The University of Manchester

Speaker Recognition

Speaker Diarization and Identification

A dissertation submitted to the University of Manchester for the degree of Master of Science in the Faculty of Engineering and Physical Sciences

By: **Ubai SANDOUK**

Supervisor: **Dr Ke CHEN**

The University of Manchester

School of Computer Science

2012

List of Contents

List of Contents **2**
List of Figures **5**
Abstract **7**
Declaration **8**
Copyright Statement **8**
Acknowledgment **9**

Chapter 1: Introduction **10**
1.1. Primer 10
1.2. Project Scope 10
1.3. Thesis Structure 12

Chapter 2: Background **14**
2.1. Speaker Recognition Tasks 14
2.2. Sound and Digital Signal Processing 15
2.2.1. Sound 15
2.2.2. Signals 16
2.2.3. Sampling 17
2.2.4. Quantization 18
2.2.5. Windowing 19
2.2.6. Filtering 19
2.2.7. Human Speech 20
2.2.8. The Source/Filter Model 22
2.2.9. The Source of Speakers' Differences 22
2.3. Speech Signal Features 23
2.3.1. Phoneme Features and Classification 23
2.3.2. Energy 24
2.3.3. Frequency Analysis and Pitch Estimation 25
2.3.4. Cepstrum/MFCC 26
2.3.5. Liner Predictive Coefficients (LPC) 27
2.3.6. MPEG-7 Features 28
2.3.7. Speaker Specific Features (DNA) 28
2.3.8. Summary 30
2.4. Statistical Modelling 30
2.4.1. Gaussian Estimation 30

2.4.2. Gaussian Mixture Models 31
2.4.3. GMM Training and Adaptation 32
2.4.4. GMM training and adaptation implementation issues....33
2.5. Summary 34

Chapter 3: Research Method....35
3.1. Voice Activity Detection....35
3.1.1. Energy Based Estimator 36
3.1.2. Statistics Based Estimator....36
3.1.3. Supervised Methods 38
3.1.4. Summary 38
3.2. Speech Diarization 38
3.2.1. Single Channel Diarization 39
3.2.2. Multi-Channel Diarization 40
3.2.3. Voice Activity Detection for speaker segmentation 40
3.2.4. Speaker Segmentation 40
3.2.5. Distance Measures....42
3.2.6. Speaker Modelling 44
3.2.7. Speaker Clustering 44
3.2.8. One Step Diarization 45
3.2.9. Summary 45
3.3. Speaker Identification....45
3.3.1. Introduction to Speaker Identification....45
3.3.2. Speaker Modelling 47
3.3.3. Relevant Issues....49
3.3.4. Summary 53
3.4. NIST Speaker Recognition Evaluation 53
3.4.1. Task Description....53
3.4.2. Data Set description....54
3.4.3. Scoring Mechanism....55
3.5. Summary 56

Chapter 4: Experiments....57
4.1. Project Tools 57
4.2. Speaker Diarization 58
4.2.1. Evaluation Criteria....58
4.2.2. Testing Corpora....60

4.2.3. System Flow 60
4.2.4. Results 62
4.2.5. Summary 65
4.3. Speaker Identification 65
4.3.1. Evaluation Criteria 65
4.3.2. Testing Corpora 67
4.3.3. System Flow 67
4.3.4. Identification Results 70
4.3.5. Summary 84
4.4. Our SRE Participation 84
4.4.1. Task Selection 84
4.4.2. Application Overview and Task Allocation 85
4.4.3. Phone VAD 86
4.4.4. Interview VAD 87
4.4.5. DNA Training Data 88
4.4.6. Universal Background Model 89
4.4.7. Other issues 89
4.4.8. Summary 90
4.5. Summary 90

Chapter 5: Conclusion 91
5.1. Summary 91
5.2. Future Suggestions 92
5.2.1. SRE 92
5.2.2. High Dimensionality 92
5.2.3. Voiced Speech 93
5.3. Self-Reflection 93

References 94
Appendix A: Time Plan 98
Appendix B: Glossary 100

Word Count: 25,136.

List of Figures

Figure 2.1 (a) The time series and (b) frequency spectrum of a signal 17
Figure 2.2 (a) Sampling at frequency f=1/T and (b) Quantization of a continuous signal. 18
Figure 2.3 The effects of different window functions on a digital signal 19
Figure 2.4 Signal Spectrogram of the speech of the sentence "This is a test" (a male speaker) 21
Figure 2.6 (A) The time, (B) frequency and (C) quefrency spectrums for the sound /o/ of the word "boy" 25
Figure 2.7 the Mel overlapping triangular filter bank 27
Figure 2.8 a DNA. The shaded layer is the code layer. The shaded neurons are the speaker specific features. 30
Figure 2.10 An example of a one-dimensional GMM fitting the MFCCs. 32
Figure 2.11 GMM adaptation to produce a speaker's model. 34

Figure 3.1 A common flow of a speech/speaker processing algorithm. 36
Figure 3.2 (a) The time series and (b) energy based VAD results. 37
Figure 3.3 (a) The time series and (b) statistic-based VAD results. 37
Figure 3.5 A common flow of a speaker identification system. 47
Figure 3.6 JFA basic idea in the feature vector space. 50
Figure 3.7 the effect of normalization over scores (a sketch to introduce the idea). 52

Figure 4.1 The main process of segmentation evaluation 61
Figure 4.2 The threshold role in analysing the distance signal 61
Figure 4.3 The speaker models' first and second PCA. 62
Figure 4.4 ROC Curve for metric-based segmentation of a (a) TIMIT-based (b) NTIMIT-based Dataset 63
Figure 4.5 ROC Curve for metric-based segmentation of a (a) TIMIT-based (b) NTIMIT-based Dataset using Anchors model 64
Figure 4.6 ROC Curve for model-based segmentation of a (a) TIMIT-based (b) NTIMIT-based Dataset compared with KL modified 64
Figure 4.7 Different speaker recognition tasks performance 66
Figure 4.8 (a) Speaker Identification training processes 68
Figure 4.8 (b) Speaker Identification testing processes 68
Figure 4.9(a) The effects of using different GMM components number using MFCC-MAP approach (for different window sizes). 71
Figure 4.9 (b) The effects of using different GMM components number using DNA-MAP approach (for different window sizes). 71
Figure 4.10 The effects of using adaptive non-target cohort methods DNA-MAP approach (for different window sizes). 72

Figure 4.11 The effects of using different training data stream lengths when using DNA-MonoG approach (for different window sizes). 72
Figure 4.12 The effects of using different divergence measures when using DNA-Dist approach (for different window sizes). 72
Figure 4.13(a) The results of using MFCC-MAP approach. 75
Figure 4.13(b) The results of using MFCC-MAP approach when using a UBM. 75
Figure 4.13(c) The results of using MFCC-MAP approach when using a when using a large cohort. 75
Figure 4.13(d) The results of using MFCC-MAP approach when using a when using a small cohort. 76
Figure 4.14(a) the results of using DNA-Dist approach 76
Figure 4.14(b) the results of using DNA-Dist approach when using a UBM. 76
Figure 4.14(c) the results of using DNA-Dist approach when using a UBM when using a large cohort. 77
Figure 4.14(d) the results of using DNA-Dist approach when using a UBM when using a small cohort. 77
Figure 4.15(a) the results of using DNA-MAP approach. 77
Figure 4.15(b) the results of using DNA-MAP approach when using a UBM. 78
Figure 4.15(c) the results of using DNA-MAP approach when using a large cohort. 78
Figure 4.15(d) the results of using DNA-MAP approach when using a small cohort. 78
Figure 4.16(a) the results of using MFCC-MAP approach (with Normalization). 80
Figure 4.16(b) the results of using MFCC-MAP approach when using a UBM (with Normalization). 80
Figure 4.16(c) the results of using MFCC-MAP approach when using a large cohort (with Normalization). 80
Figure 4.16(d) the results of using MFCC-MAP approach when using a large cohort (with Normalization). 81
Figure 4.17(a) the results of using DNA-Dist approach (with Normalization). 81
Figure 4.17(b) the results of using DNA-Dist approach using a UBM (with Normalization). 81
Figure 4.17(c) the results of using DNA-Dist approach using a large cohort (with Normalization). 82
Figure 4.17(d) the results of using DNA-Dist approach using a small cohort (with Normalization). 82
Figure 4.18(a) the results of using DNA-MAP approach (with Normalization). 82
Figure 4.18(b) the results of using DNA-MAP approach using a UBM (with Normalization). 83
Figure 4.18(c) the results of using DNA-MAP approach using a large cohort (with Normalization). 83
Figure 4.18(d) the results of using DNA-MAP approach using a small cohort (with Normalization). 83
Figure 4.19 Different components of the proposed SRE system. 85
Figure 4.20 The results of different VAD methods on the DNA-Dist, DNA-MAP and MFCC-MAP approaches 86
Figure 4.21 The results of different divergance measures on the interview VAD. 88

Figure A.1 SRE Schedule 98
Figure A.2 Time plan that was followed during the project 99

Abstract

Most current voice activated tasks depend on short focused speech or commands from certain people. However, in most environments sound exists continuously. Besides, many of the current recordings are long multi-speaker recordings. Therefore, it is essential for computing devices to distinguish speech from other types of noise; and also to distinguish one speaker from another. Clearly, they need to take advantage of all speaker recognition techniques. Recently, some good advancement has been made in that field; For instance, it is now possible to determine the gender of the speaker with accuracy that matches the human perception of genders. Nevertheless, speaker identification systems are far from perfect. This research has the main focus of studying a novel speaker specific features set (DNA) that has shown great promise in improving speaker related tasks. Two major tasks are studied in details. Namely: *speaker diarization* and *speaker identification*.

This thesis presents the most widely accepted approaches for the mentioned tasks. Also, most approaches are adapted to fit the novel set of features as to improve the results of either task. The superiority of this set over previously best representations (i.e. MFCC and LPC) is presented along the study of the speaker segmentation task as part of the speaker diarization. A major limitation of the DNA is discovered when dealing with the speaker clustering task: the overlapping speaker models. Other issues of the novel representation, such as the high dimensionality and separability, are also discussed and solutions are presented in this thesis. The speaker identification poses another challenge which is also demonstrated and the findings are going to be used in a participation in a real life evaluation. Namely: the National Institute of Science and Technology (NIST) Speaker Recognition Evaluation (SRE) 2012.

Declaration

No portion of the work referred to in the dissertation has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Copyright Statement

i. The author of this dissertation (including any appendices and/or schedules to this dissertation) owns any copyright in it (the "Copyright") and s/he has given The University of Manchester the right to use such Copyright for any administrative, promotional, educational and/or teaching purposes.

ii. Copies of this dissertation, either in full or in extracts, may be made only in accordance with the regulations of the John Rylands University Library of Manchester. Details of these regulations may be obtained from the Librarian. This page must form part of any such copies made.

iii. The ownership of any patents, designs, trademarks and any and all other intellectual property rights except for the Copyright (the "Intellectual Property Rights") and any reproductions of copyright works, for example graphs and tables ("Reproductions"), which may be described in this dissertation, may not be owned by the author and may be owned by third parties. Such Intellectual Property Rights and Reproductions cannot and must not be made available for use without the prior written permission of the owner(s) of the relevant Intellectual Property Rights and/or Reproductions.

iv. Further information on the conditions under which disclosure, publication and exploitation of this dissertation, the Copyright and any Intellectual Property Rights and/or Reproductions described in it may take place is available from the Head of School of Computer Science.

Acknowledgment

As a researcher, I was honoured to conduct my first research under the supervision of Dr. Ke Chen; who has been a great mentor during all steps of the project. For his non-stopping support and interest in my work, I extend my most honest gratitude.

As a student, I was privileged enough to complete my studies in such a nourishing environment; made available by the university resources and its lecturers. Also, under a generous scholarship awarded to me by my Home University: *Damascus University*. For all those worked hard to get me where I am right now from both universities, I express my deepest appreciation.

As an SRE participant, I am fortunate to be part of the evaluation and work under real life conditions with a team of highly motivated members and high level university researchers. For those who imparted me knowledge, I am extremely grateful.

As a person, I am most lucky to be a member of such a loving and caring family. For each member of which I sincerely say *thank you*; you made me who I am and I owe you my life.

Chapter 1

Introduction

1.1. Primer

It has become well accepted that people interact with electronic devices using natural language, whether it is English or otherwise. With cutting edge technologies such as Apple's Siri® (speech recognition software for iPhone®) and Microsoft's Kinect® (gaming device for Xbox360® and windows-based platforms) it seems unavoidable for machines to understand human language. However, to realize that mechanism, it is essential to improve the accuracy of the speech directed applications even in the most ordinary tasks, such as deciding if a person is speaking at a certain moment or not, i.e. human voice detection.

The human speech is a signal containing mixed types of information; including words, feelings, language and identity of the speaker. It is up to the receiving party to decode the signal and extract needed information, possibly with different techniques for different pieces of information. In addition, Kemp et al. (2000) emphasises that many recognition tasks depend on recieving single-speaker speech signals; however, a great number of current sound streams are long multi-speaker streams; such as meeting and news broadcast data; also, it is impossible to build hardware that can only record human clean voice with no ambient effect or overlapped speech. On the other hand, the real time conduct is one of the most important features of human speech interaction. Therefore; effective, highly-accurate and time-efficient methods are necessary to deal with large amounts of speech information.

1.2. Project Scope

This project was initially aimed at the speaker diarization task; i.e. the task of determining which parts of a speech stream is uttered by which speaker. The main objective is to improve the current speaker diarization accuracy by investigating appropriate approaches. In particular, using

earlier research done by Chen and Salman (2011a; b) leading to the construction of a novel speech representation with the greatest speaker discrimination (discussed in section 2.3.7.). This representation is expected to improve the task at hand.

The initial segmentation results were extremely promising. Therefore, the research aims were extended to include speaker identification and verification tasks, which are related to the model-based segmentation. (section 3.2.6.)

In this report the newly introduced speaker specific representation (Chen and Salman, 2011a; b), i.e. DNA features, is presented in great detail. Moreover, different speaker related tasks are reported and tested using DNA features. In short, this research aims to study the impact of using the DNA features on said speaker related tasks. Additionally, the findings presented in this thesis are used as part of a participation in the National Institute of Science and Technology, Speaker Recognition Evaluation (*NIST SRE*) 2012. Our site is testing the novel technology in a real life evaluation, under real life conditions.

These objectives may be further decomposed into a number of smaller research targets, including:

- The Survey of the current state-of-the-art methods of different speaker related tasks.
- The study of empirical evaluation methods commonly used by the research community.
- The search for the statistical metric and speaker model which is most faithful to the speaker identity when using the speaker specific representation.
- Using the findings mentioned for improving the speaker diarization and speaker identification tasks.
- Conducting the required research for the SRE participation. Including the correct VAD; MFCC and DNA representation; and speaker modelling that can perform best using the novel representation.
- Incorporating the results in a full system to participate in the SRE 2012.

Firstly, the *diarization task* is thought of as a two-step process: (a) *Speaker Segmentation* where the speaker turn points are detected; and (b) *Speaker Clustering* where the different speech segments uttered by the same speaker are grouped together. Previous research has focus on extracting speaker-related information directly from the stream. What is more is that the new representation readily contains the most identity discriminative information. Therefore, a wise use of this representation must improve the overall accuracy.

Secondly, the *identification task* is usually done by modelling enough information about the said speaker (or speakers); and use that to identify the speaker of the testing stream.

The justification of any new technology is usually done empirically, i.e. contrasting some baseline generally-approved approaches. Even if the same methods are used with the new representation, it is generally required for the new technology to prove its superiority in a real-life or a real-life mimic system (e.g. SRE).

We believe that using the DNA will enhance the performance of any text-independent speaker related task. This claim is made because of the performance of the representation obtained in the segmentation task. This proposition will be empirically assessed through a number of well-designed tests using the evaluation metrics presented (see chapter 4).

The project can be categorized under speech processing or digital signal processing. In addition to speaker diarization and identification, the project includes tasks such as voice activity detection. Another way for looking into the matter is whether the processing is done online (where only information from previous parts of the stream is available) or offline (where the information is used regardless of where it is found in the stream); i.e. studying the temporal effects. And therefore, online (similarly offline) algorithms are parts of the project as well.

One clear challenge is to find a good underlying model that faithfully represents the DNA features. Other faced challenges include the separability of the data, the high-dimensionality of the DNA feature vector and the generalization ability of one generic DNA structure.

1.3. Thesis Structure

In this **Introduction**, the major objective of the project is discussed and the main reasoning behind it is presented.

The **Background** chapter illustrates the background knowledge needed for the speech processing done in the project. Starting with sound, signals and speech representation; section 2.2 explains the basics of human speech and speech processing. It also covers one of the fundamental models, the *source/filter model*, and then explains the main differences in speakers' sound and corresponding signals. Section 2.3 discusses the most important features of a speech signal used in any speech related application, such as the frequency envelops, MFCCs and LPCs. It also describes the speaker specific representation (DNA), which is then heavily used. Section 2.4 offers the fundamentals of statistics modelling; including Gaussian Mixture Models (GMM). Later in the report those models will be used to model different feature vectors.

Parts of this chapter were presented earlier in the progress report of this MSc project. The progress report was written by the same writer and presented in May 2012. The chapter is reproduced in this dissertation, and enriched with more relevant information.

The **Research Method** chapter of the report demonstrates what the project involves as a research. Section 3.1 speaks about *voice activity detection*, how it is generally done and the reasoning behind the methods and evaluations. Section 3.2 presents different approaches used for *speaker diarization*; including evaluation techniques. Section 3.3 continues with the *identification task*. Section 3.4 describes the SRE different tasks. Later, experiments will be done for each of those tasks separately.

The **Experiments** chapter reports the empirical results of the conducted research. Experiments are explained and their results stated. Section 4.1 explains the tools used when conducting the research. Section 4.2 is concerned with the speaker diarization task. Section 4.3 is devoted for the speaker identification task. Section 4.4 describes our SRE 2012 participating system and the final settings used.

The **Conclusion** chapter includes a summary of the done work along with a view of the suggested future work.

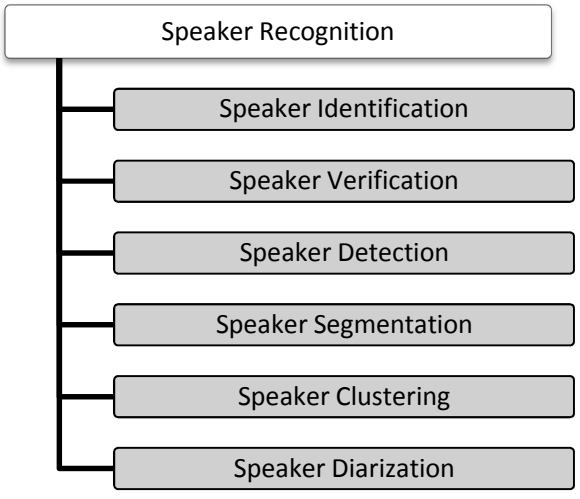
Chapter 2

Background

This chapter contains a very broad background about: a) speaker recognition and the subtasks under this wide title; b) sounds, speech, sound digital representation and manipulation; c) the most commonly extracted speech features and also the speaker specific features (DNA); and d) statistical modelling methods.

2.1. Speaker Recognition Tasks

This section attempts to give general definitions regarding different Speaker Recognition Tasks.



Speaker Recognition (SR) is a major topic which encompasses many different speaker-specific tasks. According to Reynolds (2002), the tasks can be sub categorized into text-dependent (where speakers are expected to utter a certain piece of text) and text-independent (where the speaker may speak anything they wish) tasks.

Similarly, depending on the information that the method is allowed to use and the output expected from the process; speaker recognition generally enfolds the following tasks (Doddington, 1985):

- *Speaker Identification*: A closed set of speakers is presented to the system along with a piece of testing data. The system must decide who, among the available set of speakers, had uttered the testing data. This is often referred to as *Closed-Set Identification* to avoid confusion with the verification task, or more conveniently speaker identification.

- *Speaker Verification*: Two pieces of speech are presented to the system. The system must decide whether the same speaker uttered both segments or different speakers did. This is often referred to *Open-Set Identification*.

Campbell (1997) adds the following task under the SR umbrella:

- *Speaker Detection*: One speaker's data (often called the target speaker) is offered to the system along with many testing speeches. The system is expected to correctly flag the speeches of the target speaker.

Other tasks are also related to Speaker Recognition, as they are considered of the same family of research (Kotti et al, 2008):

- *Speaker Segmentation*: A large input stream, with more than one speaker present, is offered to the system. The system is expected to find the points where the speaker changes; i.e. turn points. If knowledge about the speakers is available a priori then the system can build models for each speaker. Then the task is called *model-based speaker segmentation*. Otherwise, it is called blind speaker segmentation, or *metric-based speaker segmentation*.
- *Speaker Clustering*: A large number of utterances are presented to the system. The system must correctly cluster them according to the speaker. This task is often done online, alongside another task, as to group segments of the same speaker together.
- *Speaker Diarization*: A stream is presented to the system. The system is expected to decide who is speaking at each period of the stream. This task is often thought of as segmentation of the stream followed by clustering. Similar to the segmentation task, if knowledge is available a priori to the system then models can be built (which helps in the online clustering as well) and the task is called *model-based speaker diarization*.

2.2. Sound and Digital Signal Processing

This section describes the background knowledge regarding speech digital representation and manipulation. It also provides one of the most fundamental models for human speech, the source/filter model. Towards the end, a brief discussion about statistical modelling is laid out. Most information presented in this section is deeply discussed in (Jurafsky and Martin, 2009), (Coleman, 2005), (Huang, Acero and Hon, 2001) and (Rabiner and Bing-Hwang, 1993).

2.2.1. Sound

Sound is formed when a medium vibrates, for instance, the vibrations of the diaphragm in a computer speaker or simply the vibration of air molecules or pressure in the air. These vibrations can be modelled as two types of layers, interleaved, travelling together through the medium;

higher pressure layers (molecules compressed more than normal) and lower pressure layers (molecules relaxed more than normal). Those vibrations, when in the air, affect the ears causing people to hear. In fact, sound can be perceived as a signal; the amplitude of which corresponds to the pressure change and the length of which corresponds to the distance between two consecutive high (or two consecutive low) pressure layers.

Human speech is one form of sound; which people have developed through time to carry valuable information for communication, such as thoughts and feelings. However, it also carries other inherited characteristics such as the speaker identity, language, dialect, gender and mood.

2.2.2. Signals

A signal is the continuous measure of a quantity in terms of time. An example of a signal is the measured voltage of a certain point in an electric circuit. A signal that repeats itself every period T is called a *periodic signal*; with the value T being its *period*. The number of times that the signal repeats itself in a time unit, i.e. one second, is called the *frequency* (mathematically, the inverse of the period). An example of a periodic signal is the sinusoid given in equation 2.1:

$$x(t) = \sin(t)^{1} \tag{2.1}$$

Often signals are additive in the time domain. According to Coleman (2005, p.72) Jean Fourier provided the means where any real world signal can be represented as the sum of (maybe infinite) a number of sinusoids with different frequencies, phases and magnitudes. This technique is straightforwardly called *Fourier Transformation*. Fourier Transformation has only one condition: the original signal must be finite in length and amplitude at any time. This transformation generates a new representation of the signal in the *frequency domain*[2]. Where each frequency is associated with a value representing how much the sinusoid of that frequency contributes to the original signal.

It is common to study the *frequency spectrum* of a signal, especially a speech signal (which is composed of many mixed frequencies, caused by the shape of the human vocal tract). Figure 2.1 (a) shows the time series for the sum of two sinusoids with frequencies 100 and 220 Hertz; with some artificially added noise. It is very hard to observe the frequencies in the time domain. Figure 2.1 (b) is the result of Fast Fourier Transformation for the signal in (a). In (b) it is easier to locate the base frequencies of the mixed signal.

[1] This is the simplest form of a sinusoid. The general form is given: $x(t) = A.\sin(wt + \emptyset)$ where A is called the *Amplitude*; w is the *Angular Velocity*; and $\emptyset$ is the *Phase*.

[2] In practice, the phase component is represented in the complex part of the result. Therefore, it is often discarded and only the frequency component (the real component) is kept.

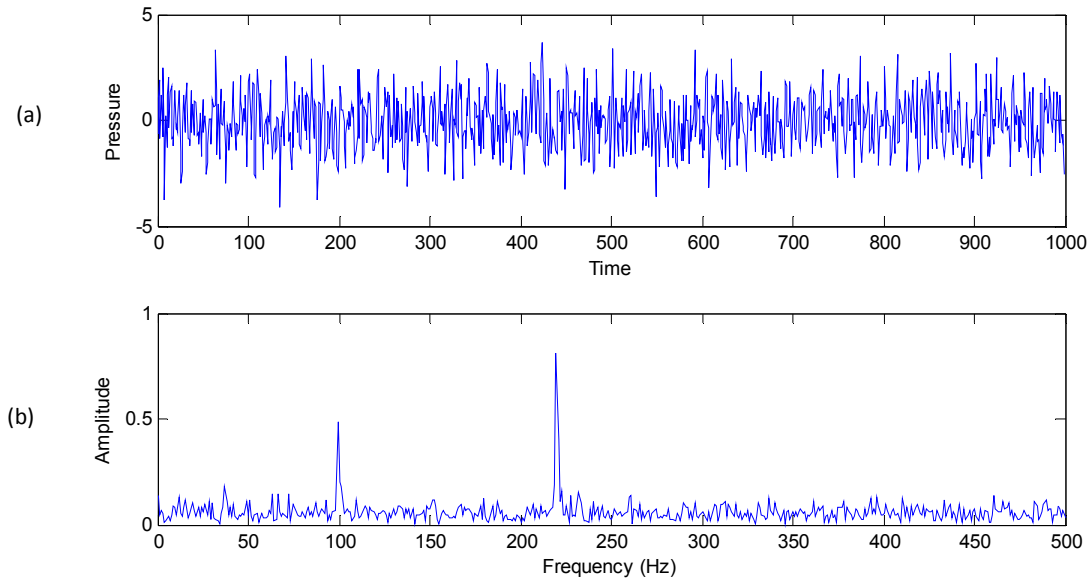


Figure 2.1 (a) The time series and (b) frequency spectrum of a signal

2.2.3. Sampling

The continuous nature of the sound signal can be captured through analogue devices, such as microphones. However, for the digital processing it is impossible to retain this continuity. The only possible way is to keep measures at distinct times. The process of retrieving those measures at equal intervals is called *sampling* and it is usually carried out by an *analogue-to-digital (AD)* convertor. The time period between two consecutive measures is called the *sampling period* T_s. Similar to an ordinary signal, the number of samples taken in one second is called the *sampling frequency* F_s. Figure 2.2(a) shows sampling of an analogue signal at *frequency* $F_s = 1/T_s$.

Evidently, the sampling processes may case losses in the original signal. For example, if only one sample was kept for each period of a sinusoid (i.e. $F_s = F$); then a flat signal would be perceived. However, keeping two samples of each period of the previous sinusoid may be enough to encode the whole signal, e.g. if the samples were taken at both maxima.

Nyquist Frequency is the highest frequency of a signal that can be faithfully preserved when sampled at a certain frequency F_s; after which the original signal may be distorted, i.e. *spatially aliased*. It can be proved that Nyquist Frequency is ½F_s (Jurafsky and Martin, 2009).

Lastly, the human auditory system has a certain frequency range of audible sounds. This range differs from person to person. Nonetheless, in general people cannot hear frequencies higher than 22,000 hertz. Therefore, in practice it is common to have sampling frequencies around F_s = 44,000 hertz. Conversely, the telephone circuit uses F_s = 8,000 hertz and therefore it distorts the original sound.

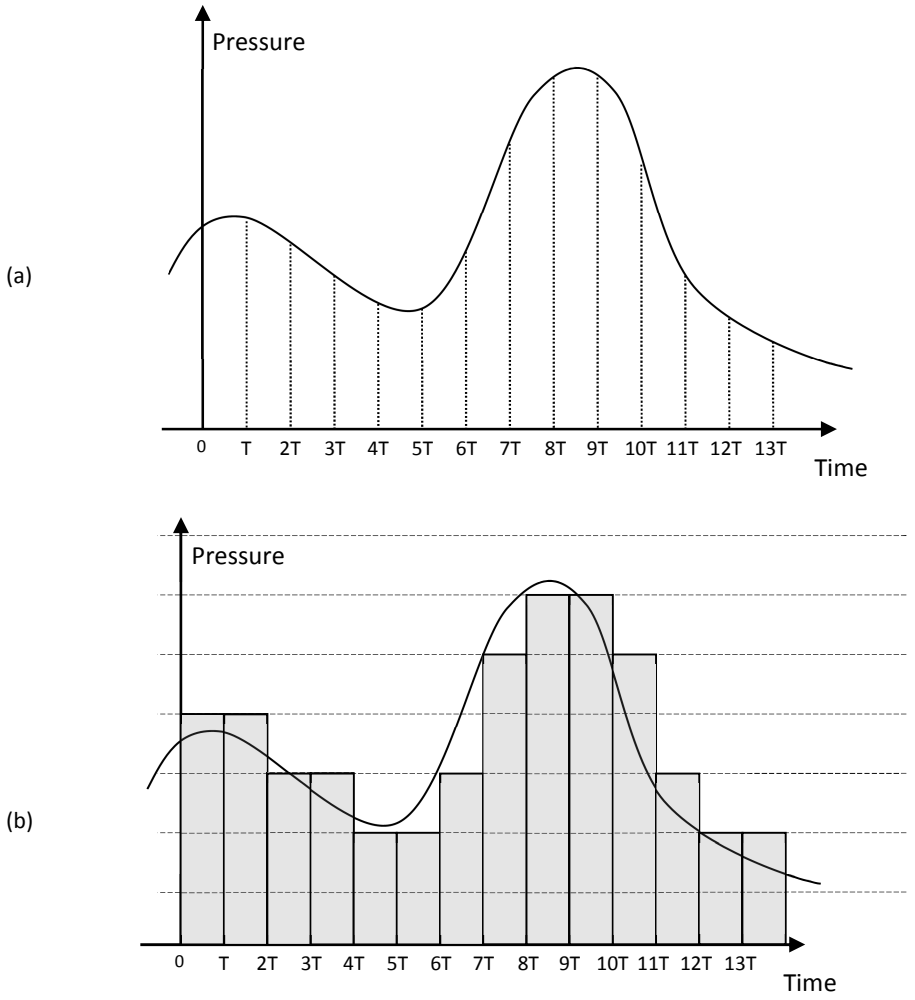


Figure 2.2 (a) Sampling at frequency f=1/T and (b) Quantization of a continuous signal.

2.2.4. Quantization

The sampling process produces a train of distinct real values $x[n]$; each representing the magnitude of the signal at time nT_s. The domain of these real values is continuous and cannot be represented by a digital machine. Therefore, an approximation of each sample value to one of distinct levels is made; and the value is then stored as the level number in a number variable. The more levels are used the closer the approximated value to the real value (the smaller the approximation error) is; then again the more space a single sample will consume. Figure 2.2(b) shows the signal sampled in figure 2.2(a) and the result of quantization as the shaded area.

The approximation errors caused by quantization are irreversible. Therefore, it is often the case of using more quantization levels than needed. In practice, it is common to use 16 bits per sample, i.e. 65536 levels. Finally, manipulating the digital signal is done by manipulating individual samples. For instance, doubling the values of a range of samples would mimic doubling the sound volume in that range.

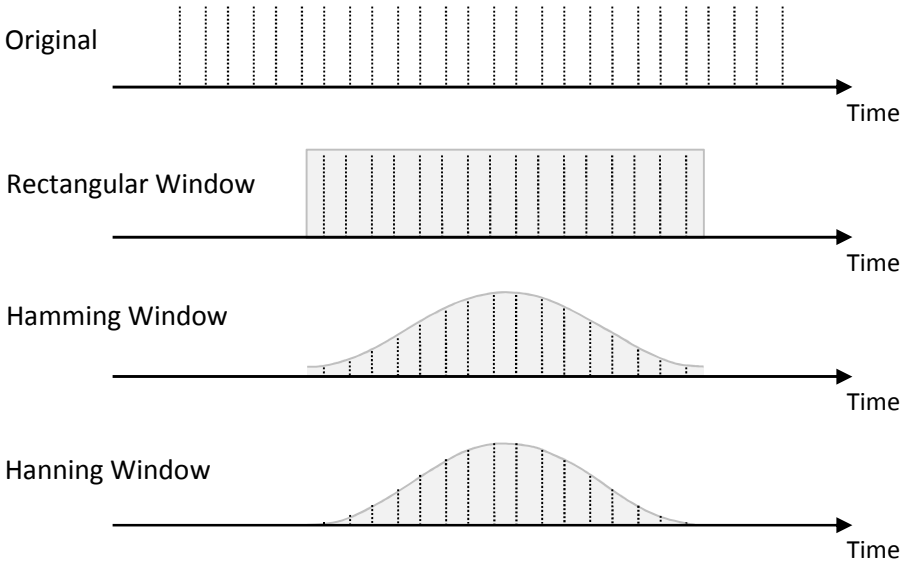


Figure 2.3 The effects of different window functions on a digital signal

2.2.5. Windowing

Window functions are used as a temporal bound on the original signal to limit the stream on the interesting range. The most common window function is the *rectangle function*. The effect of this window on the original signal is to produce a time bounded signal that is similar to the original signal inside the rectangular range and is set to null outside of it. This helps further processes to attend only to the bounded part. Other window functions sustain more attention to the central values of the window rather the boundary values. Examples include *Hamming*, *Hanning and Triangular window* functions. It is common to use overlapping windows over the same signal, i.e. one starts before the previous one ends in the time domain. Each window will have an extract of the original signal without any interference among the result windows.

In digital signal processing, windowing is thought of as a function affecting the samples of the signal to produce a new series of samples, i.e. a new signal. Figure 2.3 shows the effects of different windowing functions on a digital flat signal.

2.2.6. Filtering

A filter affects the signal in the frequency domain; i.e. it alters the components of the signal at specific frequency ranges. For instance, an ideal low pass filter only keeps low frequency components and clears any component over a certain frequency step (called the *cut-off frequency*). Just like windowing, filtering a signal will result in a new signal with sample values affected by the filter. The simplest way to build a digital filter in practice is by the use of the *difference equation* (Coleman, 2005, pp.54-60); which determines the value of the output sample as a linear combination of previous input (and/or output) sample values, as presented in equation

2.2. An example would be the *Averager*; i.e. averaging the past N input samples, which will result in a signal less influenced by quick changes (or high frequencies) of the original signal, mimicking a low pass filter. Another example is the *Differentiator*; i.e. averaging the difference of the past N input samples, which will result in a signal less influenced by the slow changes (or low frequencies) of the original signal, mimicking a high pass filter. In general the following equation is used (x is the input signal and y is the output signal):

$$y[t] = b_0x[t] + b_1x[t-1] + b_2x[t-2] + \cdots + b_kx[t-k] \\ +a_1y[t-1] + a_2y[t-2] + \ \ldots + a_jy[t-j] \tag{2.2}$$

Using the different values for the coefficients (a's and b's) results in different filter behaviours. The resulting filter is known as the *Butterworth filter*. In practice, these values are retrieved (or estimated) from lookup tables according to the ratio of the cut-off frequency to the sampling frequency of the input signal. In addition, combining two filters consecutively allows the build of a pass band filter (only passes a certain range of frequencies).

Filters are often described by their impulse response; i.e. the output corresponding to a single *Dirac impulse* $\delta(0)$. On one hand, if the output is finite then the filter is called *Finite Impulse Response (FIR) Filter*; and according to Coleman (2005, p. 60) the corresponding difference equation would only reference the input signal. On the other hand, if the output is infinite in power or in time then the filter is called *Infinite Impulse Response (IIR) Filter;* and the corresponding difference equation would reference both the input and output signals.

A *filter bank* is a collection of filters that covers the entire range of frequencies found in the original signal. Each filter emits a different signal; which can be further processed individually. An example is the sound equalizer, where each frequency is boosted separately; afterwards a final sound signal is produced by merging the results of the boosted signals.

2.2.7. Human Speech

The *vocal tract* is the part of the human body responsible for producing voice. It mainly consists of three cavities; the *pharynx cavity*, the *mouth cavity* and the *nasal cavity*; participate to shape the final voice. As a person exhales, air is pushed out of the lungs and through the *voice box*, where it either causes the *vocal cords* to vibrate or pass between them directly to the mouth cavity. When the vocal cords vibrate, they vibrate periodically at a base frequency known as the *Fundamental Frequency* (f_0). Sounds that excite the vocal cords are called *voiced* sounds. For example, all the vowels are voiced. On the other hand, if the vocal cords are extra relaxed or extra stiff, the air pass between them and the result sound is said to be *unvoiced*, for example the sound /f/. Another case is when the vocal cords are closed; then air pressure builds up behind

them and then is released at once causing a quick erupt sound. This type of sound is called a *Stop*. Stops may be either voiced (such as /d/) or unvoiced (such as /t/) depending on what happens to the vocal cords after the air escapes. Cavities change their shape by the aid of the tongue, teeth, lips, and the nasal tissues which effectively change the final sound. More details are presented in (Rabiner and Bing-Hwang, 1993, pp.14-17).

The ears are the part of the human body that is affected by sound signals. The air vibrations affect the eardrum, which in turn vibrates and causes other bones to vibrate. The higher the pressure change of the air the louder the perceived sound is. *Pitch* is the perception of the fundamental frequency. The ear can only be excited by a range of frequencies; on average, an adult can hear frequencies up to 20,000 hertz. Moreover, the ear is more sensitive to certain frequencies than others. Mel scale captures this phenomenon and is described in section 2.3.4.

The smallest part of the spoken language is called a *phoneme*. Much effort has been made to list all the phonemes a person can articulate; and also to write every word of a certain language phonetically. The most common notation is the *International Phonetic Alphabet (IPA)*. For example, the English word "fish" can be phonetically written as /fɪʃ/. Different phonemes have different characteristics that can be used to recognise them. The most observable feature of phonemes is whether it is voiced or unvoiced, which comes from the existence of the base frequency or not.

The frequencies of one's speech change dramatically over time (see 2.2.8. for more details). It is not wise to do frequency analysis over a long time of speech because it will produce wrong frequency components. Therefore, a small sliding window (<20 ms) is often analysed. The result is a 3D diagram (Time – Frequency – Intensity). A special diagram called the *spectrogram* is used; i.e.

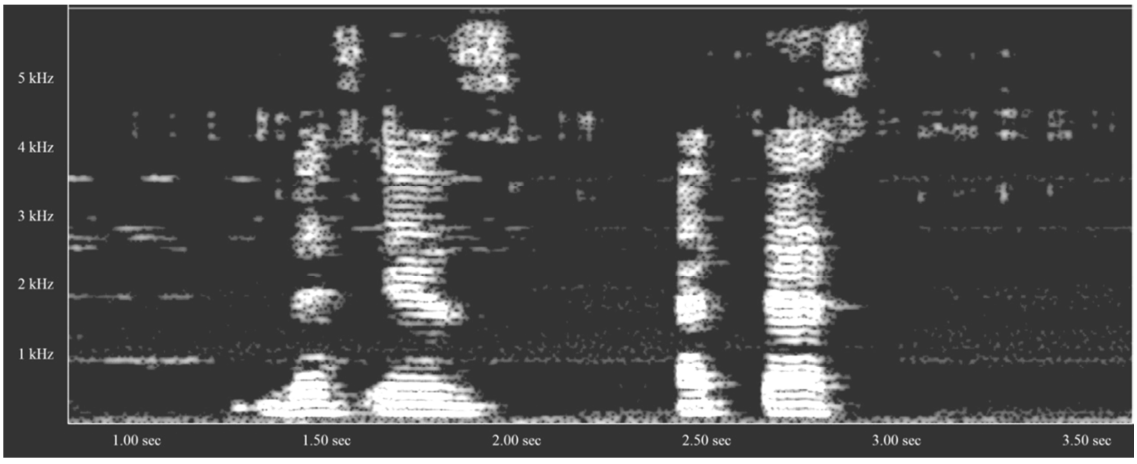


Figure 2.4 Signal Spectrogram of the speech of the sentence "This is a test" (a male speaker)[3].

[3] Figure is generated by the a free tool "*Spectrogram 16*" [online] Available at: <http://www.visualizationsoftware.com/gram> [Accessed 24 March 2012]

a 2D plot to represent the Time-Frequency dimensions, with the intensity represented by the intensity of the colour. An example of spectrogram is presented in Figure 2.4. The window used is 2ms. The speech signal did not exceed 6000 hertz at any given window.

2.2.8. The Source/Filter Model

The architecture of the vocal tract has influenced a *source/filter model* of human speech. The source is a simple base-frequency (pitch) signal generator that is used for voiced phonemes. This signal passes through the cavities of the voice tract. Much like any physical tube enclosing a signal; the different cavities' shapes cause a raise of the effect of certain frequencies and a drop of the effect of others. The result voiced sound is different because of the filtering against different frequencies that takes place in all the cavities. For example, the difference between the vowels /æ/ in "cat" and /u/ in "boot" is only the shape of the lips. This effect can be modelled using a filter bank applied to the base-frequency signal to produce the final phonemes. Different behaviour on different frequencies (i.e. different filtering) is a key feature for the detection of voiced phonemes. This is why the speech detection process (both in human and digital systems) relay heavily on voiced phonemes (Rabiner and Bing-Hwang, 1993, p.24).

This model is the basis for many speech synthesizers such as *Klatt's formant synthesizer*; presented in Klatt (1980) and further discussed and its parameter set in Coleman (2005, pp.62-68). Clearly, different banks are used for different phonemes. Hence, there will be a great number of parameters to control the synthesizer. This resembles the different positions and shapes cavities can take to shape the final voice.

2.2.9. The Source of Speakers' Differences

The *frequency of the vibrating vocal cords* (the fundamental frequency) is the source of most differences between voices. It can be used for some general classification such as gender (male voices usually have lower frequency than female voices) and age (the fundamental frequency drops with age). The *full length of the vocal tract* and the ability to change its shape is also a factor of differences between voices.

In a phoneme-based study we find differences in *class and dialect* between different people. Those changes affect the places of stress and the syllables used (Campbell, 1997). Also a source of difference is the number of uttered syllables in a period of time or *rate of speech* (Jurafsky and Martin, 2009). That also affects the speaking style of the person, such as the deletion of the last syllable of a word, the reduction of the stress in some cases and merging words for convenience (Huang, Hon and Reddy, 2001). Moreover, the same person speech may be affected by their mood, such as repetition, whispers and yells.

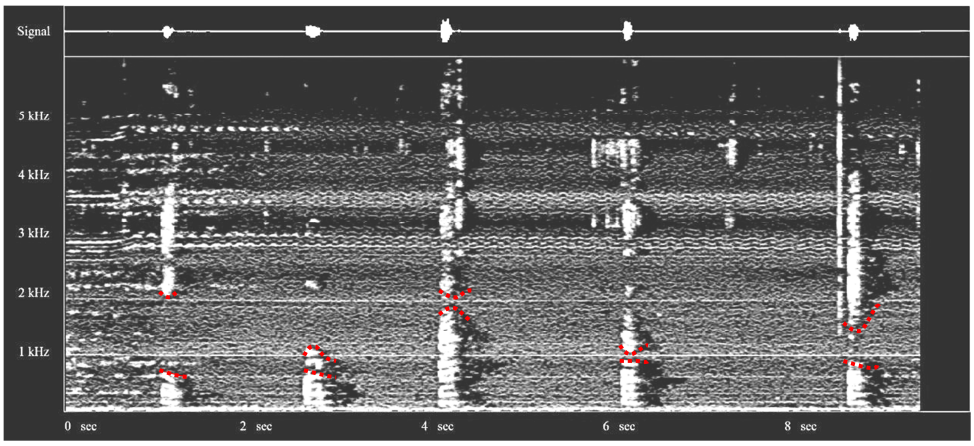
2.3. Speech Signal Features

As seen before the human voice signal mixes different types of information. However, this information is hidden in a series of sample values available to the digital process. Therefore, features are extracted from the signal first and then they are handled for certain purposes. In this section, generic speech features are presented, followed by the description of the most widely used features for speaker related tasks, i.e. MFCC features. Later in this section, speaker specific representation is described, which I will be focusing on afterwards.

2.3.1. Phoneme Features and Classification

A spectrogram of one voiced phoneme shows the boost of a certain set of frequencies. This boost is present regardless of the speaker. Figure 2.5 shows 5 vowel phonemes and their respective spectrograms. It is clear that each vowel has a set of boosted frequencies (The colour intense areas). The visual boundaries of these frequencies (the frequency of change from damp to boost and vice versa) are called *the formants*. A phoneme can have up to 5 formants called F_1 to F_5. All voiced phonemes share the lower frequencies (around the fundamental frequency F_0); then they differ on higher frequencies. For example, in figure 1 we can see the first two formants for the vowel /i/ (in "eve" /iv/) around $F_1 \approx 800hz$ and $F_2 \approx 2100hz$. Using the first two formants, a classification of the voiced vowels can be set. In practice, this classification is speaker dependant and not flawless (Rabiner and Bing-Hwang, 1993, p.27). According to Jurafsky and Martin (2009); other features of the voiced phonemes may help the synthesis of speech, such as the position of the tongue (High/Low, Front/Back) and the shape of the lips (Round/Flat).

Phonemes uttered completely from the nasal tract, such as /ŋ/ in "king" /kɪŋ/, are called *Nasal*. Whereas, Phonemes uttered completely from the mouth tract, such as /t/ in "tip", are called *Plosive*. A Phoneme that causes friction in either the mouth or the nasal cavities is called *Fricative*; for example, /ʃ/ in "fish" /fɪʃ/. Fricatives can be voiced (/v/ in "voice") or unvoiced (/Θ/ in "teeth"). *Stops* happen when the vocal cords momentarily close, causing air pressure to be built and then released as one utterance. Stops can be voiced (/g/ in "get") or unvoiced (/k/ in "kit"). Other phonemes are harder to classify, such as the sound /w/ in "wit". They are called *Semivowels* if they are close in sound to some vowel. Other than that they occupy their own category such as the only whisper phoneme /h/ as in "hard". This classification is further refined in (Rabiner and Bing-Hwang, 1993, pp.21-37) and (Huang, Acero, Hon and Reddy 2001, pp.39-47).



Vowel	/i/	/u/	/ə/	/ɔ/	/æ/
Example	Eve	Boot	Ago	Bought	Cat

Figure 2.5 The first two formants of the presented vowels.

2.3.2. Energy

The sum of the values of the samples over a period of time (in a rectangular window) is called the window's *Energy* (Coleman, 2005, p.48).

$$Energy = \sum_{n=t_1}^{n=t_2} x[n] \tag{2.3}$$

Clearly, the energy increases with the size of the frame. Therefore, it holds no inherited significance by itself. On the other hand, *Power*: the average of energy over the length of the window, does.

$$Power = \frac{\sum_{n=t_1}^{n=t_2} x[n]}{t_2 - t_1} \tag{2.4}$$

Root Mean Square (RMS) is a more used measure because it is not affected by the signs of the individual values.

$$RMS = \sqrt{\frac{\sum_{n=t_1}^{n=t_2} (x[n])^2}{t_2 - t_1}} \tag{2.5}$$

In speech processing, these measures have different uses. For example, when a window has RMS close to zero, it suggests that it must be a silence frame. Also, a sudden change in the RMS for a short period suggests an irregular event taking place at that window; such as a knock on the door or a phone ring. Furthermore, if the values stabilize after the change, it may indicate a speaker change.

The *zero-crossing rate* is the rate of the signal sign change. It can be easily approximated by the use of the power over small consecutive windows. This feature is of great importance in

distinguishing different types of voices, such as different types of music or even human speech (Gouyon and Pachet, 2000).

2.3.3. Frequency Analysis and Pitch Estimation

The fact that each voiced phoneme has certain frequency features (see section 2.3.1) influenced extending the signal study to the frequency domain. That is done by the use of Fourier Transformation. A good digital implementation for the discrete version is the *Fast Fourier Transformation* FFT, which can be made to windows of lengths 2^N; if the length is not a power of 2, the window is zero padded. An inversion function is also available, called *Inverted FFT.* Since the results of mentioned transformations are in the complex numbers domain $\mathbb{C}$, it is often the case that applying the Inverse FFT to the frequency spectrum of a signal will not retrieve the same original signal.

Typically, the voiced phonemes hold more energy in lower frequencies than unvoiced phonemes; where the energy is spread all over the spectrum (Coleman, 2005, p.86). This fact can be used to estimate voiced/unvoiced regions. Noticeably, this method needs a threshold to discriminate between voiced and unvoiced regions. Another better way of voiced region estimation is *autocorrelation* where a window of the signal is auto-correlated with itself at growing intervals. The interval with the highest autocorrelation is said to be the base period and can be used to determine the base frequency F_0. If the highest autocorrelation was at period 0 then the frame is deemed unvoiced. Parameters for this method include the lower and upper bounds for the autocorrelation process; for human voice, good values are 40 hertz and 600 hertz (Huang, Acero and Hon 2001, p.325). Figure 2.6(A) shows the time series of the /o/ sound of the word "boy" for 90ms sampled at 62K hertz. It clearly shows the voiced nature of the vowel phoneme. The frequency spectrum is presented in (B) up to 4096 hertz.

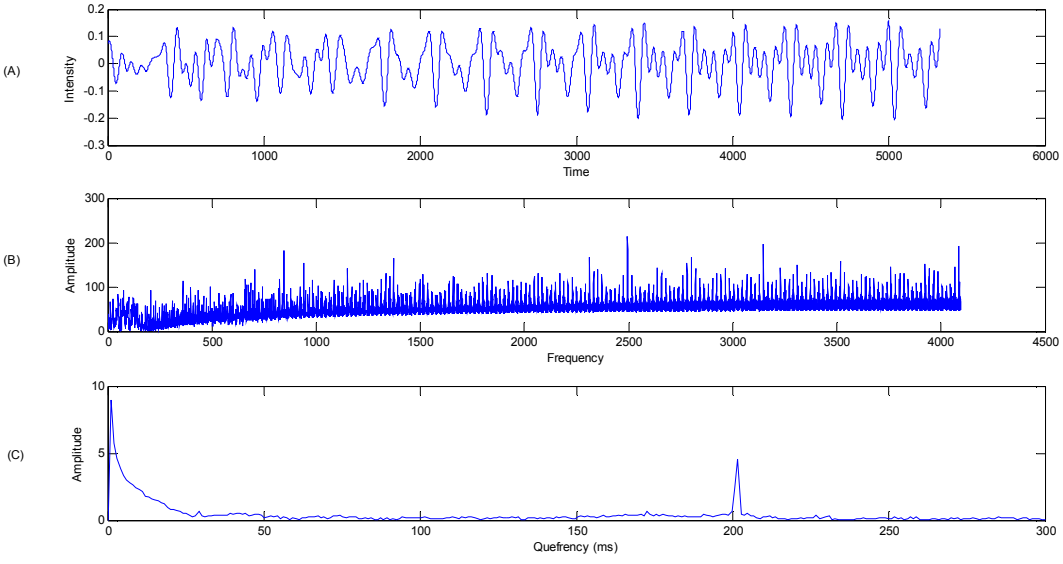


Figure 2.6 (A) The time, (B) frequency and (C) quefrency spectrums for the sound /o/ of the word "boy"

2.3.4. Cepstrum/MFCC

As described in Coleman (2005, p.78) the frequency spectrum (such as the one in Figure 2.6(B)) contains two types of oscillations. 1) The small more frequent ones; called *harmonics*; their frequency is of a multiple of the base frequency. Therefore, they are related to the speaker. 2) The large hidden ones, where the oscillations are amplified; their frequency is affected by the other formants of the speech. For a speaker related task, one may be interested in the rate of harmonics' repeat; i.e. the frequency of them. Therefore, FFT is applied on the frequency spectrum to analyse its oscillations. The result is in the samples' domain, or what is called *Cepstrum domain*. The x axis represents samples called *quefrency.* The amplitude reflects the amount by which the quefrency value contributes to the frequency spectrum. The quefrency which has the highest amplitude (and is reasonably high in value) corresponds to the harmonics frequency and also to the base frequency of the original signal[4]. An example is presented in figure 2.6(C).

Directed by a field testing of human perception of different frequency; the *Mel-Scale* handles frequencies below 1000Hz with a linear behaviour; whereas frequencies over a 1000Hz are handled logarithmically; hence giving more importance to lower frequency changes. Handling signals on the Mel-Scale can be made by the use of a filter bank with equal length regions below 1000Hz and log increasing length regions afterwards. Often, the filters used are 24 overlapping triangular filters. To produce MFCCs those filters add up the energy within (from the frequency spectrum); each will produce a single value called the *Mel value*. The logs of the Mel values are introduced to the FFT again (more formally to the *Discrete Cosine Transformation* DCT; which is the FFT in real domain). The result is called the *Mel-Frequency Cepstral Coefficients MFCCs*. Figure 2.7 shows the Mel-Scale step. The MFCCs reflects the quefrency behaviour and therefore encodes speaker related information. They reduce the effects of the source of the original signal; and keep only the effects of the filter (as in the source/filter model). Moreover, the Mel-Scale step provides more significance to the lower frequencies, and hence conveys the fact that humans are more influenced by changes at lower frequencies than changes at higher ones (Jurafsky and Martin, 2009).

The MFCC features have provided good results in both speaker and speech recognition tasks. Often only the leading L numbers are kept for each frame, such as 12 or 20. Also, the *delta MFCCs* are the difference between MFCCs of two consecutive frames; and the *delta-delta MFCCs* are the difference between delta MFCCs of two consecutive delta frames. The deltas have also shown good performance in the recognition tasks.

[4] There is no guarantee that this is F_0. However, it is in relation with F_0, hence it can be used as F_0 estimate.

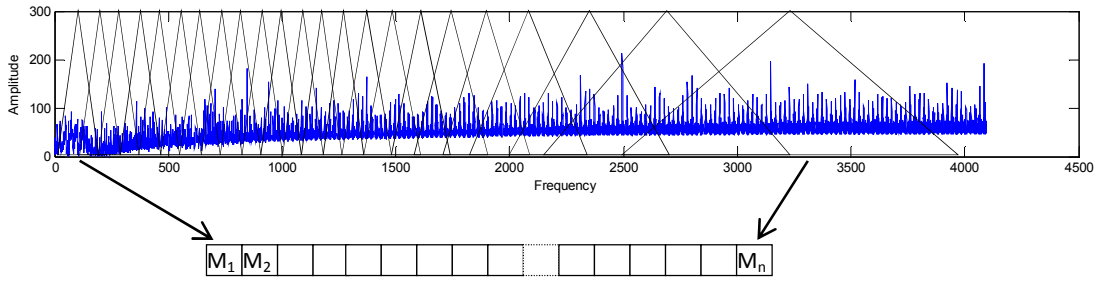


Figure 2.7 the Mel overlapping triangular filter bank

2.3.5. Liner Predictive Coefficients (LPC)

An approximation of the speech signal sample can be made by a linear combination of the p previous values:

$$\tilde{x}[n] = \sum_{i=1}^{p} a_i . x[n-p+i] \tag{2.6}$$

Moreover, if the residuals (i.e. the error $e[n] = x[n] - \tilde{x}[n]$) are saved then the full input signal can be retrieved by first computing $\tilde{x}[n]$ and then adding the residual. This fact has motivated a compression technique for speech signals known as *Liner Predictive Coding* (LPC). The error signal is stored for each frame of the signal; as well as the coefficients a_i. The compression is achieved because the error signal amplitude is smaller than the original signal amplitude. However, if p is small then the error values will be large and the compression will fail. On the other hand, if p is large then the required space for the coefficients will be considerably large and the compression will also fail. Either way, the coefficients a_i are best selected to minimise the mean squared error value for the approximation. To do so, the original signal is auto-correlated with itself to generate an auto correlation matrix, from which the coefficients can be extracted. The method is described in details in (Rabiner and Bing-Hwang, 1993, pp.97-106) and in (Huang, Acero and Hon, 2001, pp.290-300).

A simple observation of the LPC values suggests that the LPC values do not change dramatically unless a sudden event occurs, such as change in the frequency of the speech. Using Fourier Transformation on the error signal generates the *Line Spectral Frequency* (LSF) spectrum; which can be used to estimate the changes of base frequency and formants. It can be shown that they are more stable than other methods for F_0 estimation (Huang, Acero and Hon 2001, p.305).

Another use of the LPC is linked to the source/filter model; where the errors can be viewed as the distortion happening for a signal. Either signal (the source or the filter) can be linearly approximated by the previous p values of the original signal. This approximation generates a list of pairs of values (source/filter pairs) that can be further used for analysis of the signal. The list of pairs is called *Line Spectral Pairs* (LSP). Another interesting idea is *Perceptual Linear Predictive*

(PLP) presented in Hermansky (1990). PLP performs the Mel-Scale step on the input signal so to boost the effect of lower frequencies before extracting the LSF or LSP.

2.3.6. MPEG-7 Features

This standard is developed by the Moving Picture Experts Group and is standardized by the International Standards Office in (ISO 15938-4:2001). It encodes many multimedia types including audio. The standard provides the structure of the media file, along with some low-level features embedded in the file or easily extracted from the data. Additionally, the standard describes some high-level functionality for application use.

Features are many; I present the most used ones in their defined categories:

- *Basic*: the structure lists the sampled values.
  The *AudioWaveform* describes the minimum and maximum values of the time series.
  The *AudioPower* which provides a smoothed version of the samples for quick reference.
- *Frequency Spectral*:
  AudioSpectrumEnvelope the frequency analysis over the whole signal.
  AudioSpectrumCentroid and *AudioSpectrumSpread* describe the centre of gravity of the frequency spectrum and its spread respectively. The spread helps in distinguishing pure sound from noise.
- *Signal Parameters*:
  AudioFundamentalFrequency which is an estimation of the pitch.
  AudioHarmonicity describing the harmonicity of the signal, which is used to differentiate harmonic sounds (e.g. music) from inharmonic sounds (e.g. bell) from non-harmonic sounds (e.g. noise).
- *Silence Segments*: Tagging silent frames of the signal.

2.3.7. Speaker Specific Features (DNA)

A novel approach for representing speech retaining as much speaker discrimination information as possible is presented by Chen and Salman (2011a; b). A *deep network architecture* (DNA) is used to learn essential features of the speaker. The DNA is trained with labelled MFCC frames; with the objective of grouping frames uttered by the same speaker and discriminating different speakers as much as possible; without the loss of the fundamental objective in any DNA which is to minimize the difference between the input and the output layers. One of the hidden layers is called the *code layer*; which contains two types of values: speaker related and speaker non-related values. The latter are retained to be used in the reconstruction process (the rest of the layers). To be able to learn the discrimination properly, two networks are propagated

simultaneously with and the difference in the code layer is minimized if they belong to the same speaker or maximized otherwise. This multi-objective loss function can be summarized by the following ($0 < \alpha < 1$ controls the trade off between the reconstruction loss and the discrimination loss):

$$\mathcal{L}(X_1, X_2; \Theta) = \alpha[\mathcal{L}_R(X_1; \Theta) + \mathcal{L}_R(X_2; \Theta)] + (1 - \alpha)\mathcal{L}_D(X_1, X_2; \Theta) \quad (2.7)$$

$\mathcal{L}_R$ is the reconstruction error and can be straight forwardly calculated by:

$$\mathcal{L}_R(X_1; \Theta) = \frac{1}{T_B}\sum_{t=1}^{t=T_B}\|x_t - \hat{x}_t\|_2^2 \quad (2.8)$$

However, $\mathcal{L}_D$ is the discrimination loss. $\mathcal{L}_D$must be small for the same speaker and large for different speakers. The writers propose the following as the compatibility measure between two frames.

$$D(X1, X2; \Theta) = \|\mu_1 - \mu_2\|_2^2 + \|\Sigma_1 - \Sigma_2\|_F^2 = D_m + D_S \quad (2.9)$$

Where μ is the mean value of the code vector; and Σ is its covariance matrix. $\|.\|_2$ is the $\mathcal{L}_2$ norm and $\|.\|_F$ is Frobenius norm (i.e. $\|A\|_F = \sqrt{tr(AA^*)}$). Therefore the discrimination loss can be expressed as:

$$\mathcal{L}_D = \mathrm{I}\, D(X1, X2; \Theta) + (1 - \mathrm{I})(e^{-\frac{D_m}{\lambda_m}} + e^{-\frac{D_S}{\lambda_S}}) \quad (2.10)$$

$\mathrm{I} = 1$ if the frames are taken from the same speaker, and $\mathrm{I} = 0$ otherwise. λ_m, λ_S are boundary parameters that can be estimated from the training set. According to the writers, not having a reconstruction loss will result in the DNA over fitting the training data.

The speaker specific representation of the frame is obtained from the code layer. 100 features are extracted for each MFCC frame. Using the MFCC representation removes the effects of the source and generates more robust performance for the recognition task. However, using MFCC it is unclear how to determine which frames are similar enough to be deemed from the same speaker. The DNA is trained to have the representation of the frames form the same speaker close and from different speakers far. Therefore, developing more effective recognition is possible.

Figure 2.8 shows an example of the DNA. The code layer is dark shaded and the speaker specific features are also shaded in the code layer.

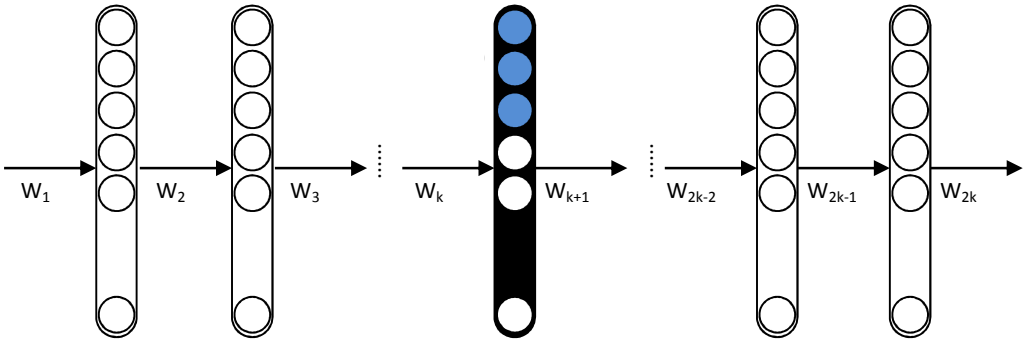


Figure 2.8 a DNA. The shaded layer is the code layer. The shaded neurons are the speaker specific features.

2.3.8. Summary

A number of sound representations were presented in this section; including speaker specific representation (DNA). The features presented are generic, i.e. they can be utilized in any speech related task, such as speaker identification or speaker diarization. Clearly, each representation has advantages and disadvantages. Moreover, each representation may need further processing. For the purpose of this project, a comparison will be done to show the performance of the DNA representation when compared with other representations in terms of accuracy. Nonetheless, the higher dimensionality of the DNA (100 features for each frame) may cause some issues and might be time consuming.

2.4. Statistical Modelling

In the field of Artificial Intelligence (AI) it is not uncommon to use statistical models as governing models for a specific set of features for the studied event, or to capture the essence of the studied phenomenon. The normal distribution (Gaussian distribution) of the measurements is most often used for its simplicity and adequacy.

In this section, the Gaussian distribution will be briefly examined and then extended in a Mixture Model. Towards the end GMM training, adaptation and implementation issues are presented.

2.4.1. Gaussian Estimation

The Gaussian distribution is the most common statistical distribution used for modelling a certain phenomenon. In one dimensional Gaussians, the Gaussian distribution is degenerated to the famous bell shaped distribution, i.e. Figure 2.9(a). In general, this model is called a *mono-modal distribution* with one mean point and a variance value on each dimension. A multi-dimension Gaussian with one mean is still a mono-modal distribution, see figure 2.9(b). Practically, out of a certain set of N observations x_i: $1 < i < N$ we can estimate the mean $\hat{\mu}$ and variance $\hat{\sigma}$ using equation 2.11.

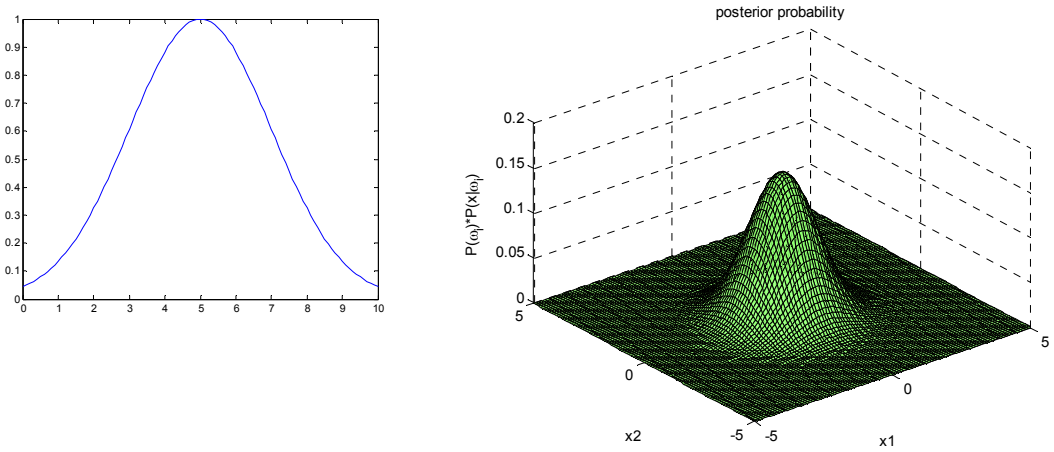


Figure 2.9 (a) single dimensional and (b) multi-dimensional single-modal Gaussian Distribution examples.

$$\hat{\mu} = \frac{1}{N}\sum_{i=1}^{N} x_i$$
$$\hat{\Sigma} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \hat{\mu})^2}. \tag{2.11}$$

Providing only the mean and variance values are enough to fully describe the model which fits the data best. The probability function is well known and given in the following closed form equation 2.12.

$$p(x) = \frac{1}{\sqrt{(2\pi)^d|\Sigma|}} e^{-\frac{1}{2}(x-\mu)^t\Sigma^{-1}(x-\mu)} \tag{2.12}$$

2.4.2. Gaussian Mixture Models

In many cases, the Gaussian distribution falls short to fully represent all the variability and distribution of the sample set because of its restricted bell shape. However, being extremely natural and easy to use gave the Gaussian distribution a certain appeal. For those reasons a more complex and mature model is usually used called the Multi-Modal normal distribution or *Gaussian Mixture Model* (GMM). The GMM is basically a combination of a number (M) of normal distributions; each having a different weight to make up a statistical distribution.

As Reynolds, Quatieri and Dunn (2000) explain: the density of a GMM distribution is a linear combination of a number of mono-modal Gaussians; i.e. $p(x|GMM) = \sum_{i=1}^{M} w_i p_i(x)$. An example of a GMM is presented in figure 2.10 with different GMM plot lines when using different numbers of components to fit the original bar-plotted data.

Therefore, the parameters of GMM are the mean values, the covariance matrices and also the weights. Hence, a training algorithm is needed to estimate good parameters to model the observed samples.

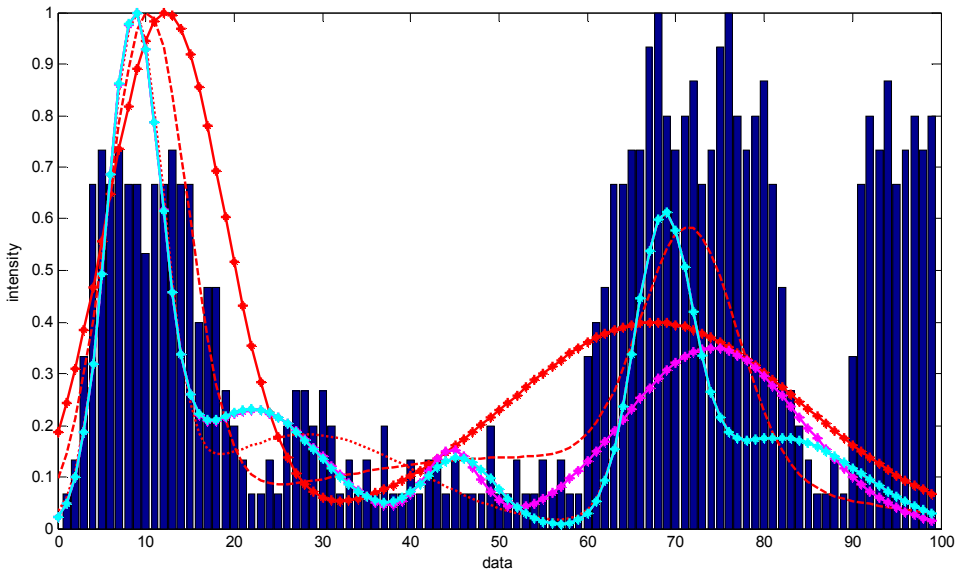


Figure 2.10 An example of a one-dimensional GMM fitting the MFCCs.

2.4.3. GMM Training and Adaptation

Unfortunately, it is not possible to calculate all the parameters of the GMM model by a closed form equation. Therefore, an Estimation Maximization (EM) algorithm, extensively explained in Dempster, Laird and Rubin (1977), is used to improve a GMM fit step by step until convergence.

The idea is to maximize the probability value of the observations by iteratively improving the model. One implicit assumption is that the observations are disjoint and therefore the probability of the observation being drawn from the GMM model at the i^{th} iteration can be expressed by:

$$p(x|GMM_i) = \prod_{t=1}^{N} p(x_t|GMM_i) \tag{2.13}$$

The following set of formulas are used to improve the parameters, they are proven to monotonically increase the probability. The full training algorithm is presented in (Reynolds, 2008).

The Mixture weight (for each component):

$$w_i = \frac{1}{N}\sum_{t=1}^{N} \Pr(i|x_t, GMM) \tag{2.14}$$

The mean (for each component):

$$\mu_i = \frac{\sum_{t=1}^{N} \Pr(i|x_t,GMM).x_t}{\sum_{t=1}^{N} \Pr(i|x_t,GMM)} \tag{2.15}$$

The covariance (for each component):

$$\sigma_i^2 = \frac{\sum_{t=1}^{N} \Pr(i|x_t,GMM).{x_t}^2}{\sum_{t=1}^{N} \Pr(i|x_t,GMM)} - \mu_i^2 \tag{2.16}$$

The probability of each component after the i[th] iteration:

$$\Pr(i|x_t, GMM) = \frac{w_i g(x_t|\mu_i, \sigma_i^2)}{\sum_{k=1}^{M} w_k g(x_t|\mu_k, \sigma_k^2)} \quad (2.17)$$

If the gain in the probability between two consecutive steps is small enough, then the process is stopped and the GMM is said to have converged.

Clearly, this process takes a long time until convergence and is not precise. Nonetheless, the more it is left to run the better the fit of the model becomes. Using the same process for every set of observation (every speaker in our case) will generate non-related models and will be very time consuming. More robustness and computational efficiency can be achieved by the training of one GMM model (usually called the Universal Background Model, UBM) and then adapting a new model for each speaker's data out of that UBM (Reynolds, 2000). Figure 2.11 expresses an example of this adaptation process.

For each set of observations, i.e. speaker, the adaptation process is also an EM algorithm. Starting from the original well-trained UBM, a number of iterations are carried out exactly like the EM used to train the GMM in the first place, except that change is done with a parameterized penalty α as to control how far each speaker model is expected to be from the original model.

2.4.4. GMM training and adaptation implementation issues

Zero Covariance Matrix

As the EM algorithm advances, it groups the points of each Gaussian component in a cluster. Those clusters change and different points change their clusters until a convergence is achieved or no more improvement is possible. During this process, some clusters may accommodate very few data points, which may lead to a problem when calculating the covariance, i.e. the number of data points is less than the dimension of the data vector. To reduce the chances of that happening, more training data is required or fewer components are used in the GMM.

Different implementations handle this issue differently. Moreover, a different initialization may prevent this issue. However, some quick-and-dirty fixes are proposed:

- Adding a small amount (ε) to the diagonal values of each covariance matrix following each EM iteration. This distorts the covariance matrices, depending on the value of ε. Nevertheless, it prevents the zero covariance matrix problems.

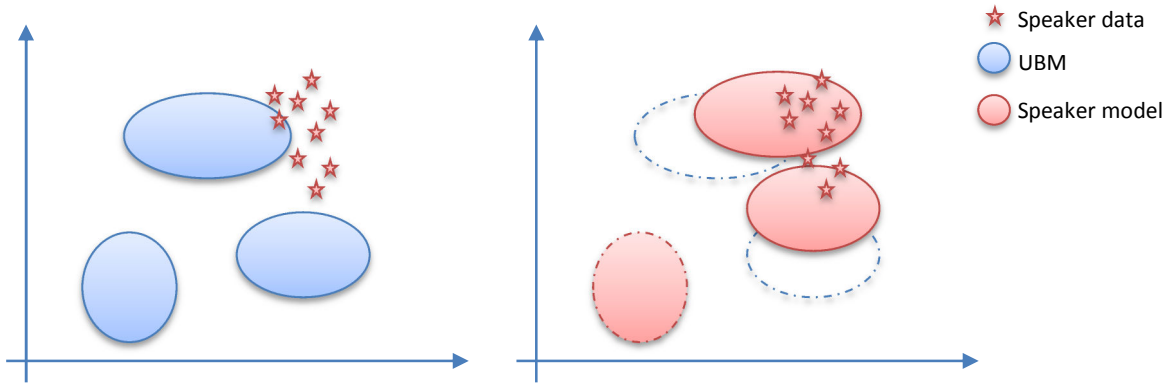


Figure 2.11 GMM adaptation to produce a speaker's model.

- Randomizing the covariance matrix when it is detected to have a zero covariance instead of estimating it from data points. This is quite a large distortion, especially at late stages of the EM and it is not recommended. Nonetheless, it also prevents the zero covariance matrix problems.

High dimensionality

The scarcity of the covariance matrix is another issue arises with very high dimensional data (such as the DNA features); which leads to very small covariance matrix determinant value; which in turn leads to absurdly large likelihood values (several hundreds of digits). This is not a wrong result, rather an incomputable result. The solution for this is working in the log domain, i.e. implement the EM training algorithm and GMM likelihood estimation functions in the log domain.

2.5. Summary

This chapter should have given the reader good background knowledge about speech digital processing and the most common features used. In the following chapters MFCC and DNA features are to be used.

Chapter 3

Research Method

This chapter introduces the problems at hand and discusses the different methods and techniques used to solve the problems. More than one method may be available to solve a problem, in that case further tests are to be done (see chapter 4). Problems introduced include: a) Voice Activity Detection; b) Speaker Diarization; and c) Speaker Identification. Towards the end, the Speaker Recognition Evaluation (SRE) is explained and our method for the task is mentioned.

3.1. Voice Activity Detection

Voice Activity Detection (VAD) is usually the first step of cleaning a speech stream, ordinarily used to remove any non-speech segments (including silences, overlapped speech, laughter, ambient noise, etc…) keeping only speech to be provided to later processing steps. Unsurprisingly, the accuracy of VAD depends on level of noise present in the stream, clarity of speech and bandwidth of the channel. The choice of a correct VAD technique is crucial for later tasks; and most SRE 2010 participants report the importance of correctly choosing a VAD method (Gonzalez-Dominguez et al., 2010; Abad et al., 2011). Nonetheless, VAD is still an open research problem (Ramírez, Górriz and Segura, 2007). In general, the output of the VAD should be clear continuous single source speech. As already discussed in section 2.3, most extracted speech features dependent on the formants (F1 through F5); therefore, it is very important for the VAD to keep the speech intact and not manipulate the speech segments. Consequently, most VAD methods are thought of as speech/non-speech classifiers or filters of the stream. Also, it is worthwhile performing a smoothing step for the end filter when appropriate as not to break continuous speech. Figure 3.1 clearly describes the flow of a speech/speaker related algorithm showing the position of VAD in the context of a complete algorithm.

The success of a VAD process is very hard to assess. That is why it is generally acceptable for VAD to be considered in the context of another task such as speech coding or recognition. And then evaluate the complete algorithm.

Figure 3.1 A common flow of a speech/speaker processing algorithm.

On the other hand, it is possible for some tasks (such as speaker segmentation) to handle non-speech on-the-fly as a separate speaker. For example, in the segmentation task, it is not uncommon to use a separate speaker for each of the following: music, overlapping speech, silences (Gauvain, Lamel and Adda, 1998). Although Anguera et al. (2012) reports this does not produce the best results. The following section describes some common techniques that are used for VAD; later sections will only refer to the techniques by name.

3.1.1. Energy Based Estimator

This method assumes that non-speech frames have lower energy than speech frames. This assumption is true in case of silences. However, it fails to detect other types of non-speech, especially overlapping speech. Nevertheless, this approach is very simple to implement and fast to run.

A window is slid across the stream and the energy for each window is computed. Either an energy threshold is set for decision making. Or, if the speech portion of the whole stream is known a priori (as a percentage of the stream length), then windows are sorted in a decreasing order according to their energies. Later, the top percentage of the sorted windows are classified as speech and thus retained, while the rest of the windows are discarded as being non- speech.

This method is parametric (needs a threshold parameter to be set manually). Some research was done to estimate the threshold according to the SNR of the stream; however, there is no guarantee for these methods to perform any better. And even if the best parameters were set, other non-speech activity (laughter, overlapping speech, etc…) will still be audible on the output stream. Another issue is that the classified frames may be interleaved, and the end stream would miss parts in the middle of the speech (e.g. unvoiced segments) if they were classified as non-speech. For that, it is possible to perform a smoothing step for the filter.

Figure 3.2 shows the results on a time series when applying a simple energy-based VAD. The strictness of the results is clear and much of the speech has been trimmed.

3.1.2. Statistics Based Estimator

Depending on research on the stationary noise behaviour on different bands Sohn, J. Kim, N. Sung, W. (1999) proposes a method for discrimination between two hypothesises for a window X:

$$H_0: speech\ absent: X = N$$

$$H_1: speech\ present: X = N + S$$

Ramírez, Górriz and Segura (2007) formulate the decision rule by the following:

$$P(H_1|X) \underset{H_0}{\overset{H_1}{\gtrless}} P(H_0|X) \tag{3.1}$$

By using Bayes Rule:

$$\frac{P(H_1|X)}{P(H_0|X)} \underset{H_0}{\overset{H_1}{\gtrless}} \frac{P(H_0)}{P(H_1)} \tag{3.2}$$

Based on knowledge about noise distribution and the amount of speech in the stream, one can classify speech and noise frames using equation 3.2. This method requires a great amount of prior knowledge about noise distribution and about the speech stream. This method can correctly classify silences and ambient noise. Nonetheless, it fails to classify overlapping speech and other unwanted sounds (paper shuffling, laughter, etc…).

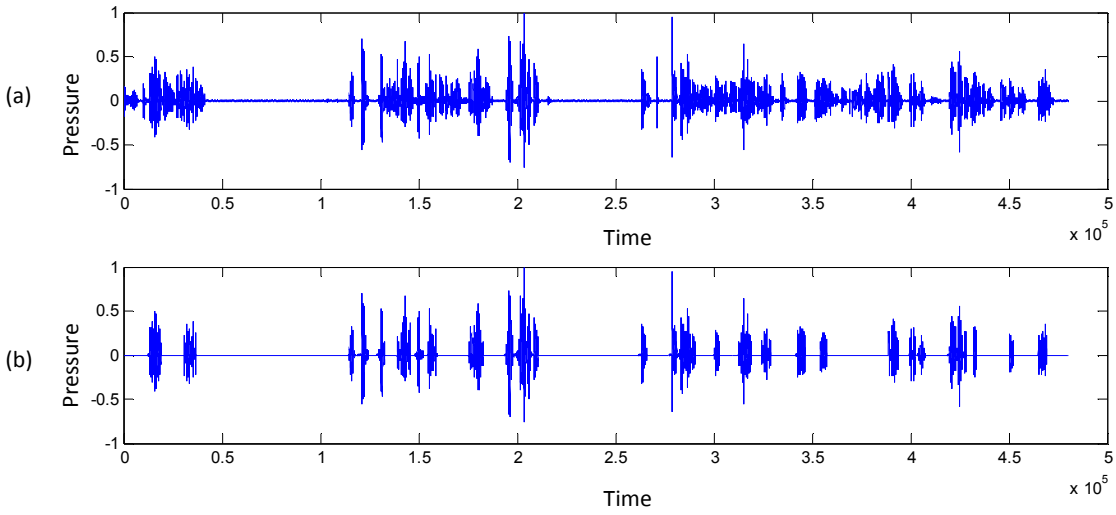


Figure 3.2 (a) The time series and (b) energy based VAD results.

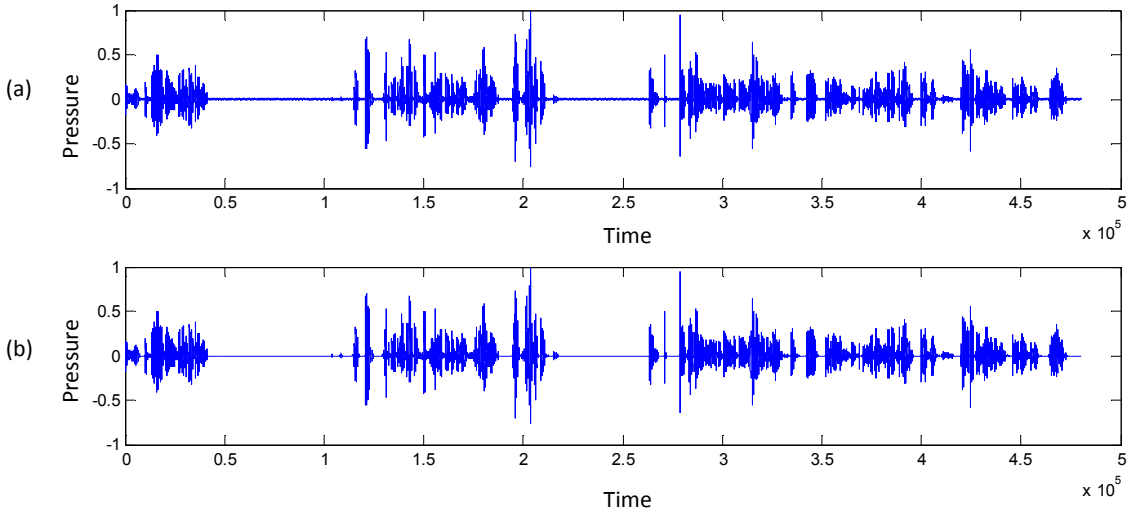


Figure 3.3 (a) The time series and (b) statistic-based VAD results.

Figure 3.3 shows the results of applying a statistical based VAD on a sound time series. The result looks better than the energy based approach in terms of retaining more speech.

3.1.3. Supervised Methods

According to Anguera et al. (2012) the best VAD results in general can be achieved by training general models for different types of non-speech and then use a discriminative method (such as LDA or SVM) to make predictions on each window. In (Wooters et al., 2004) a supervised method is used with a three state HMM trained to detect speech/non-speech/music. In (Liu and Kubala, 1999) 8 classes differentiation was made: 3 for speech phonemes and 5 for silences. The trained models are usually GMMs using LPC or MFCC features. The GMM is trained using a large amount of target type data and clearly classified frames are used for the classifier. One problem with this method is that models are trained with external data with different collecting environments than the testing data. Therefore, tuning and normalization may have to be performed. Also, there is a need for huge amounts of data for each class (such as: laughter, paper shuffling, music, etc…).

Hybrid approaches and multi-pass approaches are sometimes used, but these methods are usually slow and do not improve the overall accuracy of the task at hand.

3.1.4. Summary

VAD is very important as a pre-process technique when dealing with speech or speaker related task. However, the more noise present in the stream the harder the VAD becomes. For some tasks (such as speaker identification) good results may be achieved with little effort on the VAD. That is true because one can use a simple energy-based VAD on the training and testing streams, which will be strict and will mistake much of the speech as non-speech and discard of it. Nonetheless, if the streams are long, the retained speech will be enough to correctly build speaker models or identify the speaker.

3.2. Speech Diarization

This section discusses the different methods and approaches used for speaker diarization. The information is presented in a logical order. First, the diarization task is introduced and different parts of it are defined. Single- and Multi-Channel diarization are presented. The VAD and its role in the Diarization task are discussed. Later, metric based segmentation and different divergence measures are listed, followed by the models used for speaker representation. These models are used in model based segmentation and in speaker clustering. The section is concluded

with one step diarization technique, which saves time and may be more accurate than other techniques.

3.2.1. Single Channel Diarization

Speech diarization is the task that answers the "*who spoke when?*" question. According to Kemp et al. (2000) many recognition tasks depend on being fed short single speaker speeches and would fail if more than one speaker's speech is present. However, at the moment a great number of sound streams are long multi speaker streams; such as meeting data and news broadcast; and also it is still very hard to capture only one speaker speech in real-life environments. The speaker diarization task aims to partition the input stream into parts uttered by a single speaker and also determine the regions spoken by each individual speaker (Kotti et al., 2008). Several applications benefit from this process; such as dialogue detection, automatic annotation of broadcast news and automatic indexing of speech according to speaker.

It is common to think of the diarization task as "blind"; i.e. with no prior knowledge about the content of the stream (Tranter, 2006). However, this is not a constraint and some research do benefit from knowledge about the stream content. Furthermore, the quality of the input stream may be a source of errors in the segmentation and clustering process; such as: low signal to noise ratio (SNR), low bandwidth of the recording and other human artefacts such as heavy breathing (Almpanidis and Kotropoulos, 2008).

Speaker diarization is thought of as a two-step task (Kotti et al., 2008):

1) *Speaker segmentation*: the task of determining *turn points* in a stream. That is where the change of the speaker occurs. The output is a stream of segments, each uttered by a single speaker, and no two adjacent segments uttered by the same speaker.
2) *Speaker clustering*: the task of assigning each segment to its speaker, so that it can be determined what parts were spoken by each speaker.

A classification of diarization systems can be made by approach (Anguera et al., 2012):

1. *Bottom-Up*: the output is initialized by a very large number of clusters, each representing the speaker of a segment of the over-segmented stream. When two clusters are determined to belong to the same speaker, they are combined. An extreme example would be "frame to cluster approach"; where each frame is initially thought of as being from a different speaker.
2. *Top-Down*: the output is initialized by a few clusters, each representing a speaker of a segment of the under-segmented stream. Whenever a cluster is determined to include more than one speaker it is split into two. An example would be the E-HMM approach, where the model is initialized by only one HMM node (representing the only cluster) and

nodes are added, each representing a speaker, until the HMM faithfully represent the stream (see section 3.2.7).

3.2.2. Multi-Channel Diarization

Depending on the scenario and the available data; the diarization system may be provided with a single channel of sound to diarize, or a number of channels for the same recording to be used in the diarization process. An example of a multi-channel recording is a recording of a meeting that has many microphones in the meeting room available. More information may be extracted from the multi-channel recordings. Anguera et al. (2006) use the multi channels to build an enhanced single reference speech signal using a technique called *Beam Forming*. Beam forming uses a sliding window and the information of relative delay of each microphone to an elected reference microphone, and then aligns the different channels into one channel. Boakye et al. (2008) use the multi channels for overlapping separation. This separation helps in eliminating the effects of non-speech activity (such as door knocking or paper shuffling) on the stream and extracting more accurate features of the recording. Anliker et al. (2006) use the time delay to estimate the speaker position and use that information to improve speaker segmentation.

A single channel diarization box flowchart is presented in Figure 3.4(a) and a multi-channel diarization box flowchart is presented in figure 3.4(b). It is clear that multi-channel diarization is the same as single channel diarization in its core, except it uses multi-channel features. That is why I will focus only on single channel diarization.

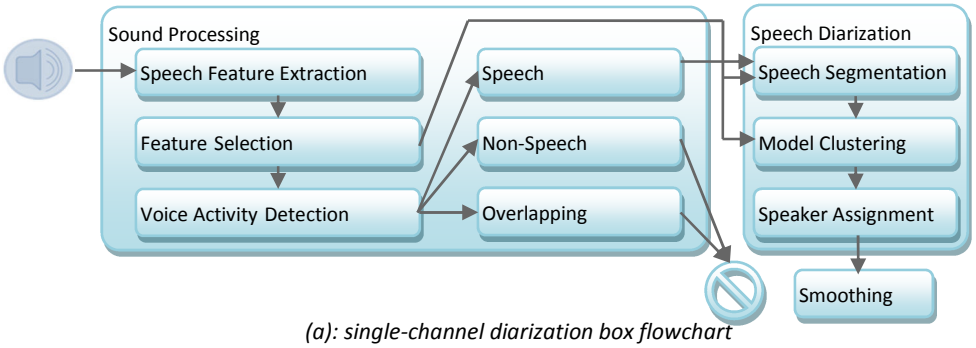
3.2.3. Voice Activity Detection for speaker segmentation

As discussed in section 3.1; VAD is used to separate speech parts from non-speech parts. In (Liu and Kubala, 1999) the writers claim that 80% of the true speaker change points take place in non-speech frames and therefore it is essential to get VAD correctly.

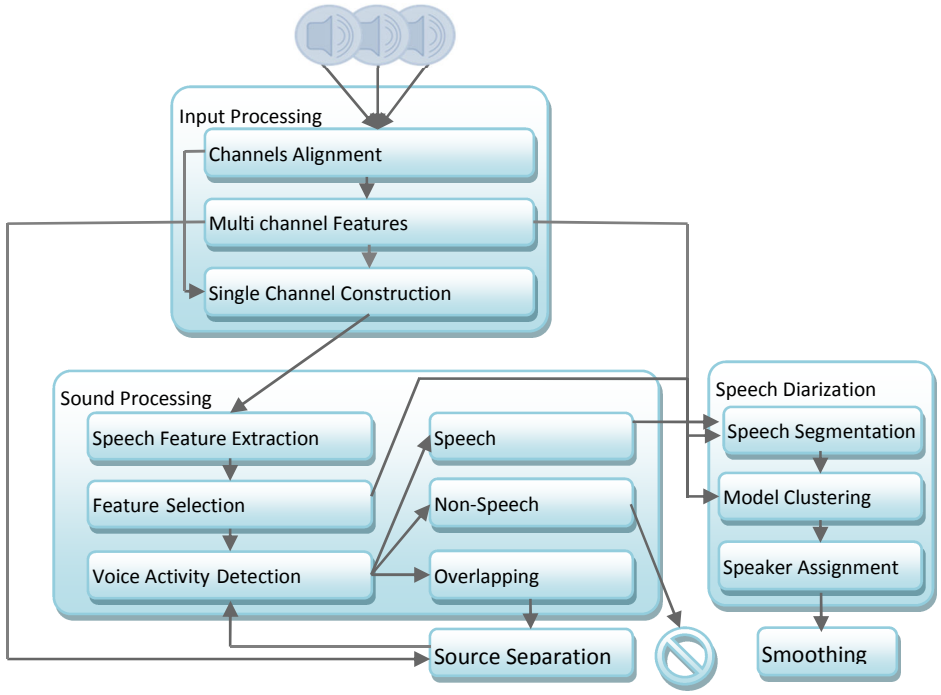
Some research tries to handle non-speech as an extra speaker, and therefore eliminates the need for an explicit VAD process. However, experiments have shown that having a separate VAD improves the overall performance of the system (Anguera et al., 2012).

3.2.4. Speaker Segmentation

Most approaches of finding *turn points* in multi speaker stream can be explicitly categorized as:



(a): single-channel diarization box flowchart



(b): multi-channel diarization box flowchart

Figure 3.4 single-channel and multi-channel diarization box flowchart

- *Metric based*: often the assumption that one speaker's speech follows a normal distribution is made. Hence; statistical metrics (measuring how far two statistical distributions are), i.e. *divergences*, can be used to check whether two speech parts are uttered by the same speaker or not. The method by which this idea is implied can be further subcategorised into two methods (Cheng and Wang, 2010).
    - *Fixed size window*: Two fixed sized windows move frame by frame through the stream. The dissimilarity is measured between them; which results in a distance curve over the entire stream. This curve can be further processed to find peaks or major dissimilarity points. The main metrics used are listed in 3.2.5. Kotti et al. (2006) provides a comparison between three systems and concludes that different measures put emphasis on different evaluation criteria such as reducing the false alarm rate of reducing the miss diarization

rate. However, some of the best performing systems need manual parameter setting.

- *Growing window*: A window is expected to contain one turn point or otherwise it is grown by one frame. Once the window is decided to have a turn point the process is repeated starting from the turn point. The metric used to make the decision of the most probable change point is usually a Bayesian Information Criterion (BIC) or any of its variations. This process is evaluated in (Cettolo et al., 2005). Clearly, the process is very computationally demanding. Therefore, some research has been done to pre evaluate regions' probabilities for containing a turn point before applying the BIC metric.

Model based: models are trained in a supervised fashion for each speaker. These models are then used to estimate turn points. In practice, this method transforms the segmentation task into an identification task where each window is separately being identified to belong to one of the known speakers. According to Kotti et al., (2008) the most common model based method is GMM/UBM presented in (Reynolds at al., 2000). It relays on the Gaussian Mixture Model (GMM) to represent one speaker (see section 2.4). Followed by a trained discrimination process (such as SVM) to discriminate among speakers. Other models are rarely used such as *anchor models* where the speaker is not represented in a statistical "cloud" form, but rather as a vector of its distance from a certain set of predefined speakers. This vector is called the speaker characterization vector (SCV) and their differences can be measured using L_p norm metrics[5]. (Anguera and Bonastre, 2010) builds on top of that a faster model called *binary keys*; which has shown comparable performance to anchor models at much lower time consumption.

3.2.5. Distance Measures

A number of distance measures can be applied to sliding fixed-sized windows P and Q. In fact, any divergence metric could be used. More statistic work is presented in (Bimbot et al., 1995); however, the following is a list of the closed forms of the most used distance measures according to Kotti et al., (2008): μ is the mean vector of the features in the window; Σ is the covariance matrix of the features and n is the number of frames in the window. tr is the trace operator.

- *Kullback divergence (KL2)*

$$D_{KL2}(P \parallel Q) = \frac{1}{2}\left(\mu_Q - \mu_P\right)^T \left({\Sigma_P}^{-1} + {\Sigma_Q}^{-1}\right)\left(\mu_Q - \mu_P\right) + \frac{1}{2} tr({\Sigma_P}^{-1}\Sigma_Q + {\Sigma_Q}^{-1}\Sigma_P - 2I) \qquad (3.3)$$

[5] $L_q(V) = \sqrt[q]{\sum |V(i)|^q}$

- *Bhattacharyya divergence (BH)*

$$D_{BH}(P \parallel Q) = \frac{1}{4}\left(\mu_Q - \mu_P\right)^T \left(\Sigma_P + \Sigma_Q\right)^{-1}\left(\mu_Q - \mu_P\right) + \frac{1}{2} log \frac{|\Sigma_P + \Sigma_Q|}{2\sqrt{|\Sigma_P \Sigma_Q|}} \quad (3.4)$$

- *Hotelling T^2 divergence (H)*

$$D_H(P \parallel Q) = \frac{n_p . n_q}{n_p + n_q}\left(\mu_Q - \mu_P\right)^T \Sigma^{-1}\left(\mu_Q - \mu_P\right) \quad (3.5)$$

- *Generalized Likelihood Ratio divergence (GLR)*

$$D_{GLR}(P \parallel Q) = \frac{n}{2}\, log|\Sigma| - \frac{n_P}{2}\, log|\Sigma_{\mathrm{P}}| - \frac{n_Q}{2}\, log\left|\Sigma_{\mathrm{Q}}\right| \quad (3.6)$$

- *Information based: Rényi divergence (EL)*

$$D_{EL}(P \parallel Q) = R_\alpha(P \parallel Q) = \frac{1}{2}\left(\mu_Q - \mu_P\right)^T \left(\alpha\Sigma_P + (1-\alpha)\Sigma_Q\right)^{-1}\left(\mu_Q - \mu_P\right) - \frac{1}{2\alpha(\alpha-1)} log \frac{|\alpha\Sigma_P + (1-\alpha)\Sigma_Q|}{|\Sigma_P|^{1-\alpha}|\Sigma_Q|^{\alpha}} \quad (3.7)$$

Rényi divergence is generic with a parameter α. Usually α is considered in the range [0,1], but that is not a restriction of the metric. When $\alpha = 1$ the metric converges to Shannon entropy $E(P - Q)$. Interestingly, when $\alpha = 0.5$ the equation will be identical to the Bhattacharyya metric.

Another widely adopted metric is the *Bayesian Information Criterion* (BIC) given in the equation.

$$\Delta BIC(P \parallel Q) = \frac{n}{2}\, log|\Sigma| - \frac{n_P}{2}\, log|\Sigma_{\mathrm{P}}| - \frac{n_Q}{2}\, log\left|\Sigma_{\mathrm{Q}}\right| - \frac{1}{2}\lambda\left(d + \frac{1}{2}d(d+1)\right) logn \quad (3.8)$$

d is the number of features considered; and λ is a human set parameter that controls the penalty related to the complexity of the model.

Chen and Gopalakrishnan (1998) formulated the speaker change problem as a model selection problem using ΔBIC; where either two speakers contributed to the speech in the window, and therefore two normal distributions fit it best. Or only one speaker contributed to the speech in the window, and therefore only one normal distribution fits best. This is currently used to select among the following two hypotheses:

H_0 no speaker change during the whole window.

H_1 a speaker change takes place in the middle of the window.

The selection usually requires a threshold which is set manually, or set to zero and controlled by the λ parameter.

3.2.6. Speaker Modelling

It is usually in the speaker identification context where research is concerned with speaker modelling. However, the metric-based approaches make the inaccurate assumption that one speaker's features are normally distributed. For example, it is not very recognizable but Almpanidis and Kotropoulos (2008) show how the generalized gamma distribution fits the MFCC features better than the mono Gaussian distribution. Moreover, it has become the norm in speaker modelling to use HMMs with nodes of GMMs with 1024 or more components using MFCC or LPC features to encode the information of one speaker in a text independent scheme (Reynolds, 2002).

Unfortunately, there is no closed formula for a divergence between two GMMs. Although some effort has been made to devise a computable distance measure (Goldberger, Gordon and Greenspan, 2003); it is still not accurate and very computationally extensive.

Consequently, model-based approaches handle the problem differently. A number of models are trained a priori (by means of EM algorithms, see section 2.4). Then a window is slid across the stream and for each window one of the speakers is identified as its speaker. See section 3.3. for better understanding of the technique. Depending on the accuracy of training, the size of the window and matching environmental effects; the accuracy of identification may change. A smoothing step may be applied afterwards as not to break one speaker's speech.

3.2.7. Speaker Clustering

The purpose of the clustering is to group all the segments uttered by one speaker in one unity. Preferably, the system would produce one cluster for each speaker. When the speaker is modelled as a single modal normal distribution, distance can be measured using any of metrics presented for segmentation such as the KL divergence. Otherwise, more complex techniques are used such as the use of HMM (a node for each GMM and a learnt transition function) with an alignment algorithm (e.g. Viterbi).

However, an inherited problem of clustering is determining the correct number of clusters. To that end, the most widespread approach for speaker clustering is hierarchical, agglomerative *bottom-up* clustering (Tranter, 2006). To do that a cluster for each speech segment is initialized after which a number of iterations are curried out. Each iteration, the two closest clusters are merged. This process is repeated until the closest two clusters are far enough (i.e. surpass a manual threshold) to describe two distinct speakers and should not be merged. Conversely, *top-down* approach initializes the model with a few clusters and split one whenever is appropriate.

When a frame-to-cluster approach is carried out; it might be the case that some frames are mistakenly assigned to a cluster that differs from the cluster of its surrounding frames. Further,

the boundary frames of a turn point may be misclassified. Therefore, there is a need for a cluster purification scheme.

3.2.8. One Step Diarization

It is clear that the same kinds of metrics are being used for both segmentation and clustering processes. Besides, the accuracy of the clustering is highly affected by the accuracy of the segmentation. Plus, the accuracy of the segmentation can be improved by clustering (detection of missed turn point or removal of a hypothesised turn point). Thus, one-step segmentation and clustering approach may be carried out. According to Anguera et al. (2012) either bottom up; or top down. A speaker is usually modelled by a GMM and the transitions by an HMM. Then a dynamic algorithm aligns the frames into clusters; such as the approach Evolutive HMM (Meignier et al., 2000).This approach has the advantage that the information gathered from the clustering process can improve the segmentation. However, it is extremely computationally demanding.

3.2.9. Summary

In this section, the speaker diarization task was formally put. All the subtasks are introduced and discussed. Two main approaches are presented. Namely: metric-based and model-based. A number of metrics, divergences, are listed in their closed form for easy use. Later, only the name of the metric will be specified.

3.3. Speaker Identification

This section provides some background about the speaker identification tasks and presents the most common methods used. Different speaker modelling approaches are presented and many issues regarding the environment modelling and background modelling are also presented.

3.3.1. Introduction to Speaker Identification

Speaker identification is the task of identifying who is speaking in a stream. In most cases, the identification task is thought of as a multi-class classification. The task is further explained in (Reynolds, 2002) where the task is sub categorized into two types:

- *Closed set*: the speaker is one of a closed apriori known set of speakers. The system's task is to decide which speaker of the mentioned set is speaking in the stream. No identity claim is given. The methods for this task are actually the same methods used for the model-based segmentation.

- *Speaker Verification*: the speaker presents an identity claim along with the stream. The claim is one of the set of the speakers already known by the system. The system is expected to verify this claim by a yes/no answer.

- *Open set*: the system assumes that the speaker is either one of the speakers known a priori or is a new speaker out of the known set. This is formalized by adding a new "none of the above" option to the set of known speakers.

Another way to classify the identification task is by spoken text. In some cases, all speakers are expected to say a certain sentence or phrase. This case is called *text-dependent*. The more general case is called *text-independent* and it happens when no restrictions are set on the speech content.

Most research has targeted the more general tasks, with no restriction on the content of the stream or the environment used to create the stream (at least it is allowed to vary between training and testing streams). A good example of this technology is the National Institute of Science and Technology. Speaker Recognition Evaluation (NIST SRE) 2012 which will be described in section 3.4.

According to Reynolds, Quatieri and Dunn (2000) the best performing systems in the text-independent task are based on GMM/UBM modelling. Their results have been the best achievable regardless of the used features for speaker modelling. That is due to the complexity of the GMM and the strong theoretical basis for the approach. Other approaches, such as supervectors classification, recently reported comparable results. An extensive results comparison is available in the MIT participation in SRE 2010 (Sturim et al., 2011). The only issue is environmental change between training and testing settings. Later, (Gonzalez-Dominguez et al., 2010) reported good results of a novel approach for environmental noise isolation called *Joint Factor Analysis* (see section 3.3.3.2). This dramatically improved the results in the SRE 2010 participations.

In the work of Reynolds and Rose (1995) a general identification algorithm is described. A window (of N frames) is slid across the testing stream. Each window is identified separately with a confidence score and a final decision is made accordingly. To eliminate the window size effect and the length of the stream effects, average functions are used. This method has become the standard method in speaker identification. Moreover, a scoring mechanism is proposed and justified. The likelihood score is basically the likelihood of selecting one model λ given the input testing data X, i.e. $P(\lambda|X)$. Those probabilities (for all models) will not form a probability distribution. For that we normalize by the probability of the testing data X not fitting the speaker model λ i.e. $P(\bar{\lambda}|X)$. Scores can be calculated according to equation 3.9

$$s = \frac{P(\lambda|X)}{P(\bar{\lambda}|X)} = \frac{\frac{P(X|\lambda).P(\lambda)}{P(X)}}{\frac{P(X|\bar{\lambda}).P(\bar{\lambda})}{P(X)}} = \frac{P(X|\lambda)}{P(X|\bar{\lambda})} \cdot \frac{P(\lambda)}{P(\bar{\lambda})} \quad (3.9)$$

The term $\frac{P(\lambda)}{P(\bar{\lambda})}$ is a constant over all models λ. Therefore, the scoring mechanism $s = \frac{P(X|\lambda)}{P(X|\bar{\lambda})}$ has become the most common scoring mechanism is speaker identification. Calculating the denominator can be done by testing X against every other speaker model in the training set. This process is a) time consuming, and b) cannot cope when an unanticipated speaker is encountered. Conversely, a general GMM model for all human speech is proposed and developed during the training process, usually called Universal Background Model (UBM).

After scoring the stream against all the speaker models another normalization technique (e.g. t-test) may be applied in order to limit the variability of scores.

3.3.2. Speaker Modelling

Figure 3.5 shows the basic flow of a speaker identification system (closed-set). Regardless of the features used (be it MFCC, PLP or DNA), we make the assumption that a model is going to fit the features and that the scoring mechanism is going to work perfectly. Clearly, this depends on the complexity of the model (which is going to affect the running time of the algorithm). In any case, the researcher has to make a decision of how to model a speaker and what is the appropriate scoring scheme.

In text-dependent identification, the most important knowledge to store in the model is the temporal changes that the features experience through the spoken sentence. On the other hand, this information is not as important in the text-independent identification. The feature vector is often thought of as forming a "cloud" which is needed to be stored (Kinnunen and Li, 2010). HMMs are the most used model in the text-dependent identification because of their inherited characteristics of encoding temporal change. However, this is a very restricted task which part of a more general task, i.e. the text-independent identification, where we try to model the whole cloud of feature vectors.

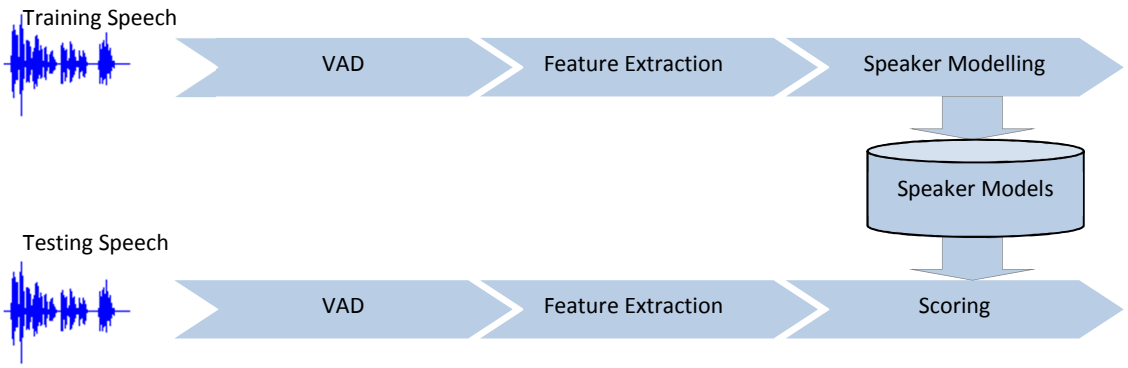


Figure 3.5 A common flow of a speaker identification system.

Two major families of models are available:

- *Discriminative*: where we try to model the boundaries of the speaker clouds (Ramachandran et al., 2002)
- *Generative*: where we try to capture the shape of each cloud (Reynolds and Rose, 1995).

The following is a list of the most commonly used models and algorithms to facilitate matching and scoring:

- *Vector Quantisation* (VQ): The earliest method of speaker identification, it depends on keeping all the vectors of a speaker as a model. The scoring system relays on a distance measure (Euclidian) of each test frame from each training model (Kinnunen and Li, 2010). In practice, only some vectors for each speaker are kept (e.g. 20 feature vectors); which are retained by clustering of the available training data of each speaker.
- *Mono Gaussian*: we assume that the data is Gaussian distributed. The scoring would be normal likelihood of the testing frames presented to the system. This model is easily trained and tested. However, depending on the variability of the features, it might be the case that Gaussians are very broad (with big covariance matrices) when modelling the speakers. Therefore, likelihoods would not really represent much information about how will a model fits the data. On the other hand, if the features are Gaussian distributed (or can be gaussianized); then this model is efficient and effective.
- *Gaussian Mixture Models* (Reynolds and Rose, 1995): As seen in section (2.4) GMM is more complex for fitting data. The scoring is also the likelihood. This approach requires a finely tuned UBM for speech data, and speakers' adaptation to create their individual models. The number of components of the GMM can be used to tune the performance, although it is very computationally demanding to build a GMM. Choosing the training data for the UBM is somewhat arbitrary. Although it is advised to cover all variability equally (e.g. equal length of male and female uttered speeches). Otherwise, the UBM will be biased towards one form of data (Reynolds, Quatieri and Dunn, 2000). If the gender of the speaker is known a priori, different gender-related UBMs may be trained for better results. This approach has shown stable results over a big number of SRE participants because it does not make assumptions about the data provided.
- *Mutli-class discrimination*: The identification task can be easily reformatted as a multi-classification task; using classifiers such as Nearest Neighbour and Neural Networks (Reynolds, 2002). The main issue with these methods is the lack of generalisation ability, along with the fact that their training is very computationally demanding. Nonetheless, they are known not to produce as good results as statistical models. Recently, better

results were achieved by using GMM super-vectors as features (Campbell et al., 2006). A super-vector is the stacked set of mean vectors of a GMM model.

In conclusion, the model partly depends on the feature vector used. Most the current research use MFCC as speaker representation. Moreover, it is not clear which of the above methods is best. That is because they depend on the training and testing recording environment and the VAD used. Nonetheless, some research has considered relating one speaker to other speakers such as the already-mentioned *anchor models* and *binary keys* (see 3.2.4). Similarly, some research is considering more speaker specific features such as (Malayath et al. 2000). Our research is of the later family.

3.3.3. Relevant Issues

3.3.3.1. Universal Background Model (UBM) Selection

The UBM is the general model for human speech. It should have the same structure as any speaker model in a certain application. However, it is trained using many speakers in an attempt to capture the human voice traits in a generative manner. Also, UBM is to be used as a normalizer for all speaker models' scores in the identification phase.

It has come to be the norm in speaker identification tasks to use a GMM as a UBM regardless of the feature vectors used. One of the main issues is the choice of the GMM number of components to be used in the UBM. A small number of components will not represent the data well, and a big number of components will require a huge training set and would take a lot of computational power. Also, it is clear that the more components we introduce the higher chance of inter correlation among the components. Lee, Lee and Lee (2006) suggest an iterative method for estimating a good number of components; by adding one component at a time and observing the mutual relationship between any two components. Nonetheless, this is very computationally demanding. One the other hand, Kampa (2010) suggests a method based on the BIC to select the correct number of components. The algorithm takes into account the likelihood of the fit and penalises the complexity as not to over-fit the data.

3.3.3.2. Environmental effect

The difference between the recording settings at training and testing times (or language used or any change of settings) distorts the results and causes miss identifications. Therefore, much research has considered this problem in order to reduce environmental effects.

- *Feature Mapping* (Reynolds, 2003): A very successful method to eliminate the environmental difference effect. After training a UBM using much speech out of many

speakers λ_{UBM}; one GMM is adapted for each environment setting by the use of much speech from many speakers in the same environment λ_{ei}. The difference between each λ_{ei} and λ_{UBM} (the means vector change) is used to isolate the effect of the channel in speaker models and in the testing stream.

- *Joint Factor Analysis* (Garcia-Romero and Espy-Wilson, 2010): State of the art technique for the time being when using MFCC features. Similar to Eigenfaces approach for lowering the dimensionality for face detection task testing and training images, an Eigenvoices approach is used for speaker recognition (see figure 3.6). The input vectors are usually the GMM supervector (Kuhn et al., 2000). Similarly, Kenny et al. (2007) suggest the use of a channel distortion factor, which is trained a priori in the Eigenvoices space.

  JFA decomposes the GMM supervector into four components:

$$s = m + Vy + Ux + Dz \tag{3.10}$$

  - m *: Speaker independent component (from UBM).*
  - y *: Speaker dependent component.*
  - x *: Channel dependent component.*
  - z *: Speaker dependent residual component.*

  V is the Eigenvoice matrix. U is the Eigenchannel matrix. D is the residual matrix. Those matrices are often in lower dimensionality that the original s supervector. In general the training is done in three steps for each one of the previous matrices.

  1- Train for V, assuming matrices U and D are zeros.
  2- Train for U, using the estimated V and assuming D is zero.
  3- Train for D, using the estimated V and U.

  Using the estimated matrices, we calculate the vectors x, y and z for each speaker. And then use this information in the testing process.

DNA Case: Because of the nature of the DNA features and the way they were learned, i.e. the structure of the code layer, they encode the identity of the speaker regardless of the environment; i.e. the DNA inherently isolates the speaker identity apart from the channel effect. Nonetheless, if the training does not cover all the environment variability, then the DNA will not learn to separate the channel effect well.

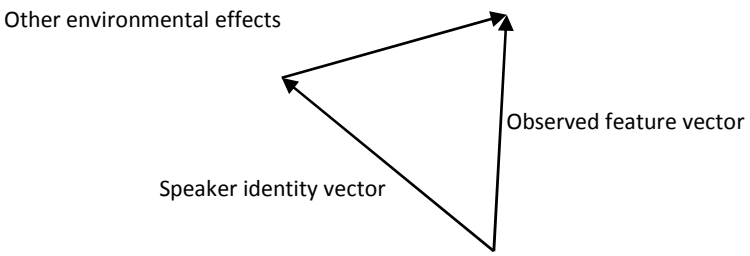


Figure 3.6 JFA basic idea in the feature vector space.

3.3.3.3. Normalization

The output scores of many tests may vary in range and degree according to many factors including the nature of the test segment, the environment and the identity of the speaker. Therefore, some speakers' scores may be very large in comparison to others. Similarly, the verification scores may vary as well.

Nonetheless, some thresholds may be needed to determine the output of the system (Yes/No answer or identity of the speaker). Those thresholds must be static or adaptive to the output scores' distribution. This is not easily set when the scores are highly variable. Some solutions are suggested for this problem and they are usually called normalization of the output scores.

According to Kinnunen and Li (2008); an internal assumption is often made: the output scores are normally distributed. Therefore, normalisation can be done using the following equation (3.11):

$$\hat{s} = \frac{s - \mu}{\sigma} \tag{3.11}$$

s is the "raw" score and $\hat{s}$ is the normalized score. Nonetheless, many test segments can estimate the Gaussian which fits the output scores for a specific test environment and a specific speaker. Two main normalization techniques are usually used in order to estimate the Gaussian and hence normalize the scores (Auckenthaler, Carey and Lloyd-Thomas, 2000).

- *Zero normalization (z-norm)*: a number of imposter speakers (from the training set or from an outer set) are used to estimate the distribution of imposter likelihoods $\mathcal{N}(\mu, \sigma)$. This can be done offline, with no need for validation or testing set. To do that, for each speaker model, some of the training data of others are used as testing data against that particular model.
- *Test normalization (t-norm)*: a number of models are selected (from the training models set) to be tested against the imposter testing data at test time. Therefore, this approach is online. The distribution is thought of as the distribution for only the current testing speaker and environment. Therefore, the environmental difference between training and testing data is minimised.

Kinnunen and Li (2008) reports good results when combining both normalization techniques in either order. Figure 3.7 is a simple sketch to show the effect of ideal score normalization. Finally, using environmental separation techniques (such as JFA) does not remove the need for this normalization. However, it has been reported in some cases that it did not improve the results.

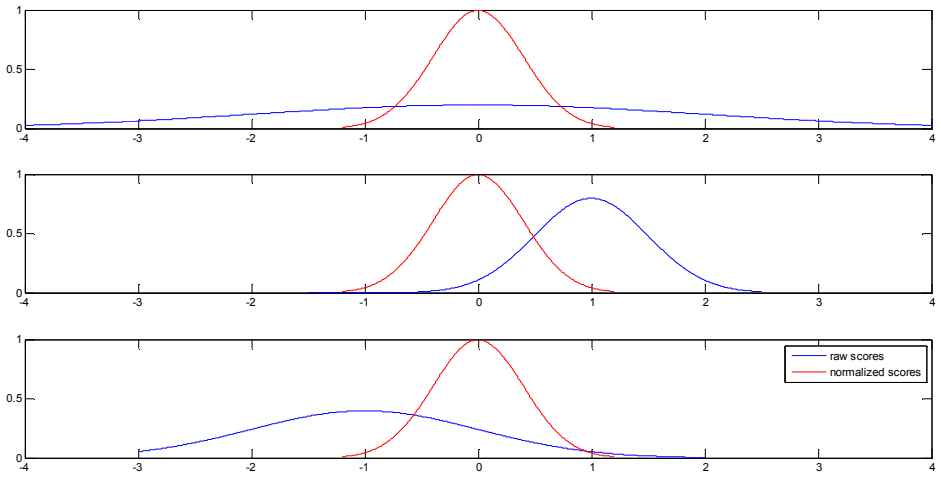


Figure 3.7 the effect of normalization over scores (a sketch to introduce the idea).

3.3.3.4. Converting Distance to Likelihood and vice versa

It is often the case that the system is required to output a likelihood ratio of how well the model λ fits the data x; i.e. $p(\lambda|x) = \frac{p(x|\lambda).p(\lambda)}{p(x)}$. Given that $p(x)$ and $p(\lambda)$ are constant for all speaker models; this likelihood is reduced to $p(x|\lambda)$. However, some speaker models are better used with a distance measure of how far the test data is from any speaker model.

Shlens (2007) shows a special case of the KL-divergence, it is shown that if the number of measurements (both at training and testing) approaches infinity then the KL-divergence between two Gaussian models p and q, and the average likelihood of each frame L (the likelihood of data points from one model to be represented by the other) are governed by the relation presented in (3.12):

$$D(p||q) = -\log_2(L) \tag{3.12}$$

Campbell (1997) suggests the following general formula as approximation for the likelihood regardless of the distance measure:

$$L = e^{-\alpha D(p||q)} \tag{3.13}$$

With the parameter α which is needed to normalize the resulting likelihood.

Instead we propose a different normalization technique for the closed set identification where we evaluate the likelihoods $\overline{L_1}, \overline{L_2}, \dots, \overline{L_N}$ for each speaker model $\lambda_1, \lambda_2, \dots, \lambda_N$. Then normalize the likelihood by their sum:

$$P(X|\lambda_j) = L_j = \frac{\overline{L_J}}{\sum_{i=1}^{i=N} \overline{L_i}} \tag{3.14}$$

3.3.4. Summary

In this chapter, Speaker Identification (in both forms) was presented; along with possible techniques and scoring mechanisms for the task at hand. Speaker Modelling was discussed in details and some relevant issues for the models were discussed, majorly the environmental change effects.

3.4. NIST Speaker Recognition Evaluation

In this section the tasks of NIST[6] SRE[7] 2012 are presented and also the input data and the scoring mechanism provided by NIST are discussed. Most of this information can be found in the SRE evaluation plan (NIST, 2012). The input data is described according to the data our site received from NIST.

3.4.1. Task Description

NIST administers an evaluation of speaker recognition systems worldwide. The evaluation is carried out almost once every two years since 1996. The specific tasks of the SRE have changed during this period of time. However, the main purposes of the evaluation are still the same, which are explained in the SRE plan (NIST, 2012):

- "Exploring promising new ideas in speaker recognition"
- "Developing advanced technology incorporating these ideas"
- "Measuring the performance of this technology"

The basic idea is: NIST provides the same testing data to different research groups around the world and specifies conditions and terms for using this data in profiles (called the tasks). In turn, different sites would explore different methods of approaching the SR tasks. Finally, a scoring mechanism is agreed on, and each system is evaluated. However, it is strictly forbidden for any team to be deemed a winner or for any team members to suggest themselves "the winners".

Nine different tasks are presented in the SRE 2012. *Core* task is one which is required by all SRE participants, all other tasks are optional but highly recommended. Tasks differ on two axes:

[6] http://www.nist.gov/

[7] http://www.itl.nist.gov/iad/mig/tests/sre/

Prior Assumption	Input Data Type: Core	Telephone	Microphone
Core	Required	Optional	Optional
Extended	Optional	Optional	Optional
Summed	Optional	-	-
Known	Optional	-	-
Unknown	Optional	-	-

Table 3.1: Tasks involved in SRE 2012

- *Input data type*: differs in the method of collecting speech. Microphone data is usually easier to handle than phone data because of the higher bandwidth over the channel. Nonetheless, the core task includes both types of data mixed together.
- *Prior Assumption*: differs in the way an imposter is handled. Whether the speaker must strictly be one of the training speakers (the *Known* task), or a totally new speaker (the *Unknown* task). In the Core task, both cases may be true (Open set identification).

Two more tasks are also presented:

- The *extended* task is the same as the core task but with a larger number of test segments. This is done to raise the confidence of the system's empirical evaluation.
- The *summed* task is concerned with summed speech data (two way conversations on the same channel).

Table 3.1 summarises the mentioned tasks of SRE 2012.

3.4.2. Data Set description

NIST releases the training data at registration time. The data includes much of previous SRE training and testing datasets (1996 up to 2010), plus newly introduced SRE 2012 data. The data is provided in SPHERE[8] file format. SPHERE file consists of a textual self-describing file header (usually 1KB), followed be raw binary data. SPHERE supports lossless compression and multi-channel recording. Moreover, some tools are published and ready to handle it.

The provided data comes in three dominant modes:

Two way conversation: such as a telephone call. The SPHERE files contain two channels, each containing one side of the speech. One of the sides is the target speaker, the other should be ignored. Echo and overlapping may be present. Different training and testing segments have different lengths and amounts of speech. The data

[8] http://www.ldc.upenn.edu/Using/

is collected in one of two methods: either using a headset (which is focused and minimizes environmental effects) or using a microphone (which can also detect other non-speech sounds and record them on the stream).

- *Interview*: The interview data is also two channel data. One channel is recorded by a headset on the interviewer part. This channel collects the speech of the interviewer and has added noise to cover the interviewee speech. The other channel is recorded by a microphone between the interviewer and the target speaker. This channel collects all the speech from both parties. All environmental effects are present. Echo and overlap is common. Also, the two channels are not perfectly aligned due to the positioning of the microphone in the room. For some earlier years interview data, an imperfect transcription of the interview is also presented in a textual form. This is to mimic Automatic Speech Recognition (ASR) approaches. The text needs to be aligned with the speech if it is to be used.
- *Summed*: the summed data is one channel phone conversation of two sides. The task is to identify the speaker even if another is present on the same channel. Overlapping is heavily present. Speaker segmentation is particularly important.

The testing data (called the evaluation data) is also presented in the same formats, although not released at the same time. The research sites are provided with the testing data at a later stage.

3.4.3. Scoring Mechanism

As described before (see section 3.3.1) the most common identification scoring mechanism is given in equation 3.9. NIST suggests using the log likelihood ratio (the log of equation 3.9) as the score throughout all SRE 2012 tasks:

$$s = \log\big(P(X|\lambda)\big) - \log\Big(P\big(X|\bar{\lambda}\big)\Big) \tag{3.15}$$

A trial is provided in the shape of a stream of non-cleared speech, along with an identity claim. Features X are extracted from the stream and the identity is mapped to one model λ. Score s represents how confident the system is of the claim. This score is a raw score for identification. However, a verification task will almost always follow to determine if the test speaker is the true speaker or an imposter. An imposter is considered to be one speaker of the training set in the known task, none of the speakers of the training set in the unknown task and either case in the core task.

Confidences are expected to be reported on each test. NIST (or the participating site) may want to conduct different evaluations on the environmental effects, the length of the streams or

the amount of speech present in them to assess their approaches or novel technologies. A maximum of 100,000 test segments will be provided (for each testing assumption case: i.e. core; extended or summed). The same core tests are used for known and unknown. A number of trials may use the same data segment with different target speaker claim. A total of 1,000,000 trails may be done in the core (also known and unknown) condition.

One interesting aspect is that no firm decision is needed as an output. The log likelihood ratio (or the score) is enough as an output because a threshold can simply be set at a later stage to maximize the number of correct associations.

A site is allowed to participate with up to three different systems for each evaluation condition. A final system's score is given by the following cost formula:

$$C_{Det} = C_{Miss} \times P_{Target} \times P_{Miss|Target} + C_{FalseAlarm} \times (1 - P_{Target}) \times \begin{pmatrix} P_{FalseAlarm|KnownNonTarget} \times P_{Known} \\ + P_{FalseAlarm|UnknownNonTarget} \times (1 - P_{Known}) \end{pmatrix} \quad (3.16)$$

The terms are defined as follows:

C_{Miss}: the cost of a miss (usually 1)

$C_{FalseAlarm}$: the cost of a false alarm (usually 1).

P_{Target}: the a priori probability that the test segment is from the target speaker (a parameter, usually in the range [0.001, 0.01]).

P_{Known}: the a priori probability that the test speaker is one of the speakers in the cohort (1 in the known task, 0 in the unknown task and ½ in the core condition).

Normalizing the scores for different participants also takes place, by dividing the raw score C_{Det} by a normalizer. This normalizer is suggested to be the score of the system with no prior knowledge about the input data, i.e. $C_{Norm} = C_{Miss} \times P_{Target}$.

3.5. Summary

This chapter has provided the most common methods for handling the tasks at hand. Namely: *speaker diarization* and *speaker identification*. It started by a discussion about the voice activity detection and how it is evaluated. Then the speaker diarization is formally introduced and approaches are described. Later, the speaker identification task was presented along with the most common approaches. Towards the end, the (SRE) tasks and rules were reviewed.

Chapter 4

Experiments

This chapter presents the results achieved while conducting this research.

- For speaker diarization, the different approaches used are discussed and the dataset explained; followed by the results and a discussion about them.
- For speaker identification and verification, the used algorithm is clearly stated and the dataset is justified; followed by the results and a discussion about them.
- Lastly, our SRE participation is presented and different issues faced are also discussed.

4.1. Project Tools

Matlab

Mathworks' **Mat**rix **Lab**oratory (Matlab) is a fourth generation programming tool primarily targeting numerical computations. Through the notion of *Toolbox*es; Matlab's functionality is extended to include many computational applications. For our certain purposes, Matlab is used to process signals and manipulate statistical data. The toolboxes used in this project are:

- *Signal Processing Toolbox*: Provided by Mathworks. The toolbox offers functionality to manipulate and visualize digital signals. Including filtering and FFT calculation.
- *Statistics Toolbox*: Provided by Mathworks. The toolbox offers algorithms for analysing and modelling data, including multidimensional analysis. The toolbox handles common techniques such as statistical distribution and hypothesis testing (Chi-squared and t-testing).
- *VoiceBox*[9]: Provided by Mike Brookes under GNU Public License. The toolbox offers functionality to deal with speech signals, such as feature extraction, speech recognition and synthesis.
- *Netlab*[10]: includes statistical techniques for data processing, including GMM modelling and adaptation, PCA and pPCA, and K-means clustering.

[9] Available online <http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox/voicebox.html> [Accessed 28 April 2012].

- *Loggmm*: a Netlab extension which contains the implementation of GMM training and adaptation in the log domain.

SoX[11]

Sound e**X**change is a cross-platform program that can read/write and interact with the NIST SPHERE file format. It is used to handle the data sets provided by NIST in the SRE 2012.

SHORTEN[12]

A piece of software that is written by SoftSound ltd. This can be incorporated with voicebox to interact with NIST's SPHERE files.

4.2. Speaker Diarization

In this section, the criterion that is usually used to evaluate a speaker diarization approach (and its sub-algorithms) is discussed. That is followed by a discussion about the data sets used to evaluate the approach empirically. Later, the found speaker diarization results are presented.

4.2.1. Evaluation Criteria

The **Voice Activity Detector** (VAD) (see section 3.1) is very hard to assess by itself because it is thought of as a first step of another system. More often than not, VAD is assessed by measuring the results of the system that VAD is part of. However, if a separate evaluation for the VAD is to be done, then it is often evaluated by the ratio of frames it miss-categorizes. Typically, using *miss rate* (VAD suggests a silence frame for a speech frame) and *false alarm rate* (VAD suggests a speech frame for a silence frame). If the VAD is parameterised (for instance, with an energy threshold) then *Receiver Operating Characteristic* (ROC) curves are used to show the effect of the parameter. The miss rate and false alarm ratios are graphed against each other for different parameter values.

The **Speaker Segmentation** (SS) task is also evaluated by false alarms and miss segmentation ratios. The output is aligned with the ground truth, and speaker turn points are expected to be close in the hypothesised output to the true turn points. Turn points that are not present in the ground truth are false alarms and true turn points that are not present in the hypothesised stream are missed turn points. In addition, ROC curves are used in a parameterized setting (metric methods). In general, it is difficult to say whether it is better to have less misses or less false alarms. Therefore, *Equal Error Rate* (EER), the value of the threshold where the false

[10] Available online <http://www1.aston.ac.uk/eas/research/groups/ncrg/resources/netlab/> [Accessed 10 August 2012].

[11] Available online < http://sox.sourceforge.net/> [Accessed 26 August 2012].

[12] Available online <http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox/shorten.zip > [Accessed 10 August 2012].

alarm rate and the miss rate are equal, is used to compare different systems which are evaluated using ROC curves (Campbell, 1997).

The **Speaker Clustering** task can be evaluated by the following measures (Kotti et al. 2008):

- *Cluster Purity*: the measure of how restricted to one speaker each cluster is.
- *Cluster Coverage*: the measure of how restricted to one cluster each speaker is.

If the clustering is used in the context of diarization, then the diarization evaluation technique might be used. That is because the clustering is usually the last step of a diarization system.

In the **Speaker Diarization** one wishes to have a one-to-one speaker/frame relation. Three kinds of errors arise from this mapping:

- *Miss*: when the system suggests a silence for a spoken frame.
- *False alarm*: when the system suggests a speaker for a silence frame.
- *Speaker error*: when the system suggests a different speaker other than the true speaker of the frame.

The final diarization error (DER) is thought of as a combination of the previous three errors' values (Tranter, 2006). Usually the speaker error is penalized by the number of error frames as to give less significance to miss classifying shorter speeches.

Another method of evaluation of a diarization system is by precision and recall (Kotti et al., 2006)

- *Precision (PREC)*: the ratio of correct associations out of all associations made.
- *Recall (RECL)*: the ratio of the correct associations out of the associations that should have been made.
- *F measure (F)*: a one value fusion of PREC and RECL given by $F_\beta = (1+\beta^2).\frac{PREC.RECL}{(\beta^2.PREC)+RECL}$.

The most widely used F measure is F_1.

In the *NIST Rich Transcription* (RT) 2009 (NIST, 2009) the error rate used is presented in the following:

$$\sum_{S\in segs}\left\{dur(S).\left(Max\left(N_{Ref}(S),N_{Sys}(S)\right)-N_{Correct}(S)\right)\right\} \quad (4.1)$$

Usually, normalized by the true segmentation error $\sum_{S\in segs}\{dur(S).N_{Ref}(S)\}$. Where $dur(S)$ is the duration of the segment; $N_{Ref}(S)$ is the number of reference speakers in the segment; $N_{Sys}(S)$ is the number of hypothesised speakers in the segment and $N_{Correct}(S)$ is the number of hypothesised speakers whom are true speakers as well. Another metric also give different weights to different speakers.

4.2.2. Testing Corpora

In the context of speech processing the *Linguistic Data Consortium* (LDC) provides a number of linguistic datasets, including *TIMIT* and *NTIMT* (LDC, 1990a; b). TIMIT has 630 speakers (438 males and 192 females) of different backgrounds and dialects; each speaking 10 English sentences in a quiet environment. NTIMIT is obtained by passing TIMIT sentences through different telephone lines. For our purposes, a random set of those utterances are coalesced to produce a mixed speech which mimics the characteristics of a normal conversation in terms of speaker change. Moreover, they can be artificially overlapped and Gaussian noise can be added to imitate ordinary speech conditions.

Other standard corpora resource is provided by the National Institute of Standards and Technology (NIST). NIST hosts campaigns for rich transcription (RT) since 2002 with an official diarization section. The corpora include news broadcast in different languages, phone calls and meetings data. In recent years, RT diarization evaluation included separate evaluations for single-channel and multi-channel diarization tasks with different settings; mounted microphones, distant microphone sets, desktop microphones with different arrangements or combinations of those settings.

Other less known corpora are sometimes also used. Such as CNET in French and KING/NKING which are similar to TIMIT/NTIMIT respectively; however, I will limit my experiments to the TIMIT/NTIMIT corpora.

In the coming experiments two datasets are used; namely: *TIMIT* and *NTIMIT*. A number of speakers are selected and then a number of their sentences are coalesced to create a dataset. The original change points and speaker identity are kept as a reference for evaluation (ground truth). The difference between the datasets is in the bandwidth, and therefore the amount of information contained in the streams. TIMIT consists of clear speech while NTIMIT is phone speech with added noise. The datasets are also extended with more sentences for each speaker. Those extra sentences are used to build individual speakers' models in order to use a model-based segmentation or diarization approach.

4.2.3. System Flow

The system for evaluating the **segmentation** process is presented in Figure 4.1. The system starts by reading the stream and removing the silence parts using an energy based approach. Then features are extracted (DNA, MFCC or any representation that is needed).

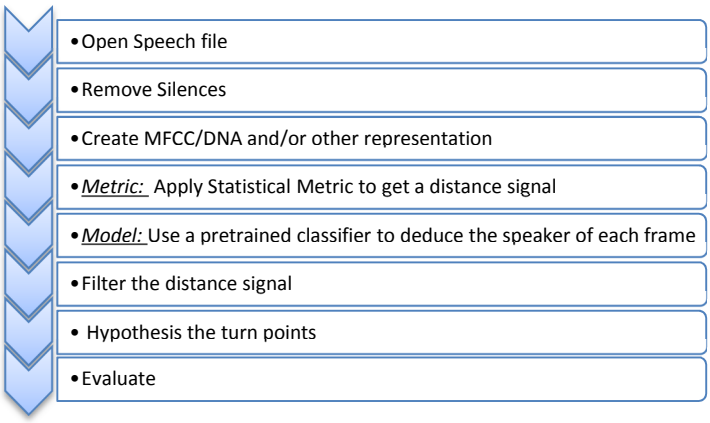


Figure 4.1 The main process of segmentation evaluation

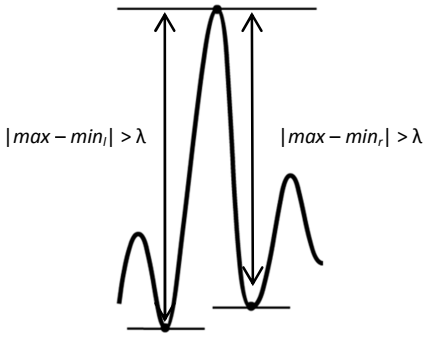


Figure 4.2 The threshold role in analysing the distance signal

In metric-based systems, a window is slid across the representation stream and a distance signal is generated; which later is filtered keeping only lower frequency changes, i.e. long range changes. Speaker changes are expected to be on the local maxima of the filtered distance signal, but only if the maximum satisfies enough change (surpasses a threshold) on both sides (see Figure 4.2), otherwise it is ignored. If the change was persistent (i.e. a long range with high distance value) it does not mean a speaker change, rather it may mean different non-speech effects on the stream.

In model-based systems, the window is used to classify its content as one of the speakers. After that a list of hypothesised turn points is evaluated. The model used in all tests is a Mono-modal Gaussian distribution $\mathcal{N}(\mu, \Sigma)$ for each speaker. This claim is made because the deep neural network is discriminatively learning the first and second order statistics. Therefore, it is not wrong to assume that the DNA representation follows the Normal distribution. Nonetheless, Using GMM speaker model may provide better accuracy. However, we will use GMM in the identification process and compare it to the normal distribution. A variation of the model is the *Anchors model* (see 3.2.4): a number of hyper-speakers are selected and each frame is represented by its distance to each of those hyper-speakers (i.e. speaker characterization vectors). Hence, using 32 speakers will result in a 32-dimensional feature vectors. The distance measure used is the Mahalanobis distance (the test segment is also modelled as a mono-modal Gaussian):

$$F(x|\mu, \Sigma) = (x - \mu)^T \Sigma (x - \mu) \tag{4.2}$$

These processes involve the correct choice of a low pass filter and its parameters. This task must be done manually because it is related to the nature of the recording. Moreover, the slight distortions caused by the filtering step are causing segmentation accuracy deterioration.

For **clustering**, offline clustering techniques might be used (such as K-means) if the number of clusters (speakers) is known a priori. Otherwise, online clustering techniques are used (see section 3.2.7). However, we could not get proper results because the speaker models are not

linearly separable. In Figure 4.3 a number of speaker models' first and second principle values are presented (the model is the mean value of the DNA). Clearly, they are very close to each other. This same problem arises in the speaker identification task, especially if a discrimination approach is to be used. Nevertheless, better DNA tuning or using only vowel segments would improve the clustering results. In this report we did not conduct more clustering research because they would provide no value to us.

The best segmentation results were achieved by following this process:

- A fixed size window of 120 frames is slid across the stream with 30 frames shift.
- For every window, a comparison to all the speakers is made and the closest model is voted the speaker of that window.
- Only the difference between each two consecutive windows is kept. That turns out to be the turn points' signal.
- To improve stability, the previous signal is low-pass filtered and further processed much like in the metric approach.

Turn points are expected to be maxima values of the signal with enough distance on both sides (this threshold is the parameter for the process).

4.2.4. Results

The segmentation is evaluated using ROC curves. By using DNA representation in a fixed-window metric approach on a TIMIT–based dataset with different metrics, in addition to the KL2 metric using MFCC representation, we get the results presented in Figure 4.4(a). Figure 4.4(b) shows the same comparison on a NTIMIT-based dataset. Different window sizes involve different information amounts which lead to different results. It is clear that the DNA representation is

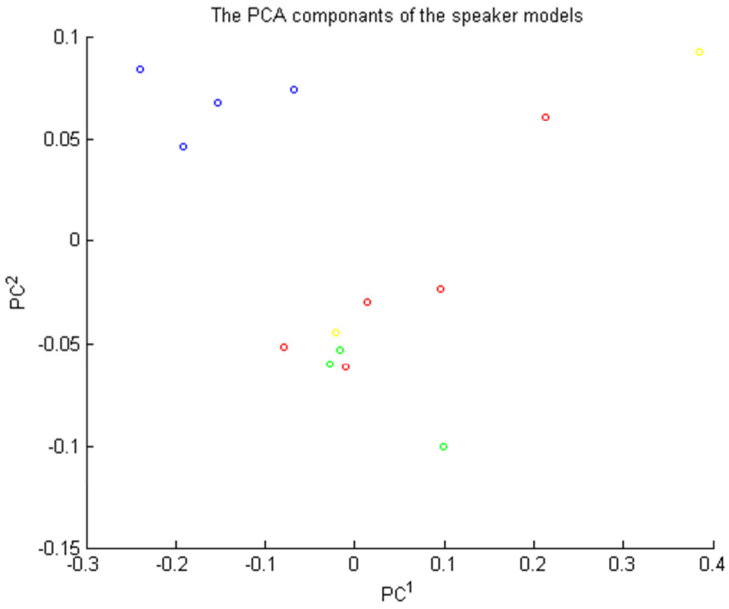


Figure 4.3 The speaker models' first and second PCA.

performing better than the MFCC representation under similar conditions. Also, that there is no one best metric for all cases. However, it seems that the modified KL metric is performing well in all cases.

Figure 4.5(a) and Figure 4.5(b) show similar test results to the ones presented in figures 4.4(a;b) using the *anchors model* (by selecting 32 hyper-speakers and relating the frames to them) in comparison to the KL metric. Figure 4.5(a) is the results on a TIMIT-based dataset and Figure 4.5(b) is the result on a NTIMIT-based dataset. The results are less accurate than ordinary metrics (with higher miss rate) because of the loss of information that takes place when reducing the number of features to only 32.

Following a model-based approach, Figures 4.6(a) and 4.6(b) show the results for TIMIT-based and NTIMIT-based datasets respectively. The model is a Normal distribution that is acquired using a separate set of sentences for each speaker in the stream. Then the data in the window is also modelled by a normal distribution and the modified KL metric is used to measure the distance to each speaker in the set. An assumption is made that the speaker is present in the provided set of speakers.

(a)

(b)

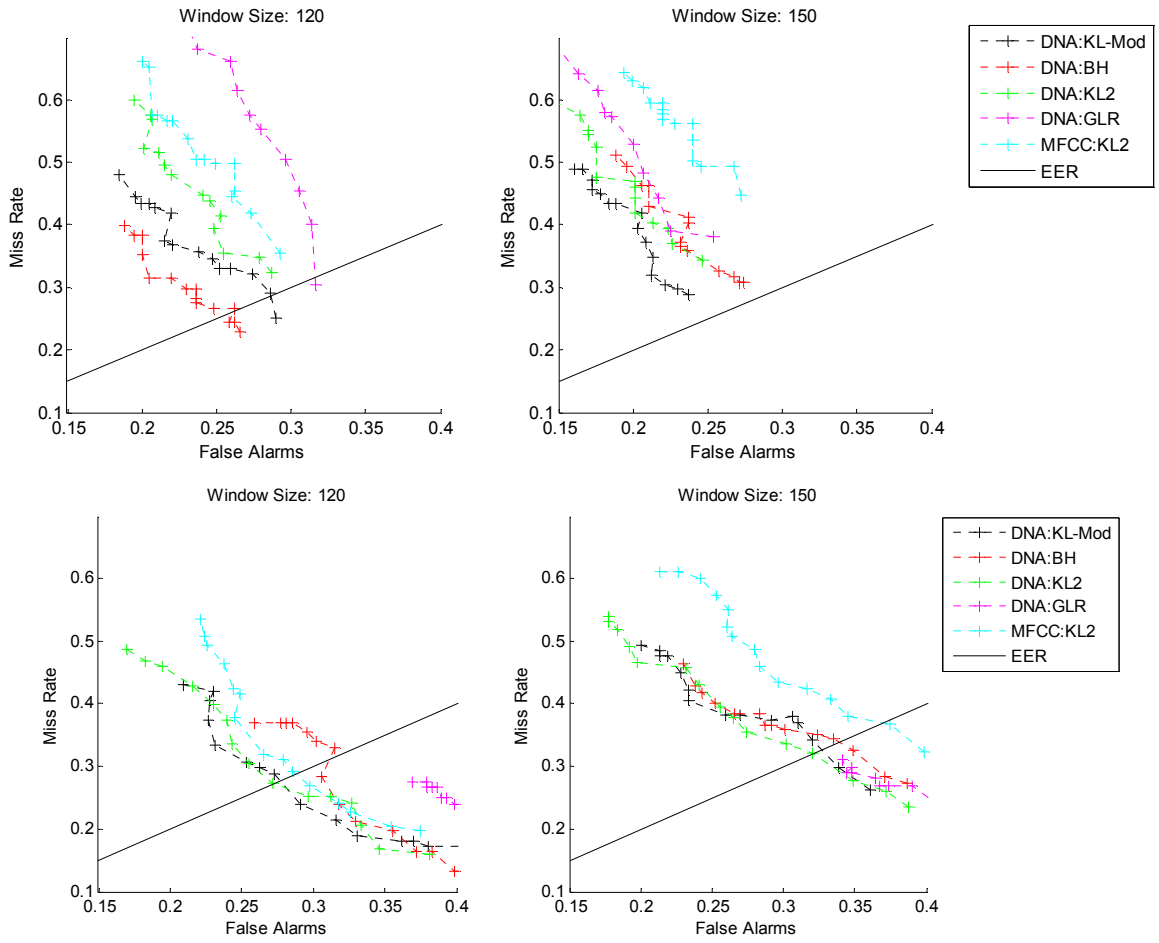


Figure 4.4 ROC Curve for metric-based segmentation of a (a) TIMIT-based (b) NTIMIT-based Dataset

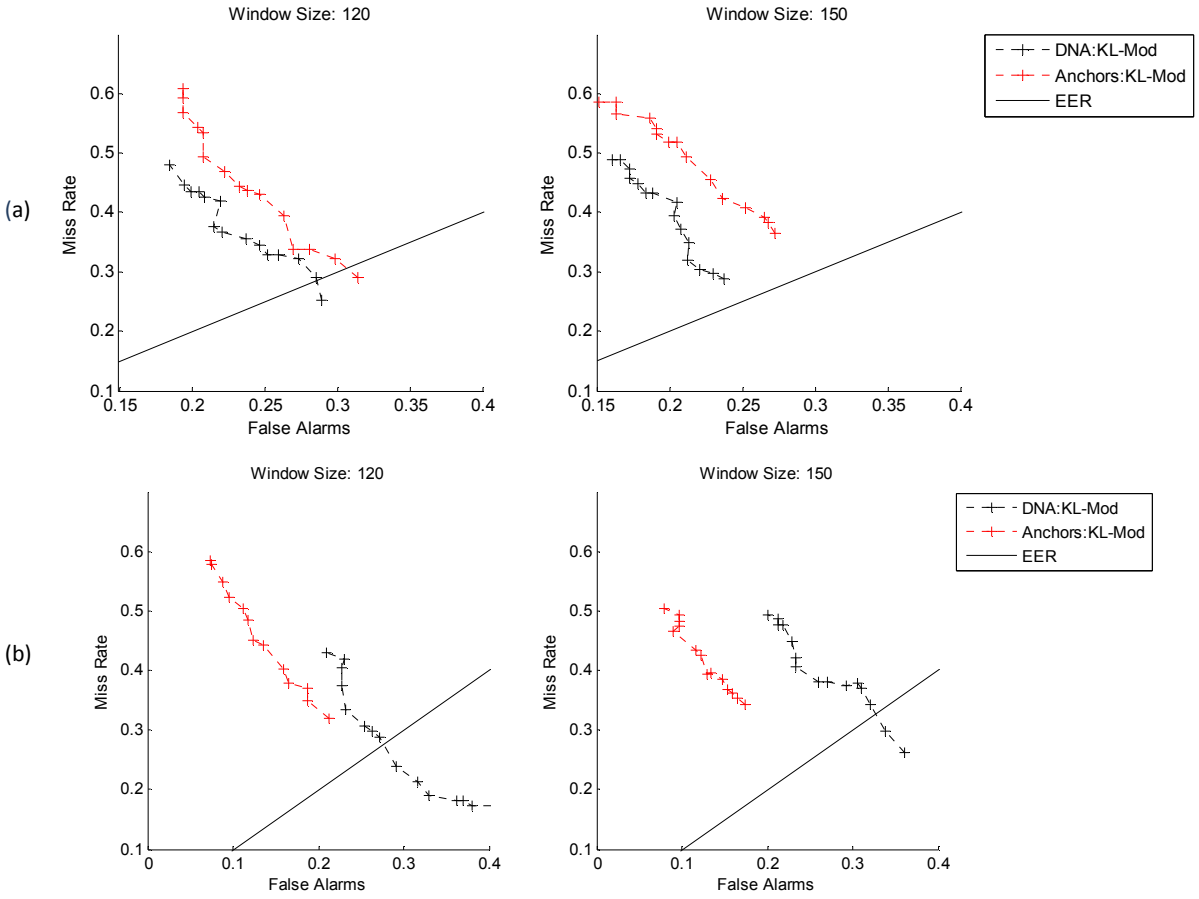


Figure 4.5 ROC Curve for metric-based segmentation of a (a) TIMIT-based (b) NTIMIT-based Dataset using Anchors model

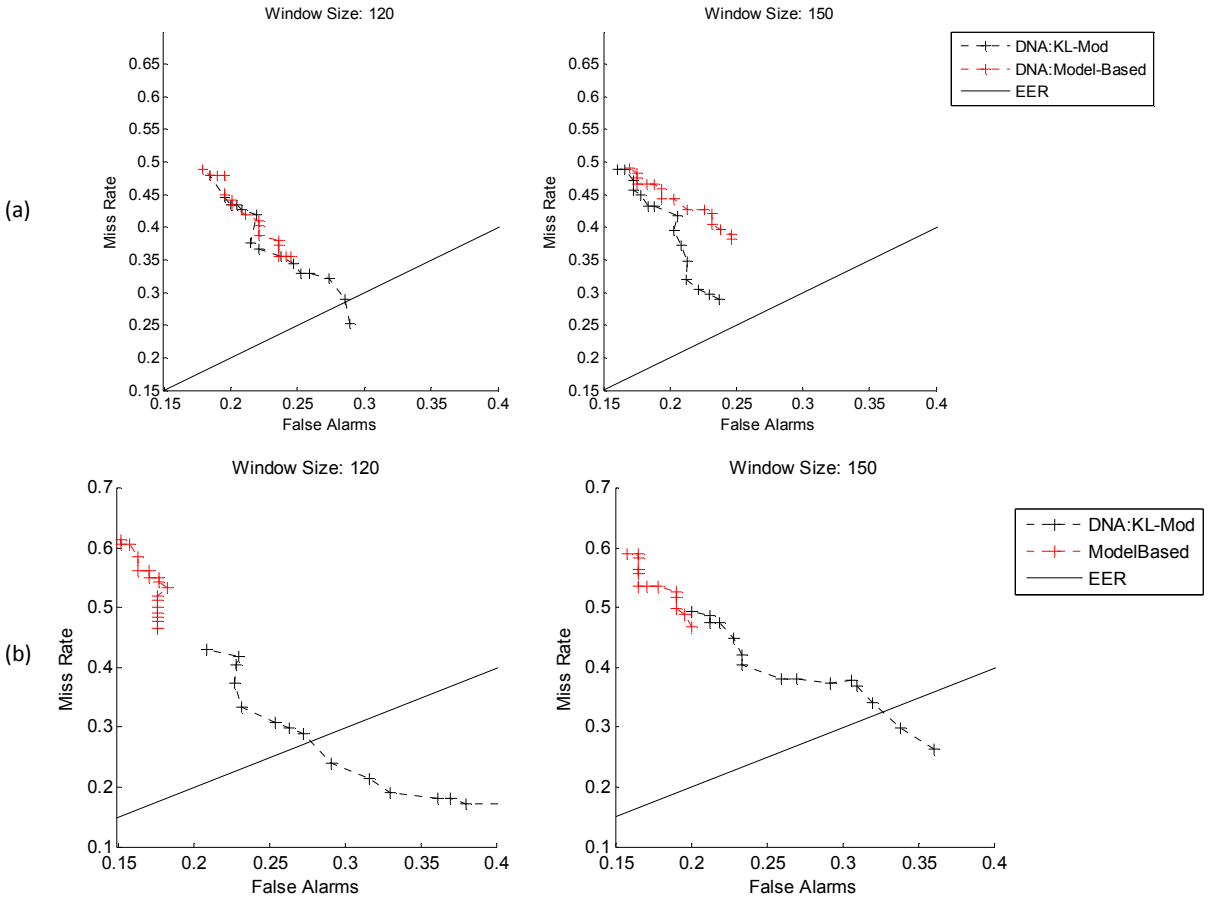


Figure 4.6 ROC Curve for model-based segmentation of a (a) TIMIT-based (b) NTIMIT-based Dataset compared with KL modified

4.2.5. Summary

In this section evaluation methods for speaker segmentation and diarization were introduced. Also, the datasets used to empirically evaluate a system were described. Many segmentation systems were presented and their results detailed. Clustering was described and its poor performance was explained. Speaker segmentation will be used in the SRE participation later (precisely in the interview VAD process). No more research is carried out for the speaker clustering task, for that would have no value towards SRE.

4.3. Speaker Identification

In this section, common evaluation criteria for speaker identification (and its sub-algorithms) are discussed. That is followed by a discussion about the data sets used to evaluate our approach empirically. Finally, the found speaker identification results are presented.

4.3.1. Evaluation Criteria

In a *closed-set* environment, it is most convenient to evaluate the system by the percentage of correct associations made using a large number of trials (test stream with an identity claim). Systems may be compared based on these numbers.

Similarly, in the open-set environment, different ratios are estimated; such as the *miss rate* (the system suggests an imposter for an inset speaker trial) and *false alarm* (the system suggests a speaker for an importer speaker trial). Usually, a threshold is needed to make a firm decision on identity. ROC curves can be used to show the effects of changing this threshold value. Similar to evaluating the segmentation task (see section 4.2), systems may be compared based on their EER. Reynolds (2002) presents a discussion for different speaker verification tasks (text dependent and text independent) using then state-of-the-art techniques. Although some advancement has taken place during the past decade; the general trend in the accuracy holds valid. Figure 4.7 is recreated from (Reynolds, 2002) and it shows the EER for the following systems:

- Sys1: text-dependent, single phrase, 3 minutes of training and 2 seconds of testing.
- Sys2: text-dependent, digits only, two phrases of training and a single phrase for testing.
- Sys3: text-independent, clean data, 2 minutes of training and 30 seconds of testing.
  Sys4: text-independent, noisy data, 30 seconds of training and 15 seconds of testing.

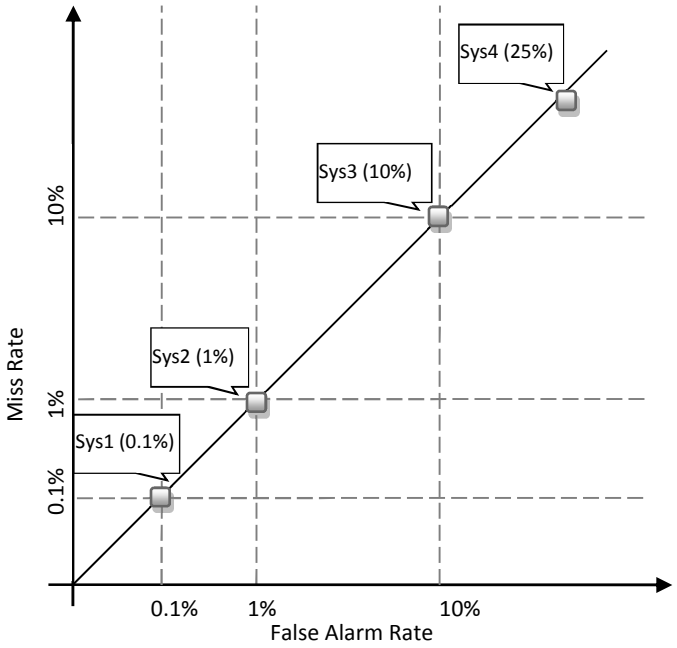


Figure 4.7 Different speaker recognition tasks performance

Common techniques make the assumption that the stream contains only one speaker (or only one identity is present in the stream). However, a more general approach is presented in (Reynolds and Rose, 1995) where the number of correctly associated windows is calculated (rather than streams). A final evaluation criterion would be the percentage of correctly identified windows (out of all the trails' windows) as in equation 4.3:

$$Accuracy = \frac{number\ of\ correctly\ classified\ windows}{number\ of\ all\ windows\ in\ the\ test\ trials} \times 100 \tag{4.3}$$

This approach copes with many speakers being present in the same stream. Nevertheless, it does not produce a firm decision of who is speaking in a certain input stream.

If it is not needed to produce a firm decision, the log likelihood may prove a good scoring technique. That is, instead of making an association of the test stream to one speaker model, we can produce the likelihood ratio (or the log likelihood ratio LLR) as a score of how confident the system is about the stream to contain the said speaker (see section 3.3.1)

$$L = log\left(\frac{P(data|\lambda)}{P(data|\bar{\lambda})}\right) = \log\big(P(data|\lambda)\big) - \log\Big(P\big(data\big|\bar{\lambda}\big)\Big) \tag{4.4}$$

The reader should realize that this score may further need normalization (see paragraph 3.3.3.3) in order to stabilise the output scores and make it easier to estimate any threshold that might be needed. Empirically this has shown to improve the accuracy as well. A full system is harder to evaluate using only the likelihoods of each trial. However, after having a list of all trials' confidence values it is easy to threshold the results and make associations at different thresholds, then evaluate accordingly.

4.3.2. Testing Corpora

NIST provides a data set for each SRE round they administer, i.e. SRE 2008, SRE 2010… etc. Those datasets are recorded to reflect the challenges of real speaker recognition applications tasks at different periods of time. For instance, in SRE 2008, the transcriptions of the interviews were supplied in textual form, while in 2012 the interview data is provided on two channels: one channel contains the interviewer (non-target) speech and added noise covering the interviewee (target) speech; while the other channel contains the full interview. There are also changes to the phone (cellular and microphone) types used to record the data as to cope with the technology changes that take place in that direction as well. The dataset is further explained in section 3.4.

NIST SRE 2012 corpora are extremely extensive. Most of them are bits of older SRE training and testing corpora. Therefore, we do not need any other dataset. Moreover, the results we achieve using this corpora would be helpful for our SRE participation.

In the coming experiments a data set of 20 speakers was selected to include both genders, all phone data and only one speaker per stream. A number of streams were merged to make at least 100 second of training. Another stream was selected for each speaker as the testing stream (may be as short as 6 seconds). Another 10 speakers were later added to the pool as imposters with one testing stream each.

The selected number of speakers is not very large. However, it is a representing set of the SRE phone data; also it is large enough to compare different approaches without causing extremely long running times.

4.3.3. System Flow

The systems for evaluating the identification processes are presented in Figures 4.8(a) and 4.8(b). Both processes include voice activity detection and feature extraction steps, which should be identical in between the training and testing; i.e. the same VAD and the same representation should be used. If a supervised method is selected for VAD (using MFCC or DNA features), then feature extracting may precede the VAD step.

Training includes two separate parts:

- *Training the DNA for a specific application*: much like any other supervised auto-associator, the best results would be achieved if the DNA was trained on the specific group of speakers; as to permit the DNA to learn the discriminative features well. Otherwise, a generic DNA might be used. In that case, the performance would be affected, i.e. the DNA features would be biased in a way which reflects the bias of the

original training set (more males vs females, or using a certain language or environment). Hence, although DNA features have shown better results than MFCC in the segmentation task (especially in a model-based approach, see section 4.2); the accuracy may dramatically deteriorate when the DNA is applied to a totally different corpora. In our case, due to lack of time, I used a generic DNA representation for the identification task. Which led to results worse than or comparable to the MFCC.

- *Training the Speaker Models*: when mono-Gaussian or direct VQ models are used, the speaker models are directly extracted from the feature vectors. When GMMs are used (which is the most common) a UBM must be trained first, which is later used to adapt individual speaker models.

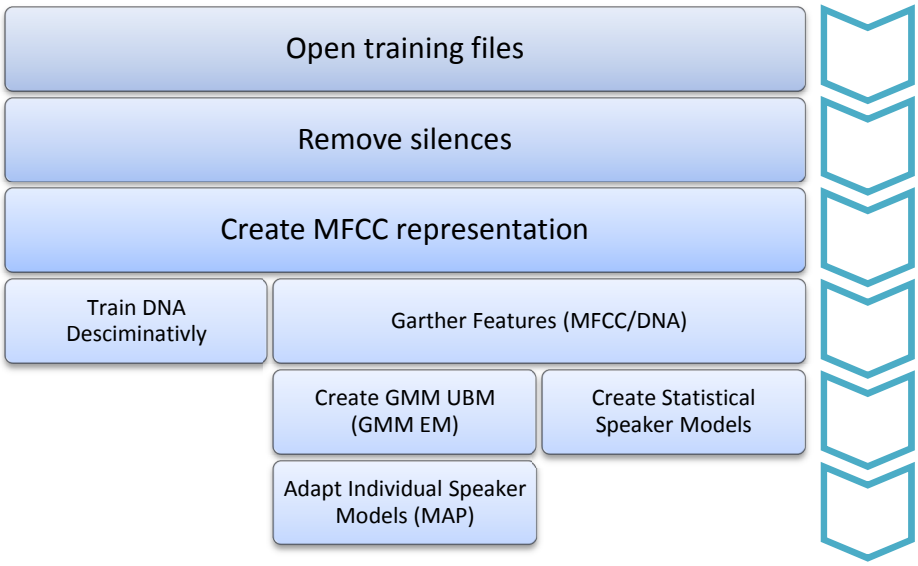


Figure 4.8 (a) Speaker Identification training processes

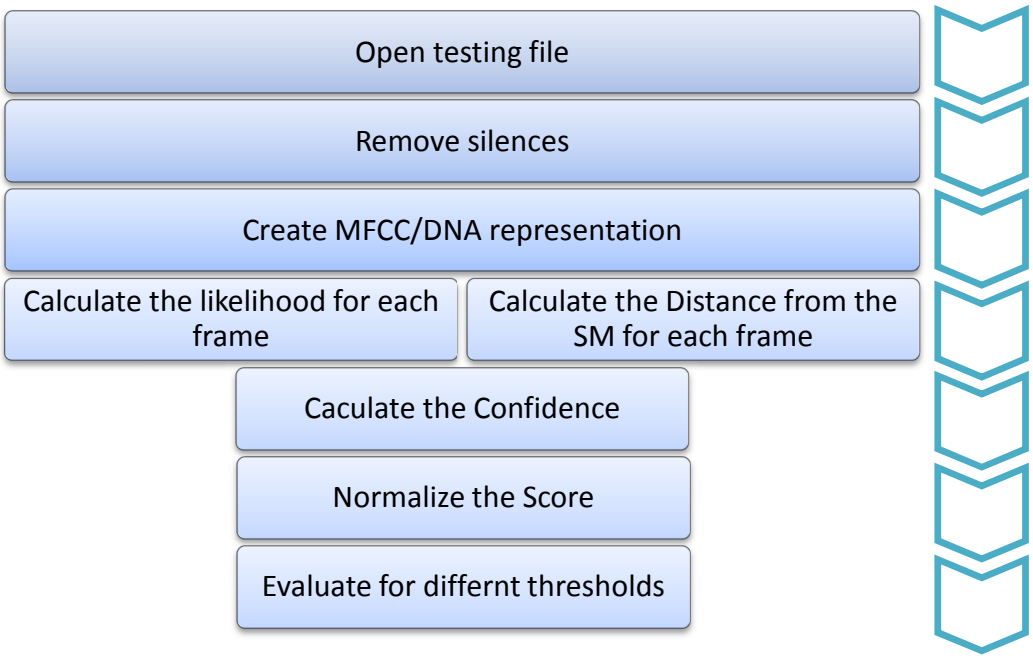


Figure 4.8 (b) Speaker Identification testing processes

During speaker training, eigen voices and eigen channel vectors may be trained depicting the training set traits. More advanced techniques may be used (such as JFA or feature mapping) to provide finer speaker models. This training part ends by storing the speaker models in a speaker model database, which would later be used in the identification task.

Testing for one window may be in one of the following forms:

- *The Likelihood approach*: individual feature vectors are thought of as independent observations. No temporal alignment is considered in the text-independent identification. Therefore, the likelihood of a window given a speaker model is the product of the likelihoods of individual frames given the same model, or the sum of logs of the likelihoods in the log domain.
- *The Distance approach*: the frames in the window are used to form a new speaker model. Later, the divergence between this model and different speaker models is calculated. As seen in paragraph 3.3.3.4 the distance may be converted to a likelihood value.

In both approaches, an aggregation function is needed to produce a confidence value for the whole stream out of the different windows' likelihoods. The general function is the average function. However, (Gish and Schmidt, 1994) proposes three different aggregation functions:

- NoSeg: the entire stream is passed as one segment to the scoring mechanism.
- SumSeg: the windows' likelihoods are summed (or averaged).
- TopNSeg: the top N scoring windows are averaged only. The idea is that in long streams a number of windows must match the training model better than others. We only keep those. If the models are not highly tuned, this approach becomes less stable than using all the windows available.

In their experiments they report comparable results between the last two approaches, are outperforming the first.

If only a firm decision is required, then the model with the highest probability is used. If the LLR score is also needed (see section 4.3.1) then a likelihood of the non-target speaker is also needed. This can be the likelihood of the Universal Background Model; or a carefully selected cohort of speakers (Sturim and Reynolds, 2005). Reynolds (1995) suggests that the "closest" N models should be used as a cohort, which may be calculated offline using a divergence approach.

The final score is usually normalized as to fit the scores' distribution to a known model, which makes it easier to set a universal threshold for verification.

4.3.4. Identification Results

4.3.4.1. Closed Set Results

The closed-set identification is done in many approaches. Namely: MFCC-MAP, DNA-MAP, DNA-Cohort, DNA-Mono_G and DNA-Dist.

MFCC-MAP: The Base-Line system. It uses MFCC (20 features) with GMM adapted speaker models. We use the simple scoring mechanism:

$$speaker\ identity = \text{argmax}_i\big(\log\big(P(data|\lambda_i)\big) - \log\big(P(data|UBM)\big)\big) \quad (4.5)$$

Depending on the number of components used in the GMM and the number of frames in a window (the size of the window), different results are achieved.

DNA-MAP: Likewise, DNA features (100 features) are used in the same scenario. Similarly, different number of components used in the GMM produces different results.

Figure 4.9(a) shows the results of using different number of components for GMM training using different window sizes. 3 different numbers of components are tested: 8, 32 and 128. The stability received against the changing window sizes is expected, because an assumption is made regarding the features' vectors being independent. Therefore, the same likelihoods are being averaged with little difference made when changing the window size. On the other hand, different number of GMM components is also causing stability. This can be explained by the fact that only few data points are used, and therefore 8 components fit them as well as 128 components.

Figure 4.9(b) shows the same results when using DNA features. Also, the stability against the window size is expected for the same reason. However, different number of GMM components is performing differently. Few components (8 components) are not fitting the high-dimensional data well. Also, many components (128 components) are over fitting the data and causing issues when generalizing and calculating the likelihood. However, for this size of data the number 32 components generated the best results; perfectly assigning the 20 testing speakers regardless of the window size we use. In the following experiments, the base-line system used is the *GMM-MAP* reported in the Figure 4.8(a). Different approaches are tested for comparison.

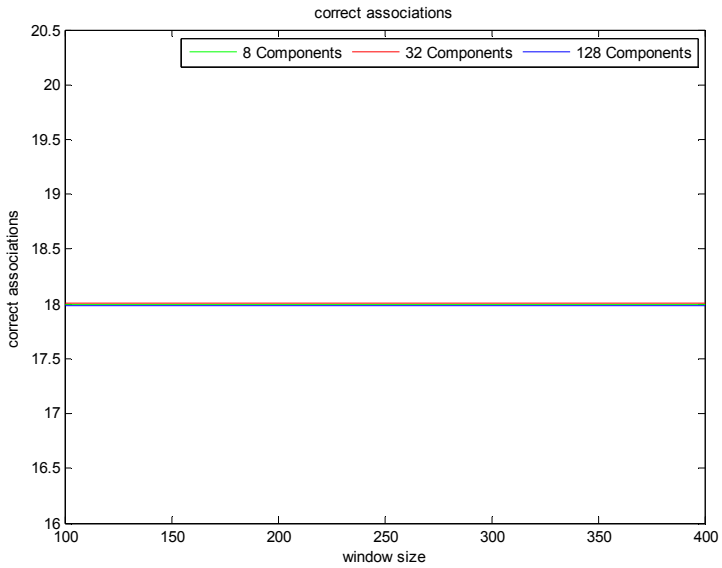


Figure 4.9(a) The effects of using different GMM components number using MFCC-MAP approach (for different window sizes).

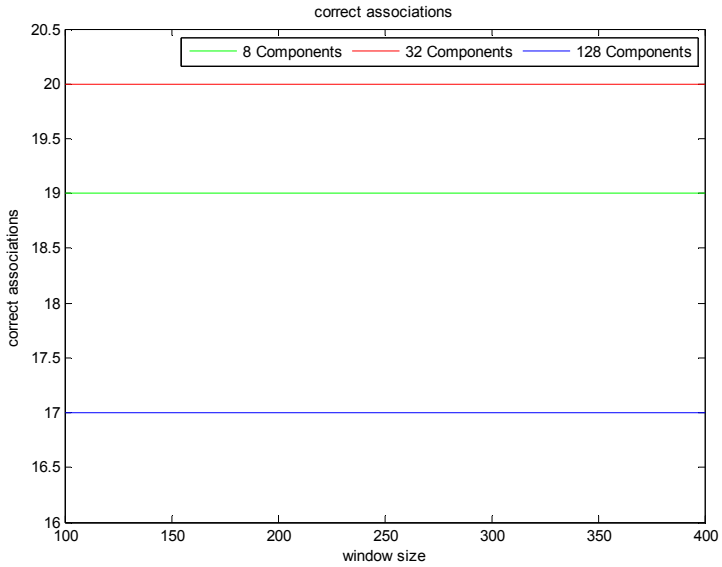


Figure 4.9 (b) The effects of using different GMM components number using DNA-MAP approach (for different window sizes).

DNA-Cohort: As suggested earlier, it is also possible to use the scores of the N closest speakers as a cohort for the calculating $P\left(data\middle|\bar{\lambda}\right) = avg\left(P(data|cohort_i)\right)$. The results are presented in Figure 4.10 against the MFCC-MAP. The closest N speakers are determined offline, by the use of KL2 divergence between the mono Gaussian models representing the speakers. For each speaker, other models are ranked by their distance. This distance may be affected by the training environment and the amount of available speech. Nonetheless, using the DNA features minimises those effects. The cohort size N may affect the results; we test for two values 5 and 10 speakers.

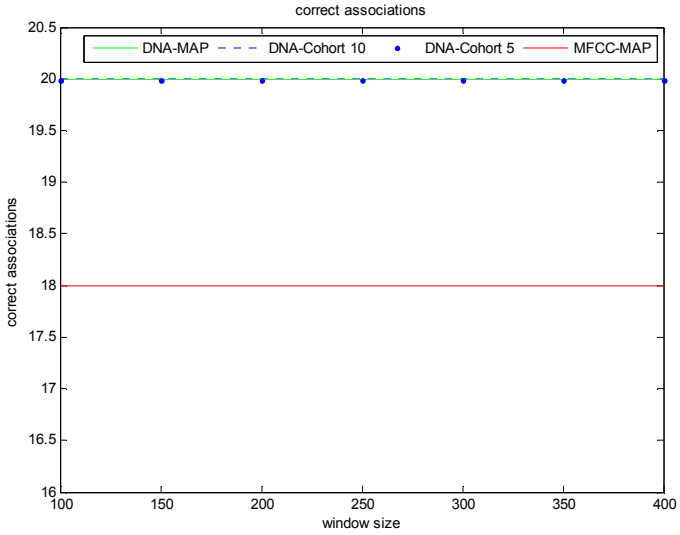


Figure 4.10 The effects of using adaptive non-target cohort methods DNA-MAP approach (for different window sizes).

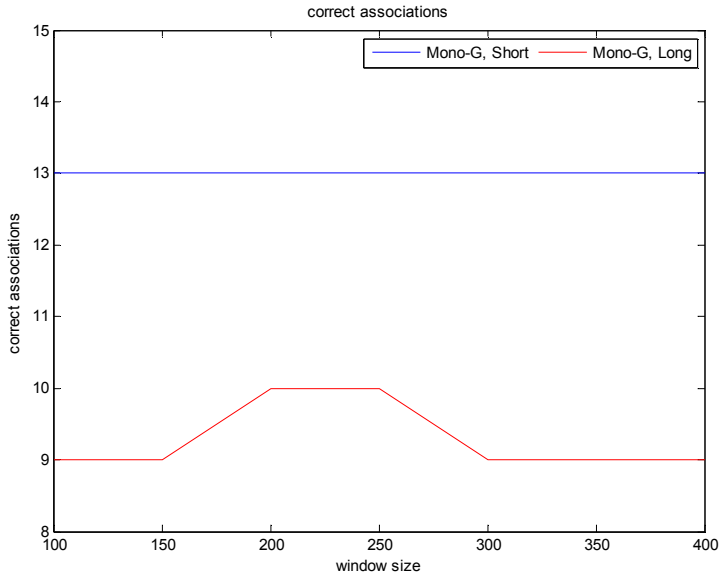


Figure 4.11 The effects of using different training data stream lengths when using DNA-MonoG approach (for different window sizes).

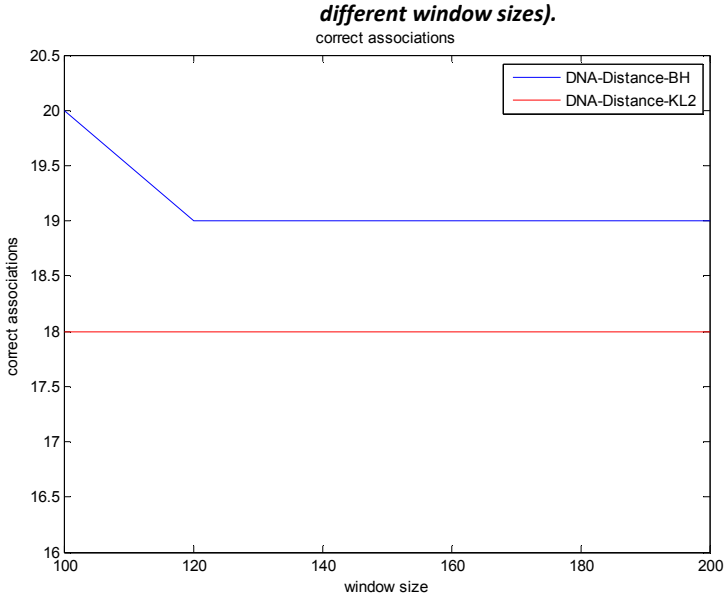


Figure 4.12 The effects of using different divergence measures when using DNA-Dist approach (for different window sizes).

The stability over different window sizes is also expected. However, the fact that *DNA-Cohort* is generating the same results as *DNA-MAP* is encouraging towards using only one method as both of them are comparably good and better than *MFCC-MAP*.

DNA-Mono_G: the speaker model is the mono Gaussian. The training can be done in two types: using all available training data or using only few points of training data. These types are compared to view the effects of using fewer training data with DNA features.

Figure 4.11 shows the results of short and long training data (the short data is 20 second of training data cropped from the middle of the training stream) to build mono-Gaussian models. Shorter training data is performing better than longer training data; that can be merely attributed to the presence of more speech and less noise in the shorter utterances. However, the remarkably poor performance of both tests is due to the fact that mono-Gaussians do not present a discriminating model for the data. In fact, mono-Gaussians are good to model one speaker, but fall short to separate the models (see section 4.2.3 the clustering results). We can make the conclusion that mono-Gaussian speaker models are much overlapped. However, as we will see, the divergence approach is achieving much better results (while using the mono Gaussian model).

DNA-Dist: The previously discussed methods are all generative models. As a comparison, a test was conducted using distance measures (divergence). Two measures are used for comparison reasons only. The results are presented in Figure 4.12. The window sizes used here are smaller than the window sizes used in the likelihood approach; because in the distance approach, we use the data in the window to build a model and compare it to different speakers' models. Including a large number of frames would distort that model (as seen in the segmentation testing; window sizes around the value of 120 frames are often used).

4.3.4.2. Open Set Results

Open-set task differs from the closed-set task by the evaluation technique. A threshold is needed to be set for the verification part of the open-set task (make the decision whether the speaker is who they claim to be or an imposter from outside of the group). Three approaches are evaluated using ROC curves (false alarms against misses' plots).

- MFCC features GMM modelled adapted from a UBM (MFCC-MAP).
- DNA mono-Gaussian models using the distance approach (DNA-Dist).
- DNA features GMM modelled adapted from a UBM (DNA-MAP).

For each approach, the system is tested for four different background speakers' settings:

- Using the results likelihood by itself with no cohort normalization. i.e. $\log P(x|\bar{\lambda}) = 0$
- Using the likelihood of the UBM as a cohort probability. i.e. $\log P(x|\bar{\lambda}) = \log P(x|UBM)$

- Selecting a cohort of the N most similar models (according to the KL2 distance) to the speaker model. i.e. $\log P(x|\bar{\lambda}) = \frac{\sum_{ci=1}^{N} \log(P(x|\lambda_{ci}))}{N}$ (Two settings are presented, 5 and 10 speakers, to check the effect of the cohort size).

ROC curves are used to present the results; the curve represents the misses and false alarms at different thresholds. Nonetheless, for different window sizes there would be different curves (coded by different colours, the different curves for different window sizes are presented in the following set of results). However, these curves do not show the number of correct associations made (trivially, if zero errors were present, then a perfect association set had been achieved). For that reason, each ROC curve is coupled with another plot that shows the number of correct associations at different window sizes and thresholds.

The following figures present the results of the said approaches:

Figure 4.13 shows the results of using *MFCC-MAP* approach (32 components GMM was used) as follows: (a) Represents the MFCC-MAP using only the likelihood; (b) represents the MFCC-MAP when using a full UBM as a non-target model; (c) represents the MFCC-MAP using a large cohort (10 speakers) as a non-target model; and (d) represents the MFCC-MAP using a small cohort (5 speakers) as a non-target model.

Figure 4.14 shows the results of using *DNA-Dist* approach (KL2 divergence was used) as follows: (a) represents the DNA-Dist using only the likelihood (distance converted to likelihood); (b) represents the DNA-Dist when using a full UBM as a non-target model; (c) represents the DNA-Dist using a large cohort (10 speakers) as a non-target model; and (d) represents the DNA-Dist using a small cohort (5 speakers) as a non-target model.

Figure 4.15 shows the results of using *DNA-MAP* approach (32 components GMM was used) as follows: (a) represents the DNA-MAP using only the likelihood; (b) represents the DNA-MAP when using a full UBM as a non-target model; (c) represents the DNA-MAP using a large cohort (10 speakers) as a non-target model; and (d) represents the DNA-MAP using a small cohort (5 speakers) as a non-target model.

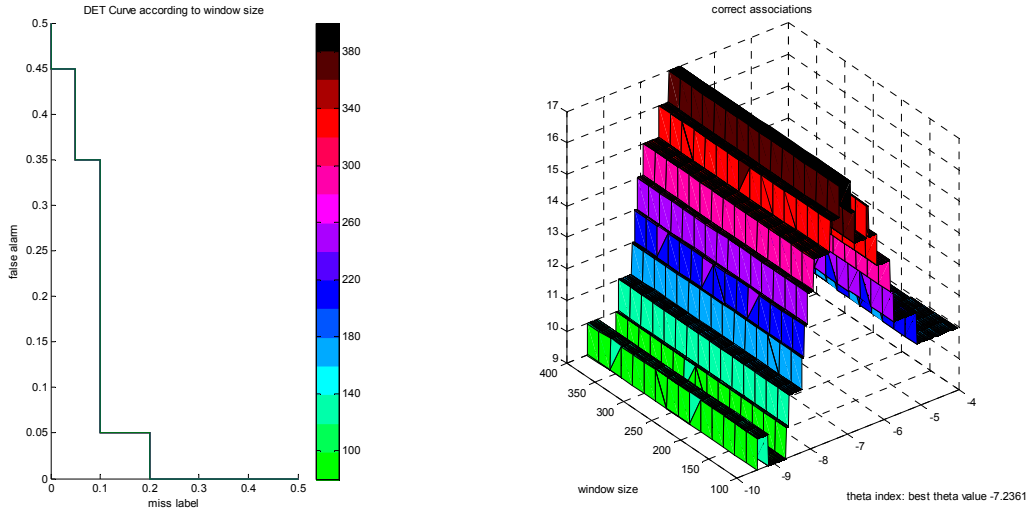


Figure 4.13(a) The results of using MFCC-MAP approach.

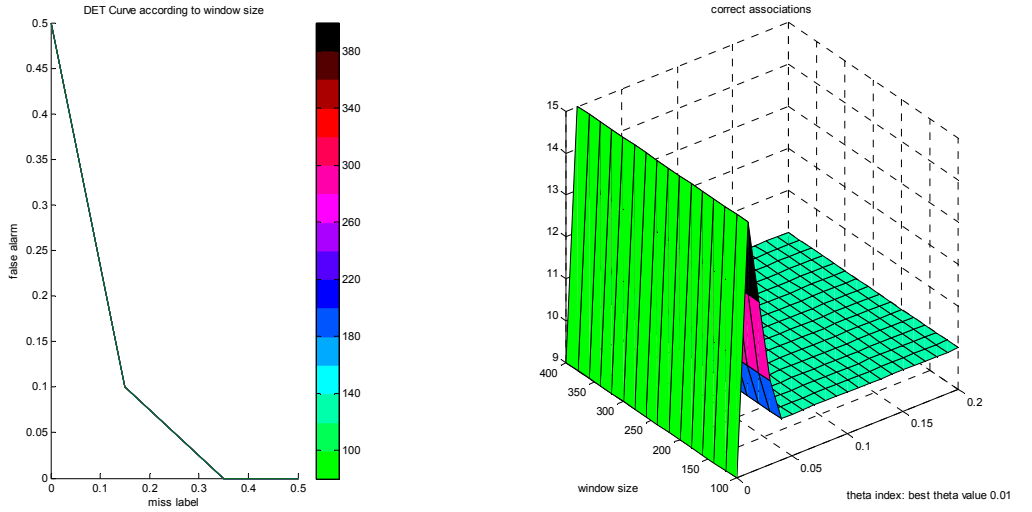


Figure 4.13(b) The results of using MFCC-MAP approach when using a UBM.

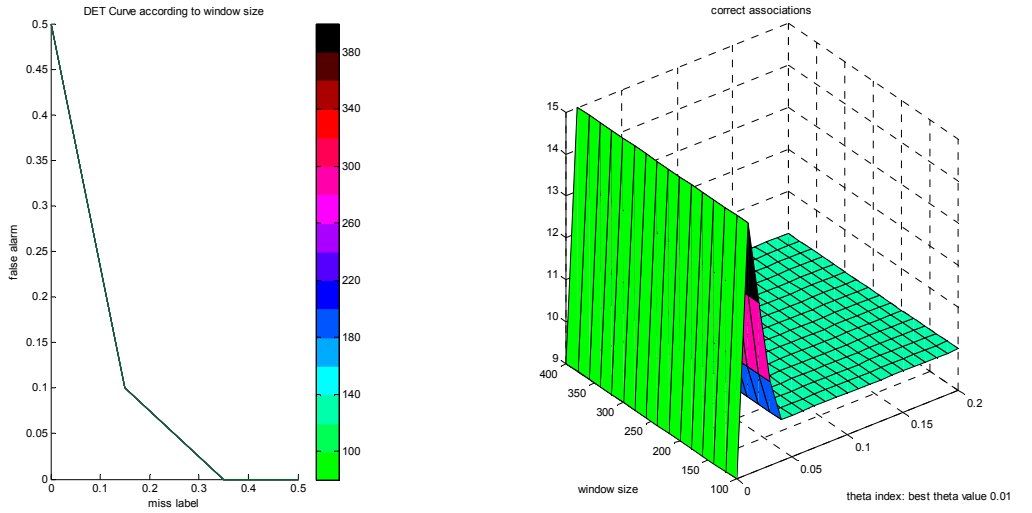


Figure 4.13(c) The results of using MFCC-MAP approach when using a when using a large cohort.

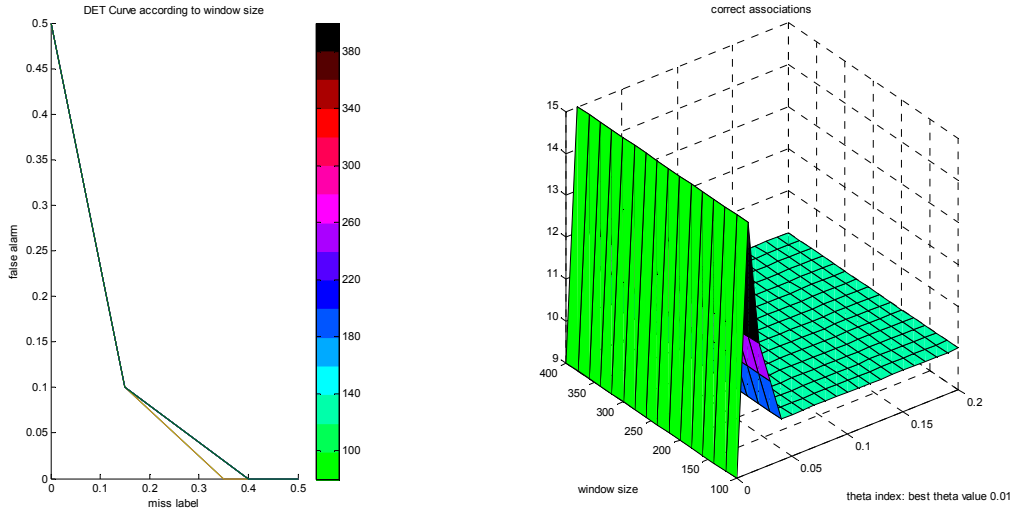


Figure 4.13(d) The results of using MFCC-MAP approach when using a when using a small cohort.

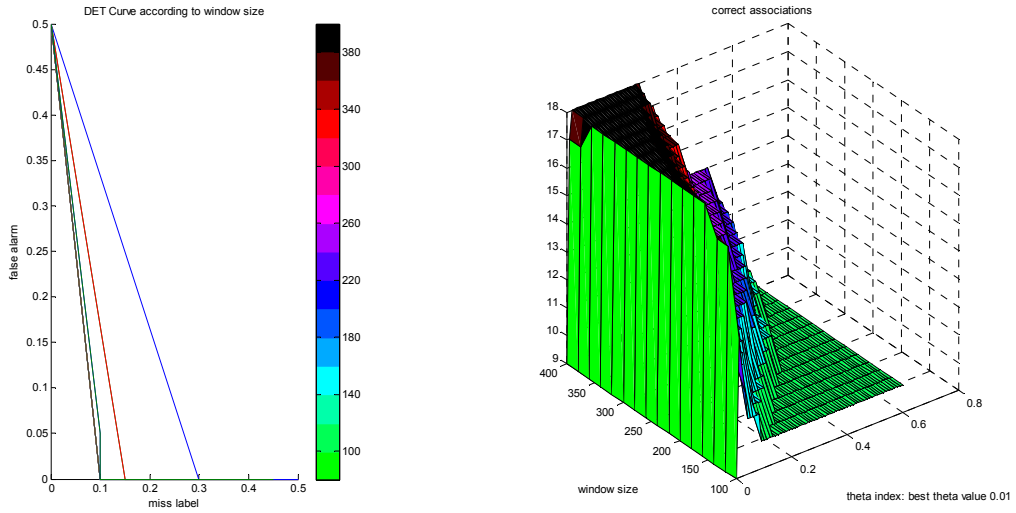


Figure 4.14(a) the results of using DNA-Dist approach

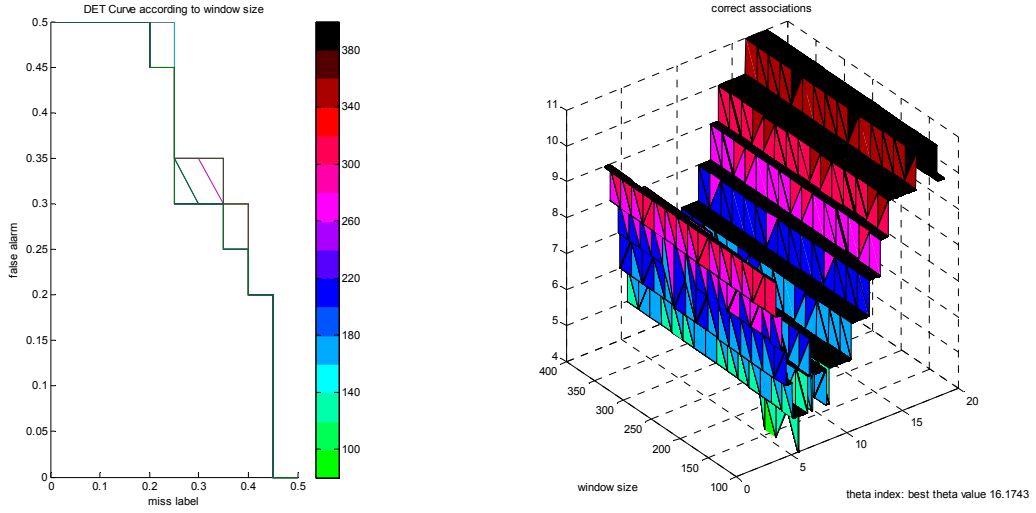


Figure 4.14(b) the results of using DNA-Dist approach when using a UBM.

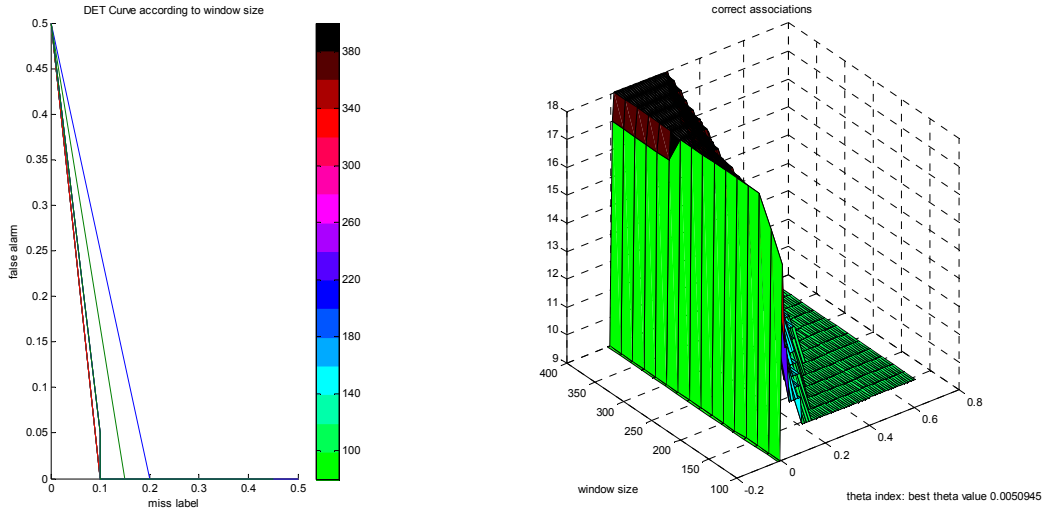


Figure 4.14(c) the results of using DNA-Dist approach when using a UBM when using a large cohort.

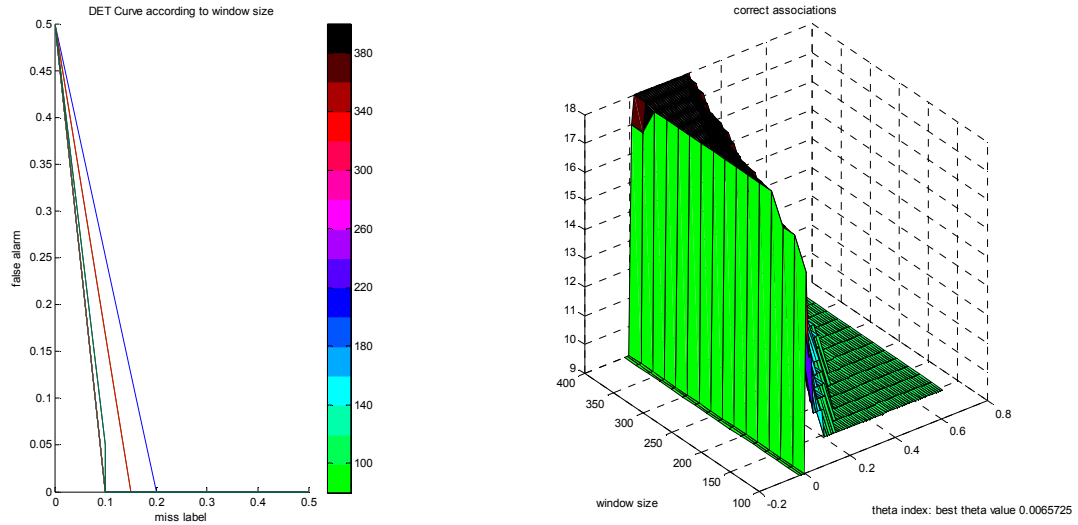


Figure 4.14(d) the results of using DNA-Dist approach when using a UBM when using a small cohort.

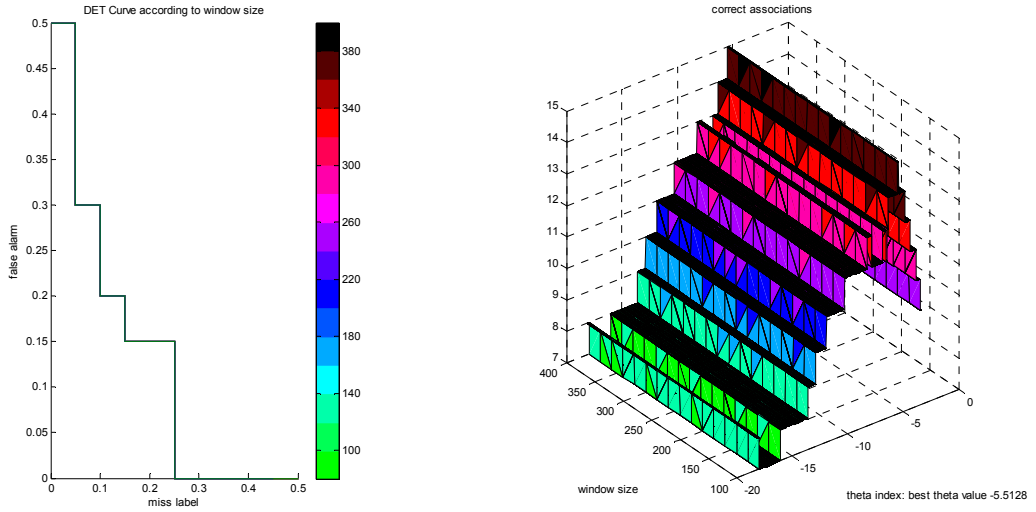


Figure 4.15(a) the results of using DNA-MAP approach.

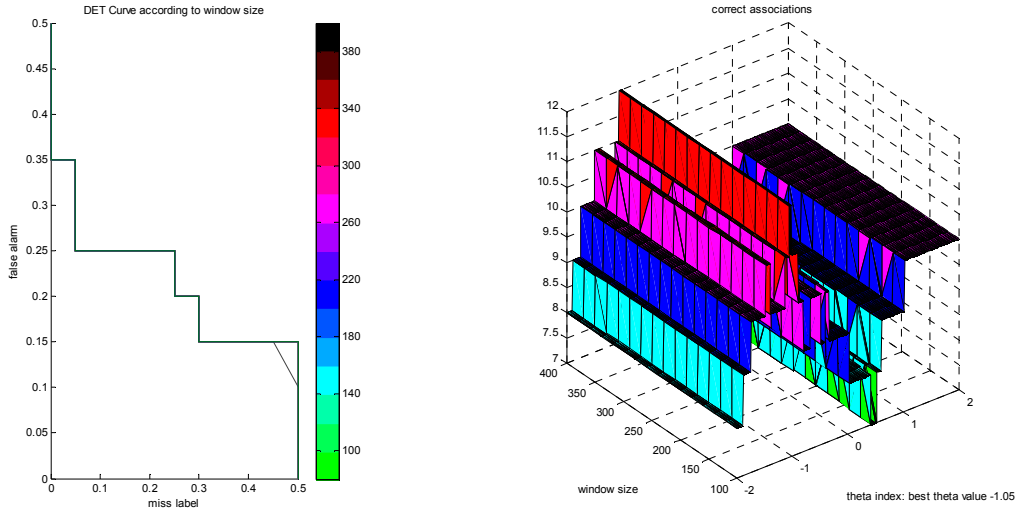


Figure 4.15(b) the results of using DNA-MAP approach when using a UBM.

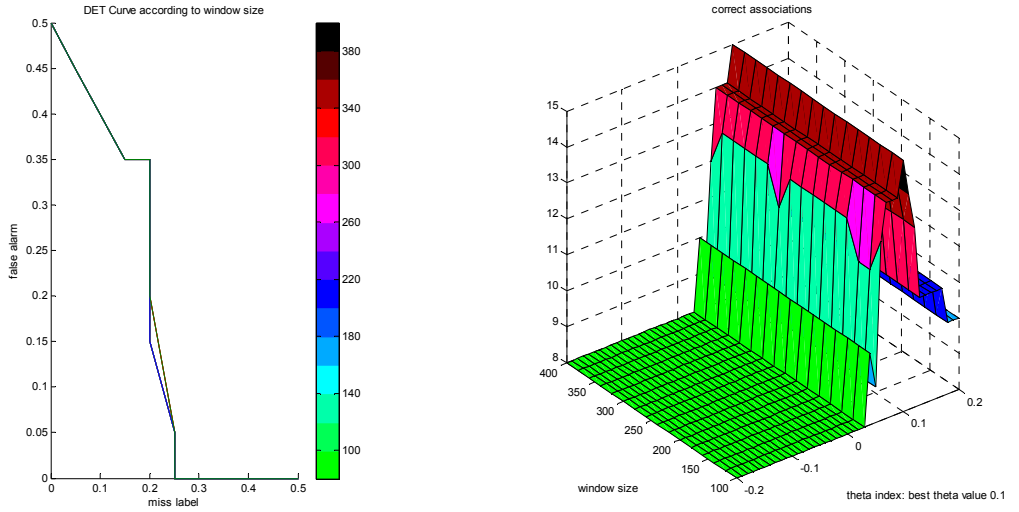


Figure 4.15(c) the results of using DNA-MAP approach when using a large cohort.

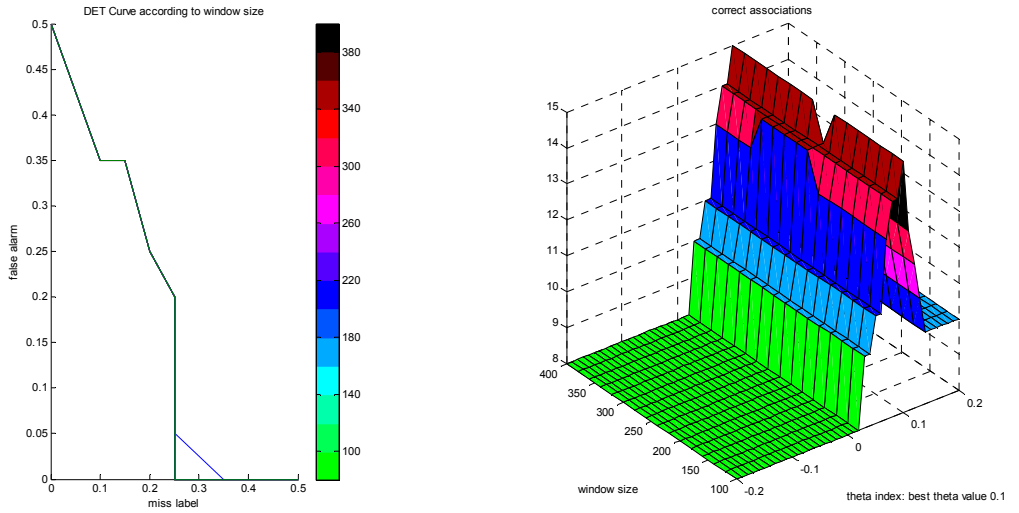


Figure 4.15(d) the results of using DNA-MAP approach when using a small cohort.

The results will be discussed by approach:

MFCC-MAP: The non-target cohort selection is not generating better results with the MAP approaches. Also, as observed before, there is almost no effect of the window size in the MAP approach.

DNA-Dist: On the other hand, the non-target cohort selection improves the results with the DNA-Dist approach. Also, it looks like the smaller the cohort size is the smaller the window size that would suffice. However, using a UBM as a non-target cohort is inaccurate due to the difference in the model used (using a GMM for UBM and mono Gaussian for speaker modelling). Thus, this set of results holds no significance.

The best results were achieved by using *DNA-Dist* with small windows (which is also faster in testing time). Moreover, careful selection of a window size (100->120 frames) generates better results than MFCC-MAP (base model).

DNA-MAP: The results are worse than MFCC-MAP counterparts. That is due to the fact that the used DNA is a general one, not finely tuned for the training data set. However, it is clear that using a cohort is not generating any better results.

4.3.4.3. Open Set Results with Normalization

The previous set of results is presented with no score normalization. Therefore, threshold ranges were manually set for each test result. This makes it particularly difficult to set a global threshold. The next set (Figures 4.16, 4.17 and 4.18) is the same as the previous set but with *t-test* normalization technique (as per equation 4.6) which evidently improves the results and makes it easier to set thresholds. The figures have the same description as the previous set of figures (4.13, 4.14 and 4.15) respectively.

$$\hat{s} = \frac{s - \mu}{\sigma} \tag{4.6}$$

The obvious remark is that different cohort settings are generating the exact same results. That is one the normalization effects. Also, the ranges for the thresholds are comparable. Although, the number of correct associations has dropped when using DNA-MAP approach.

The main advantage is that a better threshold can be set because of the stability of the results we achieve. To do so, we retain all the likelihoods for a certain verification set; And later we choose the best threshold value to maximise the correct associations.

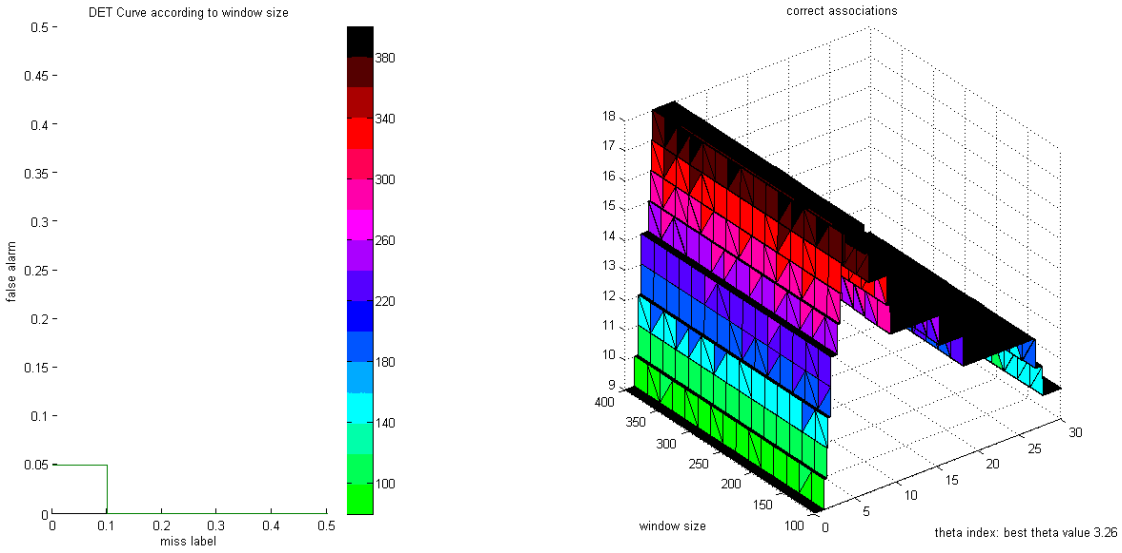


Figure 4.16(a) the results of using MFCC-MAP approach (with Normalization).

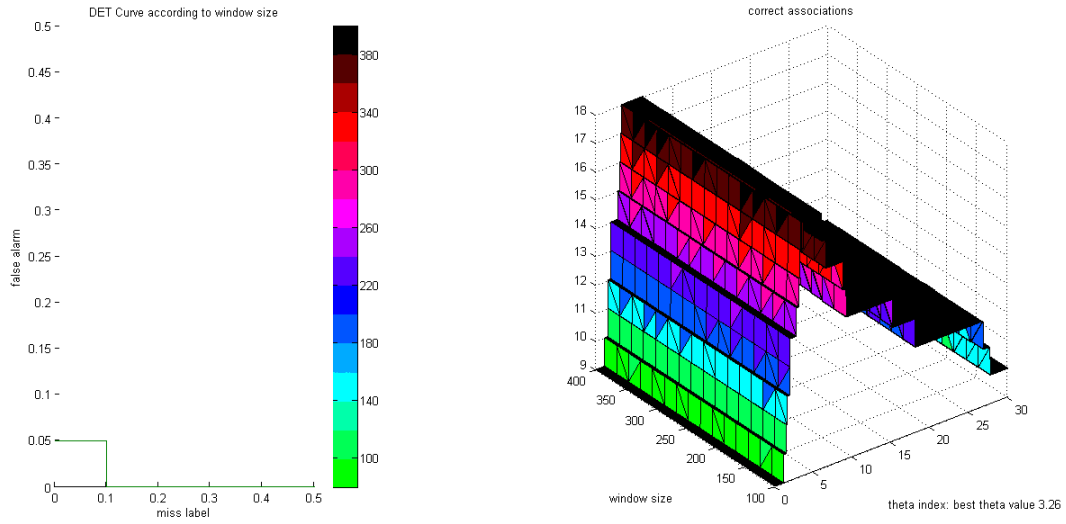


Figure 4.16(b) the results of using MFCC-MAP approach when using a UBM (with Normalization).

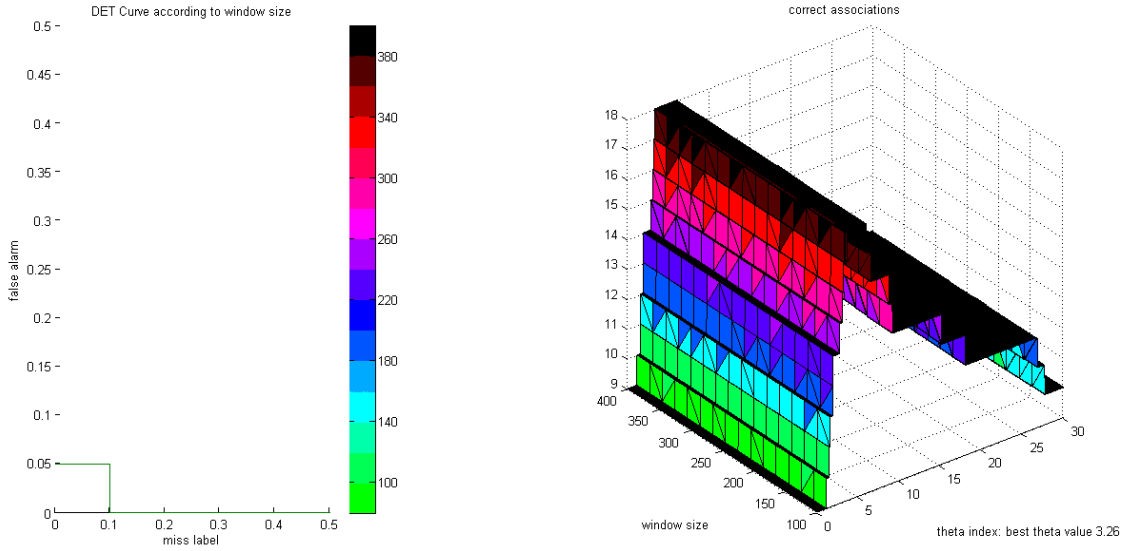


Figure 4.16(c) the results of using MFCC-MAP approach when using a large cohort (with Normalization).

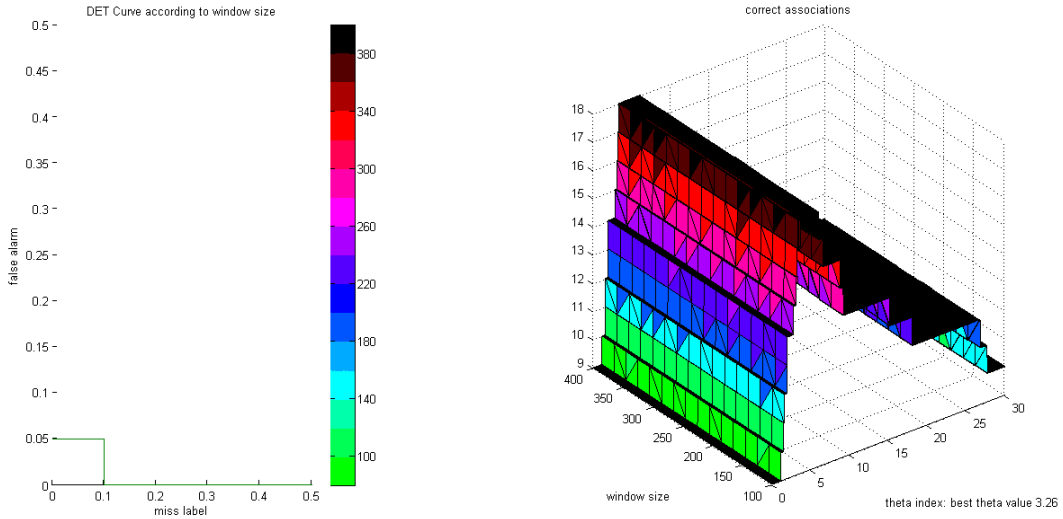


Figure 4.16(d) the results of using MFCC-MAP approach when using a large cohort (with Normalization).

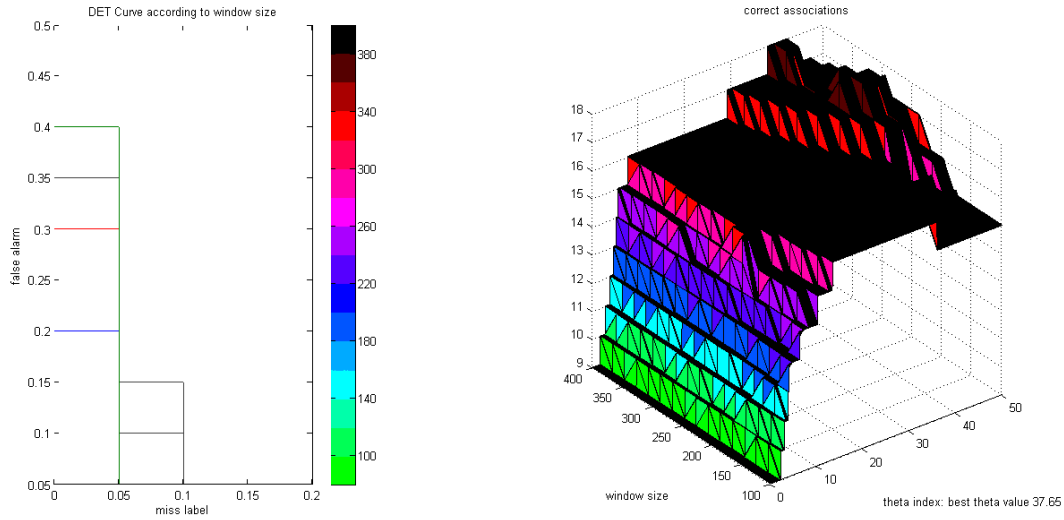


Figure 4.17(a) the results of using DNA-Dist approach (with Normalization).

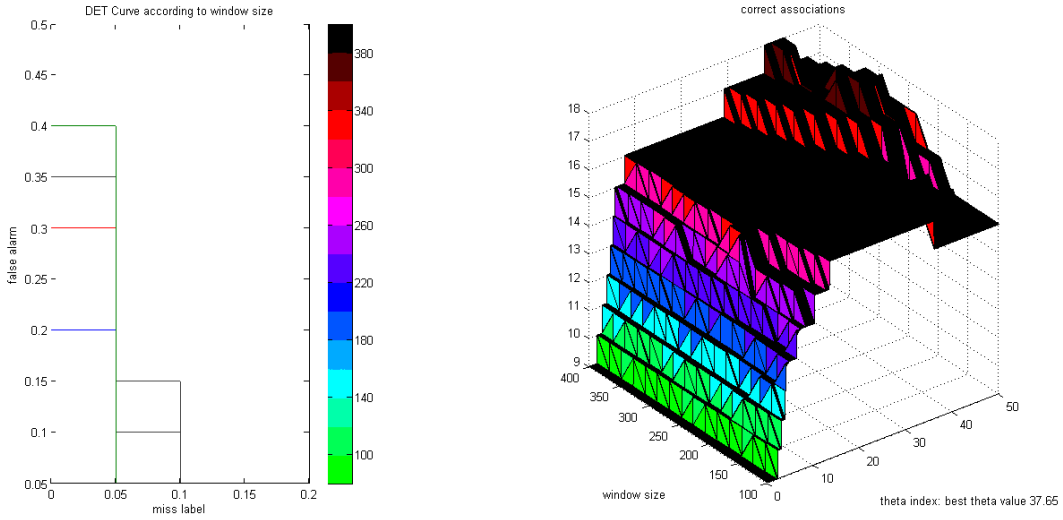


Figure 4.17(b) the results of using DNA-Dist approach using a UBM (with Normalization).

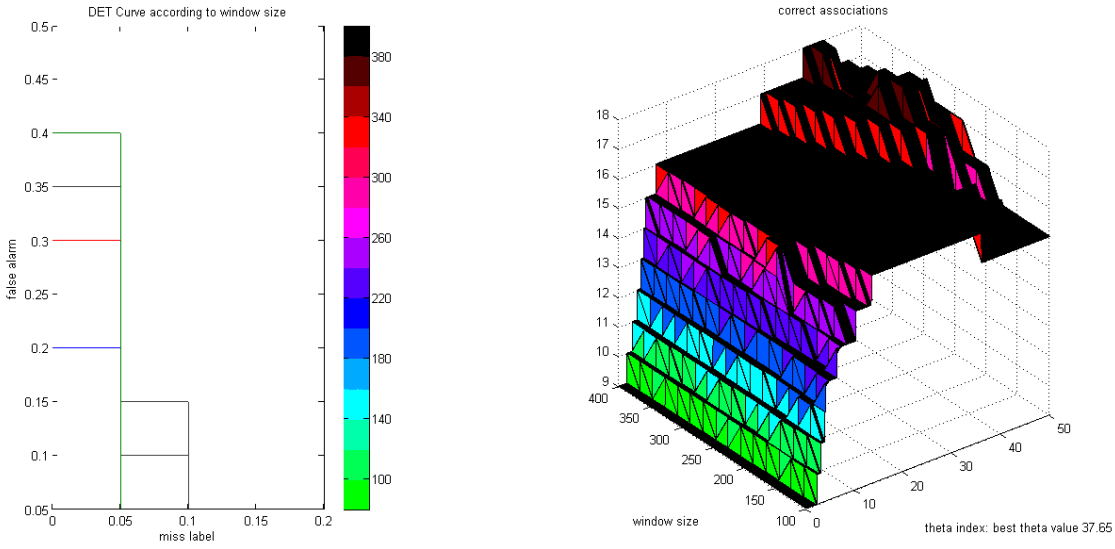


Figure 4.17(c) the results of using DNA-Dist approach using a large cohort (with Normalization).

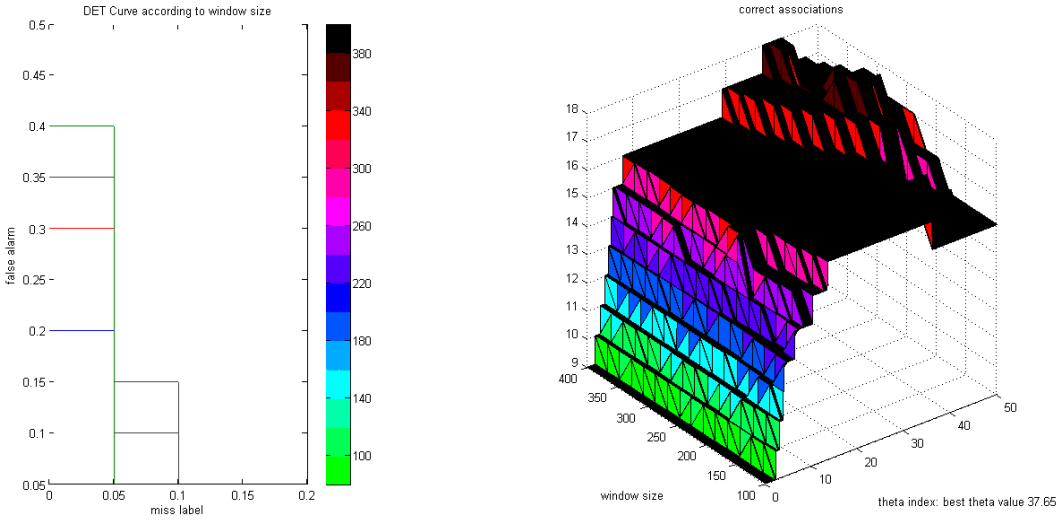


Figure 4.17(d) the results of using DNA-Dist approach using a small cohort (with Normalization).

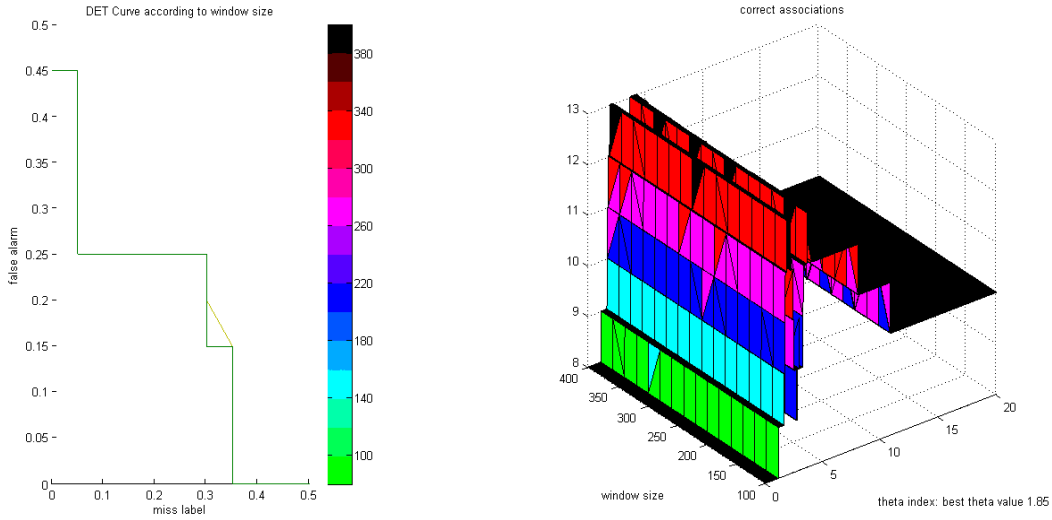


Figure 4.18(a) the results of using DNA-MAP approach (with Normalization).

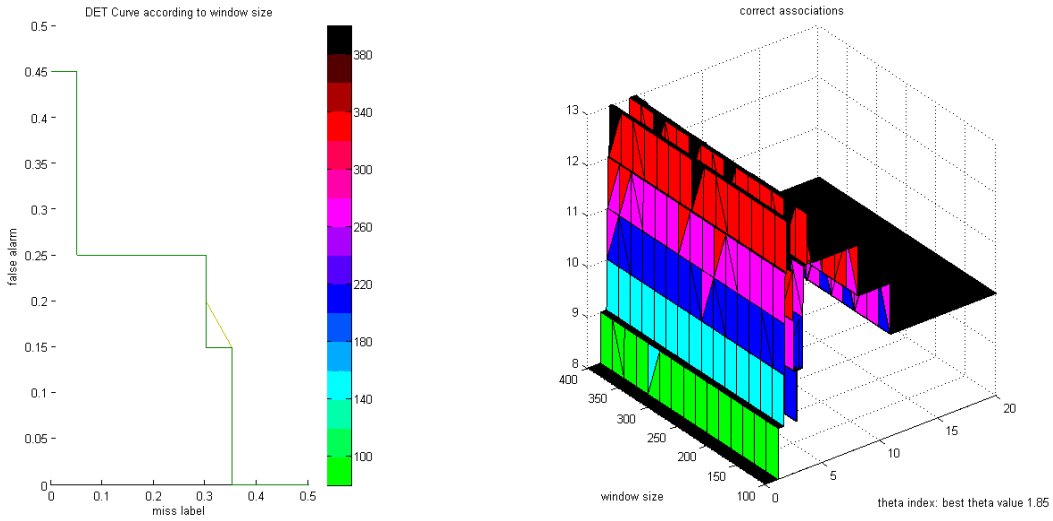


Figure 4.18(b) the results of using DNA-MAP approach using a UBM (with Normalization).

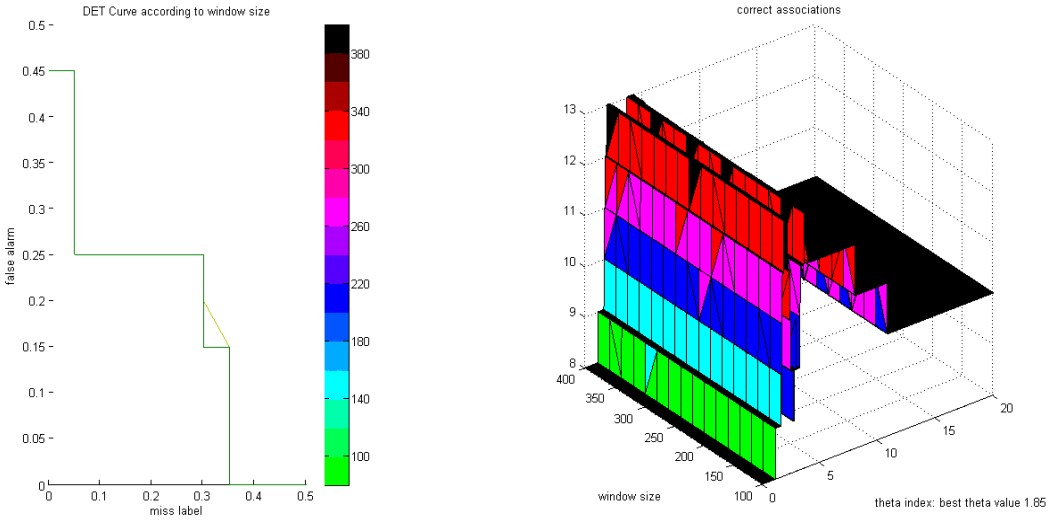


Figure 4.18(c) the results of using DNA-MAP approach using a large cohort (with Normalization).

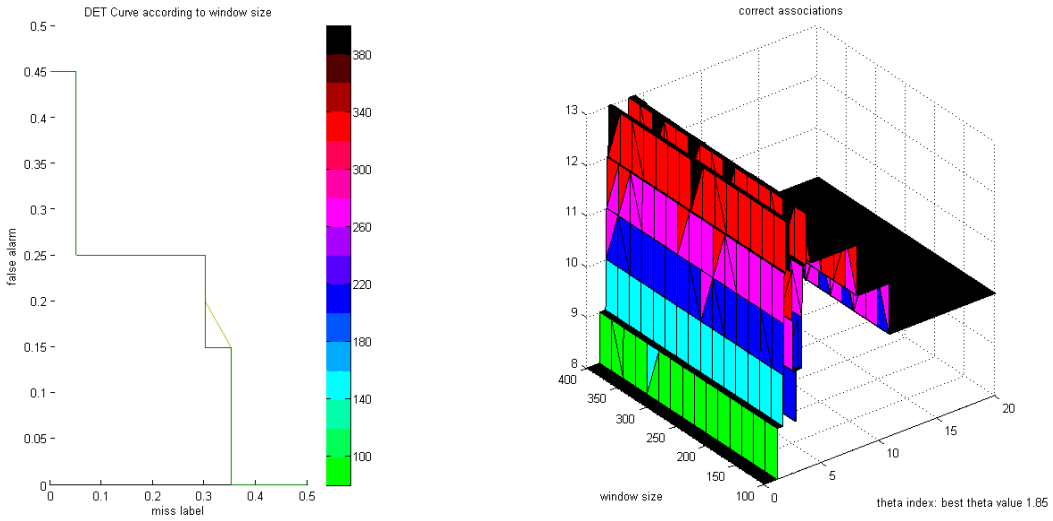


Figure 4.18(d) the results of using DNA-MAP approach using a small cohort (with Normalization).

4.3.5. Summary

In this section, evaluation methods for speaker identification tasks (open-set and closed-set) were introduced. Also, the datasets used to evaluate a system were discussed. The main flow of a typical identification system was clearly stated. This will be used later for the SRE.

The results of the closed-set were presented for selecting the number of GMM components first. Then three approaches were compared against the MFCC-MAP. Namely: DNA-MAP, DNA-Dist and DNA-MonoG. The results of the open-set were also presented. Lastly, the normalization effect was stated and shown empirically.

4.4. Our SRE Participation

This section is devoted to information about our SRE participation. Starting by the method used for selecting the tasks we are undertaking. Then, the methods used for phone VAD and interview VAD and the results of different methods used are reported. Finally, different training and testing issues are presented.

4.4.1. Task Selection

In total, SRE 2012 has *nine* different tasks that sites can choose to participate in or simply disregard. With the exception of the *Core* task, all the tasks are optional.

Tasks concerning a certain type of recording conditions (Microphones or Telephones) have no significance when presenting the new technology; unless we could only train the DNA using one type of information. The driving idea of the DNA is that it does speaker feature extraction regardless of the environment. To do so, enough training variability for the speaker has to be available so that the DNA can learn to extract only the speaker identity. Otherwise, if the variability is not available (or if it is biased) it is not possible to separate the identity from the rest of the mixed information in the stream.

Therefore, we selected the three tasks that can clearly show the superiority of the DNA representation over other representations with no explicit need for other environment separation techniques such as JFA.

- *Known*: the test stream is expected to be one of the speakers in the cohort. Therefore, if the claimed personality is not the highest probable personality, then the speaker is deemed an imposter, but one of the known speakers. This is exactly the closed-set identification task.

- *Unknown*: the test stream is either the claimed speaker or none of the cohort. Therefore, only the claimed personality is checked and compared against a UBM.
- *Core*: the test stream is either one of the speakers of the cohort, or a new speaker from outside the cohort. This is exactly the open-set identification task.

It may also be worthwhile to do a research of the training environment effect on the DNA features, i.e. Feature Mapping. That is particularly important if the DNA was biased towards a certain environment setting at training time. We did not conduct such research due to time limitation. However, if this was successfully done, then the same features could be adapted to be used in environment specific applications with no need to retrain a DNA.

4.4.2. Application Overview and Task Allocation

Our site consists of five researches led by Dr Ke Chen and administered by Mr Ahmad Salman. The proposed workflow for the speaker identification task is presented in figure 4.19. The process is split into three parts:

- Pre-processing: understanding the input structure; loading the data; extracting usable streams for DNA training, validation and then testing; and extracting MFCC representation. This was the responsibility of Mr Mate Toth and Ubai Sandouk (me).
- DNA Training: starting from the MFCC representation, DNA is trained using the specific provided labelled training data. Later, the trained deep structure would be used to extract the representation of the testing speaker. This was the responsibility of Mr Darren Hau.

Speaker Modelling: using the DNA features, different speaker models were explored and different approaches were tested. Depending on the task, UBM or cohort is selected and the final score is calculated. This was the responsibility of Mr Ubai Sandouk.

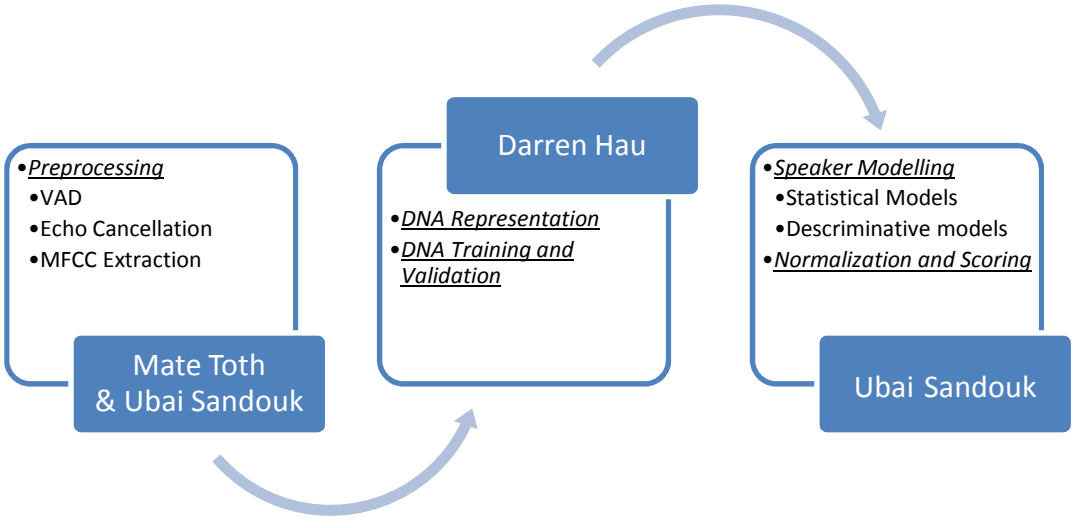


Figure 4.19 Different components of the proposed SRE system.

4.4.3. Phone VAD

Testing the phone VAD is done in the context of a closed-set identification task using the DNA-Dist and also MFCC-MAP and DNA-MAP approaches using 32-components GMMs; and different window sizes (although some stability is expected).

A fresh set of 30 male speakers where randomly selected (so that over-fitting over the previously used dataset is avoided). three different approaches are used:

- Energy based method (Energy): which requires a threshold (the amount of speech expected in the stream). This threshold is manually set. If no smoothing is applied, the output is audible but not perceivable as speech. That is because some low energy frames are removed from within the speech.
- Supervised method (Voicebox): which relays on background knowledge about noise and speech statistics, which is used to classify speech and non-speech (provided by Voicebox, see section 3.1)

Statistical method (EBVAD2): relays on the estimating the level of noise from previous frames, and using that as an energy threshold. When implemented properly, no threshold is needed and the output is smooth.

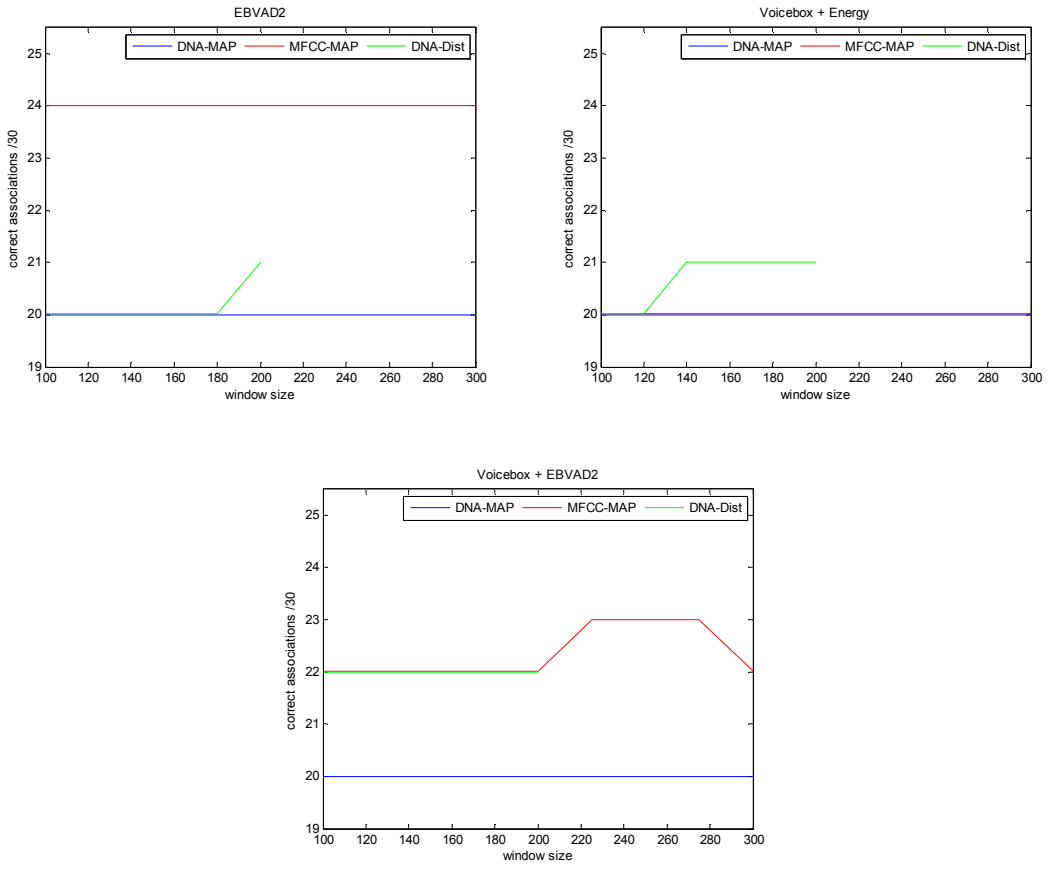


Figure 4.20 The results of different VAD methods on the DNA-Dist, DNA-MAP and MFCC-MAP approaches

Because the Energy approach breaks the continuity of the speech, we advise not to use it alone. This non-continuity of speech harms the MFCC and DNA representations. Figure 4.20 shows the results when applying different VAD tasks using MFCC-MAP, DNA-MAP and DNA-Dist approaches with different window sizes. The poor performance of the DNA features can be attributed to the use of a generic DNA. This DNA was trained using clear microphone speech of both genders. This biased effect could not be eliminated. Therefore, we trust the MFCC results better in this test, and accordingly *the EBVAD2 is used in the final application*.

4.4.4. Interview VAD

Interview data come on two streams. The first is recorded using a head-set for the interviewer (non-target speaker) with a small amount of added noise to cover the target speaker's speech. The second is recorded using a microphone between the interviewer and the interviewee, with both speakers' speech clearly presents. A very small time difference is present between the streams (they do not match on the interviewer speech). Also, echo and ambient sounds makes them incomparable. The first idea that comes to mind is to use a VAD on the first stream, reverse the selection and extract the residual from the second stream then apply the VAD one more time to remove the mutual non-voice activities. However, there are many imperfections in this process, and setting the universally correct parameters is impossible. After testing all the previously mentioned VAD approaches (see phone VAD) a novel approach based on *speaker diarization* is agreed on for the interview data. The proposed VAD follows this algorithm:

1- The non-target speaker stream is selected and MFCC feature vectors are extracted.
2- A window is slid across the stream calculating a statistical divergence (KL, KL2, GLR or BH) between the two halves of it. An assumption is made here that the MFCC feature vectors are Gaussian distributed (this inaccuracy is actually the source of any error).
3- Maximum distance points (ones that satisfy a certain threshold) are identified, and the stream is split accordingly.
4- The resulting segments are grouped using K-means offline clustering technique for two clusters: a) non-target speaker, b) added noise instead of the target speaker and real noise.
5- Filter both streams and select the part of the speaker stream that experience less change between the two streams as the non-target speech, the rest of the stream is considered the target speech.

After that, the resulting stream will contain target speech and some noise. It can either be directly admitted to the system, or further processed to remove noise using an energy based method.

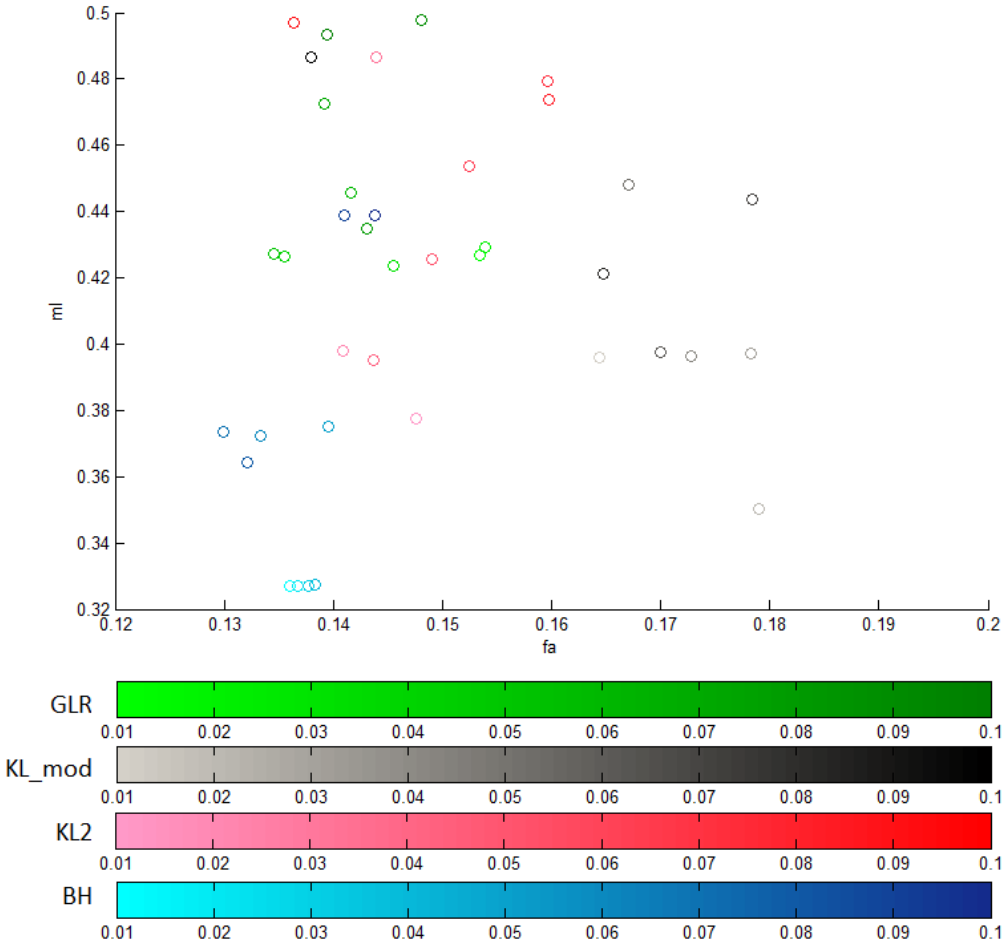


Figure 4.21 The results of different divergance measures on the interview VAD.

To evaluate the diarization based interview VAD, a number of interviews were annotated by hand; and the VAD algorithm was tested by the false alarm and miss rates. Figure 4.21 shows the results at different segmentation thresholds (the lower the thresholds, the more segments there will be. Hence, the less pure the generated clusters will be) for different divergence measures.

K-means produces "equal-sized" clusters; therefore some noise is grouped with the speech (In both clusters). This phenomenon explains why the output contains noise with the speech and then explains the high miss ratio. On the other hand, the low false alarm ratio is quite assuring and although we may miss some of the speech, what we are left with is only target speech. It is clear that the best measure is the Bhattacharyya distance (BH). However, different thresholds are generating close results.

4.4.5. DNA Training Data

In theory, the data presented to the DAN is supposed to be only speech, with as much variability as possible for all speakers. Moreover, for better results, a DNA structures can be trained for each gender. In our participation, we used different VAD techniques for different input

data types. This creates a problem because levels of noise are going to vary according to input type. One the other hand, this will add to the variability of training data.

4.4.6. Universal Background Model

Training and tuning a UBM is one of the most critical steps in the speaker identification task. Because all speaker models are later adapted from the UBM. A GMM with 32 components performed best in the identification experiments (section 4.3) because of the number of available data points, i.e. frames. Nevertheless, having more data in the real application will accommodate more components. Up to 2048 is reported in the literature when using 20 dimensions MFCC. Therefore, using a similar or lower number (e.g. 1024) of components is advised when using 100 dimensions DNA.

Moreover, a separate UBM may be tuned for each gender. This separation will reduce the size of the speaker set to the number of speakers in the said gender and hence improve the results. If the DNA is also trained on gender separate basis, it will learn better features for each gender and further improve the results.

4.4.7. Other issues

For the closed set, the best performing approach is the Distance approach using Bh divergence. However, as seen before, this cannot easily be extended for the verification task. Therefore, the MAP approach is used, but the number of components for the GMM must be carefully tuned. Up until the time of writing this report, the team had not yet trained the DNA structure. Therefore, no concrete test was done to select the best number of components.

For the verification task, a threshold is needed to be set. Once the scoring mechanism is set, a separate validation set of speakers (not used for training) can be used to validate the system. Their scores can be normalized and used to select the best threshold for the final application. The best threshold would be one that maximises the number of correct associations.

If more than one approach is found appropriate for different cases; then a hyper classifier can be built in order to make the decision of when to use each approach, or to fuse the results of different approaches. To do so, linear regression is usually used. A ready to use Matlab toolkit is available, i.e. Tools for Fusion and Calibration of automatic speaker detection systems (Brummer, 2005).

4.4.8. Summary

In this section, our SRE participation was described. The task selection was explained and the different roles the team members play were listed. The basic results needed to support out Speaker Identification results were also discussed. Different VAD for phone and interview data types are tested and their results demonstrated. DNA training and UBM tuning along with other issues such as score normalization are presented as well.

4.5. Summary

This chapter demonstrated the results of the conducted research; including the basic results of speaker diarization and identification tasks. Also, our SRE participation issues are presented and discussed.

Chapter 5

Conclusion

In recent years, MFCCs have gained enormous attention because of their relation to the human auditory system and the great performance they provide in speaker related tasks. However, it seems that improving the results of the task is not possible with MFCCs anymore. Therefore, researchers are introducing new representations, such as the speaker specific speech representation (Chen and Salman, 2011).

Following is a summary of this thesis: a quick revision of the project progress and results; future suggestions: further research direction to improve the results and refine the features; and a self-reflection conclusion: where I present a reflection of this project on my academic progress.

5.1. Summary

This project was aimed to study different speaker recognition tasks and compare the different speech representations and modelling approaches. The tasks discussed are speaker diarization (including speaker segmentation and clustering) and speaker identification and verification. Similarly, the representations examined included LPC, MFCC and DNA representations. On the other hand, statistical (generative) models and discriminative models were presented for different speech and speaker representations.

The findings of the project were also used as part of our team's participation in the NIST SRE 2012. *Voice activity detection* was extensively studied for both types of provided data; namely: phone data type and interview data type; and a novel approach for interview VAD was introduced. As per our results for the VAD, the novel approach is performing better than conventional VAD approaches.

A research regarding *speaker modelling* using the DNA features and statistical fit for them was conducted and presented. The mono Gaussian distribution can be used to model one speaker, this claim is supported by the success of the distance approaches. However, the mono Gaussian cannot separate the speakers well and it seems that different speaker models are largely

overlapped. On the other hand, the GMM using a carefully selected number of components fits the DNA features better for separability.

The DNA features have shown great superiority in accuracy in different speaker related tasks over other speaker related features, such as MFCC. However, they still suffer from some issues. Most critically, the high dimensionality of the DNA feature vector; which causes computational delays in any speaker related task. Consequently, building speaker models becomes much harder. Similarly, the DNA is trained using a certain set of speakers. This process extracts features that are related to the training set and therefore are biased towards the training set characteristics. If the same DNA was used to extract features from another set (that the DNA did not experience at training time) they will not be accurate and will not represent the sole identity of the speaker, rather they will be biased to reflect information about the training data set characteristics. Nonetheless, when trained properly, DNA features successfully isolate all the environment effects and differences between training and testing streams; depending on enough variability being provided at DNA training time. Hence, no further processing needs to be done to refine the speaker models.

5.2. Future Suggestions

5.2.1. SRE

First, the results presented in this thesis should be incorporated in our participation in the SRE. The system is expected to have much improved results than other systems; especially that the DNA will be properly trained (see section 4.4). Afterwards, if the results were not satisfactory, the problems need to be identified and overcome. There is evidence of the superiority of the DNA representation over other speech representations. This should allow for better performance in any speaker related task.

5.2.2. High Dimensionality

One of the major limitations for the DNA representation is its high dimensionality (100 features). Therefore, a complete study should be done in order to check whether the number of features can be reduced in certain or all cases. For instance, as seen in section 3.2 *anchors model* is a method to reduce dimensionality by selecting N hyper speaker models and relate all feature vectors to those models. Thus, reducing the number of feature in each vector from 100 to only N. When using 32 hyper speakers we achieved little worse results than using 100 DNA features. This result suggests that in certain cases (or probably in general) there is no need for the 100 features.

Nonetheless, if the 100 features are to be retained, then better computational algorithms should be used (working in the log domain for example).

5.2.3. Voiced Speech

It has come to be well known that voiced segments contain more information about the identity of the speaker than unvoiced segments. This is true when using the DNA features as well (Chen and Salman, 2011a; b). In the short time of this project, we did not study this aspect of the DNA features. However, we believe that using only voiced segments to do the speaker identification should further improve the results.

5.3. Self-Reflection

During the time of this project I have acquired many valuable "meta" skills, especially time management and research methodology. I got to manage my time, resources, code and results through some resources provided by the university (such as *My Drive*[13]). Moreover, I got to refine and follow my plan precisely and deliver on time. However, the most important skill I developed is the research methodology; where I was successfully guided through the research. Starting from the background knowledge phase, then through the method phase, and later the innovation phase; where I could develop my own ideas and reflect my own understanding of the technology at hand.

To successfully complete the project, I had to obtain university level knowledge about different fields such as digital speech and signal processing. Also, I had to gain extensive knowledge about statistics and its application is Machine Learning. I was exposed to many Machine Learning approaches and had to use them. Likewise, many models were new to me such as GMM and HMM.

On the technical side, I had the chance to learn Matlab professionally and use more than one toolkit for speech and signal processing. The university made it easy to access resources for that purpose including the *Matlab Academic Tour*[14].

One of the most interesting opportunities that I was given is to use my research in a real life evaluation (i.e. SRE) using real life data rather lab recorded data. The SRE evaluation is the most comprehensive evaluation for current real life speaker-recognition technologies. This opportunity is given to very few students worldwide and it is not always available. I feel privileged to have been part of SRE 2012 family and most privileged to have been part of Manchester University research community at these stimulating times.

[13] Accessible through the students' portal.

[14] July, 4th 2012. At the university of Manchester.

References

Abad, A., Luque, J., Trancoso, I. and Hernando, J., 2011. The L2F-UPC Speaker Recognition System for NIST SRE 2010. In *2010 NIST Speaker Recognition evaluation*.

Almpanidis, G. and Kotropoulos, C., 2008. Phonemic segmentation using the generalised Gamma distribution and small sample Bayesian information criterion. *Speech Communication*, 50(1), pp.38–55.

Anguera, X., Wooters, C., Peskin, B. and Aguiló, M., 2006. Robust speaker segmentation for meetings: The ICSI-SRI spring 2005 diarization system. *Machine Learning for Multimodal Interaction*, pp.402–414.

Anguera, X. and Bonastre, J.F., 2010. A novel speaker binary key derived from anchor models. In *Eleventh Annual Conference of the International Speech Communication Association*. pp. 2118–2121.

Anguera, X. et al., 2012. Speaker Diarization: A Review of Recent Research. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(2), pp.356–370.

Anliker, U., Randall, J. and Tröster, G., 2006. Speaker separation and tracking system. *EURASIP journal on applied signal processing*, 2006, pp.1–14.

Anon, 2009. The 2009 (RT-09) Rich Transcription Meeting Recognition Evaluation Plan. *Evaluation Paper*, pp.1–18. [online] Available at: http://www.itl.nist.gov/iad/mig/tests/rt/2009/docs/rt09-meeting-eval-plan-v2.pdf. [Accessed August 18, 2012].

Anon, 2012. The NIST Year 2012 Speaker Recognition Evaluation Plan. *Evaluation Paper*, pp.1–7. [online] Available at: http://www.nist.gov/itl/iad/mig/upload/NIST_SRE12_evalplan-v11-r0.pdf [Accessed August 18, 2012].

Auckenthaler, R., Carey, M. and Lloyd-Thomas, H., 2000. Score Normalization for Text-Independent Speaker Verification Systems. *Digital Signal Processing*, 10(1-3), pp.42–54.

Bimbot, F., Magrin-Chagnolleau, I. and Mathan, L., 1995. Second-order statistical measures for text-independent identification. *Speech Communication*, 17(1-2), pp.177–192.

Boakye, K., Trueba-Hornero, B., Vinyals, O. and Friedland, G., 2008. Overlapped speech detection for improved speaker diarization in multiparty meetings. In *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*. IEEE, pp. 4353–4356.

Brummer, N., 2005. FoCal. *Tools for Fusion and Calibration of automatic speaker detection systems*. [online] Available at: http://www.dsp.sun.ac.za/~nbrummer/focal/ [Accessed August 18, 2012].

Campbell, J.P., 1997. Speaker recognition: a tutorial. *Proceedings of the IEEE*, 85(9), pp.1437–1462.

Cettolo, M., Vescovi, M. and Rizzi, R., 2005. Evaluation of BIC-based algorithms for audio segmentation. *Computer Speech and Language*, 19(2), pp.147–170.

Chen, K. and Salman, A., 2011a. Extracting Speaker-Specific Information with a Regularized Siamese Deep Network. In *Advances in Neural Information Processing Systems 25 (NIPS'11)*. pp. 1–9.

Chen, K. and Salman, A., 2011b. Learning speaker-specific characteristics with a deep neural architecture. *IEEE transactions on neural networks*, 22(11), pp.1744–56.

Chen, S. and Gopalakrishnan, P.S., 1998. Speaker, environment and channel change detection and clustering via the bayesian information criterion. *Proc. DARPA Broadcast News Transcription and Understanding Workshop*, pp.127–132.

Cheng, S. and Wang, H., 2010. BIC-Based Speaker Segmentation Using Divide-and-Conquer Strategies with Application to Speaker Diarization. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(1), pp.141–157.

Coleman, J., 2005. *Introducing speech and language processing*. Cambridge, UK: Cambridge University Press.

Dempster, A., Laird, N.M. and Rubin, D.B., 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society.* 39(1), pp.1–38.

Garcia-Romero, D. and Espy-Wilson, C.Y., 2010. Joint factor analysis for speaker recognition reinterpreted as signal coding using overcomplete dictionaries. In *The Speaker and Language Recognition Workshop*. pp. 117–124.

Gauvain, J.L., Lamel, L. and Adda, G., 1998. Partitioning and transcription of broadcast news data. *The 5th International Conference on Spoken Language Processing (ICSLP'98),* pp.1335–1338.

Gish, H. and Schmidt, M., 1994. Text-independent speaker identification. *IEEE Signal Processing Magazine*, 11(4), pp.18–32.

Goldberger, J., Gordon, S. and Greenspan, H., 2003. An efficient image similarity measure based on approximations of KL-divergence between two gaussian mixtures. In *Proceedings Ninth IEEE International Conference on Computer Vision*. IEEE, pp. 487–493 vol.1.

Gonzalez-Dominguez, J. et al., 2010. ATVS-UAM NIST SRE 2010 SYSTEM. *Proceedings of FALA 2010*, pp.143–146.

Gouyon, F. and Pachet, F., 2000. On the use of zero-crossing rate for an application of classification of percussive sounds. In *Third Conference on Digital Audio Effects*. Verona (Italy), pp. 3–8.

Hermansky, H, 1990. Perceptual linear predictive (PLP) analysis of speech. *The Journal of the Acoustical Society of America*, 87(4), pp.1738–52.

Huang, X., Acero, A. and Hon, H., 2001. *Spoken Language Processing*. New York: Prentice Hall.

International Standards Office, 2001, *ISO 15938-4:2001 MPEG-7: Multimedia Content Description Interface, Part 4: Audio*. [online] Available at: <http://505606.pbworks.com/f/ISO-IEC-15938-4-Audio.pdf> [Accessed 22nd April 2012].

Jurafsky, D. and Martin, J., 2009. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition.* 2nd ed. New Jersey: Pearson Prentice Hall/Pearson education international.

Kampa, K., 2010. *Derivation of Variational Bayesian for Gaussian Mixture Models (VBGMM)*, [online] Available at: http://dl.dropbox.com/u/14115372/variational_apprx_inference/VBGMM_derivation.pdf [Accessed August 22, 2012].

Kemp, T., Schmidt, M., Westphal, M. and Waibel, A., 2000. Strategies for automatic segmentation of audio data. In *2000 IEEE International Conference on Acoustics, Speech, and Signal Processing.* Istanbul: IEEE, pp. 1423–1426.

Kenny, P., Boulianne, G., Ouellet, P. and Dumouchel, P., 2007. Joint Factor Analysis Versus Eigenchannels in Speaker Recognition. *IEEE Transactions on Audio, Speech and Language Processing*, 15(4), pp.1435–1447.

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

Klatt, D., 1980. Software for a cascade/parallel formant synthesizer. *Journal of the Acoustical Society of America*, 67(3), pp.971–995.

Kotti, M. et al., 2006. Automatic speaker segmentation using multiple features and distance measures: A comparison of three approaches. In *IEEE International Conference on Multimedia and Expo*. IEEE, pp. 1101–1104.

Kotti, M., Moschou, V. and Kotropoulos, C., 2008. Speaker segmentation and clustering. *Signal Processing*, 88(5), pp.1091–1124.

Kuhn, R., Junqua, J., Nguyen, P. and Niedzielski, N., 2000. Rapid speaker adaptation in eigenvoice space. *IEEE Transactions on Speech and Audio Processing*, 8(6), pp.695–707.

LDC, 1990a. NTIMIT Speech Corpus CD-ROMs (NTIMIT). [online] Available at: http://www.ldc.upenn.edu/Catalog/readme_files/ntimit.readme.html [Accessed August 20, 2012].

LDC, 1990b. The DARPA TIMIT Acoustic-Phonetic Continuous Speech Corpus (TIMIT). [online] Available at: http://www.ldc.upenn.edu/Catalog/readme_files/timit.readme.html [Accessed August 20, 2012].

Lee, Y., Lee, K.Y. and Lee, J., 2006. The Estimating Optimal Number of Gaussian Mixtures Based on Incremental k-means for Speaker Identification. *International Journal of Information Technology*, 12(7), pp.13–21.

Liu, D. and Kubala, F., 1999. Fast speaker change detection for broadcast news transcription and indexing. In *Sixth European Conference on Speech Communication and Technology*. Budapest, Hungary, pp. 1031–1034.

Malayath, N., Hermansky, H., Kajarekar, S. and Yegnanarayana, B., 2000. Data-Driven Temporal Filters and Alternatives to GMM in Speaker Verification. *Digital Signal Processing*, 10(1-3), pp.55–74.

Meignier, S., Bonastre, J., Fredouille, C. and Merlin, T., 2000. Evolutive HMM for multi-speaker tracking system. In *Acoustics, Speech, and Signal Processing, 2000. ICASSP'00. Proceedings. 2000 IEEE International Conference on*. IEEE, pp. II1201–II1204.

Ning, B., 2010. Developing an Isolated Word Recognition System in MATLAB. *MATLAB Digest*, pp.1–6. Available at: http://www.mathworks.com/tagteam/60673_91805v00_WordRecognition_final.pdf [Accessed August 20, 2012].

Rabiner, L. and Bing-Hwang, J., 1993. *Fundamentals of Speech Recognition*. New Jersey: Prentice-Hall.

Ramachandran, R., Farrell, K., Ramachandran, R. and Mammone, R., 2002. Speaker recognition — general classifier approaches and data fusion methods. *Pattern Recognition*, 35, pp.2801–2821.

Ramírez, J., Górriz, J. and Segura, J., 2007. Voice activity detection. fundamentals and speech recognition system robustness. In M. Grimm and K. Kroschel, eds. *Robust Speech Recognition and Understanding*. Vienna: I-Tech, pp. 1–22.

Reynolds, D., 1995. Speaker identification and verification using Gaussian mixture speaker models. *Speech communication*, 17(1-2), pp.91–108.

Reynolds, D. and Rose, R., 1995. Robust text-independent speaker identification using Gaussian mixture speaker models. *IEEE Transactions on Speech and Audio Processing*, 3(1), pp.72–83.

Reynolds, D., Quatieri, T. and Dunn, R., 2000. Speaker Verification Using Adapted Gaussian Mixture Models. *Digital Signal Processing*, 10(1-3), pp.19–41.

Reynolds, D., 2002. An overview of automatic speaker recognition technology. In *IEEE International Conference on Acoustics Speech and Signal Processing*. IEEE, p. IV–4072–IV–4075.

Reynolds, D., 2003. Channel robust speaker verification via feature mapping. *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03).*, 2, p.II–53–6.

Reynolds, D., 2008. Gaussian Mixture Models. *Encyclopedia of Biometric Recognition*, (2), pp.1–5.

Shlens, J., 2007. Notes on Kullback-Leibler Divergence and Likelihood Theory. *System Neurobiology Laboratory, Salk Institute for*, 92037, pp.1–4.

Sohn, J., Kim, N. and Sung, W., 1999. A statistical model-based voice activity detection. *IEEE Signal Processing Letters*, 6(1), pp.1–3.

Sturim, D. et al., 2011. The MIT LL 2010 speaker recognition evaluation system: Scalable language-independent speaker recognition. In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, pp. 5272–5275.

Sturim, D. and Reynolds, D., 2005. Speaker Adaptive Cohort Selection for Tnorm in Text-Independent Speaker Verification. *Proceedings. (ICASSP '05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.*, 1, pp.741–744.

Tranter, S., 2006. An overview of automatic speaker diarization systems. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(5), pp.1557–1565.

Wooters, C. et al., 2004. Towards robust speaker segmentation: The ICSI-SRI fall 2004 diarization system. In *RT-04F Workshop*.

Appendix A

Time Plan

In this appendix, the time plan (in Gantt format[15]) is presented. The milestones are the deliverables of the project: a) the Preliminary Report; b) the Background Report; and c) the Dissertation. A 20 days period is presented in orange describing the exam period, in which no progress in the project will be made.

The SRE schedule is also presented. The milestones are a) the registration deadline and training data collection; b) the evaluation dataset release; c) the deadline for any submission; and d) the live evaluation session (i.e. the workshop). Different people are expected to work on this project; hence different progress lines are available.

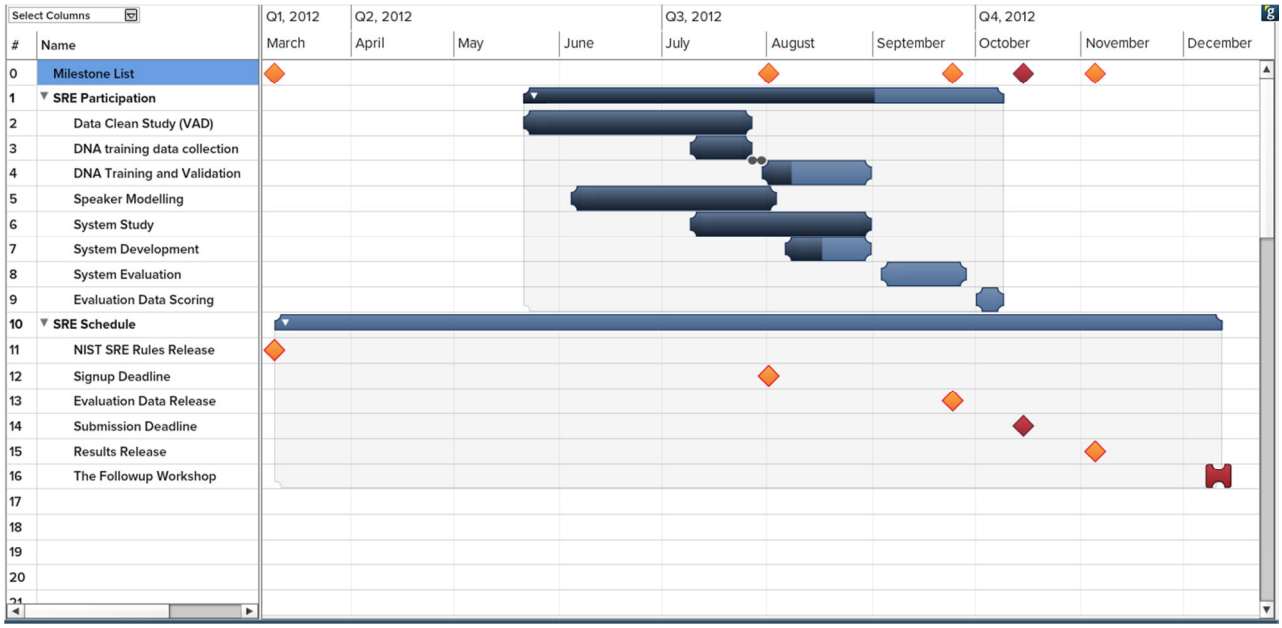


Figure A.1 SRE Schedule

[15] The plan was compiler by the free online tool "gantto" [online] Available on <http://gantto.com> [Accessed 28 April 2012]

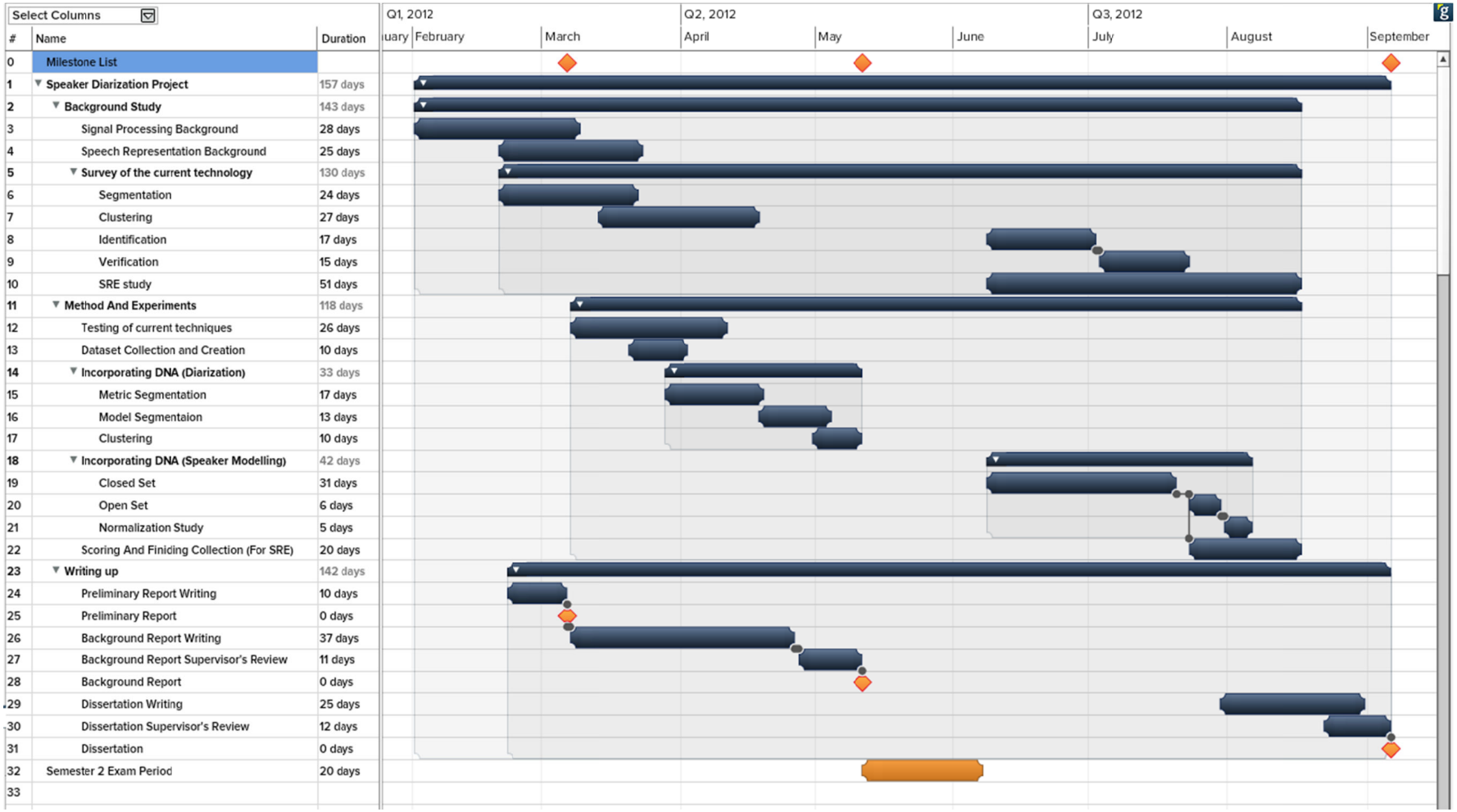


Figure A.2 Time plan that was followed during the project

Appendix B

Glossary

Averager: A difference equation to approximate a low-pass filter.

Beam Forming: An algorithm to align of many streams of the same recording.

Butterworth Filter: A filter written in the different equation format.

Cepstrum domain: A sample domain which is obtained when applying frequency analysis to the frequency domain.

Difference Equation: An approximate to a filter as a combination of previous input/output samples.

Differentiator: A difference equation to approximate a high-pass filter

Dirac Impulse: A signal consisting of one infinitely high impulse.

Divergence: A measure of the distance between two statistical models.

DNA Features: Speaker specific features extracted using the deep network architecture.

EER: Equal Error Rate, a special point on the ROC curve, where the miss ratio is equal to the false alarm ratio.

Filter Bank: A set of filters to cover the whole range of frequency found in the signal.

Formants: The higher frequency boundaries for a voiced part of speech.

Fourier Transformation: A method to decompose a real life signal into a number of sinusoids.

Frequency: The number of times the signal repeats itself in a time unit.

Fundamental Frequency: The frequency at which the vocal cords vibrate.

Gaussian Mixture Model: A mixture of more than one normal distribution, combined with different weights.

Klatt's formant synthesizer: A source/filter model based synthesizer proposed by Klatt.

Liner Predictive Coding: Compression technique using linear combinations and residual.

Mel-Scale: A non-linear scale based on human frequency perception.

NIST: National Institute for Science and Technology

Nyquist Frequency: The highest sampling frequency that can faithfully preserve the original signal.

Period: The time a periodic signal needs to repeat itself.

Periodic Signal: A signal that repeat itself every period T.

Pitch: The human perception of the fundamental frequency.

ROC Curves: Receiver Operating Characteristic, a plot to show the effect of different parameter values in a parameterised method.

Signal Filtering: Affecting certain frequency domains of the signal.

Signal Windowing: Extracting parts of the signal in the time domain.

source/filter model: A simple method to model the human voice production by using a source (base frequency generator) and a filter (filter bank to shape the final voice).

Speaker Clustering: Group different speech segments according the speaker.

Speaker Detection: Correctly flag the speeches, out of a set of speech, of a certain speaker.

Speaker Diarization: Detect who is speaking when on a stream.

Speaker Identification: Correctly identify the speaker, out of a certain set, of a certain speech.

Speaker Modelling: Different techniques used to build a model to retain enough information about the speaker identity.

Speaker Recognition SR: Any task related to automatically deal with the speaker identity.

Speaker Segmentation: Automatically identify turn points of speakers in a large stream.

Speaker Specific Features (DNA): Deep network architecture learned features for maximum discrimination among speakers.

Speaker Verification: Correctly verify that two segments are uttered by the same speaker.

SRE: Speaker Recognition Evaluation

Text-Dependent SR: Any task of SR that the speakers are restricted to a certain speech.

Text-Independent SR: Any task of SR that does not restrict the speech.