.e. the F_0 estimates and the LPC residual, are not implemented with the neural network as they yielded poor performance for the other two classifiers and so due to the computational time that was required in order to train the neural network for each new feature set, no trial time was allocated for these feature sets based on the assumption that they do not contain enough speaker-specific information to enable successful speaker identification. Results for the NN are thus obtained for the feature sets that yield the most promising results when using the k-NN and MoG classifiers. These proved to be the system based features. The preliminary trials with the NN classifier are described in Section 7.5.

9.4 Impostor Detection

The impostor detection method of Section 5.6 was implemented for the 12ΔMFCC feature set so that closed-set classification could be implemented for the speakers that were accepted as being reference speakers. The test data of each speaker was split into a validation and a test set and the likelihood estimates of each of these sets were used to determine the speaker-specific thresholds τ_i. The values that were determined for τ_i are listed in Table 9.2. The logarithm of these values is also shown as this transforms the values of τ_i to a more useable scale.

Speaker	Threshold, τ_i	$\log(\tau_i)$
1	$3.3 \cdot 10^{10}$	24.22
2	$5.4 \cdot 10^{9}$	22.41
3	$7.76 \cdot 10^{11}$	27.38
4	$8.1 \cdot 10^{9}$	22.82
5	$2.1 \cdot 10^{9}$	21.47
6	$5.0 \cdot 10^{10}$	24.64

Table 9.2: The likelihood and log-likelihood values of the speaker specific impostor detection thresholds

Impostor detection is implemented by first calculating each test frames class-conditional density estimate on the original scale and comparing it to the correpsonding τ_i values. The τ_i values are based on a minimum error criteria. This method of impostor detection, as noted in Section 5.6, resulted in a 90% correct rate of impostor detection and a 100% correct reference speaker detection rate. The process of determining the speaker-specific thresholds and then calculating the minimum error from the sum of the rejection and acceptance errors is time-consuming and the only other feature set that this method was implemented for is the 12ΔLPCC set. The results for this set are poorer than for the 12ΔMFCC set as it proved difficult to determine suitable thresholds for Speakers 1 and 3. This lead to the acceptance of a large number of impostors so that the impostor detection rate fell to 60% while the reference speaker detection rate was maintained at 100%. The problem of determining good speaker-specific threshold values is assumed to be due to the limited amount of data available for each speaker. The impostor detection method is not implemented for other feature sets but as it is assumed that this method works to a certain degree for all feature sets, so these are implemented in closed-set system setups so as to enable the evaluation of the performance of each SID system for a small set of reference speakers.

9.5 SID System Performance Using All Frames

In Chapters 5, 6 and 7, the results for each classifier using the 12MFCC+12ΔMFCC were obtained. Here, additional feature sets are implemented and each classifier's performance is measured. As the classifiers cannot all handle an equal amount of data, the training and test data are set to the values listed in Table 9.3 for all the trials that yielded the results listed in this section.

Classifier	Training Data	Test Data
MoG	ALL	8s
k-NN	10s	8s
NN	50s	8s

Table 9.3: Training and test data lengths for each classifier

Each amount of data listed in Table 9.3 is per speaker, and "ALL" indicates that all

available training data is used. The length of the training data for each reference speaker is provided in Table 8.1. The training data in the case of the k-NN and NN classifiers are restricted because an equal representation for each speaker is required and because an excessive memory usage requirement that could not be provided for was noted for some of the feature sets if more training data is included. All of the test data is limited to 8s as the lower bound for the test material of the reference speakers is above this value and using the same length in all cases provides a fair basis for comparison and the possibility for a frame-by-frame analysis that is started in Figures 5.9, 5.10, 6.3, 6.4, 7.4 and 7.5, where the different classifiers performances are visualized as the classification of each test frame from each reference speaker.

All classification tasks are based on the principle of consensus so that not only the rate of correct identification of speakers from a whole test sequence are obtained, but the percentage of correctly classified frames in each case is also recorded. This measurement reveals details as to the SID system's ability to recognize a speaker from specific frames.

The results for each test conducted for the different feature sets and the three classifiers are shown in Tables 9.4, 9.5, 9.6 , 9.7 and 9.8. The abbreviation "wLPCC" stands for warped LPCC coefficients.

The results recorded as "ID" show the total number of speakers that were identified by using consensus over all test frames. As there are 6 speakers in the set, a 100% correct identification rate is noted as 6. A complete failure to identify any of the speakers is signified by 0, and all values inbetween indicate how many reference speakers out of 6 are correctly identified. "Frames" measures the correctly classified frame rate in percentage. This is calculated from the number of frames that are assigned to the correct speaker out of the 8s(800 frames) of test speech. For the F_0 estimates, "Frames" are actually entire sentences. Although the correct frame rate in itself is not sufficient to determine the performance of the SID system, it is interesting in that it shows which feature sets contain frames that are more easily classified as belonging to the correct speaker and are therefore more rich in speaker-specific information. This knowledge introduces a measure of reliability for each system setup combining a feature set and a classifier. The distribution of correctly classified frames is also a useful performance measure. This is what was used in the comparisons of the preliminary trials with the three classifiers, where the confusion matrices were analyzed. It was revealed that although the MoG classifier had a higher correct frame classification rate than the k-NN classifier, the distribution of these was so uneven that the rate of identification of speakers was the same for both classifiers. As there is no simple way to represent this distribution however, it is not included as a performance measure in Tables 9.4-9.8. A good distribution of correctly classified frames is, however, represented by the identification of speakers rate. When all 6 speakers are correctly identified, the confusion matrix contains a large majority of the classified test frames in its diagonal.

The source based features, the system performance of which is listed in Table 9.8, prove to be unsuitable for speaker identification. The F_0 estimates for the RC method are the best source features for speaker classification when using the k-NN classifier, as for this set 4 reference speakers are identified and a large percentage of the test sentences are classified correctly. Referring to Figure 3.7 no evidence as to why this is the case

Classifier	Measure	8 MFCC	8ΔMFCC	10 MFCC	10ΔMFCC	12 MFCC	12ΔMFCC	12ΔΔMFCC
MoG	ID	4	4	4	5	5	5	5
MoG	Frames	41%	42%	45%	47%	46%	48%	43%
k-NN	ID	6	6	5	6	5	6	5
k-NN	Frames	37%	39%	40%	42%	40%	43%	41%
NN	ID	6	6	6	6	6	6	6
NN	Frames	52%	54%	53%	56%	55%	60%	61%

Table 9.4: The performance of different classifiers for MFCC feature sets

Classifier	Measure	8 LPCC	8ΔLPCC	10 LPCC	10ΔLPCC	12 LPCC	12ΔLPCC	12ΔΔLPCC
MoG	ID	6	6	6	6	6	6	6
MoG	Frames	45%	50%	50%	54%	57%	62%	68%
k-NN	ID	6	6	5	5	6	6	6
k-NN	Frames	32%	33%	34%	35%	38%	38%	38%
NN	ID	5	5	5	5	5	6	5
NN	Frames	43%	46%	48%	49%	52	54%	59%

Table 9.5: The performance of different classifiers for LPCC feature sets

Classifier	Measure	8 wLPCC	8ΔwLPCC	10 wLPCC	10ΔwLPCC	12 wLPCC	12ΔwLPCC
MoG	ID	6	6	6	6	6	6
MoG	Frames	37%	41%	37%	43%	40%	46%
k-NN	ID	4	5	4	5	6	6
k-NN	Frames	28%	29%	31%	31%	34%	34%
NN	ID	6	5	6	6	6	6
NN	Frames	40%	46%	43%	46%	48%	43%

Table 9.6: The performance of different classifiers for warped LPCC feature sets

Classifier	Measure	9 PLPCC	9ΔPLPCC	11 PLPCC	11ΔPLPCC	13 PLPCC	13ΔPLPCC	13ΔΔPLPCC
MoG	ID	6	6	6	6	6	6	6
MoG	Frames	55%	58%	55%	59%	59%	63%	71%
k-NN	ID	6	5	6	6	6	6	6
k-NN	Frames	41%	40%	41%	43%	45%	45%	45%
NN	ID	6	6	5	6	6	6	5
NN	Frames	54%	56%	55%	56%	60%	61%	68%

Table 9.7: The performance of different classifiers for PLPCC feature sets

Classifier	Measure	8 LPC residual	10 LPC residual	12 LPC residual	YIN F_0	RC F_0	F_0 Trajectory
MoG	ID	1	1	1	1	1	1
MoG	Frames	18%	17%	17%	0%	0%	6%
k-NN	ID	2	2	0	2	4	2
k-NN	Frames	18%	18%	17%	42%	67%	34%

Table 9.8: The performance of different classifiers for source based feature sets

for the real cepstrum and not for the YIN estimates can be found, as the relative differences within the two sets is not large. As there are only 2 test sentences, though, a single correct classifiation can make a big difference in the total results. The extremely small amount of points in the source-based feature sets made it impossible for the MoG classifier to estimate a density function with any precision. The LPC residual leads to poor performance in all cases. The F_0 trajectories of the real cepstrum method lead to results for both classifiers that confirm that these features are not rich in speaker-specific information, as was already observed in Figure 3.10.

From Table 9.4 the NN is seen to be the only classifier that can successfully identify all 6 speakers based on all the MFCC feature sets. The low frame classification rate of the MoG classifier may be due to the overlap in feature space of MFCC coefficents that is observed in the PCA analysis of Figure 3.17. Combined with a restricted amount of data points, the MoG classifier has difficulty in estimating speaker specific density functions. Although the highest frame classification rate is obtained for the 12$\Delta\Delta$MFCC feature set implemented with the NN classifier, the NN is capable of identifying all 6 speakers for the 8MFCC feature set, as can the k-NN classifier. Using MFCC as a feature is thus best done in a SID system setup using the NN.

From Tables 9.5 and 9.6 it is observed that warping the LPCC coefficients leads to a decrease in correctly classified frames for all classifiers. The correct identification of speakers rate, however, does not deviate much between the two types of features. These results show that for this SID task, no improvement in performance is gained from the warping of the LPC autoregressive coefficients to the bark scale. All the LPCC feature sets result in optimal speaker identification rates of 100% for the MoG classifier while the NN classifier requires the information contained within the 12ΔLPCC feature set to be able to identify all 6 speakers. The much lower dimensional 8LPCC feature set is sufficient for good classification of speakers using the MoG classifier, while the k-NN classifier, as for the MFCC set, can identify all 6 speakers for a few of the LPCC feature sets.

Of all the feature sets, the PLPCC lead to the best performance of the SID system. For almost all the combinations shown in Table 9.7, the speaker identification rate is 100% and the correct frame classification rate is higher than that for the other feature sets. The preprocessing of the PLPCC coefficients that approximates the audiological frequency analysis in the ear and places weight on the perceptually significant parts of speech thus leads to an improvement in the speaker identification system performance. For almost all the PLPCC feature sets, 100% correct speaker identification is obtained for all 3 classifiers. The preprocessing does require more computational time and so if this is

of vital importance, the MFCC or LPCC feature sets should be used instead.

The reason that the NN classification of the 12ΔΔLPCC and 13ΔΔPLPCC feature sets is not 100% correct is that the amount of training data used for the tests involving the second derivatives was limited to 30s instead of 50s as the NN otherwise experienced memory storage difficulties. To summarize the results obtained in Tables 9.4-9.7, the best performance for each classifier is listed in Table 9.9. The performance is based on which feature set yields 100% correct speaker identification with the highest level of reliability, i.e. the largest number of correctly classified frames. If a situation arises where several feature sets resulted in the same performance, the feature set of lowest dimension is chosen. The feature set or sets that generally lead to reliable performance for each classifer are also listed.

Classifier	Optimal Feature Set	Good Feature Set(s)
MoG	13ΔΔPLPCC	LPCC, wLPCC, PLPCC
k-NN	13PLPCC	LPCC, PLPCC
NN	13ΔPLPCC	MFCC

Table 9.9: The optimal feature sets for different classifiers

Although Table 9.9 shows that the optimal performance for all classifiers is achieved with the 13PLPCC feature set and its temporal derivatives, the NN classifier is most reliable when used to classify speakers using any of the MFCC feature sets, despite the slightly lower correct frame rate when compared to the PLPCC. For all 4 cepstral coefficient feature sets, the inclusion of the temporal derivatives of each feature set usually leads to a better speaker identification rate and a higher correct frame classification percentage. More speaker-specific information is thus available when the temporal variations of the speech signal are analyzed. This can f.ex. be seen in Table 9.4, where using the 12ΔMFCC feature set instead of the 12MFCC set with the NN classifier leads to a 5% increase in correct frame classification rate. The temporal derivatives are thus relevant for the speaker identification task.

In order to limit the amount of feature sets used in further trials, four feature sets that result in 100% correct speaker identification rate and high correct frame classification rates are selected for additional testing: the 12ΔMFCC, the 12ΔLPCC, the 12ΔwLPCC and the 13ΔPLP feature sets. Apart from the MFCC features, all of these sets resulted in 100% correct identification rate for all classifiers. As the NN classifier is more stable than the MoG classifier and more efficient than the k-NN classifier, it is chosen to implement the various types of tests that are presented in Sections 9.6 and 9.7.

9.6 Gender Separation

The rapid separation of genders based on a single F_0 estimate for each sentence was shown to be possible using the k-NN classifier for all F_0 feature sets in Section 6.2. It is thus interesting to measure the eventual improvement in performance for a more complex classifier if the reference speakers are split into two groups of three speakers each, based

on gender classification. Using the 12ΔMFCC, 12ΔLPCC and 13ΔPLP feature sets, the NN classifier is implemented for the male and the female speakers separately. For the test speech that is classified as belonging to a male speaker, the NN classifer that is only trained with the male speakers' training data is used, while the estimated female test speech is classified by the NN trained on the female training data. The results are obtained for a system setup using t_{train} = 50s and t_{test}=8s. For ease of comparison, the results that were obtained for classification of all 6 speakers are noted alongside the gender specific classification results in Table 9.10. As the correct speaker identification rate is 100% in each case, only the correct frame classification rate is presented as a measure of performance.

Feature Set	All	Female	Male
12ΔMFCC	60%	67%	73%
12ΔLPCC	54%	65%	74%
13ΔPLP	61%	69%	78%

Table 9.10: NN results for gender separated data sets

As seen in Table 9.10, the results of NN classification when using gender separation are greatly improved in reliability, as up to a 20% increase in the amount of correctly classified frames is obtained. This considerably reduces the possibility of classification ambiguity between speakers. Although the most substantial increase in correct classification rate is achieved for the male speakers, it was noted that this required, in each case, roughly 3 times more training time than for the female speakers. As an example, for the 12ΔLPCC set, the entire training and classification process of female speakers took a little over 2 hours, while the corresponding process occupied 6 hours for the male speakers. Similar observations were made for the MFCC and PLPCC feature sets. Once the networks were trained however, both executed rapid classification of the gender separated speech in the test phase.

As two separate, gender-specific NN classifiers are trained, this means that if one of the male speakers' test data is classified by the network trained on female speakers' data, the identification will invariably be wrong. This situation did not arise, however, as the gender separation was accurate in 100% of cases.

9.7 Voiced/Unvoiced Analysis

The voiced/unvoiced analysis is introduced as a step in the direction of eventually streamlining the number of frames that are needed in order to achieve optimal SID performance. All results have shown that a relatively large number of test data frames are misclassified, meaning that frames that cannot be identified as belonging to the correct speaker are included. The information contained within these frames is thus not speaker-dependent and can therefore be viewed as noise in the SID system.

The analysis is commenced by classifying all the training and test frames as being voiced or unvoiced. This is done by using the autocorrelation with center clipping method of

Section 3.4.1, where a frame is labelled as being voiced if the autocorrelation function has a value above 30% of the maximum peak value found at $\tau = 0$. Any frames not meeting this requirement are classified as being unvoiced. The classification of frames from a test sentence as belonging to different speakers in comparison with the same sentence divided into voiced/unvoiced frames may reveal whether a correlation exists between the voicing of a frame and its content of speaker specific information. For visualization of this comparison, the 13ΔPLPCC feature set is used, as it yielded the highest correct frame classification results in the series of tests conducted in Section 9.5. In Figure 9.1, the classification of 800 frames (8s) of test material using the 13ΔPLPCC feature set for all classifiers is shown for Speaker 1, a woman (FAML). The top row of classified frames are the results of k-NN classification, the second row the MoG model classification, the third row the NN classification and the bottom row is the sequence of voiced/unvoiced decisions for the test sequence. An analogous analysis for a male speaker, Speaker 4 (MASM), is shown in Figure 9.2. The value 0 is used to denote the unvoiced label, 1 denotes both Speaker 1 and the voiced label and numbers 2-6 each correspond to a speaker in the reference set as listed in Chapter 8.

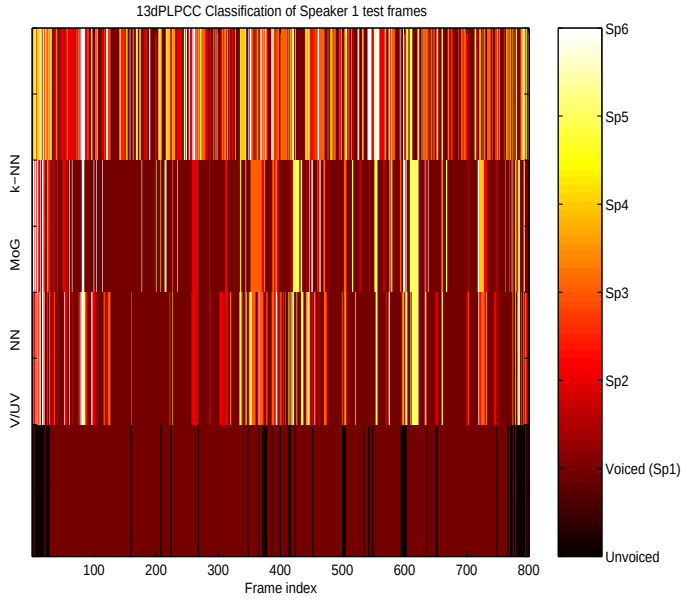


Figure 9.1: Classification results for Sp1, 13PLPCC + 13ΔPLPCC

Although it is difficult to draw conclusions from Figures 9.1 and 9.2, a few observations can be made that are relevant for both speakers. Firstly, there is no clear division in the classified frames from any of the classifiers according to the voiced/unvoiced decisions. However, it can be seen that an incorrectly assigned frame in the speaker identification results is often associated with an unvoiced frame. There are wrong classifications made for voiced frames, too, but the difference lies in the fact that it appears to be a rule that the frames that are unvoiced are incorrectly assigned to a speaker while misclassification occurs more randomly for the voiced frames. In short, a voiced frame is not sure to be classified correctly while an unvoiced frame has a high probability of being classified incorrectly.

These results are for one feature set only and dependent on the parameters of each

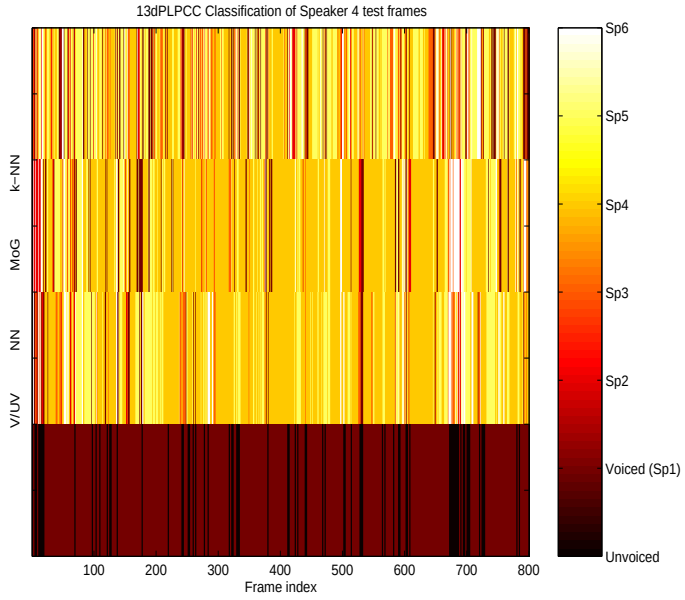


Figure 9.2: Classification results for Sp1, 13PLPCC + 13ΔPLPCC

classifier and so cannot be seen as conclusive. In order to shed more light on the classification compared with the voiced/unvoiced sequence, consensus between the results of the two best performing classifiers, the MoG and NN, is analyzed w.r.t. to the voicing of frames. The results are shown as correct classifications, so that only two options are permitted: "Correct" and "Incorrect". The voiced and unvoiced labels correspond to the colours for the correct and incorrect labels, respectively, though this is done to permit all the sequences to be shown at once and not because voiced frames are considered in any way as being "correct" and unvoiced ones as "incorrect". These results are shown in Figure 9.3 for Speaker 1 and in 9.4 for Speaker 4. The top row in both figures shows the correctly classified frames for the MoG classifier, the second row for the NN classifier, the third row for the consensus between these two classifiers and the fourth row shows the voiced/unvoiced classifications.

Although it remains problematic to observe conclusive trends, there seems to be evidence in Figures 9.3 and 9.4 that while classification tends to be difficult for unvoiced frames, the frames immediately after these are more frequently correctly identified. This may be connected to the theory that a considerable amount of speech information is contained in the acoustic *transients* of a speech signal [63]. The transients are areas of rapid change in the spectral envelope of a speech signal and the rich information that they carry may well be speaker-dependent. As it is confusing to try to decipher whether this is true from the sequences of 800 frames that have been shown, a few trials are implemented to test whether the theory holds.

Each of the four feature sets is divided into five subsets, as listed below.

1. Voiced(V): contains all the frames classified as being voiced

2. Unvoiced(UV): contains all the frames classified as unvoiced

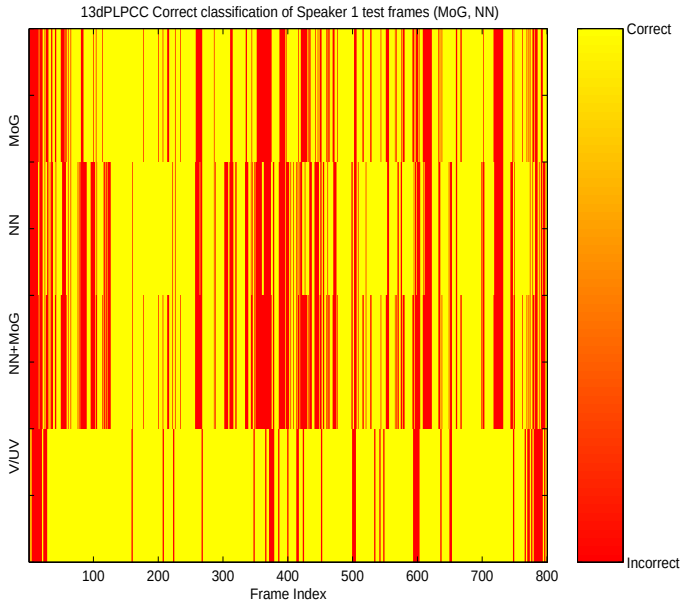


Figure 9.3: Correct Classification results for Sp1, 13PLPCC + 13ΔPLPCC, including consensus between MoG and NN classifiers

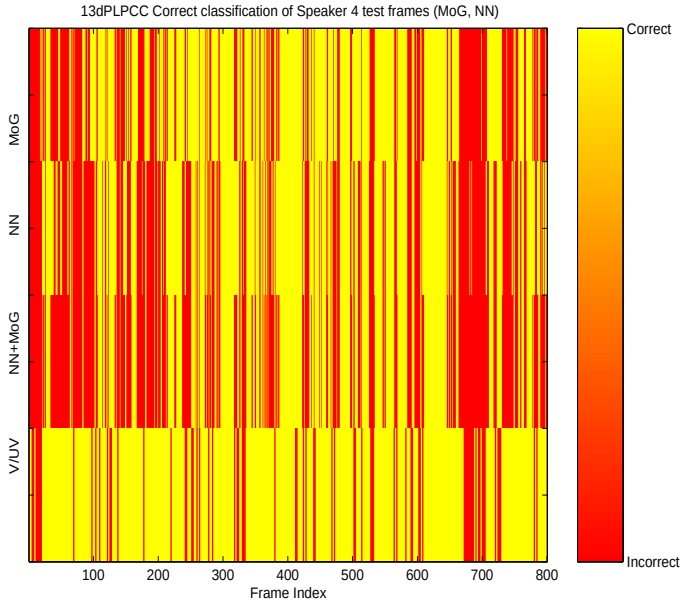


Figure 9.4: Correct Classification results for Sp4, 13PLPCC + 13ΔPLPCC, including consensus between MoG and NN classifiers

3. Unvoiced-Voiced(UVV): contains only the voiced fames that are preceded by an unvoiced frame

4. Voiced-Unvoiced(VUV): contains only the unvoiced frames that are preceded by a voiced frame

5. ALL: contains all frames not sorted according to voicing labels

The temporal changes in the speech signal may not always be represented by the transition between a voiced and unvoiced segment, but this analysis will still provide clues as to how heavily the identification depends voiced/unvoiced state of the frames and the order that these occur in. An initial experiment was conducted with the k-NN classifier which proved incapable of identifying all 6 speakers based on anything else but the mixed sequence of frames. All the available material, up to t_{train} =10s, is used in this analysis and so the limited number of frames in the UVV and VUV sets may cause a decline in identification rate. Despite this, it was observed that for all four feature sets, the percentage of correctly classified frames was highest for the VUV and UVV sets. This can be seen in Figure 9.5.

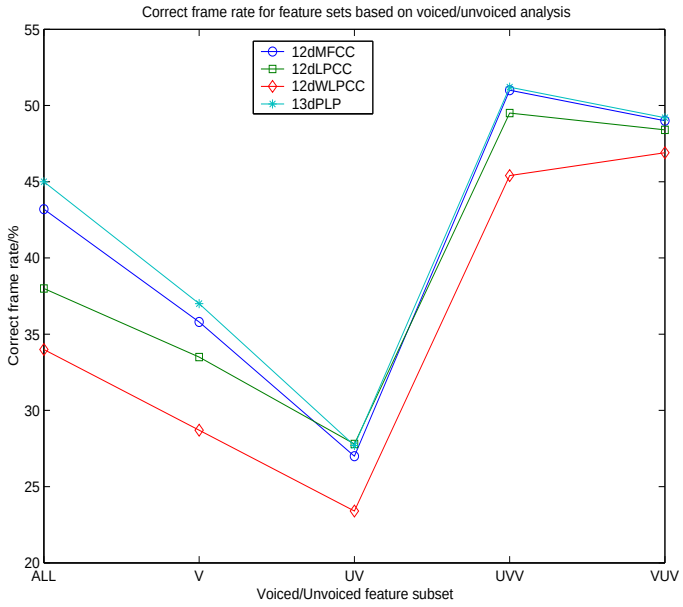


Figure 9.5: k-NN results for the voiced/unvoiced analysis

Figure 9.5 shows the same tendency for all feature sets: that the lowest amount of correctly classified frames is obtained for the unvoiced frames, while the highest rate is either for the unvoiced-voiced feature set or the voiced-unvoiced feature set. From the PCA analysis of voiced and unvoiced frames using the 12ΔMFCC feature set in Section 3.10, it is not surprising that these subsets do not provide good features for speaker identification. As none other than the complete set of frames yielded a successful identification of all 6 reference speakers, these results just provide pointers to the fact that the areas of transition between voiced and unvoiced frames certainly contain information that is vital for speaker identification and that classification based on the voiced or unvoiced frames alone performs more poorly than when there is a combination of the two (in the ALL

data set).

Classification using the MoG models could not be implemented with the reduced feature sets divided along the lines of the voicing decisions. This produced very sparse data for very high dimensionality ($D = 24$ for MFCC and LPCC, and $D = 26$ for PLPCC) and so the ability of the MoG classifier to model the distributions was greatly reduced. In the few trials that were implemented the results displayed a high level of instability and always showed overwhelming bias for just one speaker. As there is no additional data available for the reference speakers, the voiced/unvoiced analysis for the MoG classifier was not implemented.

The final series of tests is conducted with the NN classifier. Here, the training data sets were all limited to just 9s of speech for each speaker and 2.5s of test speech. These are the upper bounds set by the smaller feature sets, i.e. the UVV and VUV sets. The same amount of data for each feature set provides a platform for fair comparison of performance results. The first five feature subsets to be implemented are those pertaining to the 12ΔMFCC feature vectors, which was the original reference feature set. The results measured for this series of tests are listed in Table 9.11.

Performance measure	ALL	V	UV	UVV	VUV
ID rate	5	4	5	6	6
Correct frames	43%	41%	35%	50%	49%

Table 9.11: NN results for the voiced/unvoiced analysis using 12ΔMFCC

The most significant difference shown in Table 9.11 is the correct identification of speakers rate. The correct frame rate increases for the VUV and UVV feature sets, but this alone, as was seen in Section 9.5, is not of vital importance, while the fact that this leads to the correct identification of all six speakers in the reference set is of far greater weight. It suggests that not only are more frames correctly classified, but also that these correctly classified frames are evenly distributed among all 6 speakers.

The next step in searching a way to optimize the classification process is the implementation of the voiced/unvoiced subsets used in conjunction with gender separation. Following the implementation of gender separation based on F_0 estimates with the k-NN classifier, the NN is implemented with the five subsets of the original 12ΔMFCC feature set and the results obtained are listed in Table 9.12. As there are only 3 speakers in each group, the ID rate that represents 100% correctly identified speakers is 3.

From Table 9.12 it is observed that the previously obtained results are confirmed, both for gender separation and for the V/UV analysis. The results from the gender and voicing separation displays improved performance when compared to simply implementing the V/UV subsets for all six speakers, which shows that gender separation once more causes an increased rate of correct classification of the frames. The ID rate does not change much though, and the only two subsets that result in 100% identification for the combined set and the male and female subgroups can be seen in Tables 9.11 and 9.12 as being the UVV and VUV sets. It is interesting to note that while most of the results in Table 9.12 are similar for male and female speakers, a discrepancy exists for the "voiced" and "unvoiced"

Performance measure	Gender Group	MIX	V	UV	UVV	VUV
ID rate	Male	2	2	3	3	3
Correct frames	Male	54%	44%	55%	64%	66%
ID rate	Female	3	2	2	3	3
Correct frames	Female	56%	58%	43%	66%	65%

Table 9.12: NN results for the voiced/unvoiced analysis using gender grouped 12ΔMFCC

frames. Here, the male speakers are recognized at a higher rate for the unvoiced frames, while the opposite holds true for the female speakers. The UVV and VUV subsets yield a more substantial increase in correct frame classification rate than the case for the 6 mixed speakers. For both male and female speakers, using these subsets results in a 10% increase in correct frame classification rate compared to the unsorted feature set for each gender group.

The division of a feature set into 5 subsets labelled with V/UV details was implemented for the 12ΔLPCC, 12ΔwLPCC and 13ΔPLPCC resulted in classification of only one speaker possible in each case, showing extreme bias towards the one correctly identified speaker. The results obtained for the 12ΔMFCC feature set could thus not be reproduced using other feature sets with the NN classifier.

Chapter 10

Conclusions and Future Work

10.1 Conclusions

The aim of this thesis was to create a system that could solve an open-set, text-independent speaker identification task. Several combinations of feature sets and classifiers have been implemented so that an optimal system for a small set of reference speakers could be determined. The work comprised of researching each part of the SID system individually before creating and analyzing the system in its entirety. The ultimate use of the SID system is for implementation in hearing intruments, but this thesis focusses on determining the optimal combination of feature set and classifier and does not include an analysis of the constraints such an implementation involves. The research work was divided into three stages:

1. Preprocessing of speech signals.
2. Selection and extraction of features.
3. Selection and implementation of classifiers.

The performance of the SID system created by combining the different feature sets with the various classifiers has been analyzed in two ways:

1. A complete analysis for each system setup using the data sets as they were.
2. A comparative analysis for a few of these setups when the data sets are split into feature subsets depending on whether the frames contained within the set were voiced, unvoiced, voiced with an unvoiced frame preceding it, unvoiced with a voiced frame preceding it, or unsorted.

Six speakers were chosen from the ELSDSR database[1] to be used as reference speakers throughout this thesis. The speakers were randomly chosen but intentionally an equal amount of male and female speakers were included. The preprocessing of the speech signals that each of these speakers provided resulted in pre-emphasized signals that were divided into short-term segments from which features could be extracted. The feature extraction methods were split into two groups. One consisted of the features that model

[1] see Chapter 8

the source signal in speech, which are more robust to noise but complex to extract reliably. In this thesis the source based features that are extracted are the fundamental frequency, F_0, and the LPC residual. The other group consisted of the system based features that represent the physical characteristics of the vocal tract of a speaker. While these features are less robust to noise than their source based counterparts, they are far simpler to calculate automatically. The system based features were transformed into cepstral coefficients that implement spectral smoothing that is commonly used in speech processing applications. The cepstral coefficients proved very efficient for the speaker identification task. It must be taken into account that the speech signals in the ELSDSR database are uncontaminated by noise.

The method of classification for all classifers is one of consensus over the sequence of classified short-term frames of a speech signal.

The use of density modelling using Mixture of Gaussians models has been documented as being highly suitable for the speaker recognition task [59] and was implemented for all the feature sets. The results were unstable due to the high dimensionality of most of the feature sets and showed a large amount of bias towards the reference speaker labelled as Speaker 1. The best performance for the MoG classifier was achieved when using the 13PLPCC feature set with its first and second order temporal derivatives, as here a total classification rate of the frames from all 6 reference speakers is 71% and the speaker identification rate is 100%. The MoG classifier perfromed particularly poorly for the MFCC feature sets.

A method of impostor detection was implemented using MoG density estimation. This method determined a speaker-specific threshold τ_i and used this to compare all density estimates of a sequence of frames. For the 12ΔMFCC feature set, this method resulted in 100% reference speaker detection and 90% impostor detection. The impostor detection was implemented so that the classification stage only started if the speaker was identified as being a reference speaker. The processing of the impostor speech after it has been detected has not been treated in this thesis. The impostor detection method was not tested with all of the feature sets, due to the amount of time needed to establish each feature specific τ_i. It was tested with the 12ΔLPCC feature set and yielded a less reliable rate of 60% impostor detection. The performance of this method is thus dependent on the feature set used. A larger amount of data for each speaker would have facilitated the determination of the thresholds. It was assumed that impostor detection can be implemented for all feature sets and so the setups that were implemented were used to evaluate performance for a closed-set SID system for the 6 reference speakers.

The k-NN classifier is simple to implement and an evaluation of the SID performance using this method yielded a highest percentage of correctly classified frames when all reference speakers are identified for the 13PLPCC feature set, at 45%. k-NN is exceedingly heavy in computations due to the lack of a separate training and test phase and the high dimensionality of the feature sets used. The highest level of performance for the k-NN classifier was obtained with the PLPCC and the LPCC feature sets.

Neither the MoG nor the k-NN classifier performed satisfactorily using the source based

features, as none of these yielded 100% correct identification of speakers. These feature sets were therefore omitted from the trials with the neural network. The F_0 estimates using the real cepstrum method implemented with the k-NN classifier resulted in the best performance for the source based feature SID system, with a correct sentence classification rate of 67% and an identification rate of 4 speakers out of 6.

The NN classifier proved to be more robust than the MoG classifier when working with feature sets of high dimensionality and limited size, resulting in less bias towards specific speakers, and computationally far more effective than the k-NN classifier. The optimal feature sets used with the NN classifier were the 12$\Delta\Delta$MFCC feature set and the 13ΔPLPCC feature set that each resulted in 61% correct frame classification rate. The NN classifier was capable of identifying all reference speakers correctly for all of the lowest order cepstral coefficient feature sets that were extracted, i.e. for the 8MFCC, 8LPCC, 8wLPCC and 9PLPCC feature sets.

In order to obtain 100% correct speaker identification of the 6 reference speakers, the 12ΔLPCC, 12Δwarped LPCC and 13ΔPLPCC feature sets can be used in combination with all 3 classifiers. The 12ΔMFCC feature only yields this level of performance for the k-NN and NN classifiers. While the MFCC and LPCC features are usually implemented in speaker identification applications, the PLPCC feature set is more common in for use in speech recognition tasks and so the promising results that this feature set yielded had not been expected [65].

It was possible to classify speakers according to gender using any one of the 3 fundamental frequency estimators. 100% correct classification of male and female speakers was achieved for the autocorrelation with center clipping, the YIN and the real cepstrum estimators. Once the speakers had been divided into gender groups, the NN classifier was implemented for each group. For the 12ΔMFCC feature set, the correct frame classification rate was increased by 7% and 13% for the female and the male group, respectively. Similar increases were noted for the 12ΔLPCC and 13ΔPLPCC feature sets, thus leading to the conclusion that the SID system's performance is improved if gender separation is implemented prior to the final classification phase.

The autocorrelation with clipping algorithm also made it possible to divide the frames of each training and test sentence feature vector into five subsets depending on whether they were voiced (V), unvoiced (UV), voiced preceded by a voiced frame (UVV), unvoiced preceded by an unvoiced frame (VUV), or unsorted (ALL). A separation of speakers based on the voiced and unvoiced frames was shown both with a PCA analysis and with the k-NN and NN implementations that these 2 feature subsets do not improve the ability of the system to identify speakers.

The k-NN classifier implemented using the 12ΔLPCC, 12Δwarped LPCC, 13ΔPLPCC and 12ΔMFCC feature sets resulted in an increase in correctly classified frames for the UVV and VUV feature subsets. For the 12ΔMFCC feature set implemented with the NN classifier, the feature subsets only resulted in 100% correct identification of all 6 speakers for the UVV and VUV subsets and a correct frame classification rate increase of 7% and 6%, respectively.

Combining the gender separation method with the feature subsets resulted in the observation that 100% correct speaker identification is only obtained with the VUV and UVV subsets for both male and female speakers. For the limited amount of training data (9s) and test data (2.5s) for each reference speaker using the 12ΔMFCC feature implemented with the NN classifier, the identification rate is increased from 5 out of the 6 speakers to all 6 speakers by using the VUV and UVV feature subsets. Implementing both gender separation and the UVV set, the correct classification rate was increased by 23% for female speakers and 22% for male speakers. This confirms that there is a high level of speaker-specific information present in the transient areas of speech though it gives no indication that frames that are voiced preceded by an unvoiced frame contain more information than the unvoiced frames preceded by a voiced frame, or that the opposite is true. A similar voiced/unvoiced analysis to the one executed here could not be found in the litterature.

The results for the NN classifier could not be reproduced using the 12ΔLPCC, 12Δ warped LPCC and 13ΔPLPCC feature sets.

To summarize, the most robust feature set for this SID task was found to be the PLPCC features that model the filtering that takes place in the ear and thus places emphasis on the perceptually significant parts of speech. The different combinations of feature sets with 3 different classifiers has shown that in the case of a small reference speaker set, the NN classifier can reliably identify all speakers with limited training and test data for several system based cepstral coefficient feature sets, notably for PLPCC and MFCC, especially when these feature sets' temporal derivatives are included. A SID system setup using the k-NN and MoG classifiers performs less satisfactorily. An impostor detection method using MoG density modelling performed well with a 90% impostor detection rate but needs to be perfected. The analysis of the voicing decisions for the test sentences showed that some frames contain more speaker-specific information than others. Exploiting this information could lead to streamlining input data sets so that frames that are irrelevant for the speaker identification task can be excluded from the input data set to the SID system. It has been found that using the frames that are situated at the transient areas of the speech signals can yield up to 23% better performance for the SID system when used in conjunction with gender separation.

10.2 Future Work

Work to improve the capabilities of the MoG classifier is necessary. An eventual averaging of the models estimated from several runs of the EM-algorithm for one speakers' training data could be implemented in order to obtain more reliable results. As the amount of training data is limited, the method of adapting the Mixture of Gaussians model from a Universal Background Model (UBM) as described in [64] might be worth consideration. Using the UBM approach to speaker verification with the log-likelihood computation that is also presented in [64] could be implemented for the imposotor detection task and this should be investigated.

It would be interesting to research the voiced/unvoiced analysis in more detail and see whether there exist other subsets of frames that improve SID performance. A larger pool of data is needed for MoG classification based on the UVV and VUV sets to be possible. For the gender separated UVV and VUV sets, work should be done to implement a way to make the preprocessing of feature sets into these groups and subgroups as efficient and automated as possible so that an eventual application in hearing aids could be considered.

The performance of the MoG classifier, the impostor detector and NN classifer could all be improved if larger data sets were used, and so these methods should be implemented with speech from other databases than the ELSDSR. Testing the methods described in this thesis with speech signals from other databases would allow an analysis of how noise and mismatched training and test conditions affect the SID system's performance.

Appendix A

The Bark Scale

The Bark scale, which will shortly be explained, is used in the derivation of two feature sets: The *warped* LPCC feature set and the *perceptual linear prediction* coefficients. The Bark scale is introduced with the intent of more precisely approximating the frequency scaling that is executed in the biological ear, which may prove beneficial in the identification process.

In order to describe the processes that occur in the human auditory system, a brief description of the different parts of the ear is presented here, though for a detailed description refer to [3] and [52]. The ear is made up of three peripheral parts, the fourth part being the nerve connection from the ear to the brain. The three peripheral parts are referred to as the outer, middle and inner ear. The outer ear receives sounds and conveys them to the inner ear through the ear canal. The ear canal modifies the resonance of the frequencies that pass through it. At the end of the ear canal, the tympanic membrane is situated. The vibrations of this membrane are transmitted to the fluid that is found in the inner ear by the workings of the chain of three tiny bones (called the Hammer, Stirrup and Anvil) in the middle ear. Within the inner ear, the *Basilar Membrane*(BM) is situated, and it is the movements of this membrane that transmits impulses to the brain through hair cells that constitute the *organ of Corti*. A diagram of the ear and the different organs mentioned here can be seen in Figure A, which is taken from [52].

The hair cells of the organ of Corti are divided into the outer and inner hair cells. The outer hair cells have the ability to amplify the vibrations of the Basilar membrane if the latter is exposed to weak sounds, to ensure that the inner hair cells transmit the sound. The Basilar Membrane is of vital importance as it effectively acts as the ear's frequency analyzer. Changes in the frequency of incoming sounds cause movement along the membrane. This means that different frequencies are interpreted by the variation of the position along the BM. An envelope can be modelled to contain the pattern on the BM that is produced by a frequency and it is the width of the peak of this envelope that indicates the frequency selectivity within the inner ear. The selectivity is thus interpreted as a bank of filters. Each filter has a critical bandwidth (CB) that varies depending on the center frequency of the incoming sound signal. Each filter is labelled with a number on the Bark scale.

The Bark scale is named after Barkhausen, a German acoustician. The scale consists of values 0 to 23 Bark, as there is space for just 24 critical band filters on the basilar

membrane. Up to a center frequency of 500Hz, there exists a linear relation between the frequency and Bark scale. In this interval the CB is 100Hz, meaning, as an example, that the frequency bandwidth band from 200Hz-300Hz corresponds to 3 Bark. Above 500Hz, the relation between frequency and Bark becomes logarithmic, notably in the interval from 600Hz - 7kHz. F.ex., the 8^{th} Bark indicates the filter with a center frequency of 1kHz and that has a bandwidth of 160Hz. The critical band rate z, in Bark, is defined in [47] as:

$$z = \left[\frac{26.81}{1 + 1960/f}\right] - 0.53, \quad f \text{ in Hz} \tag{A.1}$$

In Figure A.2, the curve of the Bark scale is seen to be linear for the first few Bark values and logarithmic as the values increase. The blue curve shows the relation between the Bark scale and the frequency of incoming signals and the red curve depicts the critical bandwidth of each filter as a function of the Bark values.

The CB in Hz and the incoming frequencies that correspond to the Bark scale are listed in Table A.1.

Frequency/Hz	CB-rate/Bark	Critical Bandwidth/Hz
20	0	80
100	1	100
200	2	100
300	3	100
400	4	110
510	5	120
630	6	140
770	7	150
920	8	160
1080	9	190
1270	10	210
1480	11	240
1720	12	280
2000	13	320
2320	14	380
2700	15	450
3150	16	550
3700	17	700
4400	18	900
5300	19	1100
6400	20	1300
7700	21	1800
9500	22	2500
12000	23	3500

Table A.1: Input frequencies and the corresponding Bark values and Critical Bandwidths

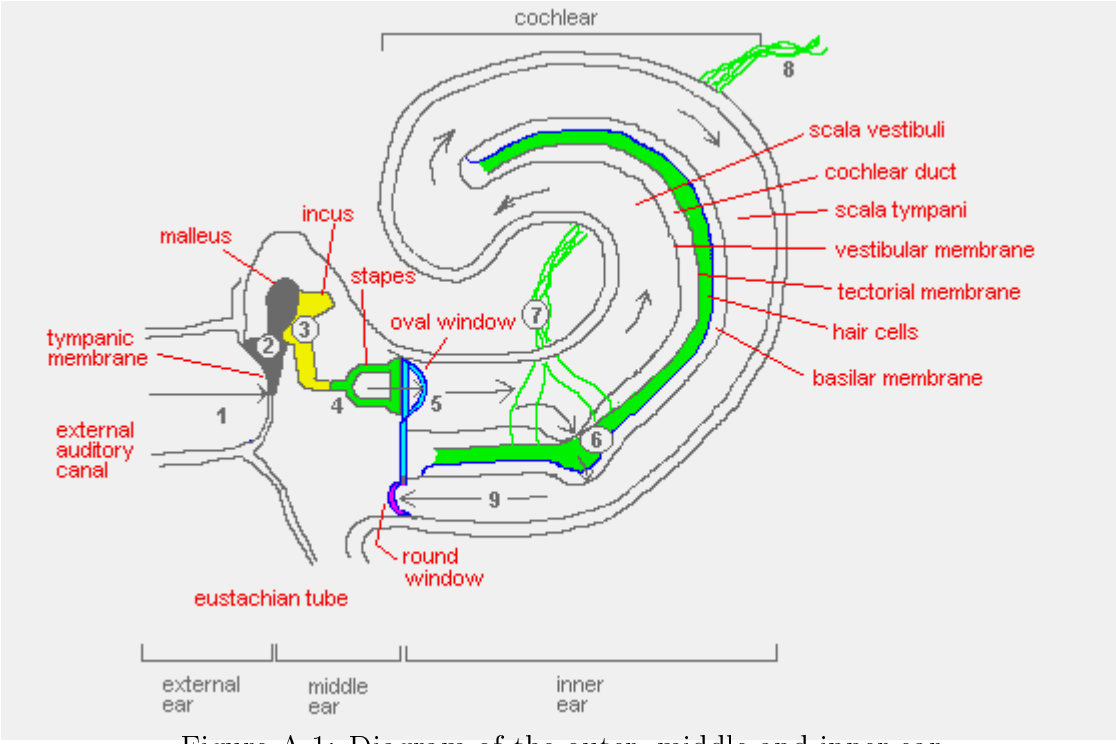


Figure A.1: Diagram of the outer, middle and inner ear

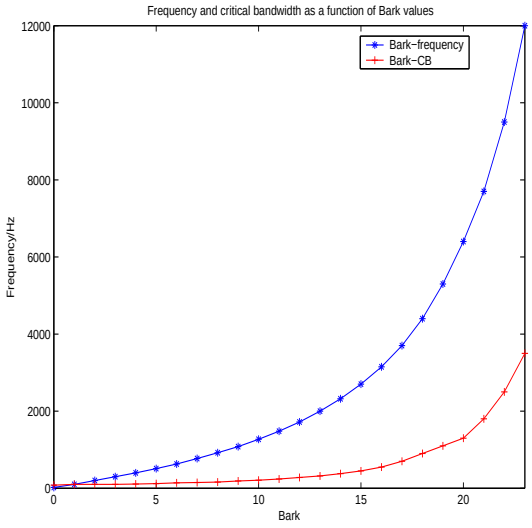


Figure A.2: The Bark scale and corresponding frequencies and critical bandwidths

Appendix B

Parameter Estimation using the EM-algorithm

First, the parameters of the MoG model - the mean vector μ, covariance matrix $\mathbf{\Sigma}$ and mixture weights $P(j)$ - are initialised. The initial values can be found either randomly or by a scheme such as the k-means algorithm [42]. The former alternative is not desirable, though, as the EM-algorithm is never guaranteed to converge at a global maximum of the likelihood function rather than a local one, especially in a high-dimensional case. The choice of initial values determines which local maximum the algorithm converges towards, and the quality of the resultant parameter estimation depends on this. If the initialisation is in an area far from any local maximum, there is a risk that the EM-algorithm converges before the extremum is reached. This can occur because as the likelihood increases, the covergence slows down and is often stopped by some stopping criteria, such as a maximum number of steps being reached or a sufficiently small change in likelihood is recorded.

During the first iteration, the initial model is referred to as the old model. The parameters of this model are kept constant during the E-step. For each iteration of the EM algorithm, the new parameter values are estimated during the M-step and then kept constant and used as the old parameter set for the derivation of an updated expectation function in the E-step.

From Section 5.3, it is established that the training data feature vectors make up the incomplete data set that does not have a class label assigned to each training sample. To fill in as a class label, the component j that has the highest probability of having produced the training data sample $\mathbf{x}_n$ is determined. This probability corresponds to a posterior probability and is therefore obtained using Bayes' Theorem, which for convenience is shown again here:

$$P(j|\mathbf{x}_n) = \frac{p(\mathbf{x}_n|j)P(j)}{p(\mathbf{x}_n)} \tag{B.1}$$

The computation of the posterior probability, $P(j|\mathbf{x}_n)$, is executed in the E-step for each Gaussian component, using parameter values that are obtained in the previous iteration (or, in the case if the first iteration, the initial model parameter set).

After determining the posterior probabilities $P(j|\mathbf{x}_n)$, the E-step consists of computing the conditional expectation of the complete data set log-likelihood, Q, given the data

point $\mathbf{x}_n$ and the current parameter set.

For a complete data set, including both $\mathbf{x}_n$ and z_n, the likelihood, $\mathcal{L}^c$, is denoted in Eq.(B.2).

$$\mathcal{L}^c = \prod_{n=1}^{N} p^{new}(\mathbf{x}_n, z_n) \tag{B.2}$$

By maximizing $\mathcal{L}^c$ with respect to each of the model's adjustable parameters, one can obtain the model parameter set that is most likely to have generated the given training data. Instead of finding the maximum likelihood, it is common practice to determine the maximum log-likelihood. Taking the logarithm of a multiplicative problem, we obtain an additive one, and as the logarithm is a monotonic function, it is analytically easier to manipulate in order to determine the minimum negative log-likelihood, yet its solution corresponds to finding the parameter set that yields the maximum likelihood.

The negative log-likelihood of the complete data set is derived in Eq.(B.3).

$$-\ln \mathcal{L}^c = -\sum_{n=1}^{N} \ln \left\{ p^{new}(\mathbf{x}_n, z_n) \right\} = -\sum_{n=1}^{N} \ln \left\{ P^{new}(z_n) p^{new}(\mathbf{x}_n | z_n) \right\} \tag{B.3}$$

For the actual data set, where the class labels are replaced by the posterior probabilities $P(j|\mathbf{x}_n)$, the negative log-likelihood for each component j is:

$$-\ln \mathcal{L}^j = -\sum_{n=1}^{N} P^{old}(j|\mathbf{x}_n) \ln \left\{ P^{new}(j) p^{new}(\mathbf{x}_n | j) \right\} \tag{B.4}$$

Finally, the expectation of the negative log-likelihood is obtained by evaluating the sum of the negative log-likelihood functions over all M components:

$$Q = -\sum_{n=1}^{N} \sum_{j=1}^{M} P^{old}(j|\mathbf{x}_n) \ln \left\{ P^{new}(j) p^{new}(\mathbf{x}_n | j) \right\} \tag{B.5}$$

where the negative log-likelihood expectation function, Q, is the error function that must be minimized in order to obtain the optimal MoG model parameter set.

During the M-step, the minimum of the expectation function Q is found by determining the first derivative of Q w.r.t the *new* parameters and setting it to zero, then finding the solution with provisions made for the constraints that are listed in connection with Eq.(5.6). This leads to the equations for the updated values of the MoG model parameters. These updates are shown in Eq.(B.6)-(B.8).

$$P(j)^{new} = \frac{1}{N} \sum_{n=1}^{N} P^{old}(j|\mathbf{x}_n) \tag{B.6}$$

$$\mu_j^{new} = \frac{\sum_{n=1}^{N} P^{old}(j|\mathbf{x}_n)\mathbf{x}_n}{\sum_{n=1}^{N} P^{old}(j|\mathbf{x}_n)} \tag{B.7}$$

$$\mathbf{\Sigma}_j^{new} = \frac{\sum_{n=1}^{N} P^{old}(j|\mathbf{x}_n)(\mathbf{x}_n - \mu_j^{new})(\mathbf{x}_n - \mu_j^{new})^T}{\sum_{n=1}^{N} P^{old}(j|\mathbf{x}_n)} \tag{B.8}$$

The above equations execute the E-step and the M-step simultaneously. The estimation of the new component weight values belongs to the E-step, while the mean and covariance updates are performed in the M-step.

The updates of the various parameters are quite logical: In Eq.(B.6), it can be seen that when using the maximum likelihood solution to obtain the new prior probability for the j^{th} component, this is given by the average posterior probability for that component, given $\mathbf{x}_n$, over the entire data set. The updated mean in Eq.(B.7) is just the mean of the data vectors, weighted by the posterior probabilities that the data vectors were generated by the j^{th} component. Finally, in Eq.(B.8), the updated covariance matrix is also weighted by the posterior probabilities and derived from the distance of the data samples from the j^{th} component mean.

Appendix C

The Biological and Artificial Neuron

C.1 The Biological Neuron

The brain consists of billions of nerve cells called neurons. The biological neuron consists of four main parts: the soma, dendrites, axon and the synapses, which is the area that transmits signals between neurons. In addition, the neuron contains a cell nucleus and a hillock. All of these parts are shown in the biological neuron schematic in Figure C.1, taken from [55].

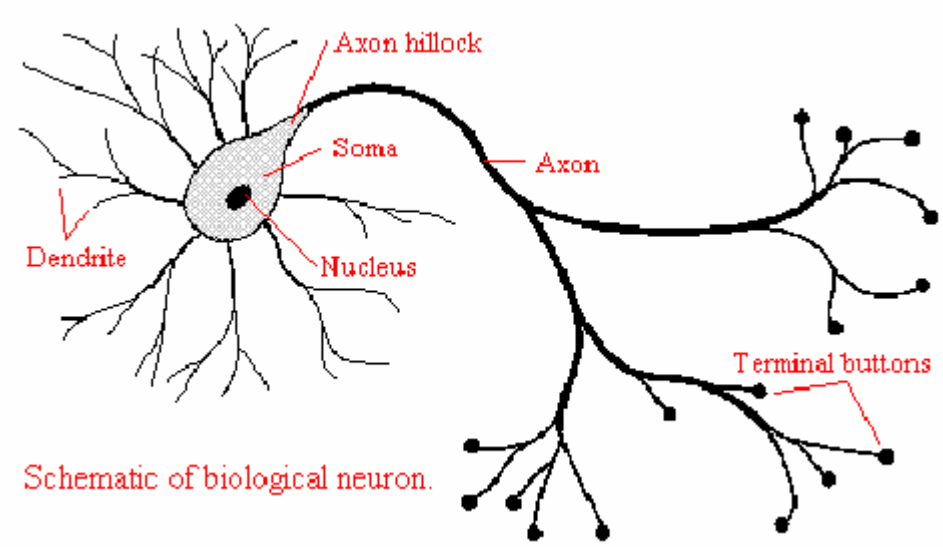


Figure C.1: Schematic of a biological neuron

The body of the neuron is the soma. The dendrites that extend from the soma are the neuron's input channels. They receive signals from other neurons through the synapses, which can connect them to thousand of other neurons. These signals are in the form of electrically charged ions. The input from the dendrites are added together in the soma, which then decides how to react. The neuron can be in two states: the resting and the "firing" state. The states change depending on the input that is received and once a certain threshold is reached, the neuron "fires" - sending an impulse towards the synapse.

The hillock is situated at the origin of the axon and generates the outgoing pulses. At the ends of the axon, the terminal buttons, otherwise called boutons, are found. Here, chemical neurotransmitters are produced that activate the synapses. The synapses may cause the neurons that are connected to it to fire, or prevent some of them from firing. It is the combination of impulses received through the synapses between the axon of one neuron and the dendrite of another neuron that is then summed up with numerous other signals received in the latter neuron's remaining dendrites, and so forth. The operation of the neuron can be altered by continuously stimulating it with a certain combination of input signals, which can cause its resting potential to be modified. This means that the neuron can learn to recognize an input combination so that connections that the neuron is exposed to frequently may cause a more rapid transfer of impulses to other neurons. In this way, the operation of the neuron is defined by comparing its internal parameters with the incoming signals. The structure and workings of the biological neuron are complex and this streamlined description is only meant to provide a brief introduction in order to understand the motivation behind the structure of artificial neural networks. For a detailed description of the operation of the biological neural network, one can refer to [53].

C.2 The Artificial Neuron

The artificial neuron is modelled according to the biological neuron, and therefore imitates some of the latter's functionality. Inputs from several preceding neurons are weighted by the connections (weights) between neurons and summed up in the receiving neuron. The sum of these inputs is referred to as the *activity*, a_k, of the neuron.

$$a_k = \sum_{i=0}^{I} w_{ki} z_i \tag{C.1}$$

w_{ki} denotes the weight connecting unit i to unit k and z_i denotes the output from unit i.

A transfer function, g, "fires" when the summed input increases above a certain threshold if it is a step function. A transfer function like *tanh* that is used in the perceptron that is discussed in Section 7.3 results in different values in the range $[-1, 1]$ for different input combinations.

$$z_k = g(a_k) \tag{C.2}$$

The output, z_k, is then weighted by the next layer of connections and used as input to the following neurons. This is repeated for as many layers of neurons as are necessary for the solution of a problem. A diagram of the artificial neuron is shown in Figure C.2.

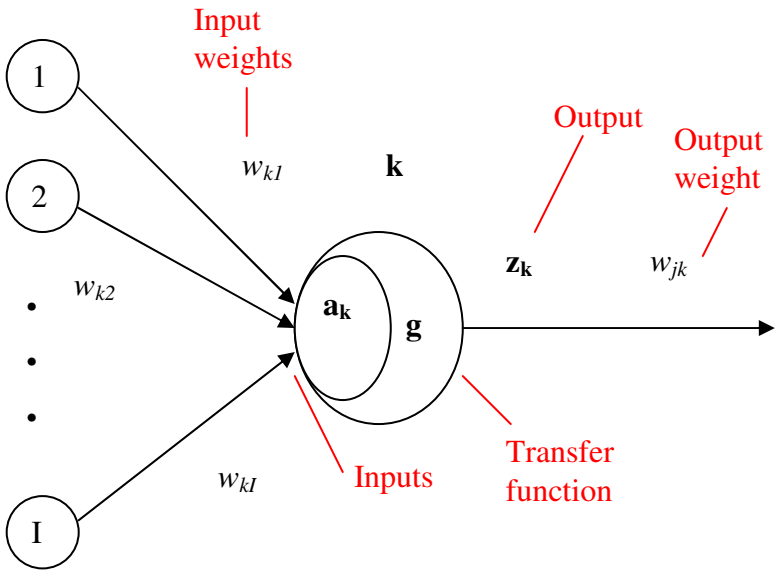


Figure C.2: Diagram of an artificial neuron

In the multi-layer perceptron, the weight values are changed according to how much error is found at the output of the entire network, and thus changes the weight combinations to each neuron until a good approximation of the input pattern is determined, as described in Sections 7.3 and 7.4.

Appendix D

BFGS algorithm to train network weights

In order to determine the optimal weight values during the training phase, the BFGS algorithm [45] is implemented. This stands for Broyden, Fletcher, Goldfarb and Shanno, who are responsible for the creation of this iterative updating algorithm. When called with the network parameters (the cost function and its derivatives w.r.t the network weights), this algorithm returns the optimal weight values of the network for the given input data set. The way that this training is executed cannot be adequately explained without a brief introduction to the theory behind the determination of optimal weight values as the process of minimizing a function, and so this introduction is presented in what follows. For a complete derivation, refer to [45].

The process of determining an optimal weight matrix entails determining the minimum of the given cost function. Often a local minimum is found, as it is too complex to implement a search method to find the global minimum.

The fundamental aim of the optimization algorithm is to find $\mathbf{w}^*$ in Eq.(D.1)

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} F(\mathbf{w}), \qquad F : \Re^m \mapsto \Re \tag{D.1}$$

where F is the cost function and m is the number of elements in $\mathbf{w}$. A simple illustration of $\mathbf{w}^*$ is given in Figure D.1, where the function shown is

$$F(\mathbf{w}) = (w_{21} - w_{32} - 2)^2 + 50 \cdot (w_{21} - w_{32})^2$$

In Figure D.1, the minimum is global and is found at $F(\mathbf{0})$. These conditions often do not prevail when more complex, high-dimensional data is used to define the function F. It is known that the first derivative of a 1-dimensional continuous differentiable function defines the slope of that function, and that when this slope is zero, the function is at a *stationary point*, i.e. either a maximum, minimum or a saddle point. When working in more than one dimensions, the slope of the multivariate function is called the *gradient*, $\nabla\mathbf{F}$, and is defined in Eq.(D.2).

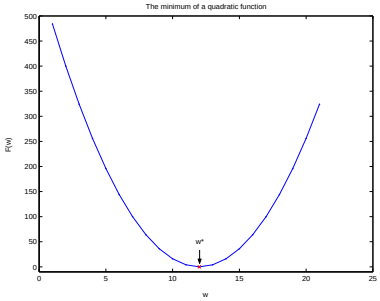


Figure D.1: The minimum $\mathbf{w}^*$ of the quadratic function, $F(\mathbf{x}) = (w_{21} - w_{32} - 2)^2 + 50 \cdot (w_{21} - w_{32})^2)$

$$\nabla \mathbf{F} = \mathbf{F}'(\mathbf{w}) \equiv \begin{bmatrix} \frac{\partial F}{\partial w_{21}}(\mathbf{w}) \\ \frac{\partial F}{\partial w_{32}}(\mathbf{w}) \\ \vdots \\ \frac{\partial F}{\partial w_{m(m-1)}}(\mathbf{w}) \end{bmatrix} \tag{D.2}$$

It is now possible to establish a condition that must be met for a point to be a local minimum:

$$\nabla \mathbf{F} = 0 \tag{D.3}$$

Using this as a criteria does however not only locate local minimums, but can determine any stationary point. It is therefore necessary to introduce an additional condition in order to ensure that the stationary point is, indeed, a minimum. For this objective, the second order derivatives that define the surface of the cost function $\mathbf{F}$ are needed. These derivatives make up the *Hessian matrix*, $\mathbf{H}$, and define the gradient of the gradient function:

$$\mathbf{H} = \nabla(\nabla \mathbf{F}) \equiv \left[\frac{\partial^2 F}{\partial w_{kh} \partial w_{lk}} \right] \tag{D.4}$$

When the Hessian matrix that is evaluated for a stationary point proves to be *positive definite*, the stationary point is a local minimum. A positive definite matrix is defined in Eq.(D.5).

$$\mathbf{g}^T \mathbf{H} \mathbf{g} > 0 \qquad \forall \mathbf{g} \tag{D.5}$$

in which case $\mathbf{H}$ is a positive definite matrix.

Using the above mathematical derivations, it is now possible to briefly outline the process of optimization of weight values when using the BFGS algorithm. First, however, as the formulae for the BFGS updating are used in combination with a *soft line search*, the latter is described below:

soft line search -
Apart from the conditions that need to be satisfied in order to ensure that a local minimum is found, an optimization algorithm must also determine a *search direction*, which defines

the direction to follow in order to reach the desired minimum. From Figure D.1 it is clear that in order to find $\mathbf{w}^*$, starting from an arbitrary point, the cost function must decrease for each iterate. The search direction is denoted as $\mathbf{r}^T$. The cost function that is distance α from the point $\mathbf{w}$ in the direction $\mathbf{r}$ can be approximated by the first order Taylor series in Eq.(D.6), given that α is not very large and that $\alpha > 0$.

$$F(\mathbf{w}) + \alpha\mathbf{r} = F(\mathbf{w} + \alpha\mathbf{r}^T\nabla F) \tag{D.6}$$

The direction $\mathbf{r}$ is a descent direction from $\mathbf{w}$ if $\alpha\mathbf{r} < 0$. The soft line search strives to determine the value α_s, which must meet the criteria to be an acceptable argument for the function $\nu(\alpha)$, where:

$$\nu(\alpha) = F(\mathbf{w} + \alpha\mathbf{r}) \tag{D.7}$$

This acceptable argument, α_s, must satisfy the following criteria:

$$\alpha_s = \arg\min_{\alpha}(F(\mathbf{w} + \alpha \cdot \mathbf{r})) \tag{D.8}$$

Another function, $\omega(\alpha)$, is defined as a point on the cost function that goes through the starting point and moves away from it by a fraction of the starting point slope:

$$\omega(\alpha) = \nu(0) + \varrho \cdot \nu'(0) \cdot \alpha, \qquad 0 < \varrho < 0.5 \tag{D.9}$$

This can be used in order to determine an upper limit for α_s:

$$\nu(\alpha_s) \leq \omega(\alpha_s) \tag{D.10}$$

There must also be a limit as to how small the search step is, as a step size that is too small may cause convergence before the region of the minimizer is reached. The condition in Eq.(D.11) must thus also be satisfied.

$$\nu'(\alpha_s) \geq \beta \cdot \nu'(0), \qquad \varrho < \beta < 1 \tag{D.11}$$

In each iteration, the cost function $F(\mathbf{w})$ is approximated by a quadratic function that yields a parabolic form for the cost function. The search determines an interval containing acceptable points, and a point α is found within this region. When both criteria in Eq.(D.10) and Eq.(D.11) are met, convergence is reached and $\alpha_s = \alpha$. In the case where one or both of the criteria are not met, the interval is refined and a new α is determined.

In Figure D.2, the interval $[a, c]$ contains the minimizer α_s of the shown quadratic function. The interval $[a, b]$ represents the area that is found when the condition in Eq.(D.11) is not satisfied, i.e. the step size is too small and thus the local minimum cannot be determined in the interval $[a, b]$. The step size that defines the interval $[a, d]$ is too large, and the cost function increases in this case. The interval must be refined as the condition in Eq.(D.10) is not satisfied.

We return now to the updating of the BFGS parameters. In order to complete the introduction to the BFGS updating process, the quadratic function $Q(\mathbf{w})$ is shown in Eq.(D.12). This function approximates the cost function $F(\mathbf{w})$ and it is the minimizer of $Q(\mathbf{w})$ that must be determined in each iteration.

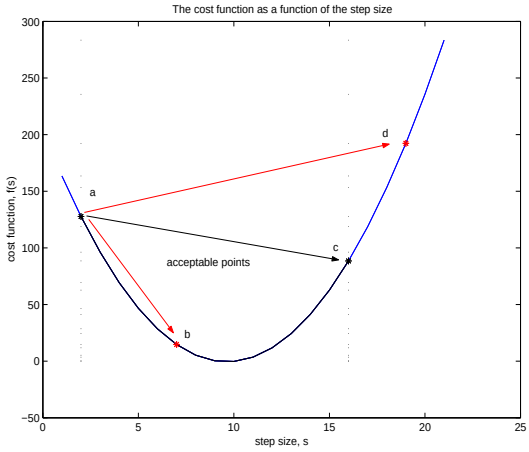


Figure D.2: The interval $[a, c]$ containing acceptable points

$$Q(\mathbf{w}) = a + \mathbf{b}^T\mathbf{w} + \frac{1}{2}\mathbf{w}^T\mathbf{H}\mathbf{w} \tag{D.12}$$

As the evaluation of the Hessian matrix can prove to be extremely complex, an approximation to it is introduced: $\mathbf{B} \simeq \mathbf{H}$. The BFGS algorithm approximates the inverse of $\mathbf{B}$, $\mathbf{D} = \mathbf{B}^{-1}$.

The following update for $\mathbf{D}$ is taken from [45].

$$\begin{aligned}
\mathbf{D}_{new} &= \mathbf{D} + a\mathbf{r}\mathbf{r}^T - b(\mathbf{r}\mathbf{v}^T + \mathbf{v}\mathbf{r}^T), && \text{(D.13)}\\
\text{where} & && \text{(D.14)}\\
\mathbf{r} &= \mathbf{w}_{new} - \mathbf{w}, && \text{(D.15)}\\
\mathbf{y} &= \nabla F(\mathbf{w}_{new}) - \nabla F(\mathbf{w}), && \text{(D.16)}\\
\mathbf{v} &= \mathbf{D}\mathbf{y}, && \text{(D.17)}\\
b &= \frac{1}{\mathbf{r}^T\mathbf{y}}, && \text{(D.18)}\\
a &= b(1 + b(\mathbf{y}^T\mathbf{v})) && \text{(D.19)}
\end{aligned}$$

The initial $\mathbf{D}$ is checked for symmetry and positive definiteness. It follows that if the Hessian matrix of Eq.(D.12) is positive definite, then there is a single minimizer for $Q(\mathbf{w})$. $\nabla F(\mathbf{w})$ are the cost function derivatives w.r.t. weight values that are provided by the calling back propagation algorithm. The update for $\mathbf{D}$ is only implemented if the following condition is satisfied:

$$\mathbf{r}^T\nabla F(\mathbf{w})_{new} > \mathbf{r}^T\nabla F(\mathbf{w}) \tag{D.20}$$

as the increased curvature of the cost function shows that the search is approaching the minimum.

For each iteration, the search direction is initialized at $\mathbf{D} \cdot (-\nabla\mathbf{F})$. The value of α is estimated by the soft line search, and the value of $\mathbf{D}$ is updated if the condition in

Eq.(D.20) is satisfied. The algorithm continues until convergence is reached, i.e. the gradient falls below a certain very small threshold or the stepsize becomes very small, indicating the proximity of a stationary point and, due to the positive definiteness of $\mathbf{D}$, thus a minimum. However, if the convergence is very slow or if the algorithm diverges, the iterations stop when a certain preset maximum number of iterations is reached, as it is then assumed that further updates will not aid convergence.

The finer details of the theory behind the discussed optimization techniques have been omitted in the above overview, but it remains to be mentioned that the BFGS is a popular updating algorithm because it can determine the quadratic minimizer faster than the *conjugate gradient* methods and is yet computationally less heavy than the *Newton* method, in which the actual Hessian matrix must be calculated. The BFGS method belongs to the *Quasi-Newton* methods. The general theory of nonlinear optimization algorithms, as well as a detailed description of the Conjugate Gradient, Newton and Quasi-Newton methods are given in chapter 7 of [15], as well as in [45], while a complete description of the program that implements the BFGS algorithm can be found in [46].

Bibliography

[1] John R. Deller, Jr., John H.L. Hansen, and John G. Proakis. *Discrete-Time Processing of Speech Signals*. IEEE Press, 2000.

[2] Bob Dunn. "Speech Signal Processing and Speech Recognition". Technical report, IEEE Signal Processing Society, 2003.

[3] Torben Poulsen. *Ear, Hearing and Speech - A short introduction*. Ørsted DTU, 2001.

[4] Torben Poulsen. *Lydopfattelse*. Ørsted DTU, 1998.

[5] F. Bimbot, J-F. Bonastre, C. Fredouille, G. Gravier, I. Magrin-Chagnolleau, S. Meignier, T. Merlin, J. Ortega-Garcia, D. Petrovska-Delacretaz, and D.A.Reynolds. "A Tutorial on Text-Independent Speaker Verification". *EURASIP Journal on Applied Signal Processing*, 4:430–451, 2004.

[6] F.Farahani, P.G. Georgiou, and S.S. Narayanan. "Speaker Identification using Supra-Segmental Pitch Pattern Dynamics". *Proceedings of ICASSP, SP-L4.5*, 2004.

[7] B.Yegnanarayana, C. d'Alessandro, and V. Darsinos. "Iterative Algorithm for Decomposition of Speech Signals into Periodic and Aperiodic Components". *IEEE Transactions on Speech and Audio Processing*, 6(1), 1998.

[8] B. R. Wildermoth. "Text-Independent Speaker Recognition using Source Based Features". Master's thesis, Griffith University, Australia, January 2001.

[9] J.P. Campbell. "Speaker Recognition: A Tutorial". *Proceedings of the IEEE*, 85(9):1437–1462, September 1997.

[10] K.K. Paliwal and B.S. Atal. "Frequency-related Representation of Speech". *Eurospeech, Geneva*, 2003.

[11] J. Fry. *"Acoustics of Vowels"*. Linguistics 124, Computers and Spoken Language, SJSU, 2003.

[12] A. N.Iyer, M. Gleiter, B.Y. Smolænski, and R.E. Yantorno. "Structural Usable Speech Measure Using LPC Residual". *ISPACS C2-3*, December 2003.

[13] J.G. Proakis and D.G.Manolakis. *Digital Signal Processing - Principles, Algorithms and Applications*. Prentice Hall, 3 edition, 1996.

[14] D.Rentzos. "Inter and Intra Speaker Voice Characteristics and Models", Chapter 3. Master's thesis, Brunel University, Dept. of Electronic and Computer Engineering, London, 2003.

[15] C.M. Bishop. *Neural Networks for Pattern Recognition.* Oxford University Press, 2002.

[16] P.W. Wagacha. "Instance-Based Learning: k-Nearest Neighbour". Technical report, Institute of Computer Science, University of Nairobi, 2003.

[17] D. Reynolds, W. Andrews, J. Campbell, J. Navrati, B. Peskin, A. Adami, Q. Jin, D. Klusacek, J. Abrahamson, R. Mihaescu, J. Godfrey, D. Jones, and B. Xiang. "The SuperSID Project: Exploiting High-Level Information for High-accuracy Speaker Recognition". *SuperSID Workshop*, 2002.

[18] D.A Reynolds D. Klusacek, J. Navratil and J.P.Campbell. "Conditional Pronounciation Modeling in Speaker Detection". *SuperSID Workshop*, 2002.

[19] M.K. Sönmez, L. Heck, M. Weintraub, and E. Shriberg. "A Lognormal Tied Mixture Model of Pitch for Prosody-Based Speaker Recognition". *Eurospeech97*, 1997.

[20] S. Guruprasad, N.Dhananjaya, and B.Yegnanarayana. "AANN Models for Speaker Recognition Based on Difference Cepstrals". *Proceedings of the International Joint Conference on Neural Networks*, 1:692–697, 2003.

[21] S. Molau, M. Pitz, R. Schlüter, and H. Ney. "Computing Mel-Frequency Cepstral Coefficients on the Power Spectrum". Technical report, Computer Science Dept., University of Technology, Aachen, Germany, 2003.

[22] M. Faúndez-Zanuy and D. Rodríguez-Porcheron. "Speaker Recognition using Residual Signal of Linear and Nonlinear Prediction Models". *ICSLP*, 2:121–124, 1998.

[23] Dr. J.W. Koolwaaij. "Speech Processing", http://www.iSpeak.nl, 2001.

[24] O. Ibarra and F. Curatelli. *A Brief Introduction to Speech Analysis and Recognition, An Internet Tutuorial*, 2000.

[25] Malcolm Slaney. "Auditory Toolbox", version 2. Technical report, Interval Research Corporation, 1998.

[26] B. Wildermoth and K. Paliwal. *Use of Voicing and Pitch Information for Speaker Identification.* School of Micorelectronic Engineering, Griffith University, Australia, 2001.

[27] D. Gongaza da Silva, J. A. Apolinário Jr., and C. B.de Lima. "On the Effect of the Language in CMS Channel Normalization". *International Telecommunications Symposium ITS, Natal, Brazil*, 2002.

[28] Qi Li and Biing-Hwang Juang. "Fast Discriminative Training for Sequential Observations with Application to Speaker Identification". *ICASSP*, 2:217–220, 2003.

[29] K. Chen. "On the Use of Different Speech Representations for Speaker Modeling". *IEEE Transactions of Systems, Man and Cybernetics (Part C)(Special issue on Biometric Systems)*, 34, 2004.

[30] A. Cohen and Y. Zigel. "On Feature Selection for Speaker Verification". *Proceedings of COST 275 workshop on The Advent of Biometrics on the Internet*, pages 89–92, 2002.

[31] N. Jhanwar and A.K. Raina. "Clustering for Speaker Identification using Pitch Correlogram". *IJCI Proceedings of International Conference on Signal Processing, ISSN 1304-2386*, 1(2), 2003.

[32] B. Yegnanrayana, K. Sharat Reddy, and S. P. Kishore. "Source and System Features for Speaker Recognition using AANN Models". *ICASSP*, 2000.

[33] Tae-Yun Kim. "Speech Production". Technical report, Intelligent Information & Signal Processing Lab, Korea University, 2003.

[34] Ling Feng. "Speaker Recognition". Master's thesis, IMM, Denmarks Technical University, 2004.

[35] L. P. Cordella, P. Foggia, C. Sansone, and M. Vento. "A Real-Time Text-Independent Speaker Identification System". *Proceedings of the ICIAP*, page 632, 2003.

[36] The UNKNOWN group. "LPC Vocoder and Spectral Analysis". *Rice University*, 2000.

[37] E. W. Weisstein. *Distance.* From MathWorld–A Wolfram Web Resource. http://mathworld.wolfram.com/Distance.html, 2004.

[38] D. Ellis. "EE E6820, Lecture 5: Speech Modeling and Synthesis". *Columbia University Dept. of Electrical Engineering*, Spring 2004.

[39] S. Cassidy. "COMP449: CH.7, The Source Filter Model of Speech Production". *Dept. of Computing, Macquarie University, Sydney, Australia*, 2002.

[40] S.B. Davis and P. Mermelstein. "Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Resources". *IEEE transactions on Acoustics, Speech, and signal Processing*, 28:357–366, 1980.

[41] L. Scharenbroich. "A First Tutorial on the MLX Schema". *JPL Machine Learning Systems Group, California Institute of Technology*, 2002.

[42] E.W. Weisstein. *K-means clustering algorithm.* From MathWorld–A Wolfram Web Resource. http://mathworld.wolfram.com/K-MeansClusteringAlgorithm.html, 2004.

[43] J.A. Blimes. "A Gentle Tutorial of the EM algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models". *Dept. of Electrical Engineering anf Computer Science, U.C Berkely, TR-97-021*, 1998.

[44] O. Winther. *VBMoG.m - Variational Bayes Mixture of Gaussians.* Department og Mathematical Modelling, Technical University of Denmark, 2003.

[45] P.E. Frandsen, K. Jonasson, H.B. Nielsen, and O. Tingleff. "Unconstrained Optimization". Technical report, Department og Mathematical Modelling, Technical University of Denmark, 1999.

[46] H.B. Nielsen. "UCMINF - An Algorithm for Unconstrained, Nonlinear Optimization". Technical report, Department of Mathematical Modelling, Technical Univeristy of Denmark, 2000.

[47] H. Traunmüller. "Auditory Scales of Frequency Representation". *Phonetics at Stokholm University*, August 1997.

[48] A. de Cheveigné and H. Kawahara. "YIN, a Fundamental Frequency Estimator for Speech and Music". *Acoustical Society of America, Vol.111(4)*, 2002.

[49] A. Härmä and U.K. Laine. "A Comparison of Warped and Conventianl Linear Predictive Coding". *IEEE Transactions on Speech and Audio Proceedings*, 9(5), 2001.

[50] David Gerhard. "Pitch Extraction and Fundamental Frequency: History and Current Techniques", 2003.

[51] D. Chow and W. H. Abdulla. "Robust Speaker Identification Based on Perceptual Log Area Ratio and Gaussian Mixture Models". *Electrical and Electronic Engineering Dept., University of Auckland, New Zealand*, 2002.

[52] Tim Jacob. *The Ear*. School of Biosciences, Cardiff University, 2002.

[53] Jeanette Lawrence. *Introduction to Neural Nets: Design, Theory, and Applications, 6th Edition*. California Scientific Software, 1994.

[54] S.Sigurdsson, J.Larsen, L.K.Hansen, P.A.Philipsen, and H.C. Wulf. "Outlier Estimation and Detection Application to Skin Lesion Classification". *ICASSP, Neural-P01, nr.2127*, 2002.

[55] Niel Fraser. *Introduction to Neural Networks*. Carleton University, Ottawa, Canada, 1998.

[56] D. MacKay. "The Evidence Framework Applied to Classification Networks". *Neural Computation*, 4(5):720–736, 1992.

[57] Sigurd Sigurdsson. "nc-multiclass Neural Network", http://mole.imm.dtu.dk/toolbox/ann. *Department og Mathematical Modelling, Technical University of Denmark*, 2004.

[58] R. Canuana and T. Joachims. "Description of Performance Metrics", http://kodiak.cs.cornell.edu/kddcup/metrics, 2004.

[59] D. Reynolds. "Robust Text-Independent Speaker Identification Using Gaussian Mixture Models". *IEEE transactions on Speech and Audio Processing, Vol.3, no.1*, 1995.

[60] N. Laird A. Dempster and D.Rubin. "Maximum Likelihood from Incomplete Data via the EM algorithm". *Journal of the Royal Statitical Society, B, vol.39, no.1*, 1977.

[61] L. Xu and M. Jordan. "On Convergence Properties of the EM Algortihm for Gaussian Mixtures". *Neural Computation, no.8*, pages 129–151, 1996.

[62] H. Hermansky. "Perceptual Linear Predictive (plp) Analysis of Speech". *J.Acoust. Soc. Am.*, 87(4):1738–1752, 1990.

[63] S. Yoo, J.R. Boston, J. Durrant, K.Kovacyk, S. Karn, S. Shaiman, A.El-Jaroudi, and C.C. Li. "Relative Energy and Intelligibility of Transient Speech Information". *ICASSP SP-L3.6*, 2005.

[64] D.A. Reynolds, T.F. Quatieri, and R.B. Dunn. "Speaker Verification Using Adapted Gaussians". *Digital Signal Processing*, 1:19–41, 2000.

[65] J. Smolders and D. Van Compernolle. "In Search for the Relevant Parameters for Speaker Independent Speech Recognition". *KUL/RUG/VUB Speech Seminar*, May 1993.